

Oracle® Multitenant 管理者ガイド 19c

F16108-10(原本部品番号:E96136-16)

2022年6月

タイトルおよび著作権情報

Oracle Multitenant管理者ガイド, 19c

F16108-10

[Copyright ©](#) 2017, 2022, Oracle and/or its affiliates.

原著者: Randy Urbano, Lance Ashdown

原著協力者: Patricia Huey, Donna Keesling, Roopesh Kumar, Bert Rich, Richard Strohm

原著協力者: Penny Avril, Thomas Baby, Hermann Baer, Yasin Baskan, Dominique Djeunot, Andre Kruglikov, Kishy Kumar, Sue Lee, Siyu Liu, Bryn Llewellyn, Colin McGregor, John McHugh, Valarie Moore, Muthu Olagappan, Bhavesh Patel, Kumar Rajamani, Giridhar Ravipati, Can Tuzla, Patrick Wheeler

目次

- [タイトルおよび著作権情報](#)
- [はじめに](#)
 - [対象読者](#)
 - [ドキュメントのアクセシビリティについて](#)
 - [関連ドキュメント](#)
 - [表記規則](#)
- [Oracle Multitenant管理者ガイドのこのリリースでの変更点](#)
 - [Oracle Databaseリリース19c、バージョン19.1での変更点](#)
 - [新機能](#)
 - [Oracle Databaseリリース18c、バージョン18.1での変更点](#)
 - [新機能](#)
- [第I部 マルチテナント・アーキテクチャ](#)
 - [1 マルチテナント・アーキテクチャの紹介](#)
 - [マルチテナント・アーキテクチャについて](#)
 - [CDBのコンテナについて](#)
 - [マルチテナント・アーキテクチャのユーザー・インタフェースについて](#)
 - [マルチテナント・アーキテクチャの利点](#)
 - [非CDBアーキテクチャの課題](#)
 - [マルチテナント・アーキテクチャのデータベース統合に対する利点](#)
 - [マルチテナント・アーキテクチャの管理性に対する利点](#)
 - [データベース統合の方法](#)
 - [CDBの作成](#)
 - [PDBの作成](#)
 - [クローニングによるPDBの作成](#)
 - [シードからのPDBの作成](#)
 - [PDBまたは非CDBのクローニングによるPDBの作成](#)
 - [PDBスナップショットからのクローン](#)
 - [スナップショット・コピーPDB](#)
 - [リフレッシュ可能なクローンPDB](#)
 - [接続によるPDBの作成](#)
 - [切断されたPDBの接続によるPDBの作成](#)
 - [非CDBからのPDBの作成](#)
 - [再配置によるPDBの作成](#)
 - [プロキシPDBとしてのPDBの作成](#)
 - [マルチテナント環境のドキュメント・ロードマップ](#)
 - [2 マルチテナント・アーキテクチャの概要](#)
 - [CDBのコンテナの概要](#)
 - [CDBルートとシステム・コンテナ](#)
 - [PDB](#)
 - [PDBのタイプ](#)
 - [PDBの目的](#)
 - [プロキシPDB](#)

- [PDBの名前](#)
 - [PDB間のデータベース・リンク](#)
- [CDBのデータ・ディクショナリ・アーキテクチャ](#)
 - [データ・ディクショナリ分割の目的](#)
 - [メタデータ・リンクとデータ・リンク](#)
 - [CDBのコンテナ・データ・オブジェクト](#)
 - [CDBのデータ・ディクショナリ・ストレージ](#)
- [現在のコンテナ](#)
- [コンテナ間操作](#)
- [CDBの共通性の概要](#)
 - [CDBの共通性について](#)
 - [共通性の原則](#)
 - [CDBのネームスペース](#)
 - [CDBの共通ユーザーおよびローカル・ユーザーの概要](#)
 - [CDBの共通ユーザー](#)
 - [CDBのローカル・ユーザー](#)
 - [CDBの共通ロールおよびローカル・ロールの概要](#)
 - [CDBの共通ロール](#)
 - [CDBのローカル・ロール](#)
 - [CDBでの権限およびロール付与の概要](#)
 - [CDBにおける権限およびロールの付与の原則](#)
 - [CDBでローカルに付与される権限およびロール](#)
 - [ロールまたは権限がローカルに付与される条件](#)
 - [ローカルに付与されるロールおよび権限](#)
 - [CDBで共通に付与されるロールおよび権限](#)
 - [付与が共通になる条件](#)
 - [共通に付与されるロールおよび権限](#)
 - [CDBでのPUBLICへの付与](#)
 - [権限およびロールの付与：シナリオ](#)
 - [CDBの共通オブジェクトおよびローカル・オブジェクトの概要](#)
 - [共通監査構成の概要](#)
 - [PDBロックダウン・プロファイルの概要](#)
- [アプリケーション・コンテナ内のアプリケーションの概要](#)
 - [アプリケーション・コンテナについて](#)
 - [アプリケーション・コンテナの目的](#)
 - [アプリケーション・コンテナの主な利点](#)
 - [アプリケーション・コンテナのユースケース：SaaS](#)
 - [アプリケーション・コンテナのユースケース：論理データ・ウェアハウス](#)
 - [アプリケーション・ルート](#)
 - [アプリケーションPDB](#)
 - [アプリケーション・シード](#)
 - [アプリケーション共通オブジェクト](#)
 - [アプリケーション共通オブジェクトの作成](#)
 - [メタデータリンク・アプリケーション共通オブジェクト](#)
 - [メタデータ・リンク](#)

- [データリンク・アプリケーション共通オブジェクト](#)
 - [拡張データリンク・アプリケーション・オブジェクト](#)
 - [アプリケーション・メンテナンス](#)
 - [アプリケーション・メンテナンスについて](#)
 - [アプリケーションのインストール](#)
 - [アプリケーションのアップグレード](#)
 - [アプリケーションのアップグレードの動作](#)
 - [異なるバージョンのアプリケーション](#)
 - [アプリケーション・パッチ](#)
 - [既存のアプリケーションの移行](#)
 - [暗黙的に作成されるアプリケーション](#)
 - [アプリケーションの同期化](#)
 - [1つのアプリケーションの同期](#)
 - [複数のアプリケーションの同期](#)
 - [コンテナ・マップ](#)
 - [CDBのサービスの概要](#)
 - [CDBのサービス作成](#)
 - [CDBのデフォルト・サービス](#)
 - [CDBの非デフォルト・サービス](#)
 - [CDBのコンテナへの接続](#)
 - [CDBの表領域およびデータベース・ファイルの概要](#)
 - [CDBの可用性の概要](#)
 - [CDBのバックアップおよびリカバリの概要](#)
 - [CDBでのフラッシュバックPDBの概要](#)
 - [CDBでのOracle Resource Managerの概要](#)
- [第II部 マルチテナント環境の作成および構成](#)
 - [3 マルチテナント環境の構成と管理の概要](#)
 - [マルチテナント環境の構成と管理について](#)
 - [共通ユーザーとローカル・ユーザー](#)
 - [CDB管理とPDB管理の業務の分離](#)
 - [マルチテナント環境の前提条件](#)
 - [マルチテナント環境のタスクおよびツール](#)
 - [マルチテナント環境のタスク](#)
 - [マルチテナント環境のツール](#)
 - [4 CDBの作成および構成](#)
 - [CDBの作成について](#)
 - [CDB作成計画](#)
 - [CDBの構成方法の決定](#)
 - [PDBの計画](#)
 - [物理レイアウトの計画](#)
 - [初期化パラメータの管理方法の学習](#)
 - [文字セットの選択](#)
 - [サポートするタイム・ゾーンの決定](#)
 - [データベース・ブロック・サイズおよびREDOログ・ブロック・サイズの選択](#)
 - [SYSTEMおよびSYSAUX表領域の作成](#)

- [一時表領域の計画](#)
 - [UNDOモードの選択](#)
 - [アプリケーションのサービスの計画](#)
 - [CDBの起動と停止方法の学習](#)
 - [Oracle RACの計画](#)
 - [CDB作成の前提条件](#)
 - [CDBの作成](#)
 - [DBCAを使用したCDB作成について](#)
 - [SQL文を使用したCDB作成について](#)
 - [PDBの有効化について](#)
 - [CDBルートおよびPDB\\$SEEDのファイルの名前と場所について](#)
 - [SEED FILE_NAME_CONVERT句](#)
 - [Oracle Managed Files](#)
 - [PDB_FILE_NAME_CONVERT初期化パラメータ](#)
 - [PDB\\$SEEDのデータファイル属性について](#)
 - [CDBのUNDOモードについて](#)
 - [CREATE DATABASE文を使用したCDBの作成](#)
 - [CREATE DATABASE文を使用したCDBの作成: 例](#)
 - [Oracle Managed Filesを使用しないCDBの作成](#)
 - [Oracle Managed Filesを使用したCDBの作成: 例](#)
 - [CDB用EM Expressの構成](#)
 - [CDBの作成後](#)
- [第III部 PDBおよびアプリケーション・コンテナの作成および削除](#)
 - [5 PDB作成の概要](#)
 - [PDBを作成する手法](#)
 - [現在のコンテナおよびPDB作成](#)
 - [非CDBからPDBを作成するためのオプション](#)
 - [PDB記憶域](#)
 - [ストレージ制限](#)
 - [デフォルト表領域](#)
 - [ユーザー表領域](#)
 - [PDBファイルの場所](#)
 - [FILE_NAME_CONVERT句](#)
 - [CREATE_FILE_DEST句](#)
 - [PDBファイルの場所に対する制限](#)
 - [サービス名変換](#)
 - [PDB作成のための句の概要](#)
 - [PDB作成の一般的な前提条件](#)
 - [6 PDBを最初から作成](#)
 - [PDBを最初から作成する方法について](#)
 - [PDBの作成](#)
 - [PDBの作成: 例](#)
 - [句を使用しないPDBの作成: 例](#)
 - [PDBの作成、およびPDB管理者へのOracleの事前定義されたロールの付与: 例](#)
 - [複数の句を使用したPDBの作成: 例](#)

- 7 PDBまたは非CDBのクローニング
 - PDBまたは非CDBのクローニングについて
 - クローニングの仕組み
 - PDBクローニングのユーザー・インタフェース
 - ローカルPDBのクローニング
 - ローカルPDBのクローニングについて
 - ローカルPDBのクローニング: 基本的なステップ
 - ローカルPDBのクローニング後
 - ローカルPDBのクローニング: 例
 - 句を使用しないローカルPDBのクローニング: 例
 - DBCAを使用したローカルPDBのクローニング: 例
 - PATH_PREFIX句を使用したローカルPDBのクローニング: 例
 - STORAGE句を使用したローカルPDBのクローニング: 例
 - NO DATA句を使用したローカルPDBのクローニング: 例
 - リモートPDBのクローニング
 - リモートPDBのクローニングについて
 - リモートPDBのクローニング: 基本的なステップ
 - リモートPDBのクローニング後
 - リモートPDBのクローニング: 例
 - 句を使用しないリモートPDBのクローニング: 例
 - DBCAを使用したリモートPDBのクローニング: 例
 - 非CDBのクローニング
 - 非CDBのクローニングについて
 - 非CDBのクローニング: 基本的なステップ
 - リモート非CDBのクローニング: 例
 - リフレッシュ可能なクローンPDBについて
 - リフレッシュ可能なクローンPDBの目的
 - 自動および手動リフレッシュ・モード
 - リフレッシュ可能なクローンPDBの要件
 - リフレッシュ可能なクローンPDBの作成: シナリオ
 - PDBスナップショットからのPDBのクローニング
 - PDBスナップショットからのPDBのクローニングについて
 - PDBスナップショットからのPDBのクローニング: シナリオ
 - スナップショット・コピーPDBの作成およびマテリアライズ
 - スナップショット・コピーPDBについて
 - スナップショットのコピーPDBの記憶域要件
 - スナップショット・コピーPDBの制限
 - スナップショット・コピーPDBの作成: シナリオ
 - スナップショット・コピーPDBのマテリアライズ
 - 分割ミラーによるクローンPDB の作成
- 8 PDBの再配置
 - PDBの再配置について
 - PDBの再配置の目的
 - PDB再配置の仕組み
 - PDBの再配置または停止時のサーバー・セッションの排出

- [PDBの再配置のステージ](#)
 - [共通のリリスナー・ネットワークでのPDBの再配置](#)
 - [独立したリスナー・ネットワークでのPDBの再配置](#)
 - [PDBの再配置のユーザー・インタフェース](#)
 - [CREATE PLUGGABLE DATABASEを使用したPDBの再配置](#)
 - [PDBの再配置: 例](#)
 - [リモートCDBからのPDBの再配置](#)
 - [DBCAを使用したPDBの再配置: 例](#)
- [9 切断されたPDBの接続](#)
 - [PDBのプラグイン操作について](#)
 - [XMLファイルとアーカイブ・ファイルについて](#)
 - [切断されたPDBをプラグインする場合のソース・ファイルの場所](#)
 - [SOURCE_FILE_NAME_CONVERT句](#)
 - [SOURCE_FILE_DIRECTORY句](#)
 - [PDBとしての非CDBの適用について](#)
 - [切断されたPDBの接続](#)
 - [PDBとしての非CDBの適用](#)
 - [切断されたPDBの接続後](#)
 - [切断されたPDBの接続: 例](#)
- [10 プロキシPDBとしてのPDBの作成](#)
 - [プロキシPDBの作成について](#)
 - [プロキシPDBおよびSQL文](#)
 - [プロキシPDBおよびデータベース・リンク](#)
 - [プロキシPDBおよび認証](#)
 - [プロキシPDBおよびリスナー](#)
 - [HOST句](#)
 - [PORT句](#)
 - [プロキシPDBの作成](#)
- [11 PDBの削除](#)
 - [CDBからのPDBの切断](#)
 - [PDBの切断について](#)
 - [PDBの切断](#)
 - [PDBの削除](#)
- [12 アプリケーション・コンテナおよびシードの作成と削除](#)
 - [アプリケーション・コンテナの作成および削除](#)
 - [アプリケーション・コンテナの作成](#)
 - [アプリケーション・コンテナの作成について](#)
 - [アプリケーション・コンテナの準備](#)
 - [アプリケーション・コンテナの作成](#)
 - [CDBからのアプリケーション・コンテナの切断](#)
 - [アプリケーション・コンテナの切断について](#)
 - [アプリケーション・コンテナの切断](#)
 - [アプリケーション・コンテナの削除](#)
 - [アプリケーション・シードの作成および削除](#)
 - [アプリケーション・シードの作成](#)

- [アプリケーション・シードの作成について](#)
 - [アプリケーション・シードの準備](#)
 - [アプリケーション・シードの作成](#)
 - [アプリケーション・コンテナからのアプリケーション・シードの切断](#)
 - [アプリケーション・シードの切断について](#)
 - [アプリケーション・シードの切断](#)
 - [アプリケーション・シードの削除](#)
 - [アプリケーションPDBの作成](#)
- [第IV部 マルチテナント環境の管理](#)
 - [13 CDBの管理](#)
 - [CDB管理について](#)
 - [現在のコンテナについて](#)
 - [CDBにおける管理タスクについて](#)
 - [CDBにおける管理性機能の使用について](#)
 - [CDB内表領域の管理について](#)
 - [CDB内永続表領域の管理について](#)
 - [CDB内の一時表領域の管理について](#)
 - [CDBにおけるデータベース・オブジェクトの管理について](#)
 - [PDBのフラッシュ・バックについて](#)
 - [セキュリティを強化するためのPDBユーザーの制限について](#)
 - [PDBロックダウン・プロファイル](#)
 - [PDB_OS_CREDENTIAL初期化パラメータ](#)
 - [PATH_PREFIXおよびCREATE_FILE_DEST PDB作成句](#)
 - [Oracle RACを含むOracle Multitenantの概要](#)
 - [CDB内のコンテナへのアクセス](#)
 - [CDB内のコンテナへのアクセスについて](#)
 - [CDBのサービス](#)
 - [CDBでのセッション制限](#)
 - [マルチテナント環境のユーザー名](#)
 - [管理ユーザーのパスワード・ファイルに対するマルチテナント・オプションの影響](#)
 - [CDB内のコンテナへのアクセス](#)
 - [SQL*Plus CONNECTコマンドを使用したコンテナへの接続](#)
 - [SQL*Plus CONNECTコマンドを使用したルートへの接続](#)
 - [SQL*Plus CONNECTコマンドを使用したPDBへの接続](#)
 - [ALTER SESSION文を使用したコンテナへの切替え](#)
 - [システム・レベルでのCDBの変更](#)
 - [CDBのシステム・レベルの変更について](#)
 - [ALTER SYSTEMを使用したCDBの変更](#)
 - [CDBルートへの接続時のコンテナの変更](#)
 - [CDBルートへの接続時のコンテナの変更について](#)
 - [ALTER DATABASEを使用したCDB全体の変更](#)
 - [ALTER DATABASEを使用したCDBのUNDOモードの設定](#)
 - [CDBのUNDOモードについて](#)
 - [ローカルUNDOモードについて](#)
 - [共有UNDOモードについて](#)

- [ローカルUNDOモードを使用するようにCDBを構成する方法](#)
 - [共有UNDOモードを使用するようにCDBを構成する方法](#)
 - [ALTER DATABASEを使用したCDBルートの変更](#)
 - [別のコンテナでのSQLの実行](#)
 - [CDB内のコンテナに対するDML文の発行](#)
 - [CDB内のコンテナに対するDML文の発行について](#)
 - [CDBにおけるDML文のデフォルト・コンテナの指定](#)
 - [CDBでのDDL文の実行](#)
 - [CDBにおけるDDL文の実行について](#)
 - [現在のコンテナにおけるDDL文の実行](#)
 - [CDB内のすべてのコンテナにおけるDDL文の実行](#)
 - [CDBでのオラクル社が提供するSQLスクリプトの実行](#)
 - [CDBでのオラクル社が提供するSQLスクリプトの実行について](#)
 - [catcon.plの構文およびパラメータ](#)
 - [catcon.plスクリプトの実行](#)
 - [DBMS_SQLパッケージを使用したコンテナのコードの実行](#)
 - [CDBインスタンスの停止](#)
- [14 CDBフリートの管理](#)
 - [CDBフリートについて](#)
 - [CDBフリートの目的](#)
 - [CDBフリートへのリードCDBの設定](#)
 - [CDBフリート・メンバーの指定](#)
- [15 PDBの管理](#)
 - [PDB管理について](#)
 - [PDBと非CDBに共通するタスク](#)
 - [CDBに固有のタスク](#)
 - [PDBへの接続の管理](#)
 - [PDBへの接続](#)
 - [PDBのサービスの管理](#)
 - [PDBのサービスについて](#)
 - [PDBプロパティ](#)
 - [デフォルトおよびユーザー定義のサービス](#)
 - [サービスを管理するためのツール](#)
 - [SRVCTLおよびDBMS_SERVICEを使用したPDBのサービスの管理](#)
 - [参照先PDBのリスナー設定の変更](#)
 - [参照PDBのリスナー・ホスト名の変更](#)
 - [参照先PDBのリスナー・ポート番号の変更](#)
 - [システム・レベルでのPDBの変更](#)
 - [PDBのシステム・レベルの変更について](#)
 - [ALTER SYSTEMを使用したPDBの変更](#)
 - [データベース・レベルでのPDBの変更](#)
 - [PDBのデータベース・レベルの変更について](#)
 - [記憶域の句](#)
 - [ロギングおよびリカバリの句](#)
 - [その他の句](#)

- [ALTER PLUGGABLE DATABASE文を使用したPDBの変更](#)
- [PDBのグローバル・データベース名の変更](#)
- [リフレッシュ可能なクローンPDBの管理](#)
 - [PDBのリフレッシュ](#)
 - [リフレッシュ可能なクローンPDBのスイッチオーバー](#)
- [PDBのオープン・モードの変更](#)
 - [PDBのオープン・モードについて](#)
 - [PDBのオープン状態を変更する句](#)
 - [OPEN句とCLOSE句](#)
 - [SERVICES句](#)
 - [INSTANCES句](#)
 - [RELOCATE句](#)
 - [ALTER PLUGGABLE DATABASEを使用したPDBのオープン・モードの変更](#)
 - [CDB再起動時のPDBのオープン・モードの保持または破棄](#)
- [STARTUPおよびSHUTDOWNを使用したPDBのオープン・モードの変更](#)
 - [SQL*PlusのSTARTUPコマンドを使用したPDBのオープン・モードの変更について](#)
 - [STARTUPコマンドを使用したPDBの起動](#)
 - [SQL*PlusのSTARTUPコマンドを使用したPDBのオープン・モードの変更](#)
 - [SHUTDOWNコマンドを使用したPDBの停止](#)
- [Oracle RACでのPDBの起動および停止](#)
- [16 PDBスナップショット・カルーセルの管理](#)
 - [PDBスナップショット・カルーセルについて](#)
 - [PDBスナップショット・カルーセルの目的](#)
 - [PDBスナップショット・カルーセルの仕組み](#)
 - [PDBスナップショットの内容](#)
 - [PDBスナップショット・カルーセル の内容](#)
 - [PDBスナップショット・カルーセル のユーザー・インタフェース](#)
 - [PDBスナップショット・カルーセル 内のスナップショットの最大数の設定](#)
 - [自動PDBスナップショットの構成](#)
 - [PDBスナップショットの手動作成](#)
 - [PDBスナップショットの削除](#)
 - [PDBスナップショットのメタデータの表示](#)
- [17 アプリケーション・コンテナの管理](#)
 - [アプリケーション・コンテナの管理について](#)
 - [アプリケーション・ルートの変更について](#)
 - [アプリケーション・コンテナ内のアプリケーションの管理](#)
 - [アプリケーション管理について](#)
 - [アプリケーション・メンテナンスの基本的なステップ](#)
 - [アプリケーションのバージョン](#)
 - [アプリケーション・モジュール名およびサービス名](#)
 - [アプリケーション・コンテナへのアプリケーションのインストール](#)
 - [アプリケーション・コンテナ内のアプリケーションのインストールについて](#)
 - [自動伝播によるアプリケーション・コンテナ内のアプリケーションのインストール](#)
 - [アプリケーション・コンテナ内のアプリケーションのアップグレード](#)
 - [アプリケーション・コンテナ内のアプリケーションのアップグレードについて](#)

- [アプリケーション・アップグレードの目的](#)
 - [アプリケーションのアップグレードの動作](#)
 - [アプリケーション・アップグレードのためのユーザー・インタフェース](#)
 - [アプリケーション・コンテナ内のアプリケーションのアップグレード](#)
 - [アプリケーション・コンテナ内のアプリケーションへのパッチ適用](#)
 - [アプリケーション・コンテナ内のアプリケーションへのパッチ適用について](#)
 - [自動伝播によるアプリケーション・コンテナ内のアプリケーションへのパッチ適用](#)
 - [アプリケーション・コンテナへの既存アプリケーションの移行](#)
 - [アプリケーション・コンテナへの既存アプリケーションの移行について](#)
 - [既存のPDBを使用したアプリケーション・ルートの作成](#)
 - [既存のPDBを使用したアプリケーションPDBの作成](#)
 - [アプリケーションPDB内のアプリケーションの同期](#)
 - [アプリケーション・ルート・レプリカのプロキシPDBとの同期](#)
 - [アプリケーション・ルート・レプリカのプロキシPDBとの同期について](#)
 - [アプリケーション・ルート・レプリカを参照するプロキシPDBの作成](#)
 - [アプリケーションの互換バージョンの設定](#)
 - [アプリケーションのインストール、アップグレードおよびパッチ適用の操作中の一括挿入の実行](#)
 - [アプリケーション・コンテナからのアプリケーションのアンインストール](#)
 - [アプリケーション・コンテナからのアプリケーションのアンインストールについて](#)
 - [アプリケーション・コンテナからのアプリケーションのアンインストール](#)
- [アプリケーション共通オブジェクトの管理](#)
 - [アプリケーション共通オブジェクトについて](#)
 - [アプリケーション共通オブジェクトの作成](#)
 - [メタデータリンク・アプリケーション共通オブジェクトについて](#)
 - [拡張データリンク・アプリケーション共通オブジェクトについて](#)
 - [拡張データリンク・アプリケーション共通オブジェクトについて](#)
 - [アプリケーション共通オブジェクトの制限](#)
 - [アプリケーション共通オブジェクトの作成](#)
 - [アプリケーション共通オブジェクトに対するDML文の発行](#)
 - [メタデータリンク共通オブジェクトに対するDMLの発行](#)
 - [CONTAINERS句を使用した問合せ](#)
 - [デフォルトのコンテナまたはDMLの設定](#)
 - [データリンク共通オブジェクトに対するDMLの発行](#)
 - [DDL文を使用したアプリケーション共通オブジェクトの変更](#)
- [アプリケーション・コンテナ内のコンテナに対するDML文の発行](#)
 - [アプリケーション・コンテナ内のコンテナに対するDML文の発行について](#)
 - [アプリケーション・コンテナにおけるDML文のデフォルト・コンテナの指定](#)
- [コンテナ・マップを使用したPDBでのパーティション化](#)
 - [コンテナ・マップについて](#)
 - [マップ・オブジェクト](#)
 - [リスト・パーティション化されたコンテナ・マップ: 例](#)
 - [レンジ・パーティション化されたコンテナ・マップ: 例](#)
 - [コンテナ・マップの作成](#)
- [18 マルチテナント環境のセキュリティの管理](#)
 - [共通およびローカルに付与される権限の管理](#)

- [Oracleマルチテナント・オプションが権限に影響を与えるしくみ](#)
- [共通およびローカルに付与される権限について](#)
- [共通に付与されるシステム権限の使用法](#)
- [共通に付与されるオブジェクト権限の使用法](#)
- [PDBへのアクセス権限の付与または取消し](#)
- [例: マルチテナント環境での権限の付与](#)
- [共通ユーザーによるCONTAINER_DATAオブジェクトの情報の表示](#)
 - [ルートに接続中のルート、CDBおよびPDBに関するデータの表示](#)
 - [特定のPDBのデータを問い合わせる共通ユーザーの有効化](#)
- [共通ロールおよびローカル・ロールの管理](#)
 - [共通ロールおよびローカル・ロールの管理について](#)
 - [共通ロールの使用法](#)
 - [マルチテナント環境でPUBLICロールが機能するしくみ](#)
 - [共通ロールの作成、変更または削除に必要な権限](#)
 - [共通ロールの作成の規則](#)
 - [共通ロールの作成](#)
 - [ローカル・ロールの作成の規則](#)
 - [ローカル・ロールの作成](#)
 - [共通ユーザーとローカル・ユーザーに対するロールの付与と取消し](#)
- [PDBロックダウン・プロファイルを使用したPDBでの操作の制限](#)
 - [PDBロックダウン・プロファイルについて](#)
 - [デフォルトのPDBロックダウン・プロファイル](#)
 - [PDBロックダウン・プロファイルの作成](#)
 - [PDBロックダウン・プロファイルの有効化または無効化](#)
 - [PDBロックダウン・プロファイルの削除](#)
- [PDBのオペレーティング・システム・ユーザーの構成](#)
 - [PDBのオペレーティング・システム・ユーザーの構成について](#)
 - [PDBのオペレーティング・システム・ユーザーの構成](#)
 - [PDBでのデフォルトの資格証明の設定](#)
- [マルチテナント環境でのアプリケーション・コンテキストの使用](#)
 - [アプリケーション・コンテキストとは](#)
 - [マルチテナント環境でのアプリケーション・コンテキスト](#)
- [マルチテナント環境でのOracle Virtual Private Databaseの使用](#)
 - [Oracle Virtual Private Database](#)
 - [マルチテナント環境でのOracle Virtual Private Database](#)
- [マルチテナント環境でのTransport Layer Securityの使用](#)
- [マルチテナント環境におけるOracle Data Redaction](#)
- [マルチテナント環境での監査の概要](#)
 - [マルチテナント環境での統合監査](#)
 - [例: マルチテナント環境でのDBAロールの監査](#)
 - [マルチテナント環境での統合監査ポリシーまたはAUDIT設定](#)
 - [ローカル、CDB共通およびアプリケーション共通監査ポリシーについて](#)
 - [マルチテナント環境での従来の監査](#)
 - [ローカル統合監査ポリシーまたは共通統合監査ポリシーの構成](#)
 - [例: ローカル統合監査ポリシー](#)

- [例: CDB共通統合監査ポリシー](#)
 - [例: アプリケーション共通統合監査ポリシー](#)
 - [監査証跡でのローカルまたは共通監査ポリシーまたは設定の表示方法](#)
- [マルチテナント環境でのファイングレイン監査](#)
- [19 CDBおよびPDBの監視](#)
 - [ビュー内のCDBおよびコンテナの情報について](#)
 - [現在のコンテナがCDBルートではない場合の情報の表示について](#)
 - [現在のコンテナがCDBルートである場合の情報の表示について](#)
 - [CDBのビュー](#)
 - [データベースがCDBであるかどうかの判断](#)
 - [CDB内のコンテナに関する情報の表示](#)
 - [PDBに関する情報の表示](#)
 - [各PDBのオープン・モードの表示](#)
 - [コンテナ・データ・オブジェクトの問合せ](#)
 - [CONTAINERS句を使用したコンテナ間の問合せ](#)
 - [CONTAINERS句を使用したコンテナ間の問合せについて](#)
 - [すべてのコンテナ間でのユーザーが作成した表およびビューの問合せ](#)
 - [アプリケーションPDB間のアプリケーション共通オブジェクトの問合せ](#)
 - [現在のコンテナのIDまたは名前の判別](#)
 - [PDBの変更可能な初期化パラメータをリスト](#)
 - [PDBの履歴の表示](#)
 - [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)
 - [アプリケーションに関する情報の表示](#)
 - [アプリケーション・ステータスに関する情報の表示](#)
 - [アプリケーション文に関する情報の表示](#)
 - [アプリケーション・バージョンに関する情報の表示](#)
 - [アプリケーション・パッチに関する情報の表示](#)
 - [アプリケーション・エラーに関する情報の表示](#)
 - [アプリケーション・コンテナ内の共有データベース・オブジェクトのリスト](#)
 - [アプリケーション・コンテナ内の拡張データリンク・オブジェクトのリスト](#)
- [第V部 マルチテナント環境でのOracle機能の使用](#)
 - [20 CDBとPDBのバックアップおよびリカバリ](#)
 - [CDBとPDBのバックアップおよびリカバリの概要](#)
 - [CDBのバックアップおよび完全リカバリ](#)
 - [PDBのバックアップおよび完全リカバリ](#)
 - [マルチテナント環境でのPoint-in-Timeリカバリ](#)
 - [マルチテナント環境でのフラッシュバック・データベース](#)
 - [21 マルチテナント環境でのデータベース・ユーティリティの使用](#)
 - [CDBでのデータのインポートおよびエクスポート](#)
 - [マルチテナント環境でのデータ・ポンプの使用について](#)
 - [データ・ポンプを使用したCDBへのデータの移動](#)
 - [データ・ポンプの使用によるCDB内または間のPDBの移動](#)
 - [CDBのLogMinerの使用](#)
 - [CDBでのLogMiner V\\$ビューおよびDBAビュー](#)
 - [CDBでのV\\$LOGMNR_CONTENTSビュー](#)

- [CDBでのサプリメンタル・ロギングの有効化](#)
 - [CDBでのフラット・ファイル・ディクショナリの使用](#)
- [DBNEWIDのCDBおよびPDBに関する考慮事項](#)
- [22 PDBに対するOracle Resource Managerの使用](#)
 - [マルチテナント環境でのOracle Resource Managerの概要](#)
 - [マルチテナント環境でのリソース管理の目的](#)
 - [リソース・プラン・ディレクティブの概要](#)
 - [PDBパフォーマンス・プロファイル](#)
 - [リソース・プラン・ディレクティブ](#)
 - [バックグラウンド・タスクと管理タスクおよびコンシューマ・グループ](#)
 - [PDBレベルのリソースの初期化パラメータ](#)
 - [PDBのCPU関連の初期化パラメータ](#)
 - [PDBのメモリー関連の初期化パラメータ](#)
 - [PDBのセッション関連の初期化パラメータ](#)
 - [PDBのI/O関連の初期化パラメータ](#)
 - [CDBリソース・プランの管理](#)
 - [CDBリソース・プランについて](#)
 - [PDBへのリソース割当ての共有](#)
 - [PDBの使用率制限](#)
 - [PDBのデフォルト・ディレクティブ](#)
 - [PDBを管理するためのCDBリソース・プランの作成](#)
 - [PDBを管理するためのCDBリソース・プランの作成: 使用例](#)
 - [PDBパフォーマンス・プロファイルを使用するCDBリソース・プランの作成](#)
 - [PDBパフォーマンス・プロファイルに対するCDBリソース・プランの作成: 使用例](#)
 - [CDBリソース・プランの有効化](#)
 - [CDBリソース・プランの変更](#)
 - [CDBリソース・プランの更新](#)
 - [PDBのCDBリソース・プラン・ディレクティブの管理](#)
 - [PDBの新規CDBリソース・プラン・ディレクティブの作成](#)
 - [PDBのCDBリソース・プラン・ディレクティブの更新](#)
 - [PDBのCDBリソース・プラン・ディレクティブの削除](#)
 - [PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの管理](#)
 - [PDBパフォーマンス・プロファイルの新規CDBリソース・プラン・ディレクティブの作成](#)
 - [PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの更新](#)
 - [PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの削除](#)
 - [CDBリソース・プランにおけるPDBのデフォルト・ディレクティブの更新](#)
 - [CDBリソース・プランにおけるメンテナンス・タスクのデフォルト・ディレクティブの更新](#)
 - [CDBリソース・プランの削除](#)
 - [CDBリソース・プランの無効化](#)
 - [CDBのプランおよびディレクティブに関する情報の表示](#)
 - [CDBリソース・プランの表示](#)
 - [CDBリソース・プラン・ディレクティブの表示](#)

- [PDBリソース・プランの管理](#)
 - [PDBリソース・プランについて](#)
 - [PDBリソース・プランを作成する場合のCDBリソース・プランの要件](#)
 - [PDBリソース・プラン: 例](#)
 - [PDBリソース・プランの作成](#)
 - [PDBリソース・プランの有効化](#)
 - [PDBリソース・プランの変更](#)
 - [PDBリソース・プランの無効化](#)
- [Oracle Database Resource Managerで管理されるPDBの監視](#)
 - [PDBに対するリソース・マネージャのビューについて](#)
 - [PDBのCPU使用率の監視](#)
 - [PDBの паралел実行の監視](#)
 - [PDBによって生成されるI/Oの監視](#)
 - [PDBのメモリー使用状況の監視](#)
- [23 CDBでのOracle Schedulerの使用](#)
 - [CDBにおけるDBMS_SCHEDULERの起動](#)
 - [CDBにおけるジョブ・コーディネータおよびスレーブ・プロセス](#)
 - [DBMS_JOBおよびDBMS_SCHEDULER](#)
 - [PDBをクローズするプロセス](#)
 - [新規および変更されたCDBビュー](#)
- [24 CDBでのOracle Database Vaultの使用](#)
 - [Oracle Database Vaultについて](#)
 - [マルチテナント環境におけるOracle Database Vaultの動作について](#)
 - [Database Vaultが構成および有効化されていることの確認](#)
 - [マルチテナント環境でのOracle DatabaseへのOracle Database Vaultの登録](#)
 - [CDBルートへのDatabase Vaultの登録](#)
 - [個別PDBを管理するためのDatabase Vault共通ユーザーの登録](#)
 - [個別PDBを管理するためのDatabase Vaultローカル・ユーザーの登録](#)
 - [Database Vault対応PDBへの接続](#)
 - [マルチテナント環境でのOracle Database Vaultの手動インストール](#)
 - [レルムの構成](#)
 - [レルムの概要](#)
 - [レルムについて](#)
 - [マルチテナント環境におけるレルム](#)
 - [マルチテナント環境におけるレルム認可](#)
 - [マルチテナント環境におけるルール・セットとルール](#)
 - [マルチテナント環境におけるコマンド・ルール](#)
 - [マルチテナント環境におけるOracle Database Vaultポリシー](#)
 - [Database Vault操作の制御を使用したローカルPDBデータへのマルチテナント共通ユーザー・アクセスの制限](#)
 - [Database Vault操作の制御の使用について](#)
 - [Database Vault操作の制御の有効化](#)
 - [例外リストへの共通ユーザーおよびパッケージの追加](#)
 - [例外リストからの共通ユーザーおよびパッケージの削除](#)
 - [Database Vault操作の制御の無効化](#)

- [スタンドアロンのOracle DatabaseをPDBに変換してCDBにプラグイン](#)
- [25 CDBでのXStreamの使用](#)
 - [XStreamについて](#)
 - [システム作成ルールとマルチテナント環境](#)
 - [CDBにおけるシステム生成ルールとXStream Out](#)
 - [CDBのシステム作成ルールとXStream In](#)
 - [XStream Outとマルチテナント環境](#)
 - [CDBでのXStream Outの構成](#)
 - [CDBでのローカル取得を使用したXStream Outの構成](#)
 - [CDBでのダウンストリーム取得を使用したXStream Outの構成](#)
 - [XStream Inとマルチテナント環境](#)
- [用語集](#)
- [索引](#)

はじめに

このマニュアルでは、Oracle Databaseを作成、構成および管理する方法について説明します。

- [対象読者](#)
- [ドキュメントのアクセシビリティについて](#)
- [関連ドキュメント](#)
- [表記規則](#)

対象読者

このマニュアルは、次のタスクを実行するデータベース管理者を対象にしています。

- 1つ以上のOracle Databaseの作成および構成
- Oracle Databaseの監視およびチューニング
- Oracle Databaseに対する日常的なメンテナンス操作の監視
- 表、索引、ビューなどのスキーマ・オブジェクトの作成およびメンテナンス
- システム・ジョブおよびユーザー・ジョブのスケジュール
- 問題の診断、修復およびレポート

このマニュアルを使用するにあたって、リレーショナル・データベースの概念を理解しておく必要があります。また、Oracle Databaseを実行するオペレーティング・システム環境についても知っておく必要があります。

親トピック: [はじめに](#)

ドキュメントのアクセシビリティについて

Oracleのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWebサイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートを購入したオラクル社のお客様は、My Oracle Supportを介して電子的なサポートにアクセスできます。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

親トピック: [はじめに](#)

関連ドキュメント

詳細は、次のOracleドキュメントを参照してください。

- [Oracle Database概要](#)
- [Oracle Database管理者ガイド](#)
- [Oracle Database 2日でデータベース管理者](#)
- [Oracle Database SQL言語リファレンス](#)
- [Oracle Database!リファレンス](#)

- [Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)
- [Oracle Automatic Storage Management管理者ガイド](#)
- [Oracle Database VLDBおよびパーティショニング・ガイド](#)
- [Oracle Databaseエラー・メッセージ・リファレンス](#)
- [Oracle Database Net Services管理者ガイド](#)
- [Oracle Databaseバックアップおよびリカバリ・アドバンスト・ユーザーズ・ガイド](#)
- [Oracle Databaseパフォーマンス・チューニング・ガイド](#)
- [Oracle Database SQLチューニング・ガイド](#)
- [Oracle Database開発ガイド](#)
- [Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)
- [SQL*Plusユーザーズ・ガイドおよびリファレンス](#)

このマニュアルの多くの例では、サンプル・スキーマを使用しています。これらのスキーマの詳細は、『[Oracle Databaseサンプル・スキーマ](#)』を参照してください。

親トピック: [はじめに](#)

表記規則

このマニュアルでは、次の表記規則を使用します。

規則	意味
太字	太字は、操作に関連する Graphical User Interface 要素、または本文中で定義されている用語および用語集に記載されている用語を示します。
イタリック体	イタリックは、ユーザーが特定の値を指定するプレースホルダ変数を示します。
固定幅フォント	固定幅フォントは、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、または入力するテキストを示します。

親トピック: [はじめに](#)

Oracle Multitenant管理者ガイドのこのリリースでの変更点

このドキュメントは、Oracle Databaseの最新リリース用に変更されています。

- [Oracle Databaseリリース19c、バージョン19.1での変更点](#)
Oracle Databaseリリース19c、バージョン19.1向け『Oracle Multitenant管理者ガイド』には、次のような変更点があります。
- [Oracle Databaseリリース18c、バージョン18.1での変更点](#)

Oracle Databaseリリース19c、バージョン19.1での変更点

Oracle Databaseリリース19c、バージョン19.1向け『Oracle Multitenant管理者ガイド』には、次のような変更点があります。

- [新機能](#)
このリリースの主な新機能は次のとおりです。

親トピック: [Oracle Multitenant管理者ガイドのこのリリースでの変更点](#)

新機能

今回のリリースの主な新機能は次のとおりです。

- PDBでのワークロードの取得およびリプレイ
ローカル・ユーザーは、PDBレベルでワークロードを取得、リプレイおよびレポートできます。
[「CDBにおける管理性機能の使用について」](#)および[『Oracle Databaseテスト・ガイド』](#)を参照してください。
- PDBのADDM分析
ADDMを使用して、PDB内で取得したAWRスナップショットを通じてPDB内に格納されているAWRデータを分析できます。非CDB、CDBルート、またはPDBのAWR記憶域にインポートされたPDBのAWRデータも分析できます。PDBの自動ADDMはデフォルトで無効にされています。これは、自動AWRスナップショットを有効にすることで、PDBに対して有効にできます。
[「CDBにおける管理性機能の使用について」](#)および[『Oracle Databaseパフォーマンス・チューニング・ガイド』](#)を参照してください。
- インフラストラクチャ・データベース管理者向けのDatabase Vault操作の制御
Oracle Database Vaultを使用して、共通ユーザー（インフラストラクチャDBAなど）がPDBのローカル・データにアクセスするのをブロックできます。したがって、共通ユーザーはローカル・データへのアクセスをブロックされます。Oracle Database Vaultを使用すると、ビジネス・アプリケーションの機密データを格納でき、重要な顧客データにアクセスすることなくデータベース・インフラストラクチャを管理できます。
[「Database Vault操作制御を使用したローカルPDBデータへのマルチテナント共通ユーザー・アクセスの制限」](#)を参照してください。
- 同じCDBでの複数のPDBシャードのサポート
CDBには、複数のPDBをシャード・カタログ・データベースとして含めることができます。また、CDBには異なるシャード・

データベース(SDB)からシャードPDBを含めることもできます。各シャードPDBは、個別のカatalog・データベースで管理されます。

[「Oracle Shardingの使用」](#)を参照してください。

- PDBの自動再配置

Oracle Grid Infrastructureでは、Oracle Fleet Patching and Provisioningを使用して、あるCDBから別のCDBへのPDBの再配置を自動化できます。自動再配置により、個々のPDBへのパッチ適用が迅速化され、そのパッチ適用によって他のPDBが変更される可能性がなくなります。

[Oracle Clusterware管理およびデプロイメントに関するガイド](#)を参照してください。

- DBCAを使用したリモートPDBのクローニング

サイレント・モードでDBCAを使用して、リモートPDBをクローニングできます。

[「リモートPDBのクローニングについて」](#)を参照してください。

- リモートPDBの再配置

Database Configuration Assistant(DBCA)を使用して、PDBをリモートCDBからローカルCDBに再配置できます。

[「DBCAを使用したPDBの再配置: 例」](#)を参照してください。

- データ・ポンプ・インポートに対するクラウドのオブジェクト・ストアのサポート

`impdb`の`credential`パラメータに、オブジェクト・ストア・バケットへのアクセスに必要なユーザー名とパスワードを含む資格証明オブジェクトの名前を指定します。データベース・プロパティ`DEFAULT_CREDENTIAL`を使用して、デフォルトの資格証明を指定することもできます。

[「PDBでのデフォルトの資格証明の設定」](#)および[「データ・ポンプを使用したCDBへのデータの移動」](#)を参照してください。

関連項目:

各種エディションおよびサービスでサポートされる機能の詳細は、[『Oracle Databaseライセンス情報ユーザー・マニュアル』](#)を参照

親トピック: [Oracle Database 19cリリース19.1での変更点](#)

Oracle Databaseリリース18c、バージョン18.1での変更点

Oracle Databaseリリース18c、バージョン18.1のOracle *Multitenant*管理者ガイドには、次の変更点があります。

- [新機能](#)

親トピック: [Oracle Multitenant管理者ガイドのこのリリースでの変更点](#)

新機能

このリリースの新機能は次のとおりです。

- CDBフリート

CDBフリートは、1つの論理CDBとして管理できる様々なCDBの集合です。

[「CDBフリートの管理」](#)を参照してください。

- PDBスナップショット・カールセル

PDBスナップショットはPDBのPoint-in-Timeコピーです。スナップショットの作成中に、ソースPDBを読み取り専用または読み取り/書き込みでオープンできます。スナップショットは、CREATE PLUGGABLE DATABASE (またはALTER PLUGGABLE DATABASE)のSNAPSHOT句を使用して手動で作成するか、またはEVERY間隔句を使用して自動的に作成できます。PDBのスナップショットが有効になっている場合は、PDBの複数のスナップショット(Point-in-Timeコピー)を作成できます。スナップショットのライブラリは、PDBスナップショット・カールセルと呼ばれます。カールセル内のスナップショットに基づいて、新しいPDBを迅速にクローニングできます。この方法で、カールセル内の任意のスナップショットへのPoint-in-Timeリカバリを実行したり、スナップショットをクローニングしてPDBをすぐに作成できます。

[「PDBスナップショット・カールセルのユーザー・インタフェース」](#)および[「PDBスナップショット・カールセルの管理」](#)を参照してください。

- 論理パーティション化

コンテナ・マップを使用することにより、セッションで発行したSQL文が、SQL文に使用された条件の値に応じて適切なPDBにルーティングされるようになります。マップ表のパーティション化列は、メタデータリンク表の列と一致している必要はありません。たとえば、sales表がpdb_map_tblコンテナ・マップに対して有効になっていて、pdb_map_tblをパーティション化するために使用される列がsalesにない場合は、CONTAINERS (sales) 述語による問合せが、マップ表に指定されているPDBにそのままルーティングされます。

[「コンテナ・マップ」](#)を参照してください。

- リフレッシュ可能なPDBスイッチオーバー

リフレッシュ可能なクローンPDBは、ソースPDBと定期的に同期できる読み取り専用クローンです。ソースPDBをクローンに、クローンをソースに変換して、その役割を交代できます。この手法はロード・バランシングを行う場合に便利です。さらに、ソースPDBが失敗した場合はクローンPDBで操作を再開でき、CDBレベルのOracle Data Guardのフェイルオーバーのレンダリングが不要になります。

[「リフレッシュ可能なクローンPDBについて」](#)および[「リフレッシュ可能なクローンPDBのスイッチオーバー」](#)を参照してください。

- ロックダウン・プロファイルの拡張機能

アプリケーション・コンテナのロックダウン・プロファイルを作成、変更または削除できます。また、静的または動的な基本プロファイルに基づいてロックダウン・プロファイルを作成できます。

[「PDBロックダウン・プロファイルの概要」](#)、[「セキュリティを強化するためのPDBユーザーの制限について」](#)および[「PDBロックダウン・プロファイルを使用したPDBでの操作の制限」](#)を参照してください。

- DBCAの拡張機能

DBCAを使用して、ローカルPDBをクローニングまたはCDBを複製できます。複製はサイレント・モードでのみサポートされます。

[「DBCAを使用したCDB作成について」](#)および[「ローカルPDBのクローニングについて」](#)を参照してください。

- 非CDBおよび再配置されたPDBの使用可能なバックアップ

非CDBをPDBとしてクローニングする場合またはPDBを再配置する場合は、DBMS_PDB.EXPORTMANBACKUPプロシージャを使用して、RMANバックアップ・メタデータをPDBディクショナリにエクスポートできます。このメタデータにより、ソース非CDBまたはPDBのバックアップをターゲットPDBのリストアおよびリカバリに使用できるようになります。

[「PDB作成の一般的な前提条件」](#)を参照してください。

- 別のCDBへのPDBのRMAN複製

ソースCDBから読取り/書き込みでオープンされた既存のCDBに、PDBをクローニングできます。

[「PDBを作成する手法」](#)を参照してください。

- 計画メンテナンス中のセッションの再配置

アプリケーション・コンティニューイティでは、リクエスト境界のフェイルオーバーに適した場所でアプリケーションが接続テストを発行した場合は、計画メンテナンス中にデータベース・セッションを排出できます。再配置はアプリケーションに対して透過的です。この機能は、データベース・サービス・レベルおよびPDBレベルで呼び出されるすべてのメンテナンス操作(サービスの停止、サービスの再配置、PDBの再配置およびPDBの停止)に対して、デフォルトでオンになっています。

[「PDBのサービスの管理」](#)、[「PDB再配置の仕組み」](#)および[「Oracle Real Application Clusters管理およびデプロイメント・ガイド」](#)を参照してください。

- Oracle Data Guard環境でのPDBのコピー

プライマリ・データベースでリモート・クローンを実行したり、プライマリ・データベースでPDBを接続する場合は、スタンバイ・データベースの初期化パラメータを設定して、新しく作成されたPDBのデータファイルのコピーを自動化できます。

[「リモートPDBのクローニング: 基本的なステップ」](#)および[「切断されたPDBの接続」](#)を参照してください。

- PDBレベルでのパラレル文のキューイング

PARALLEL_SERVERS_TARGET初期化パラメータを使用して、PDB以外の場合と同様に、PDBに対するパラレル文のキューイングを構成できます。PDBレベルでは、デフォルトはPDBのCPU_COUNT設定に基づきます。CDBレベルでは、デフォルト値はPARALLEL_MAX_SERVERS初期化パラメータの値です。

[「PDBの使用率制限」](#)を参照してください。

- 分割ミラーによるクローンPDB

PDBがOracle ASMにある場合は、分割ミラー化手法を使用してPDBをクローニングできます。クローニングされたPDBは元のPDBに依存しません。主要なユースケースでは、Oracle ASM環境でテストおよび開発のPDBを迅速にプロビジョニングします。

[「分割ミラーによるクローンPDBの作成」](#)を参照してください。

親トピック: [Oracle Database 18cリリース18.1での変更点](#)

第I部 マルチテナント・アーキテクチャ

マルチテナント・アーキテクチャを使用すると、Oracle Databaseをマルチテナント・コンテナ・データベース(CDB)として機能させることができます。

- [マルチテナント・アーキテクチャの紹介](#)

Oracle Multitenantオプションを理解します。

- [マルチテナント・アーキテクチャの概要](#)

この章では、マルチテナント・アーキテクチャの最も重要なコンポーネントについて説明します。

1 マルチテナント・アーキテクチャの紹介

Oracle Multitenantオプションを理解します。

- [マルチテナント・アーキテクチャについて](#)
マルチテナント・アーキテクチャを使用すると、Oracle Databaseをマルチテナント・コンテナ・データベース(CDB)として機能させることができます。
- [マルチテナント・アーキテクチャの利点](#)
マルチテナント・アーキテクチャによって、従来の非CDBアーキテクチャが引き起こす複数の問題が解決されます。
- [データベース統合の方法](#)
存在期間中、データベースはCDBまたは非CDBのいずれかです。
- [マルチテナント環境のドキュメント・ロードマップ](#)
このトピックに、CDBの理解および使用にとって最も重要なトピックと、該当するドキュメントへの相互参照を示します。

親トピック: [マルチテナント・アーキテクチャ](#)

マルチテナント・アーキテクチャについて

マルチテナント・アーキテクチャを使用すると、Oracle Databaseをマルチテナント・コンテナ・データベース(CDB)として機能させることができます。

[CDB](#)は、顧客が作成した0以上のプラグブル・データベース(PDB)を含みます。[PDB](#)は、Oracle Netクライアントに[非CDB](#)として表示されるスキーマ、スキーマ・オブジェクトおよび非スキーマ・オブジェクトの移植可能な集合です。Oracle Database 12cまでのOracle Databaseはすべて非CDBでした。

- [CDBのコンテナについて](#)
コンテナは、マルチテナント・アーキテクチャ内のデータまたはメタデータの論理集合です。
- [マルチテナント・アーキテクチャのユーザー・インターフェースについて](#)
CDBおよび非CDBのどちらにも同じ管理ツールを使用できます。

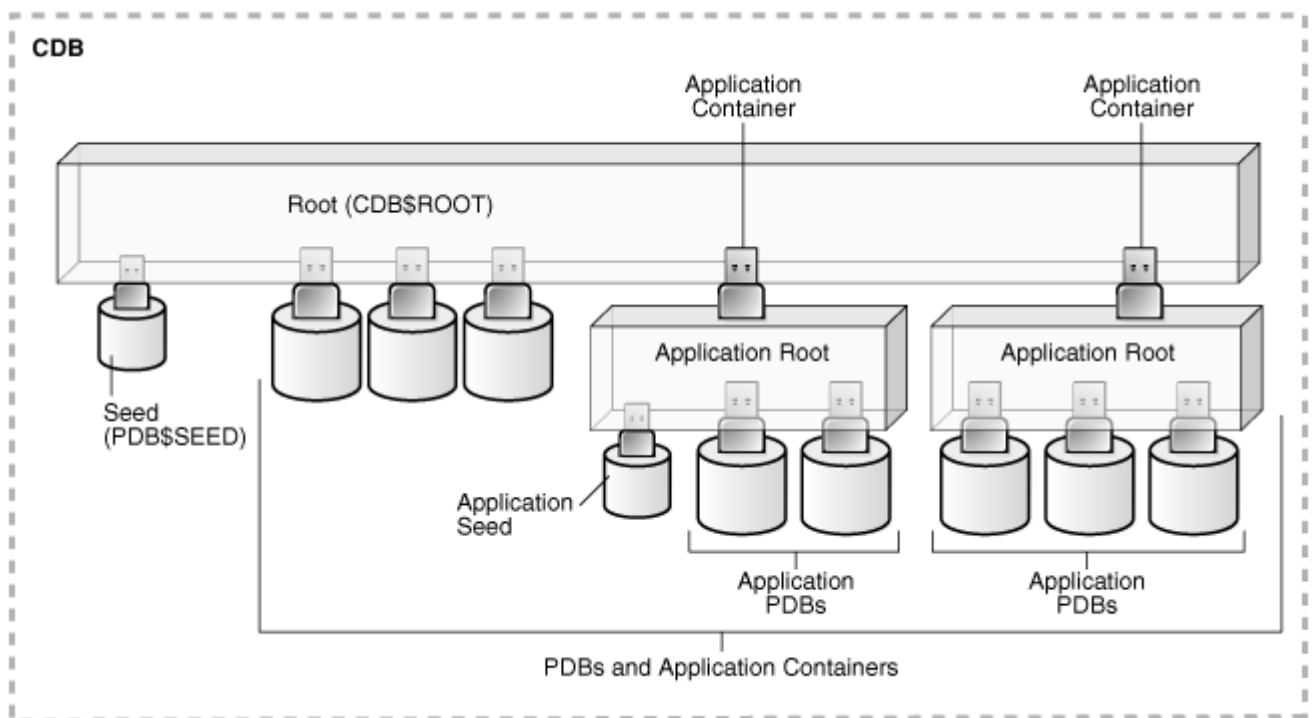
親トピック: [マルチテナント・アーキテクチャの紹介](#)

CDBのコンテナについて

コンテナは、マルチテナント・アーキテクチャ内のデータまたはメタデータの論理集合です。

次の図は、CDBで使用可能なコンテナを示しています。

図1-1 CDB内のコンテナ



CDBごとに、次のコンテナがあります。

- 1つのみのCDBルート・コンテナ(単純にルートとも呼ばれる)

[CDBルート](#)は、スキーマ、スキーマ・オブジェクトおよび非スキーマ・オブジェクトの集合で、すべてのPDBが属しています([「CDB内のコンテナの概要」](#)を参照)。ルートには、Oracle提供のメタデータおよび共通ユーザーが格納されます。メタデータの例として、Oracle提供のPL/SQLパッケージのソース・コードがあります([「CDBのデータ・ディクショナリ・アーキテクチャ」](#)を参照)。共通ユーザーとは、すべてのコンテナで認識されるデータベース・ユーザーです([「CDBの共通ユーザー」](#)を参照)。ルート・コンテナには、CDB\$ROOTという名前が付いています。

- 1つのみのシステム・コンテナ

[システム・コンテナ](#)にはルートCDBおよびCDB内のすべてのPDBが含まれます。このように、システム・コンテナはCDB自体の論理コンテナです。

- 0以上のアプリケーション・コンテナ

[アプリケーション・コンテナ](#)は1つの[アプリケーション・ルート](#)と、このルートに接続しているPDBで構成されます。システム・コンテナにはCDBルートおよびCDB内のすべてのPDBが含まれますが、アプリケーション・コンテナにはアプリケーション・ルートに接続しているPDBのみが含まれます。アプリケーション・ルートはCDBルートに属しており、他のコンテナには属しません。

- 0以上のユーザー作成PDB

PDBには特定の機能セットに必要なデータおよびコードが格納されています([「PDB」](#)を参照してください)。たとえば、PDBでは、人事管理または販売アプリケーションなど、特定のアプリケーションをサポートできます。CDBの作成時にはPDBは存在しません。ビジネスの要件に基づいてPDBを追加します。

PDBはゼロまたは1つのアプリケーション・コンテナに属しています。PDBがアプリケーション・コンテナに属す場合、これは[アプリケーションPDB](#)になります。たとえば、cust1_pdbおよびcust2_pdbのアプリケーションPDBがsaas_sales_acアプリケーション・コンテナに属す場合、これらは他のアプリケーション・コンテナには属していません。[アプリケーション・シード](#)はユーザー作成のPDBテンプレートとして機能するオプションのアプリケーションPDBで、これによって新規のアプリケーションPDBを迅速に作成できます。

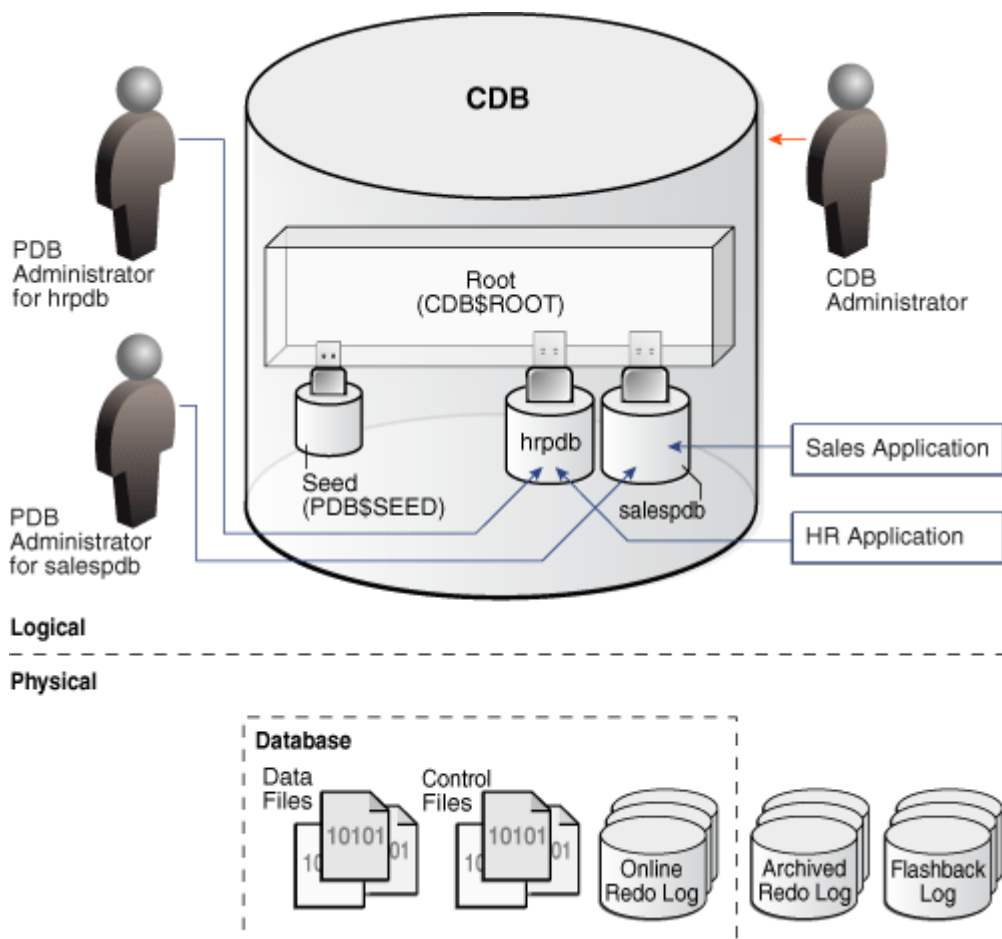
- 1つの[シードPDB](#)

シードPDBは、システム提供のテンプレートで、CDBではこれを使用して新しいPDBを作成できます。シードPDBには、PDB\$SEEDの名前が付いています。PDB\$SEEDでは、オブジェクトの追加や変更はできません。

例1-1 アプリケーション・コンテナを使用しないCDB

この例は、システム・コンテナ(CDB全体)、CDBルート、PDBシード(PDB\$SEED)および2つのPDBという5つのコンテナがあるシンプルなCDBを示しています。各PDBには、独自の専用アプリケーションがあります。各PDB管理者はそれぞれのPDBを管理します。[共有ユーザー](#)は、1つのCDB全体に1つのIDで存在します。この例では、共有ユーザーSYSは、ルートおよびすべてのPDBを管理できます。物理レベルでは、このCDBには、非CDBと同様に、1つのデータベース・インスタンスと複数のデータベース・ファイルがあります。

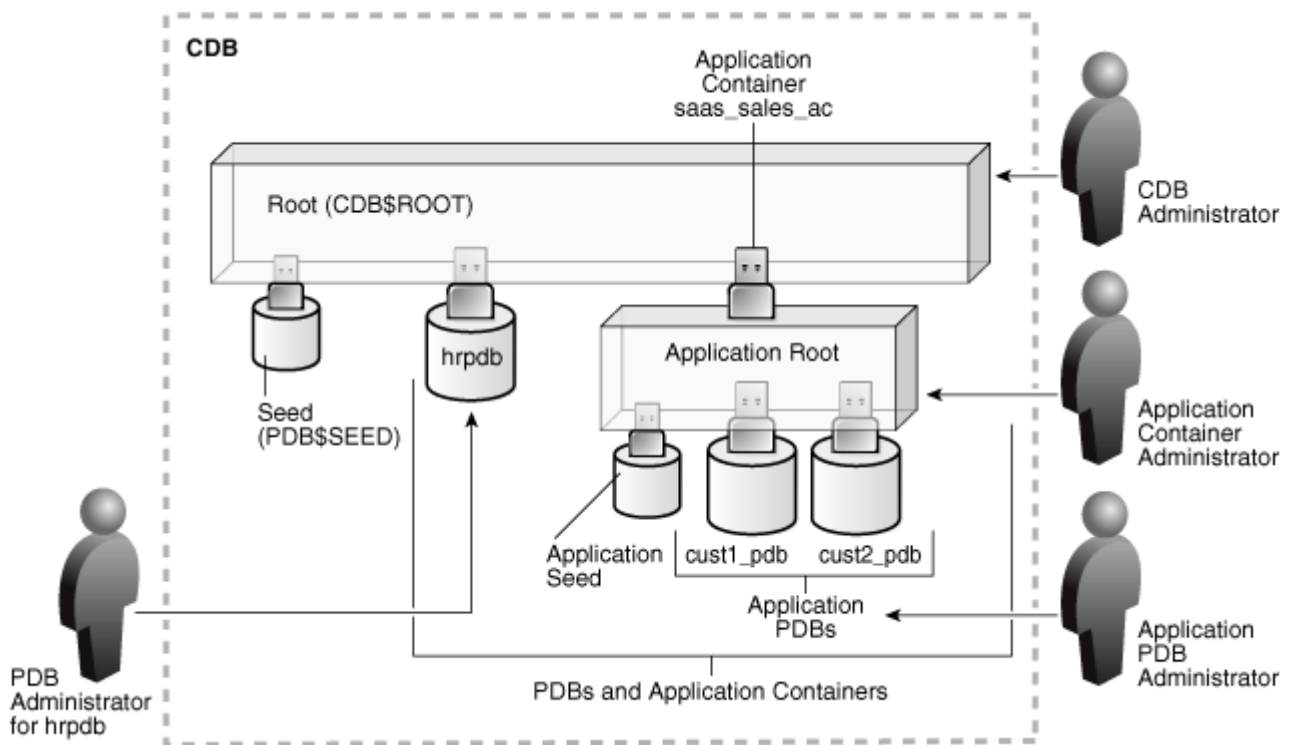
図1-2 アプリケーション・コンテナを使用しないCDB



例1-2 アプリケーション・コンテナを使用するCDB

この例では、CDBにsaas_sales_acというアプリケーション・コンテナが含まれます。アプリケーション・コンテナ内で、アプリケーションPDBのcust1_pdbは1人の顧客のアプリケーションをサポートし、アプリケーションPDBのcust2_pdbは別の顧客のアプリケーションをサポートします。CDBにはhrpdbというPDBも含まれており、これはHRアプリケーションをサポートしますが、アプリケーション・コンテナには属していません。

図1-3 アプリケーション・コンテナを使用するCDB



この例では、複数のDBAがCDB環境を管理します。

- CDB管理者はCDB自体を管理します。
- アプリケーション・コンテナ管理者は、アプリケーションのインストールおよびアップグレードも含めてsaas_sales_acコンテナを管理します。
- アプリケーションPDB管理者は、cust1_pdbおよびcust2_pdbという、saas_sales_acコンテナ内の2つのPDBを管理します。
- PDB管理者はhrpdbを管理します。

関連項目:

[「マルチテナント環境の構成と管理の概要」](#)

親トピック: [マルチテナント・アーキテクチャについて](#)

マルチテナント・アーキテクチャのユーザー・インターフェースについて

CDBおよび非CDBのどちらにも同じ管理ツールを使用できます。

表1-1 マルチテナント環境のツール

インタフェース	説明	関連項目
コマンドライン・アクセス用の SQL*Plus および SQL Developer	SQL*Plus は、Oracle Database とともにインストールされる対話型のバッチ問合せツールです。	SQL*Plus ユーザーズ・ガイドおよびリファレンス
Oracle Enterprise Manager	Cloud Control は、GUI を備えた Oracle Database の管理ツールで	Cloud Control のオンライン・ヘルプ

インタフェース	説明	関連項目
Cloud Control (Cloud Control)	す。Cloud Control では、Oracle Database 12c のターゲット(PDB、CDB および非 CDB を含む)をサポートしています。	
Oracle Enterprise Manager Database Express (EM Express)	EM Express は、Oracle Database に組み込まれた Web ベースの管理製品です。EM Express では PDB をプロビジョニングおよび管理でき、次の操作を行うことができます。 <ul style="list-style-type: none"> ● PDB の作成および削除 ● PDB の接続および切断 ● PDB のクローニング ● PDB のリソース制限の設定 	EM Express を使用して CDB と PDB を管理する方法についてさらに学習するには、 『Oracle Database パフォーマンス・チューニング・ガイド』 を参照してください
Oracle Database Configuration Assistant(DBCA)	DBCA は、CDB の作成および複製が可能なグラフィカル・ユーザー・インタフェースを備えたユーティリティです。また、PDB を作成、再配置、クローニング、接続および切断することもできます。	DBCA の詳細は、 『Oracle Database パフォーマンス・チューニング・ガイド』 および 『Oracle Database 管理者ガイド』 を参照してください

関連項目:

データベース管理者用のツールの詳細は、[『Oracle Database概要』](#)を参照してください

親トピック: [マルチテナント・アーキテクチャについて](#)

マルチテナント・アーキテクチャの利点

マルチテナント・アーキテクチャによって、従来の非CDBアーキテクチャが引き起こす複数の問題が解決されます。

- [非CDBアーキテクチャの課題](#)
大企業では、使用するデータベースが数百または数千にのぼることがあります。多くの場合、これらのデータベースは、多数の物理サーバー上の様々なプラットフォームで実行されます。
- [マルチテナント・アーキテクチャのデータベース統合に対する利点](#)
データベース統合とは、複数のデータベースから1台のコンピュータ上の1つのデータベースにデータを統合するプロセスです。Oracleマルチテナント・オプションでは、既存のスキーマまたはアプリケーションを変更することなくデータおよびコードを統合できます。
- [マルチテナント・アーキテクチャの管理性に対する利点](#)

マルチテナント・アーキテクチャのマルチテナント・アーキテクチャには、データベース統合の他にも複数の利点があります。これらの利点は、すべてのディクショナリ・メタデータを1箇所に格納するのではなく、PDB固有のデータおよびメタデータをPDB自体に格納することで得られます。

親トピック: [マルチテナント・アーキテクチャの紹介](#)

非CDBアーキテクチャの課題

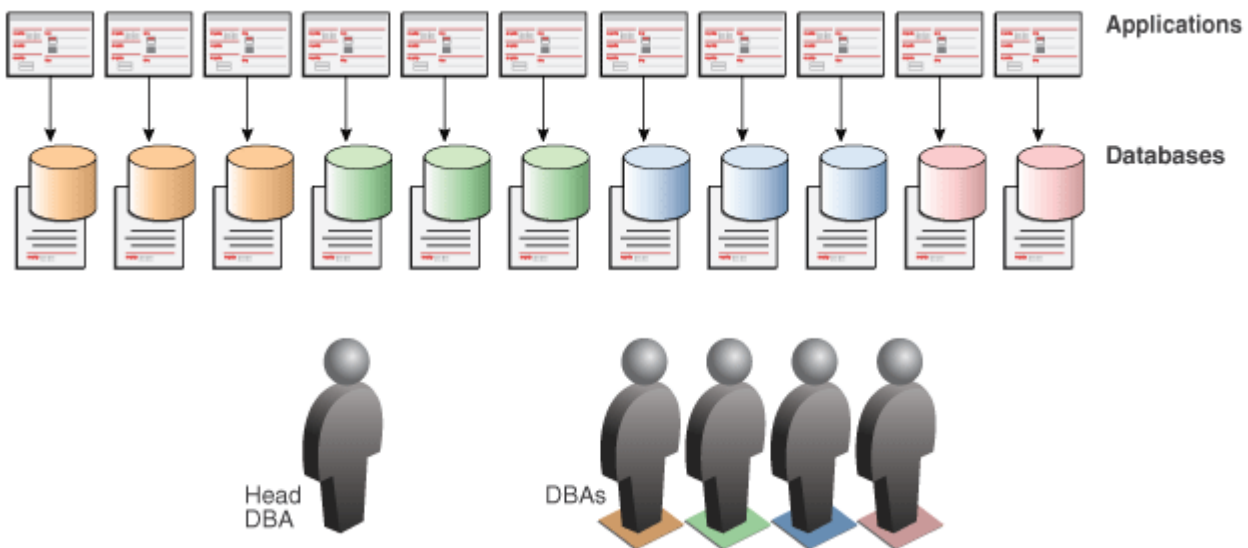
大企業では、使用するデータベースが数百または数千にのぼることがあります。多くの場合、これらのデータベースは、多数の物理サーバー上の様々なプラットフォームで実行されます。

ハードウェア・テクノロジーの向上、特にCPU数の増加により、サーバーでは以前より多くのワークロードを処理できるようになりました。1つのデータベースで使用するサーバー・ハードウェアの容量はごく一部です。この方法では、ハードウェアと人事管理の両方が無駄になります。

たとえば、100のサーバーにそれぞれ1つのデータベースがあり、各データベースがハードウェア資源の10%と管理者の時間の10%を使用するとします。DBAのチームは、各データベースのSGA、データベース、データベース・ファイル、アカウント、セキュリティなどを別々に管理する必要がある一方、システム管理者は100台の異なるコンピュータをメンテナンスする必要があります。

この問題を縮小して示すために、[図1-4](#)では、それぞれのアプリケーションとサーバーを持つ11のデータベースを表しています。1人の主任DBAが4人のDBAのチームを管理し、各DBAが2つまたは3つのデータベースを担当します。

図1-4 データベース統合前のデータベース環境



一般的な対応策は、次のとおりです。

- 仮想マシン(VM)を使用します。

このモデルでは、物理サーバーのオペレーティング・インフラストラクチャ(オペレーティング・システムとデータベース)を仮想マシンにレプリケートします。VMは機動的であるものの、技術リソースが非効率的に使用され、個別の管理が必要となります。仮想スプールは、管理にかかるコストは同様で、既存の物理スプールに置き換わるものです。

- 各サーバー上に複数のデータベースを配置します。

個別のデータベースを使用する場合、オペレーティング・システムのレプリケーションの必要性はなくなりますが、バックグラウンド・プロセス、システムおよびプロセス・メモリーやOracleメタデータは共有されません。データベースを個別に管理する必要があります。

- スキーマまたは仮想プライベート・データベース(VPD)にデータを論理的に分割します。

この方法では、技術リソースが効率的に使用されます。複数のスキーマまたはVPDを一括して管理できます。ただし、

このモデルは他の方法と比較して機動性が低く、管理、保護および転送に必要な労力が増加します。また、通常、論理モデルには広範囲にわたるアプリケーションの変更が必要となるため、採用はお勧めしません。

親トピック: [マルチテナント・アーキテクチャの利点](#)

マルチテナント・アーキテクチャのデータベース統合に対する利点

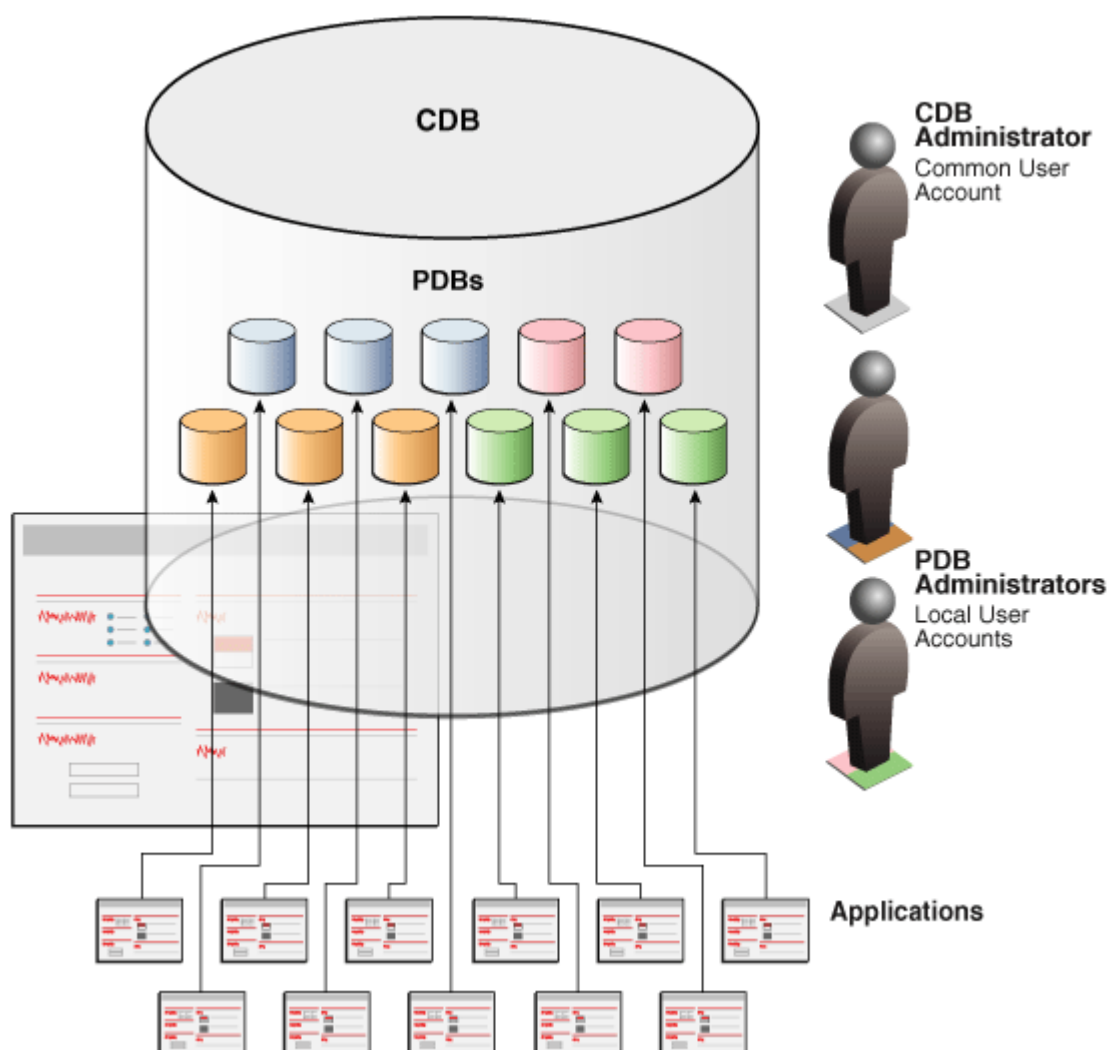
データベース統合とは、複数のデータベースから1台のコンピュータ上の1つのデータベースにデータを統合するプロセスです。Oracleマルチテナント・オプションでは、既存のスキーマまたはアプリケーションを変更することなくデータおよびコードを統合できます。

PDB/非CDB互換性保障とは、Oracle Netで接続しているクライアントから、PDBと非CDBの動作が同じように見えることです。非CDBに対して実行されるアプリケーション定義(たとえば、表とPL/SQLパッケージ)のインストール体系は、PDBに対して同じように実行され、同じ結果が生成されます。また、アプリケーション定義を格納するPDBに接続するクライアント・コードの実行時の動作も、このアプリケーション定義を格納する非CDBに接続されるクライアント・コードの動作と同じです。

完全な非CDBで動作する操作は、完全なCDBの場合と同様に動作します(Oracle Data Guardとデータベースのバックアップおよびリカバリの使用時など)。したがって、非CDBのユーザー、管理者および開発者は、データベースの統合後に実質的に同じ経験をします。

次の図に、[図1-4](#)のデータベースを1台のコンピュータに統合した後の状態を示します。DBAチームは5人から3人に縮小され、1人のCDB管理者でCDBを管理する一方、2人のPDB管理者はPDBの管理を分担します。

図1-5 単一のCDB



Oracle Database 12cリリース2 (12.2)以降では、アプリケーションPDBを含むアプリケーション・コンテナを作成できます。この方法により、このコンテナの中でアプリケーションを作成および管理できます。CDBへの統合に当てはまる利点の多くは、アプリ

ケーション・コンテナでの統合にも当てはまります。

データベース統合のためにマルチテナント・アーキテクチャを使用すると、次のような利点があります。

- コスト削減

ハードウェアおよびデータベース・インフラストラクチャを1つのセットのバックグラウンド・プロセスに統合し、計算リソースおよびメモリー・リソースを効率的に共有することで、ハードウェアおよびメンテナンスのコストを削減できます。たとえば、1つのサーバー上の100個のPDBが1つのデータベース・インスタンスを共有します。

- より簡単かつ高速なデータおよびコードの移動

計画的に、PDBを素早くCDBに接続し、そのPDBをCDBから切断して、別のCDBに接続できます。PDBが使用可能な間にそれをクローニングすることもできます。任意の文字セットを使用してPDBに接続して、文字セットを変換せずにアクセスすることができます。CDBの文字セットがAL32UTF8の場合、異なるデータベース文字セットを使用するPDBが同じCDB内に存在する可能性があります。

- 物理データベースの容易な管理および監視

[CDB管理者](#)は、ホストされているすべてのテナントとCDBルートに対して1つの操作(パッチ適用やRMANバックアップの実行など)を実行することにより、環境を1つの集合体として管理できます。バックアップ計画および障害回復が簡単になります。

- データおよびコードの分離

1つの物理データベースに統合しても、PDBは非CDBと似た動作をします。たとえば、ユーザーのエラーにより重大なデータが失われた場合、PDB管理者はOracle FlashbackまたはPoint-in-Timeリカバリを使用して、他のPDBに影響を及ぼすことなく、失われたデータを回復できます。

- 管理業務の安全な分担

[共通ユーザー](#)は十分な権限を持つどのコンテナにも接続できますが、[ローカル・ユーザー](#)は特定のPDBに制限されます。管理者は業務を次のように分担できます。

- 管理者は、共通アカウントを使用してCDBまたはアプリケーション・コンテナを管理します。権限は付与先のコンテナに含まれているため、1つのPDBのローカル・ユーザーには、同じCDB内の他のPDBに対する権限はありません。
- 管理者は、ローカル・アカウントを使用して個々のPDBを管理します。

- 容易なパフォーマンス・チューニング

複数のデータベースより1つのデータベースのパフォーマンス・メトリックを収集する方が簡単です。100のSGAより1つのSGAのサイズを変更する方が容易です。

- データベース・パッチおよびアップグレードの削減

100のデータベースよりも1つのデータベースにパッチを適用する方が簡単であり、100のデータベースよりも1つのデータベースをアップグレードする方が簡単です。

関連項目:

- [「マルチテナント環境の構成と管理の概要」](#)
- 共通ユーザーについて学習するには、[Oracle Databaseセキュリティ・ガイド](#)を参照してください

マルチテナント・アーキテクチャの管理性に対する利点

マルチテナント・アーキテクチャには、データベース統合の他にも次のような利点があります。これらの利点は、すべてのディクショナリ・メタデータを1箇所に格納するのではなく、PDB固有のデータおよびメタデータをPDB自体に格納することで得られます。

独自のディクショナリ・メタデータを格納することにより、PDBを別個の構成単位として管理しやすくなります。この利点は、CDBにPDBが1つしかなくても得られます。別個に管理されるアプリケーション・コンテナ内にPDBをグループ化することで、管理性がさらに向上します。

CDBでは、データ・ディクショナリ・メタデータは、ルートとPDBの間で分割されています。データ・ディクショナリの分離には、次の利点があります。

- データおよびコードのアップグレードの簡略化

たとえば、あるデータベース・リリースから別のリリースにCDBをアップグレードするかわりに、PDBを既存のCDBから迅速に切断し、より新しいリリースから新たに作成されたCDBに接続することができます。

- サーバー間の移行の簡略化

ロード・バランシングの実行やSLAの遵守を目的として、オンプレミスのデータ・センターからクラウドへ、または同じ環境内の2つのサーバー間で、アプリケーション・データベースを移行できます。

- PDB内のデータ破損の防止

他のPDBに影響を及ぼさずに、SCNまたはPDB固有のリストア・ポイントまでPDBをフラッシュ・バックできます。この機能は非CDB用のフラッシュバック・データベース機能に類似しています。

- アプリケーション固有のデータおよびメタデータを1つの場所でインストール、管理およびアップグレードする機能

アプリケーション固有のPDBのセットを、[アプリケーション・コンテナ](#)と呼ばれる単一のコンポーネントとして定義できます。そして1つ以上のアプリケーションをこのコンテナ内で定義できます。各アプリケーションはこのアプリケーション・コンテナ内で共有される、共通メタデータおよびデータの名前付けおよびバージョンングされたセットです。

たとえば、SaaSベンダーの各顧客は独自の[アプリケーションPDB](#)を所有できます。各アプリケーションPDBは、各PDBには異なるデータがある、sales_mltという同一に定義された表が存在する場合があります。PDBはcountries_oltという[データリンク共通オブジェクト](#)を共有し、各PDBで同一のデータを保持できます。アプリケーション管理者としてマスター・アプリケーション定義を管理することで、すべての新規顧客が同じオブジェクトを使用するPDBを取得したり、既存のスキーマへのすべての変更(新規表の追加や、表の定義内の変更など)がアプリケーションを共有するすべてのPDBに適用されるようになります。

- Oracle Database Resource Managerとの統合

マルチテナント環境では、同じサーバーで稼働するPDBでのシステム・リソースの競合が問題です。もう1つの問題は、より一貫性があり、予測可能なパフォーマンスを実現するために、リソースの使用が制限されることです。こうしたリソースの競合、使用方法および監視の問題に対処するには、Oracle Database Resource Managerを使用します。

関連項目:

- [「CDBでのOracle Resource Managerの概要」](#)
- [「CDBのデータ・ディクショナリ・アーキテクチャ」](#)
- [「アプリケーション・コンテナの管理」](#)

親トピック: [マルチテナント・アーキテクチャの利点](#)

データベース統合の方法

存在期間中、データベースはCDBまたは非CDBのいずれかです。

データベースを作成時にCDBとして定義してから、このCDB内にPDBおよびアプリケーション・コンテナを作成する必要があります。非CDBをCDBに、またはCDBを非CDBに後で変換することはできません。

- [CDBの作成](#)
CREATE DATABASE ... ENABLE PLUGGABLE DATABASE SQL文で新規のCDBを作成します。
- [PDBの作成](#)
CREATE PLUGGABLE DATABASE SQL文でPDBを作成します。

親トピック: [マルチテナント・アーキテクチャの紹介](#)

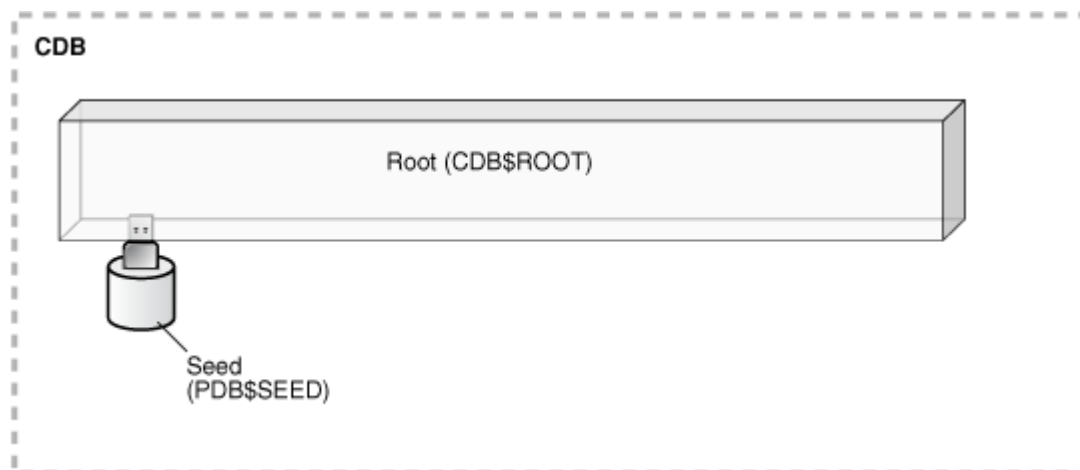
CDBの作成

CREATE DATABASE ... ENABLE PLUGGABLE DATABASE SQL文で新規のCDBを作成します。

ENABLE PLUGGABLE DATABASE句を指定しない場合、新規に作成されるデータベースは非CDBです。この場合、非CDBにPDBを含めることはできません。

CDBを作成する場合、Oracle Databaseではルート・コンテナ(CDB\$ROOT)およびシードPDB (PDB\$SEED)が自動的に作成されます。次の図に、新規に作成されたCDBを示します。

図1-6 シードPDBが含まれるCDB



例1-3 データベースがCDBであるかどうかの判断

次の簡単な問合せでは、管理ユーザーが現在接続しているデータベースが非CDBか、それともCDB内のコンテナかを確認します。

```
SQL> SELECT NAME, CDB, CON_ID FROM V$DATABASE;
```

NAME	CDB	CON_ID
CDB1	YES	0

関連項目:

- [「CDBの作成および構成」](#)
- CREATE DATABASE文で指定できる句およびパラメータ値の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。

親トピック: [データベース統合の方法](#)

PDBの作成

CREATE PLUGGABLE DATABASE SQL文でPDBを作成します。

作成されたPDBでは、メタデータとシステム提供オブジェクトへの内部リンクをはじめとする完全なデータ・ディクショナリが、CDBルートに自動的に含まれます。すべてのPDBを単一のルート([CDBルート](#)または[アプリケーション・ルート](#))から定義する必要があります。

各PDBにはグローバル一意識別子(GUID)があります。PDB GUIDは、PDBのファイルを格納するディレクトリ(Oracle Managed Filesディレクトリおよび非Oracle Managed Filesディレクトリの両方を含む)の名前を生成するために主に使用されます。

- [クローニングによるPDBの作成](#)
PDBを作成する方法として、クローニングと呼ばれる方法があります。
- [接続によるPDBの作成](#)
切断されたPDBの接続、または非CDBのPDBとしての接続によってPDBを作成できます。
- [再配置によるPDBの作成](#)
PDBをあるCDBから別のCDBに再配置するには、CREATE PLUGGABLE DATABASE ... RELOCATE文またはDBCAを使用します。
- [プロキシPDBとしてのPDBの作成](#)
プロキシPDBは、[参照先PDB](#)と呼ばれるリモートCDB内の別のPDBへのアクセスを提供します。

関連項目:

[「PDBおよびアプリケーション・コンテナの作成および削除」](#)

親トピック: [データベース統合の方法](#)

クローニングによるPDBの作成

PDBを作成する方法として、クローニングと呼ばれる方法があります。

PDB\$SEED、アプリケーション・シード、リモートまたはローカルのPDB、あるいは非CDBからPDBをクローニングできます。

- [シードからのPDBの作成](#)
CREATE PLUGGABLE DATABASE文を使用して、PDBをシードから作成できます。
- [PDBまたは非CDBのクローニングによるPDBの作成](#)
PDBまたは非CDBをクローニングするには、FROM句とともにCREATE PLUGGABLE DATABASE文を使用します。

親トピック: [PDBの作成](#)

シードからのPDBの作成

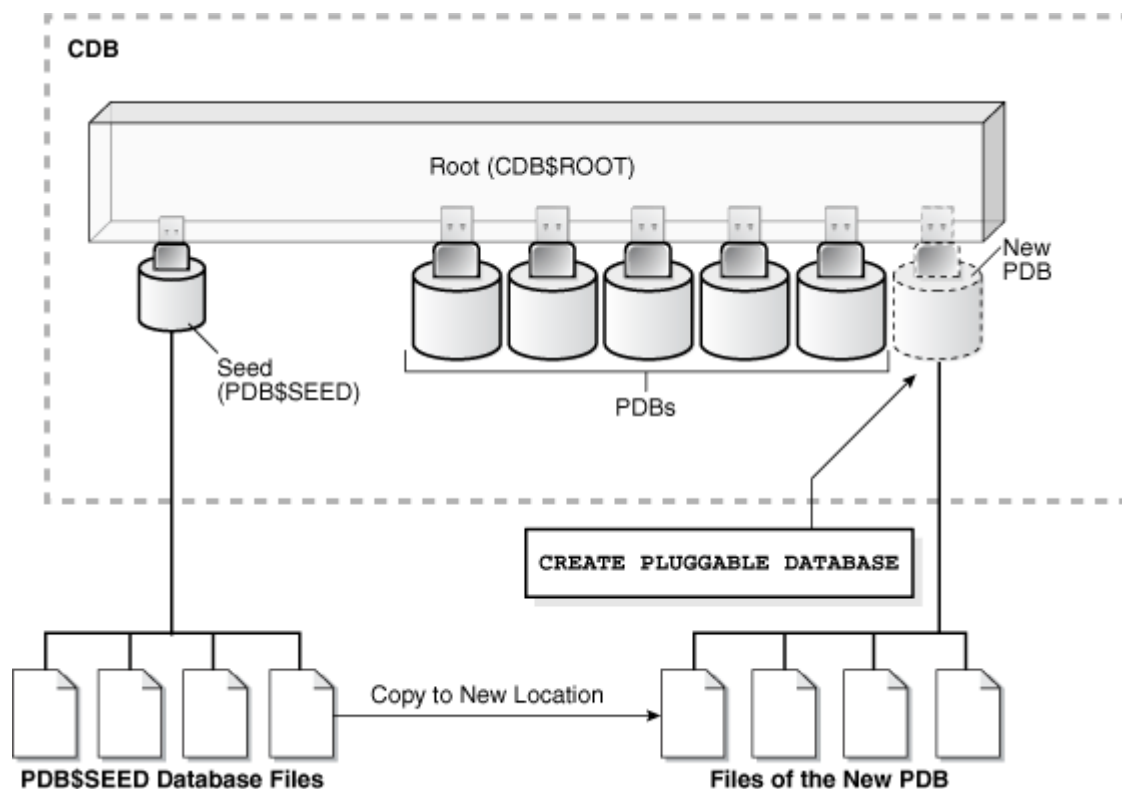
CREATE PLUGGABLE DATABASE文を使用して、PDBをシードから作成できます。

シードは、別のPDBの作成用テンプレートとして機能するPDBです。シードからのPDBの作成では、PDBの内容の一部または全部がコピーされ、新しい一意識別子が割り当てられます。

シードPDBは次のいずれかです。

- PDB作成用のシステム提供のテンプレートである、PDBシード(PDB\$SEED)
各CDBにはPDB\$SEEDが1つずつあり、これは変更することも削除することもできません。
- 指定された[アプリケーション・ルート](#)用のユーザー作成PDBである、[アプリケーション・シード](#)
[アプリケーション・コンテナ](#)内で、CREATE PLUGGABLE DATABASE AS SEED文を使用してアプリケーション・シードを作成してから、それを使用して新規アプリケーションPDBの作成を促進できます。

図1-7 PDB\$SEEDからの作成



例1-4 PDB\$SEEDからのPDBの作成

次のSQL文は、Oracle Managed Filesを使用してPDB\$SEEDからhrpdbという名前のPDBを作成します。

```
CREATE PLUGGABLE DATABASE hrpdb  
ADMIN USER dba1 IDENTIFIED BY password;
```

関連項目:

[「PDBを最初から作成」](#)

親トピック: [クローニングによるPDBの作成](#)

PDBまたは非CDBのクローニングによるPDBの作成

PDBまたは非CDBをクローニングするには、FROM句とともにCREATE PLUGGABLE DATABASE文を使用します。

この方法では、ソースは非CDB、あるいはローカルまたはリモートのCDB内のPDBです。ターゲットはソースからコピーされるPDBです。クローニング操作により、ソースに関連付けられているファイルが新しい場所にコピーされ、PDBを作成するために新しい

GUIDが割り当てられます。

この手法は、テストおよび開発用にPDBを短時間で作成する場合に便利です。たとえば、新しいアプリケーションまたは変更されたアプリケーションを、本番PDBにデプロイする前に、クローニングPDBでテストできます。PDBがローカルUNDOモードの場合、操作中にソースPDBを読み取り/書き込みモードでオープンできます。これを[ホット・クローニング](#)と呼びます。

ノート:

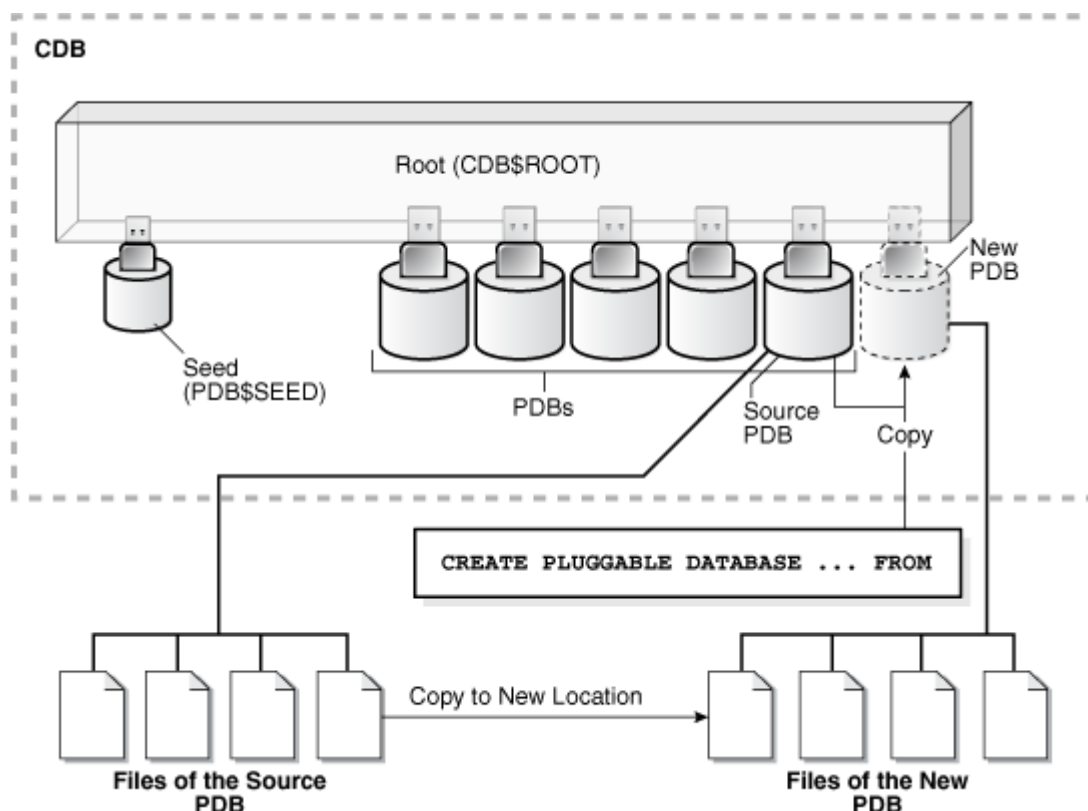


リモート CDB から PDB をクローニングする場合は、データベース・リンクを使用する必要があります。

CREATE PLUGGABLE DATABASE文をアプリケーション・ルートで実行する場合、クローニングされたPDBはアプリケーション・テナ内に作成されます。この場合、アプリケーション名およびソースPDBのバージョンはアプリケーション・テナのアプリケーション名およびバージョンと互換性がある必要があります。

次の図に、ソースおよびターゲットの両方が同じCDB内にある場合のPDBのクローニングを示します。

図1-8 PDBのクローニング



Oracle Database 19c以降では、DBCAを使用してリモートPDBをクローニングできます。

例1-5 PDBのクローニング

次のSQL文は、hrpdbという名前のプラグインPDBからsalespdbという名前のPDBのクローンを作成します。

```
CREATE PLUGGABLE DATABASE salespdb FROM hrpdb;
```

- [PDBスナップショットからのクローン](#)

CREATE PLUGGABLE DATABASEコマンドのUSING SNAPSHOT句を指定することで、**PDBスナップショットからクローン**を作成します。

- [スナップショット・コピーPDB](#)

スナップショット・コピーPDBは、基礎となるストレージ・システムのコピーに基づいています。スナップショット・コピーPDB

を使用すると、テストのために必要な記憶域が削減され、作成時間が大幅に短縮されます。

- [リフレッシュ可能なクローンPDB](#)

リフレッシュ可能なクローンPDBは、ソースPDBと定期的に同期できる読取り専用クローンです。

関連項目:

- [「PDBまたは非CDBのクローニング」](#)
- [「CDBの表領域およびデータベース・ファイルの概要」](#)
- [アプリケーション・メンテナンス](#)

親トピック: [クローニングによるPDBの作成](#)

PDBスナップショットからのクローン

CREATE PLUGGABLE DATABASEコマンドのUSING SNAPSHOT句を指定することで、**PDBスナップショットからクローン**を作成します。

SNAPSHOT句の使用によるPDBスナップショットの作成

PDBスナップショットはPDBのPoint-in-Timeコピーです。スナップショットの作成中に、ソースPDBを読取り専用または読取り/書き込みでオープンできます。ソースPDBのオープン中に取得されるPDBスナップショットは、**ホット・クローン**と呼ばれます。

PDBスナップショットからクローンを作成できます。これらのクローンPDBは開発やテストで役立ちます。

スナップショットは、CREATE PLUGGABLE DATABASE (またはALTER PLUGGABLE DATABASE)のSNAPSHOT句を使用して手動で作成するか、EVERY interval句を使用して自動的に作成できます。次の文は、pdb1_wed_4_1201というPDBスナップショットを作成します。

```
ALTER PLUGGABLE DATABASE SNAPSHOT pdb1_wed_4_1201;
```

ストレージ・システムでスパース・クローンがサポートされる場合、前述のコマンドではスパース・コピーが作成されます。それ以外の場合は、コマンドによって完全なコピーが作成されます。

すべてのPDBスナップショットは、スナップショット名と、スナップショット作成時のSCNおよびタイムスタンプに関連付けられています。

USING SNAPSHOT句の使用によるPDBの作成

PDBスナップショットからのクローンは、完全なスタンドアロンのPDBです。記憶域管理スナップショットに基づいているスナップショット・コピーPDBとは異なり、PDBスナップショットから作成されたクローンをマテリアライズする必要はありません。

PDBスナップショットからクローンを作成するには、CREATE PLUGGABLE DATABASE文のUSING SNAPSHOT句を指定します。たとえば、次の文は、pdb1_wed_4_1201というPDBレベル・スナップショットからpdb1_copyというPDBをクローニングします。

```
CREATE PLUGGABLE DATABASE pdb1_copy FROM pdb1
USING SNAPSHOT pdb1_wed_4_1201;
```

関連項目:

- [「PDBスナップショットからのPDBのクローニングについて」](#)
- 各種エディションおよびサービスでサポートされる機能の詳細は、『[Oracle Databaseライセンス情報ユーザー・マニュアル](#)』を参照

親トピック: [PDBまたは非CDBのクローニングによるPDBの作成](#)

スナップショット・コピーPDB

スナップショット・コピーPDBは、基礎となるストレージ・システムのコピーに基づいています。スナップショット・コピーPDBを使用すると、テストのために必要な記憶域が削減され、作成時間が大幅に短縮されます。

ファイル・システムで記憶域スナップショットがサポートされている場合は、`CREATE PLUGGABLE DATABASE ... FROM ... SNAPSHOT COPY`では、ソースPDBからPDBがコピーされます。この操作中の読取り/書込みは可能です。スナップショット・コピーPDBファイルには、copy-on-writeテクノロジーが使用されます。ディスク上の記憶域がさらに必要になるのは、変更されたブロックのみとなります。ファイル・システムで記憶域スナップショットがサポートされていないか、Oracle Exadataスパース・ファイルが使用されている場合、`CLONEDB`初期化パラメータはtrueである必要があり、スナップショット・コピーPDBが存在するがぎりソースPDBは読取り専用である必要があります。

スナップショット・コピーPDBは記憶域管理スナップショットに依存しているため、スナップショット・コピーPDBをCDBルートまたはアプリケーション・ルートから切断することはできません。スナップショット・コピーPDBの基となっている記憶域スナップショットは削除できません。

スパース・ファイルを使用するスナップショット・コピーPDBを完全PDBに変換できます。このプロセスは、スナップショット・コピーPDBの**マテリアライズ**と呼ばれます。マテリアライズされたPDBはソースPDBに依存しないため、削除できます。`ALTER PLUGGABLE DATABASE MATERIALIZE`コマンドを実行することでPDBをマテリアライズします。

ノート:



`USING SNAPSHOT` 句を使用して作成された PDB と `SNAPSHOT COPY` 句を使用して作成された PDB のプロパティは異なります。1 つの `CREATE PLUGGABLE DATABASE` コマンドに両方の句を指定することはできません。`CREATE PLUGGABLE DATABASE ... FROM ... USING SNAPSHOT` 句では、マテリアライズする必要がない完全なスタンドアロン PDB が作成されます。`CREATE PLUGGABLE DATABASE ... FROM ... SNAPSHOT COPY` 句は、基となる記憶域レベルのスナップショットを削除する場合にマテリアライズする必要があるスパース PDB を作成します。

ノート:



[スナップショット・コピーPDB の作成とマテリアライズ](#)

親トピック: [PDBまたは非CDBのクローニングによるPDBの作成](#)

リフレッシュ可能なクローンPDB

リフレッシュ可能なクローンPDBは、ソースPDBと定期的に同期できる読取り専用クローンです。

`REFRESH MODE`句に指定された値に応じて、同期は自動的に、または手動で行われます。たとえば、`hrpdb_re_clone`が`hrpdb`のクローンである場合は、`hrpdb`による変更で`hrpdb_re_clone`を毎月手動でリフレッシュできます。あるいは、24時間ごとに`hrpdb_re_clone`に変更が自動的に伝播されるように`hrpdb`を構成できます。

ソースPDBとそのリフレッシュ可能なクローンのロールを切り替えることができます。このスイッチオーバーは、CDB間のロード・バランシングや、ソースPDBに障害が発生した場合に役立ちます。

ノート:



REFRESH MODE 句を使用した PDB のクローニング方法を学習するには、「[PDB または非 CDB のクローニングについて](#)」を参照

親トピック: [PDBまたは非CDBのクローニングによるPDBの作成](#)

接続によるPDBの作成

切断されたPDBの接続、または非CDBのPDBとしての接続によってPDBを作成できます。

- [切断されたPDBの接続によるPDBの作成](#)

切断されたPDBは、自己完結型のデータ・ファイルのセットと、PDBファイルの場所を指定するXMLメタデータ・ファイルです。切断されたPDBを接続するには、USING句を使用してCREATE PLUGGABLE DATABASE文を使用します。

- [非CDBからのPDBの作成](#)

非CDBをPDBに移動できます。

親トピック: [PDBの作成](#)

切断されたPDBの接続によるPDBの作成

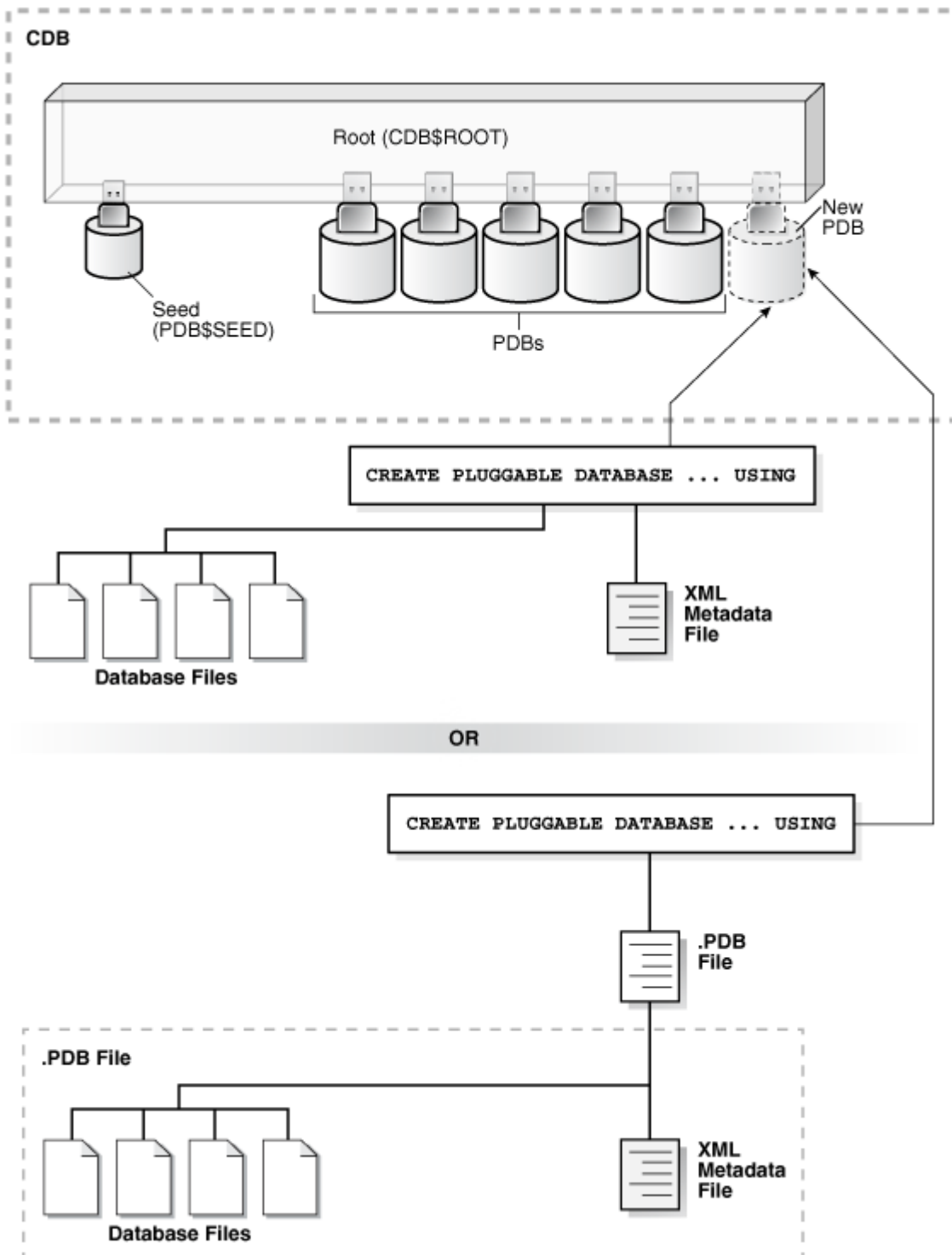
切断されたPDBは、自己完結型のデータ・ファイルのセットと、PDBファイルの場所を指定するXMLメタデータ・ファイルです。切断されたPDBを接続するには、USING句を使用してCREATE PLUGGABLE DATABASE文を使用します。

切断されたPDBを接続する際、次のようなオプションがあります。

- PDBと、そのPDBに関連付けられたファイルが記述されるXMLメタデータ・ファイルを指定します。
- XMLファイルおよびPDBデータ・ファイルの両方が含まれる圧縮されたファイルである、[PDBアーカイブ・ファイル](#)を指定します。アーカイブ・ファイルを指定することでPDBを作成し、その結果、XMLファイルおよびデータ・ファイルを別々にコピーすることを回避できます。

次の図に、XMLファイルを使用した切断されたPDBの接続を示します。

図1-9 切断されたPDBの接続



例1-6 PDBの接続

次のSQL文は、指定したXMLファイルに格納されているメタデータに基づき、salespdbという名前のPDBに接続し、切断されたPDBのファイルは新しい場所へ移動する必要がないため、NOCOPYを指定します。

```
CREATE PLUGGABLE DATABASE salespdb USING '/disk1/usr/salespdb.xml' NOCOPY;
```

関連項目:

[「切断されたPDBの接続」](#)

親トピック: [プラグインによるPDBの作成](#)

非CDBからのPDBの作成

非CDBをPDBに移動できます。

このタスクは、次の方法で実行できます。

- Oracle Database 12cで非CDBに対してDBMS_PDB.DESCRIBEを実行する

非CDBをトランザクショナルに一貫した状態にし、DBMS_PDB.DESCRIBEファンクションを実行して、このデータベースに関するXMLメタデータを生成します。CDBのルートに接続中に、CREATE PLUGGABLE DATABASE文を実行して、既存の非CDBからPDBを作成します。最後に、PDBデータ・ディクショナリ内の定義をCDB\$ROOT内のオブジェクトへの参照に変換するために、PDBにログインして、noncdb_to_pdb.sqlスクリプトを実行します。

- トランスポータブル表領域の有無にかかわらず、Oracle Data Pumpを使用します。

Oracle Data Pumpを使用して、非CDBでデータ・セットを定義できます。この非CDBは、現行または以前のOracle Databaseリリース(Oracle Database 10gなど)のどちらにあるものでもかまいません。既存のCDB内に空のPDBを作成し、Oracle Data Pumpを使用して、このPDBにデータ・セットをインポートします。

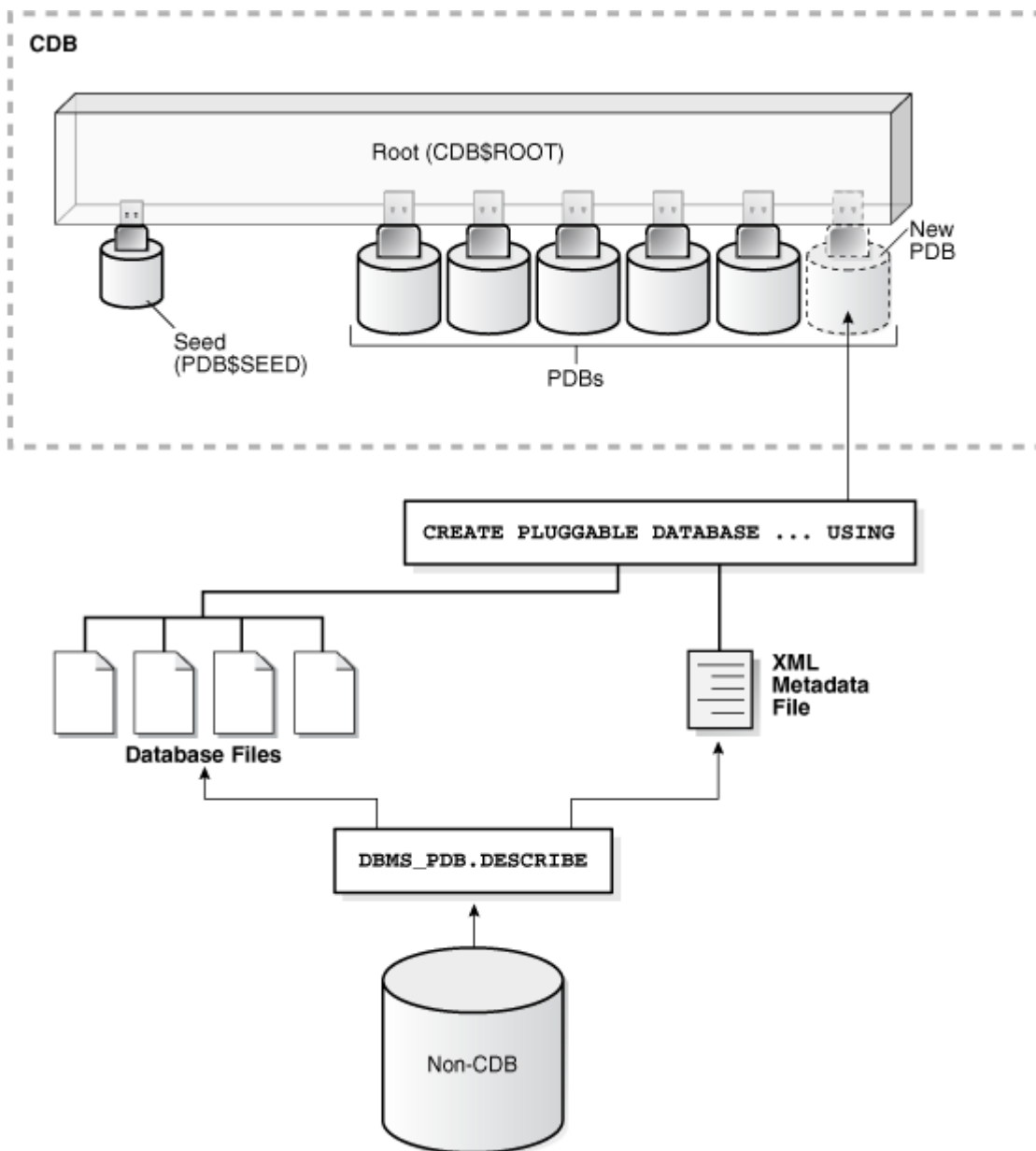
Oracle Data Pumpを使用した完全トランスポータブル・エクスポートにより、データベースの完全なコピーの作成に必要なすべてのオブジェクトとデータをエクスポートします。Oracle Data Pumpにより、ダイレクト・パス・アンロードと外部表を使用してオブジェクトをエクスポートしてから、ダイレクト・パスのINSERTと外部表を使用してオブジェクトをインポートします。完全トランスポータブル・ダンプ・ファイルには、表関連オブジェクトばかりでなく、データベース内のすべてのオブジェクトが含まれます。完全トランスポータブル・エクスポートは、Oracle Database 11gリリース2(11.2.0.3)から、Oracle Database 12c以降へのインポートに使用できます。

- Oracle GoldenGateレプリケーションを使用します。

非CDBからPDBにデータをレプリケートします。PDBが非CDBと同じ最新の状態になったら、PDBにスイッチオーバーします。

次の図は、非CDBでDBMS_PDB.DESCRIBEファンクションを実行してから、非CDBファイルを使用してPDBを作成する方法を示しています。

図1-10 非CDBからのPDBの作成



関連項目:

- [「非CDBからPDBを作成するためのオプション」](#)
- Oracle GoldenGateの詳細は、[『Oracle Database概要』](#)を参照してください

親トピック: [プラグインによるPDBの作成](#)

再配置によるPDBの作成

PDBをあるCDBから別のCDBに再配置するには、CREATE PLUGGABLE DATABASE ... RELOCATE文またはDBCAを使用します。

この方法には次のメリットがあります。

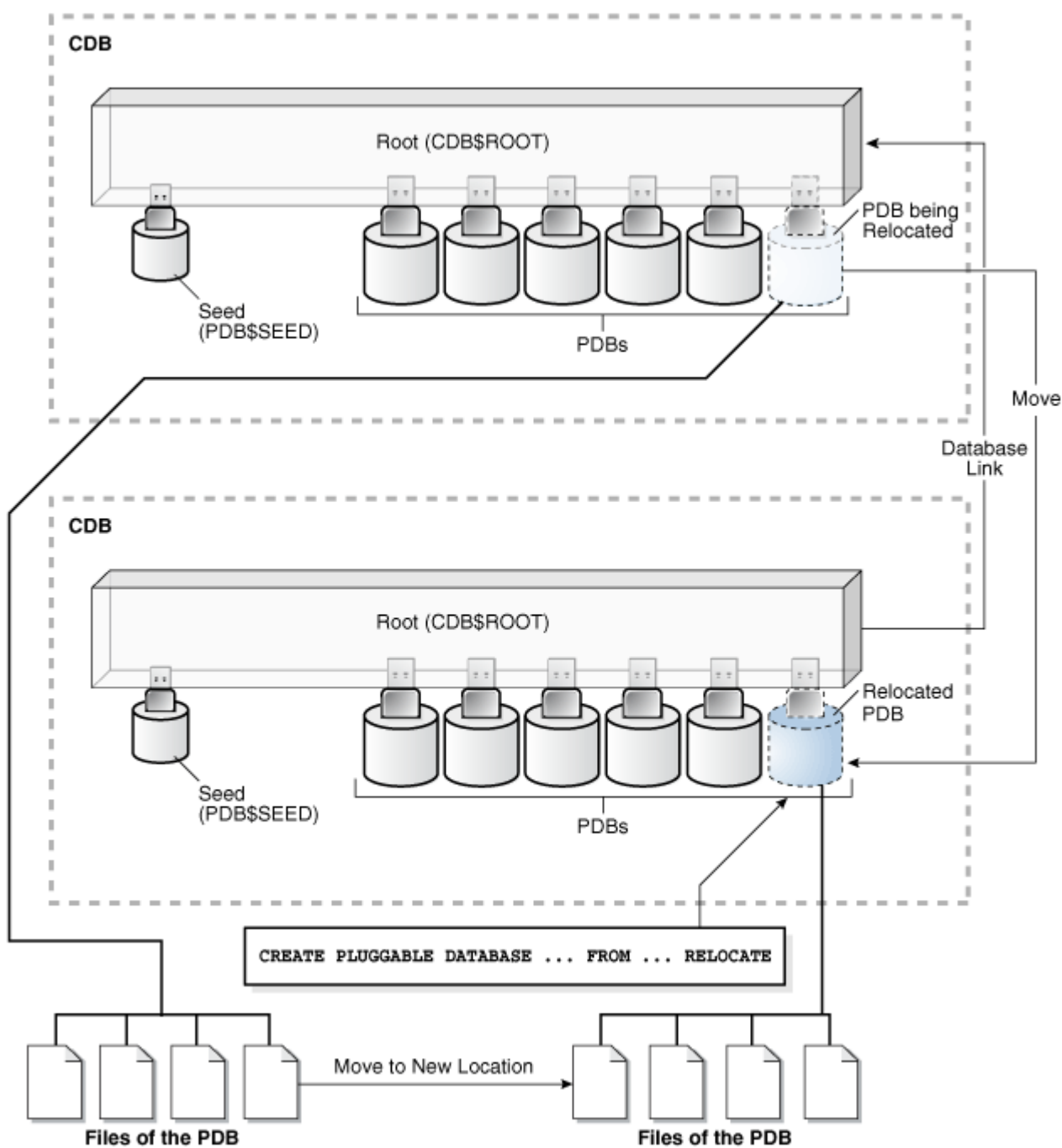
- 再配置が最小の停止時間内で行われます。
- この手法によって再配置中のPDBは再配置の間は読取り/書込みモードで開かれたままになり、その新しい場所ではPDBがオンラインに戻されます。

ターゲットCDB(再配置されたPDBが含まれるCDB)では、データベース・リンクを作成する必要があります。また、ソースPDBは

ローカルUNDOデータを使用する必要があります。

次の図は、PDBの再配置を示しています。

図1-11 PDBの再配置



Oracle Database 19c以降では、サイレント・モードでDBCAを使用してリモートPDBを再クローニングできます。

例1-7 PDBの再配置

ターゲットCDBで発行される次の文によって、hrpdbがソースCDBからターゲットCDBに再配置されます。

```
CREATE PLUGGABLE DATABASE hrpdb FROM hrpdb@lnk_to_source RELOCATE;
```

関連項目:

- [「CDBの表領域およびデータベース・ファイルの概要」](#)
- [「PDBの再配置」](#)

親トピック: [PDBの作成](#)

プロキシPDBとしてのPDBの作成

プロキシPDBは、**参照先PDB**と呼ばれるリモートCDB内の別のPDBへのアクセスを提供します。

プロキシPDBによって複数のソースからデータを集約することができます。プロキシPDBで実行のために発行されたSQL文は、参照先PDB内で実行されます。

一般的なユースケースは、アプリケーション・ルート・レプリカを参照するプロキシPDBです。複数のCDBに同じアプリケーション定義(たとえば、同じ表とPL/SQLパッケージ)がある場合、マスター・アプリケーション・ルートのアプリケーション・コンテナでプロキシPDBを作成できます。プロキシPDBに対する参照先PDBは、異なるCDB内のアプリケーション・ルートです。インストール・スクリプトをマスター・ルートで実行することで、他のCDB内のアプリケーション・ルートがマスター・アプリケーション・ルートのレプリカになります。

プロキシPDBを作成するには、FROM句(リモートCDB内の参照先PDBへのデータベース・リンクを指定する必要があります)およびAS PROXY句とともにCREATE PLUGGABLE DATABASE文を使用します。

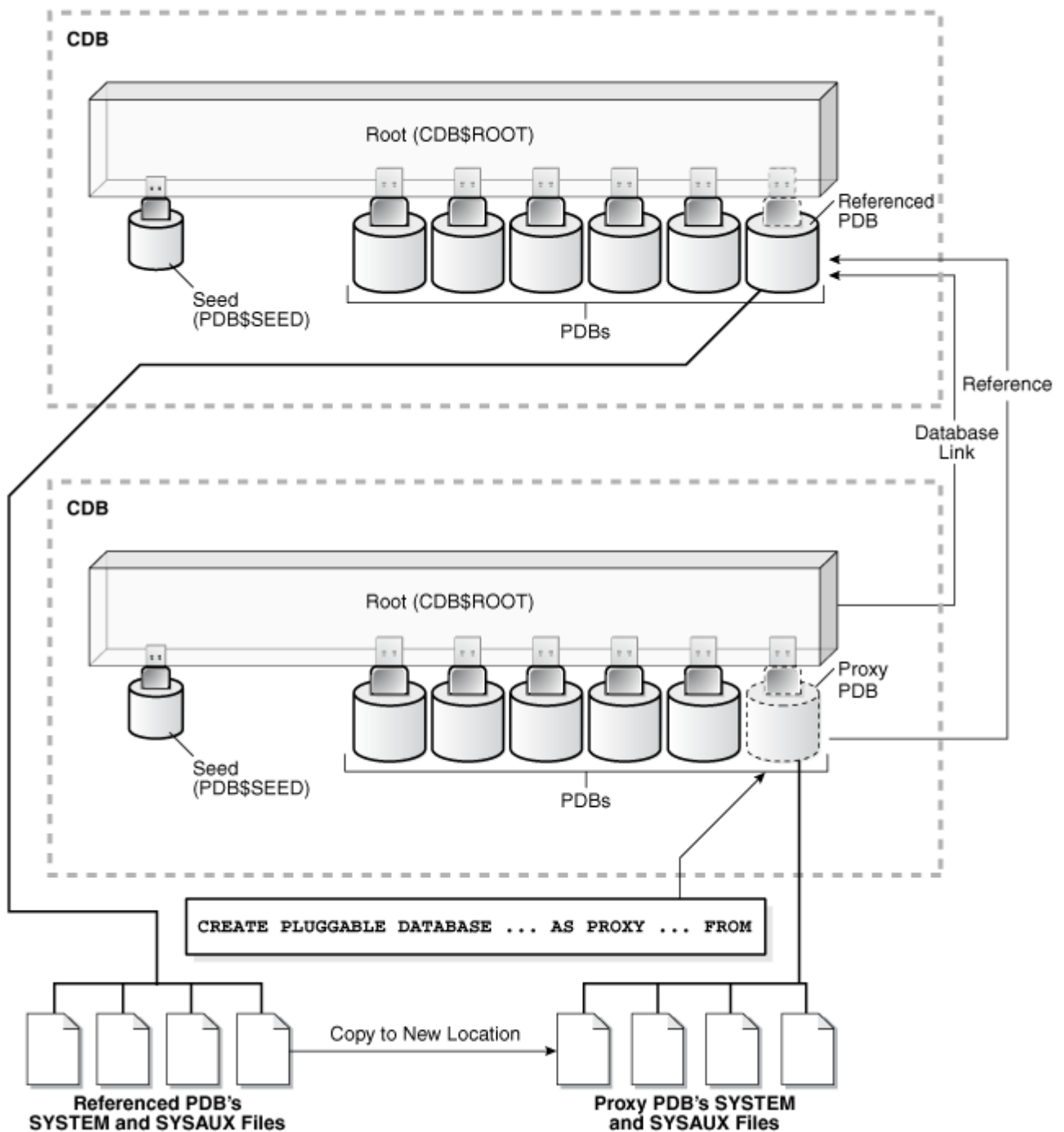
ノート:



プロキシ PDB を CDB\$ROOT に直接接続する場合は、CDB\$ROOT にプロキシを作成しておく必要があります。アプリケーション PDB のプロキシも、アプリケーション・ルートに接続されている必要があります。

次の図に、リモートCDB内のPDBを参照するプロキシPDBの作成方法を示します。

図1-12 プロキシPDBの作成



例1-8 プロキシPDBの作成

この例では、pdb1というプロキシPDBを作成します。参照先PDBは、データベース・リンクを使用して指定されます。

```
CREATE PLUGGABLE DATABASE pdb1 AS PROXY FROM pdb1@pdb1_link;
```

ノート:



[「プロキシ PDB としての PDB の作成」](#)

親トピック: [PDBの作成](#)

マルチテナント環境のドキュメント・ロードマップ

このトピックに、CDBの理解および使用にとって最も重要なトピックと、該当するドキュメントへの相互参照を示します。

表1-2 マルチテナント・アーキテクチャ・ドキュメントのロードマップ

カテゴリ	トピック	ドキュメント
概要	CDB および PDB の概要	「マルチテナント・アーキテクチャの概要」
管理	CDB の作成および構成	「CDB の作成および構成」
管理	CDB の管理	「CDB の管理」
管理	PDB の作成および構成	「PDB およびアプリケーション・コンテナの作成および削除」
管理	PDB の管理	「PDB の管理」
管理	アプリケーション・コンテナの作成および削除	「アプリケーション・コンテナの作成および削除」
管理	アプリケーション・コンテナの管理	「アプリケーション・コンテナの管理」
パフォーマンス	PDB のトラブルシューティング	Oracle Database パフォーマンス・チューニング・ガイド
監視	CDB および PDB に関する情報の表示	「CDB および PDB の監視」
バックアップおよびリカバリ	CDB でのバックアップおよびリカバリの実行	『Oracle Database バックアップおよびリカバリ・アドバンスド・ユーザーズ・ガイド』
セキュリティ	CDB での共通ユーザー、ロールおよび権限の管理	『Oracle Database セキュリティ・ガイド』
その他	CDB または PDB の管理、Oracle RAC のインクルード、リソース管理、データ転送などに関連するその他すべてのタスク	このガイドは、CDB 管理用の主要なタスク指向の中級および上級ドキュメントです。このガイドには、様々な CDB のトピックを取り上げている書籍への参照リンクも含まれています。たとえば、 『Oracle Database ユーティリティ』 では、Oracle Data Pump を使用する際の PDB 固有の概念およびタスクについて説明しています。

親トピック: [マルチテナント・アーキテクチャの紹介](#)

2 マルチテナント・アーキテクチャの概要

この章では、マルチテナント・アーキテクチャの最も重要なコンポーネントについて説明します。

- [CDBのコンテナの概要](#)
コンテナは、マルチテナント・コンテナ・データベース(CDB)内のスキーマ、オブジェクトおよび関連構造の集合です。CDB内では、各コンテナは一意のIDと名前を持ちます。
- [CDBの共通性の概要](#)
CDBでは、すべてのユーザー、ロールまたはオブジェクトが共通またはローカルのいずれかになります。同様に、特権は共通に付与されるか、ローカルに付与されるかのいずれかです。
- [アプリケーション・コンテナ内のアプリケーションの概要](#)
アプリケーション・コンテナ内で、**アプリケーション**はアプリケーション・ルートに格納される共通データおよびメタデータの名前付けおよびバージョンングされたセットです。
- [CDBのサービスの概要](#)
クライアントは、サービスを使用してPDBまたはアプリケーション・ルートに接続する必要があります。
- [CDBの表領域およびデータベース・ファイルの概要](#)
CDBの構造は非CDBと同じですが、各PDBおよびアプリケーション・ルートに独自の表領域のセット(独自のSYSTEM、SYSAUXおよびUNDO表領域を含む)がある点が異なります。
- [CDBの可用性の概要](#)
非CDBについて存在する可用性機能の多くは、CDB内の個々のPDBについても存在します。
- [CDBでのOracle Resource Managerの概要](#)
Oracle Resource Manager (Resource Manager)を使用して、CDBリソース・プランを作成し、PDBにリソースを割り当てるための初期化パラメータを設定できます。

親トピック: [マルチテナント・アーキテクチャ](#)

CDBのコンテナの概要

コンテナは、マルチテナント・コンテナ・データベース(CDB)内のスキーマ、オブジェクトおよび関連構造の集合です。CDB内では、各コンテナは一意のIDと名前を持ちます。

- [CDBルートとシステム・コンテナ](#)
CDBルート(単純にルートとも呼ばれている)は、スキーマ、スキーマ・オブジェクトおよび非スキーマ・オブジェクトの集合で、すべてのPDBはこれに属します。
- [PDB](#)
PDBは、ユーザーが作成したスキーマ、オブジェクトおよび関連構造体のセットで、論理的にクライアント・アプリケーションでは個別のデータベースとして表示されます。
- [CDBのデータ・ディクショナリ・アーキテクチャ](#)
ユーザーおよびアプリケーションから見て、CDBの各コンテナのデータ・ディクショナリは、非CDBの場合と同様に分かれています。
- [現在のコンテナ](#)
指定されたセッションで、現在のコンテナは、セッションが実行されているコンテナです。現在のコンテナはCDBルート、アプリケーション・ルートまたはPDBになります。
- [コンテナ間操作](#)
コンテナ間操作は、1回で複数のコンテナに影響を及ぼすDDL文またはDML文です。

親トピック: [マルチテナント・アーキテクチャの概要](#)

CDBルートとシステム・コンテナ

CDBルート(単純にルートとも呼ばれている)は、スキーマ、スキーマ・オブジェクトおよび非スキーマ・オブジェクトの集合で、すべてのPDBはこれに属します。

どのCDBにも、CDB\$ROOTという名前のルート・コンテナが1つだけあります。ルートにはPDBの管理に必要なシステム・メタデータが格納されます。すべてのPDBはこのルートに属します。[システム・コンテナ](#)は、CDBルートとこのルートに属するすべてのPDBです。

CDBルートにはユーザー・データは格納されません。共通オブジェクトをルートに追加したり、ルート内のOracle提供スキーマを変更したりしないことをお勧めします。ただし、データベース管理用の共通ユーザーおよびロールを作成できます。必要な権限を持った共通ユーザーは、コンテナ間の切替えが可能です。

ルート文字セットとして、AL32UTF8をお勧めします。文字セット変換をしなくても、異なる文字セットを使用するPDBが同じCDB内に存在できます。

例2-1 CDBのすべてのコンテナ

CDBルートに接続された管理ユーザーによって発行される次の問合せによって、CDB内のすべてのコンテナ(シードおよびCDBルートを含む)がCON_IDの順序でリストされます。

```
COL NAME FORMAT A15
SELECT NAME, CON_ID, DBID, CON_UID, GUID
FROM V$CONTAINERS ORDER BY CON_ID;
```

NAME	CON_ID	DBID	CON_UID	GUID
CDB\$ROOT	1	1895287725	1	2003321EDD4F60D6E0534E40E40A41C5
PDB\$SEED	2	2795386505	2795386505	200AC90679F07B55E05396C0E40A23FE
SAAS_SALES_AC	3	1239646423	1239646423	200B4CE0A8DC1D24E05396C0E40AF8EE
SALESPDB	4	3692549634	3692549634	200B4928319C1BCCE05396C0E40A2432
HRPDB	5	3784483090	3784483090	200B4928319D1BCCE05396C0E40A2432

関連項目:

- [「CDBの共通ユーザー」](#)
- [「マルチテナント環境の構成と管理の概要」](#)

親トピック: [CDBのコンテナの概要](#)

PDB

PDBは、ユーザーが作成したスキーマ、オブジェクトおよび関連構造体のセットで、論理的にクライアント・アプリケーションでは個別のデータベースとして表示されます。

あらゆるPDBは、PDBを作成したユーザーに関係なくSYSによって所有されます。CDBでは、SYSは[共通ユーザー](#)です。

- [PDBのタイプ](#)
すべてのPDBは、Oracle提供のPDB\$SEEDを除いて、CREATE PLUGGABLE DATABASE文でユーザー作成されます。
- [PDBの目的](#)
アプリケーションでは、PDBは、自己完結型の、すべての機能を利用可能なOracleデータベースです。PDBを1つのCDBに統合することで、PDBの分離を維持しながら規模の経済性が得られます。

- [プロキシPDB](#)
プロキシPDBは、参照先PDBと呼ばれるリモートPDBを参照します。
- [PDBの名前](#)
CDB内のコンテナは同じ名前スペースを共有します。つまり、この名前スペース内では一意の名前を保持する必要があります。
- [PDB間のデータベース・リンク](#)
デフォルトで、あるPDBに接続しているユーザーが、別のPDBのオブジェクトにアクセスするには、データベース・リンクを使用する必要があります。この動作は、まさに別の非CDBのオブジェクトにアクセスする非CDBのユーザーに似ています。

親トピック: [CDBのコンテナの概要](#)

PDBのタイプ

すべてのPDBは、Oracle提供のPDB\$SEEDを除いて、CREATE PLUGGABLE DATABASE文でユーザー作成されます。

次のタイプのPDBを作成できます。

標準PDB

このタイプのPDBは、シード、プロキシPDBまたは[アプリケーション・ルート](#)としてPDBを指定せずに、CREATE PLUGGABLE DATABASEを実行することによって生成されます。その機能は作成する対象のコンテナによって異なります。

- [CDBルート](#)に接続されているPDB
このPDBはCDBルート・コンテナに属しており、[アプリケーション・コンテナ](#)には属していません。このタイプのPDBは、アプリケーション共通オブジェクトを使用できません。「[アプリケーション共通オブジェクト](#)」を参照してください。
- アプリケーションPDB
[アプリケーションPDB](#)は1つのアプリケーション・コンテナのみに属しています。CDBルートに接続しているPDBとは異なり、アプリケーションPDBはアプリケーション・コンテナ内のマスター・[アプリケーション](#)定義を共有できます。たとえば、アプリケーション・ルート内のusa_zipcodes表は[データリンク共通オブジェクト](#)になる場合があり、ここに含まれているデータはこのルートに接続しているすべてのアプリケーションPDBによってアクセスが可能です。アプリケーション・コンテナに存在しないPDBは、そのアプリケーション共通オブジェクトにアクセスできません。

アプリケーション・ルート

アプリケーション・ルートは、アプリケーション固有のルート・コンテナとみなされます。アプリケーション・バックエンドのマスター定義(共通データとメタデータを含む)用のリポジトリとして機能します。アプリケーション・コンテナを作成するには、CDBルートに接続し、CREATE PLUGGABLE DATABASE文にAS APPLICATION CONTAINER句を指定します。「[アプリケーション・ルート](#)」を参照してください。

シードPDB

標準PDBとは異なり、シードPDBはアプリケーションのサポートを目的としていません。むしろ、シードはアプリケーションをサポートするPDBを作成するためのテンプレートです。シードは次のいずれかです。

- CDBルートに接続しているシードPDB(PDB\$SEED)
このシステム付属テンプレートを使用して、アプリケーション・コンテナまたはシステム・コンテナのいずれかで新規のPDBを作成できます。システム・コンテナに含まれるPDBシードは1つのみです。PDB\$SEEDは、削除したり、シードPDBに対してオブジェクトを追加または変更することはできません。
- アプリケーション・シードPDB
アプリケーション・コンテナ内のアプリケーションPDBの作成を促すため、オプションの[アプリケーション・シード](#)を作成できま

す。アプリケーション・コンテナにはゼロまたは1つのアプリケーション・シードが含まれます。

アプリケーション・シードは、アプリケーション・コンテナに接続してCREATE PLUGGABLE DATABASE ... AS SEED文を実行することで作成します。[「アプリケーション・シード」](#)を参照してください。

プロキシPDB

[プロキシPDB](#)は、データベース・リンクを使用してリモートCDB内のPDBを参照するPDBです。PDBが開いている間にプロキシPDBで文を発行すると、その文は参照先PDB内で実行されます。

プロキシPDBは、CDBルートまたはアプリケーション・ルートへの接続中に作成する必要があります。プロキシPDBは、標準PDBと同じように変更または削除ができます。

関連項目:

[「PDB作成の概要」](#)

親トピック: [PDB](#)

PDBの目的

アプリケーションでは、PDBは、自己完結型の、すべての機能を利用可能なOracleデータベースです。PDBを1つのCDBに統合することで、PDBの分離を維持しながら規模の経済性が得られます。

PDBを使用すると、次の特定の目標を達成できます。

- アプリケーション固有のデータの格納
たとえば、販売アプリケーションは独自の専用PDBを持ち、人事管理アプリケーションも独自の専用PDBを持つことができます。もう1つの方法として、[アプリケーション・コンテナ](#)(PDBの名前付きコレクション)を作成して、共通のデータおよびメタデータが含まれるアプリケーション・バックエンドを格納できます([「アプリケーション・コンテナについて」](#)を参照)。
- 異なるCDBへのデータの移動
データベースがプラグブルなのは、それを自己完結型のユニットとしてパッケージ化([切断されたPDB](#)と呼ばれる)し、別のCDBに移動できるからです。
- 迅速なアップグレードの実行
旧リリースのOracle DatabaseのCDBからPDBを切断してから、新リリースのCDBにそのPDBを接続できます。
- 可用性を維持しながらも迅速なデータのコピー
テストおよび開発のために、PDBを開いている状態のままでクローニングして、そのクローンを同じまたは別のCDB内に格納できます。オプションで、[リフレッシュ可能クローンPDB](#)としてPDBを指定できます。また、Oracle提供の[シードPDB](#)またはユーザー作成[アプリケーション・シード](#)を使用して新しいPDBをコピーすることもできます。
- 異なるCDB内の参照データ
同じCDBまたは別のCDB内のいずれかで、異なるPDBを参照する[プロキシPDB](#)を作成できます。プロキシPDBで文を発行すると、その文は参照先PDB内で実行されます。
- PDB内での付与の分離
適切な権限を持つローカルまたは共通ユーザーは、スキーマ・オブジェクトのEXECUTE権限を個々のPDB内のPUBLICに付与できます。

関連項目:

- [「マルチテナント・アーキテクチャの利点」](#)
- CDBでロールおよび権限を付与する方法を学習するには、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [PDB](#)

プロキシPDB

プロキシPDBは、参照先PDBと呼ばれるリモートPDBを参照します。

プロキシ(参照元)PDBでSQL文を発行する場合でも、その文は参照先PDB内で実行されます。この点において、プロキシPDBはLinuxのシンボリック・リンク・ファイルと大まかに似ています。

プロキシPDBには、次の利点があります。

- 複数のアプリケーション・モデルからデータを集計します
プロキシPDBを使用すると、複数のソースからデータを集計できる位置透過的なアプリケーションを構築できます。これらのソースは、同じデータ・センター内にあっても、複数のデータ・センターに分散していてもかまいません。
- 1つのCDB内のアプリケーション・ルートが、別のアプリケーション・ルートにアプリケーションの変更を伝播できるようになります
CDBのcdb_prodおよびcdb_testのアプリケーション・モデルが同じであると仮定します。cdb_testのアプリケーション・ルートを参照するプロキシPDBをcdb_prodのアプリケーション・コンテナに作成します。cdb_prodのアプリケーション・ルートでインストールおよびアップグレードの скрипт を実行すると、Oracle Databaseではこれらの文がプロキシPDBに伝播され、これは次にcdb_testのアプリケーション・ルートにリモートで送信されます。このように、cdb_testのアプリケーション・ルートがcdb_prodのアプリケーション・ルートのレプリカになります。

プロキシPDBを作成するには、AS PROXY FROM句を含むCREATE PLUGGABLE DATABASEを実行します(FROMは参照先PDB名およびデータベース・リンクを指定します)。作成文では、SYSTEMおよびSYSAUX表領域に属すデータ・ファイルのみがコピーされます。

例2-2 プロキシPDBの作成

この例ではローカル本番CDBのコンテナsaas_sales_acに接続します。sales_admin共通ユーザーは、sales_sync_pdbというプロキシPDBを作成します。このアプリケーションPDBは、cdb_dev_remデータベース・リンクを使用してアクセスするリモート開発CDBの、saas_sales_test_acというアプリケーション・ルートを参照します。本番CDBのsaas_sales_acでアプリケーションのアップグレードが発生すると、リモート開発CDBのアプリケーション・ルートsaas_sales_test_acにアップグレードが自動的に伝播されます。

```
CONNECT sales_admin@saas_sales_ac  
Password: *****
```

```
CREATE PLUGGABLE DATABASE sales_sync_pdb AS PROXY FROM saas_sales_test_ac@cdb_dev_rem;
```



ノート:

親トピック: [PDB](#)

PDBの名前

CDB内のコンテナは同じネームスペースを共有します。つまり、このネームスペース内では一意の名前を保持する必要があります。次のコンテナの名前は同じCDB内で競合しないことが必要です。

- CDBルート
- CDBルートに接続されているPDB
- アプリケーション・ルート
- アプリケーションPDB

たとえば、同じCDBにアプリケーション・コンテナsaas_sales_acとsaas_sales_test_acが含まれる場合、両方ともcust1と名付けられた2つのアプリケーションPDBが両方のコンテナに同時に存在することはできません。このネームスペース・ルールではCDBルートでのcust1pdbというPDBの作成と、アプリケーション・ルートでのcust1pdbというPDBの作成も妨げられます。

PDBおよびアプリケーション・ルート・コンテナの名前は、ネット・サービス名と同じルールに従う必要があります。さらに、PDBまたはアプリケーション・ルートは独自の名前のサービスを持つため、コンテナ名は、特定のリスナーを介して公開されるサービスを所有するすべてのCDBにわたって一意にする必要があります。ユーザー作成のコンテナ名は最初の文字を英字にし、それ以降の文字を英数字またはアンダースコア(_)にする必要があります。サービス名には大文字小文字の区別がないため、コンテナ名にも大文字と小文字の区別はなく、区切り識別子を使用して指定しても、大文字になります。

関連項目:

サービス名のルールの詳細は、[『Oracle Database Net Servicesリファレンス』](#)を参照

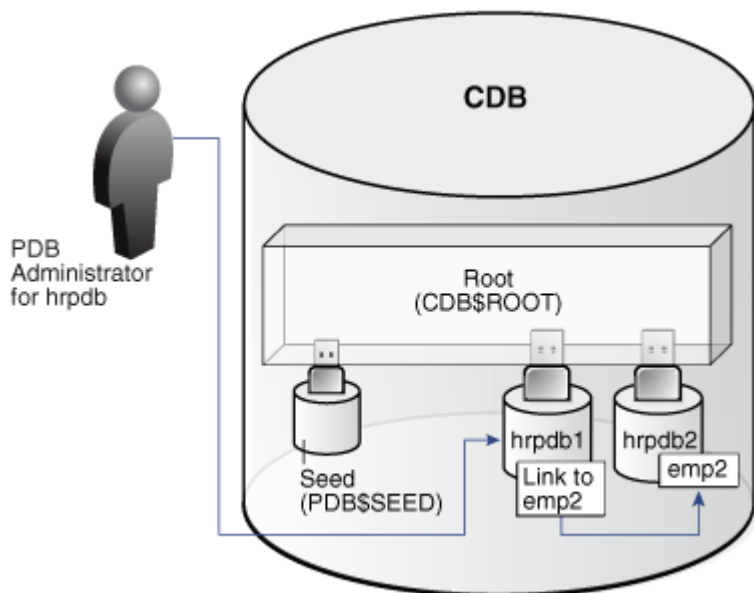
親トピック: [PDB](#)

PDB間のデータベース・リンク

デフォルトで、あるPDBに接続しているユーザーが、別のPDBのオブジェクトにアクセスするには、データベース・リンクを使用する必要があります。この動作は、まさに別の非CDBのオブジェクトにアクセスする非CDBのユーザーに似ています。

図2-1 PDB間のデータベース・リンク

この図で、PDB管理者はhrpdb1というPDBに接続されています。デフォルトで、このユーザー・セッション中に、c##dbaがデータベース・リンクを指定せずにhrpdb2のemp2表に問合せすることはできません。



このルールには、次の例外があります。

- [データリンク共通オブジェクト](#)は、このオブジェクトを示す[データ・リンク](#)が含まれるすべてのアプリケーションPDBによってアクセス可能です。たとえば、アプリケーション・コンテナsaas_sales_acではそのアプリケーション内にデータリンク表 usa_zipcodesが含まれる場合があります。この場合、共通CDBユーザーc##dbaはこのコンテナ内のアプリケーションPDBに接続して、実際の表がアプリケーション・ルートに存在する場合でもusa_zipcodesに問合せすることができます。この場合は、データベース・リンクは必要ありません。
- CDBルートまたはアプリケーション・ルートから発行されるSQLのCONTAINERS () 句。この句を使用すると、ルートに接続されたすべてのPDB間でデータを問い合わせることができます。

プロキシPDBを作成するとき、CREATE PLUGGABLE DATABASE ... AS PROXY文のFROM句でデータベース・リンク名を指定する必要があります。プロキシPDBと参照先PDBが別々のCDBに存在する場合は、プロキシPDBが含まれるCDBのルート内でデータベース・リンクが定義されている必要があります。データベース・リンクは、リモート参照先PDB、またはリモートCDBのCDBルートのいずれかに接続する必要があります。

関連項目:

- [「CDBの共通オブジェクトおよびローカル・オブジェクトの概要」](#)
- データベース・リンクを使用して別のPDBのオブジェクトにアクセスする方法を学習するには、[「PDB管理について」](#)を参照してください

親トピック: [PDB](#)

CDBのデータ・ディクショナリ・アーキテクチャ

ユーザーおよびアプリケーションから見て、CDBの各コンテナのデータ・ディクショナリは、非CDBの場合と同様に分かれています。たとえば、各PDBのDBA_OBJECTSビューでは、異なる行数を表示できます。このようにディクショナリを分けることで、Oracle DatabaseではPDBを、お互いから別々に、そしてルートから管理できます。

- [データ・ディクショナリ分割の目的](#)
ユーザー・データが含まれていない新規に作成された非CDBでは、データ・ディクショナリにはシステム・メタデータのみが含まれています。たとえば、TAB\$表には、Oracle提供の表(TRIGGER\$およびSERVICE\$など)のみを説明する行が含ま

れます。

- [メタデータ・リンクとデータ・リンク](#)

CDBは、内部リンク・メカニズムを使用して、データ・ディクショナリ情報を分けます。

- [CDBのコンテナ・データ・オブジェクト](#)

コンテナ・データ・オブジェクトは、複数のコンテナまたはCDB全体に関連するデータを含む表またはビューです。

- [CDBのデータ・ディクショナリ・ストレージ](#)

CDB全体のメタデータを格納するデータ・ディクショナリは、システム表領域にのみ格納されます。

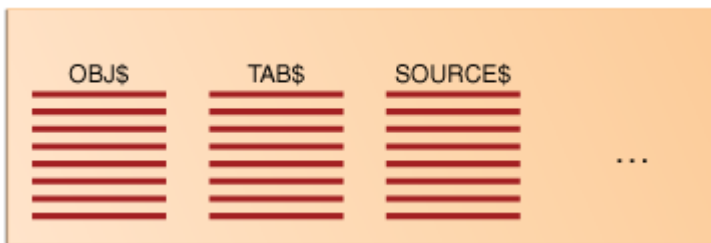
親トピック: [CDBのコンテナの概要](#)

データ・ディクショナリ分割の目的

まだユーザー・データが含まれていない新規に作成された非CDBでは、データ・ディクショナリにはシステム・メタデータのみが含まれています。たとえば、TAB\$表には、Oracle提供の表(TRIGGER\$およびSERVICE\$など)のみを説明する行が含まれます。

次の図は、3つの基礎的なデータ・ディクショナリ表で、赤い棒はシステムを説明する行を示しています。

図2-2 非CDBの非混合データ・ディクショナリ・メタデータ



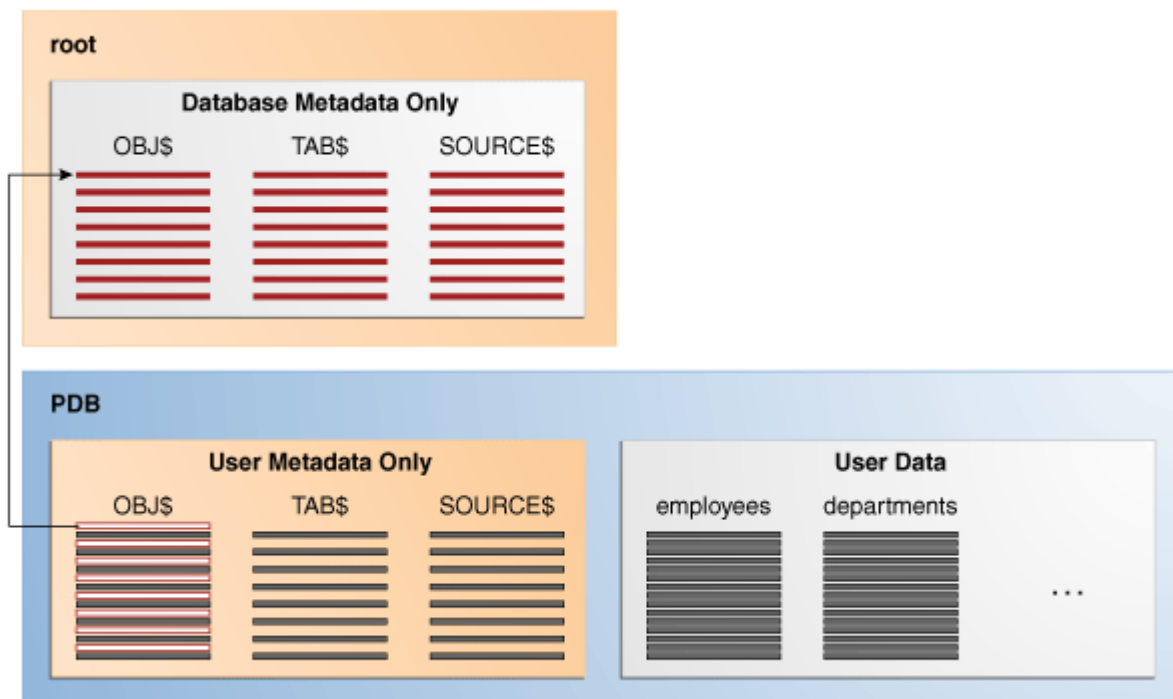
ユーザーがこの非CDBに自身のスキーマおよび表を作成する場合、データ・ディクショナリにはOracle提供のエンティティを説明する行と、ユーザー作成エンティティを説明するその他の行が含まれます。たとえば、TAB\$ディクショナリ表には、employeesを説明する行とdepartmentsを説明する行があります。

図2-3 非CDBの混合データ・ディクショナリ・メタデータ



CDBでは、データ・ディクショナリ・メタデータは、ルートとPDBの間で分割されています。次の図では、employees表とdepartments表がPDB内に存在します。このユーザー・データのデータ・ディクショナリも、PDBに常駐します。したがって、PDBのTAB\$表には、employees表の行とdepartments表の行があります。

図2-4 CDBのデータ・ディクショナリ・アーキテクチャ



前図は、PDBのデータ・ディクショナリには、ルートにあるデータ・ディクショナリに対するポインタが含まれていることを示しています。内部では、データ・ディクショナリ表の定義やPL/SQLパッケージなど、Oracle提要オブジェクトは、ルートでのみ表されます。このアーキテクチャは、CDBで2つの主な目的を達成します。

- 重複の減少

たとえば、すべてのPDBにDBMS_ADVISOR PL/SQLパッケージのソース・コードを格納するかわりに、CDBではそれがCDB\$ROOTにのみ格納され、ディスク領域が節約されます。

- データベースのアップグレードの容易さ

データ・ディクショナリ表の定義がすべてのPDBに存在し、その定義が新リリースで変わる場合、その変更を反映するには、各PDBを別々にアップグレードする必要があります。表定義をルートに1回格納すれば、この問題はなくなります。

親トピック: [CDBのデータ・ディクショナリ・アーキテクチャ](#)

メタデータ・リンクとデータ・リンク

CDBは、内部リンク・メカニズムを使用して、データ・ディクショナリ情報を分けます。

具体的には、Oracle Databaseでは、自動的に管理される次のポインタを使用します。

- メタデータ・リンク

Oracle Databaseでは、ディクショナリ・オブジェクトのメタデータは、CDBルート内にしか格納されません。たとえば、DBA_OBJECTSデータ・ディクショナリ・ビューの基礎であるOBJ\$ディクショナリ表の列定義は、ルートにしか存在しません。[図2-4](#)に示すように、各PDB内のOBJ\$表は[メタデータ・リンク](#)と呼ばれる内部メカニズムを使用して、ルートに格納されているOBJ\$の定義を指し示します。

メタデータ・リンクに対応するデータは、ルートではなく、そのPDBに常駐します。たとえば、hrpdbに表mytableを作成し、これに行を追加する場合、その行はPDBデータ・ファイルに格納されます。PDBおよびルートのデータ・ディクショナリ・ビューには、異なる行が含まれます。たとえば、mytableを説明する新しい行は、hrpdbのOBJ\$表に存在し、CDBルートのOBJ\$表には存在しません。したがって、CDBルートのDBA_OBJECTSの問合せと、hrpdbのDBA_OBJECTSでは、異なる結果が表示されます。

- データ・リンク



ノート:



データ・リンクは Oracle Database 12c リリース 1 (12.1.0.2)でオブジェクト・リンクと呼ばれていました。

場合によっては、Oracle Databaseで(メタデータのみではなく)オブジェクトのデータがアプリケーション・ルートに1回しか格納されません。アプリケーションPDBでは、[データ・リンク](#)と呼ばれる内部メカニズムを使用してアプリケーション・ルートのオブジェクトが参照されます。データ・リンクが作成されたアプリケーションPDBには、データ・リンクの説明も格納されます。データ・リンクは、参照先のオブジェクトのデータ型を継承します。

● 拡張データ・リンク

拡張データ・リンクは、データ・リンクとメタデータ・リンクを合成したものです。データ・リンクと同じように、拡張データ・リンクはアプリケーション・ルートのオブジェクトを参照します。しかし、拡張データ・リンクはアプリケーションPDBの対応するオブジェクトも参照します。メタデータ・リンクと同じように、アプリケーションPDBのオブジェクトはアプリケーション・ルートの対応するオブジェクトからメタデータを継承します。

アプリケーション・ルートで問合せを実行すると、拡張データリンク・オブジェクトはアプリケーション・ルートからのみ行をフェッチします。しかし、アプリケーションPDBで問合せを実行すると、拡張データリンク・オブジェクトはアプリケーション・ルートとアプリケーションPDBの両方から行をフェッチします。

Oracle Databaseでは、CDB\$ROOTに対するメタデータ・リンクとデータ・リンクが自動的に作成および管理されます。ユーザーは、これらのリンクを追加、変更または削除できません。

関連項目:

- [「アプリケーション共通オブジェクト」](#)
- データ・ディクショナリの概要は、[『Oracle Database概要』](#)を参照してください

親トピック: [CDBのデータ・ディクショナリ・アーキテクチャ](#)

CDBのコンテナ・データ・オブジェクト

コンテナ・データ・オブジェクトは、複数のコンテナまたはCDB全体に関連するデータを含む表またはビューです。

コンテナ・データの権限は、複数のPDBが1つのCDBに存在する一般的な要件をサポートしますが、異なるローカル管理の要件があります。たとえば、ローカル管理を行わないアプリケーションDBAは、適切なビューに対するコンテナ・データの権限を共有ユーザーに付与できます。この場合、CDB管理者はこれらのPDBのデータにアクセスできます。これに対して、CDB管理者がデータにアクセスしないようにする必要があるPDB管理者は、コンテナ・データの権限を付与しません。

名前がV\$およびCDB_で始まるOracle提供ビューは、コンテナ・データベース・オブジェクトの例です。すべてのコンテナ・データベース・オブジェクトには、CON_ID列があります。次の表に、この列の値の意味を示します。

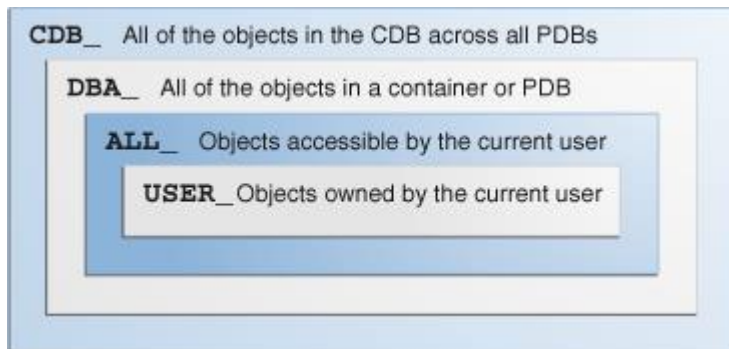
表2-1 コンテナID値

コンテナID	関連行
0	CDB 全体、または非 CDB

コンテナID	関連行
1	CDB\$ROOT
2	PDB\$SEED
その他すべての ID	ユーザー作成の PDB、アプリケーション・ルートまたはアプリケーション・シード

CDBでは、各DBA_ビューに、対応するCDB_ビューが存在します。CDB_ビューの所有者は、対応するDBA_ビューの所有者です。次の図は、異なるカテゴリのディクショナリ・ビューの関係を示しています。

図2-5 CDBのディクショナリ・ビュー



現在のコンテナがPDBである場合、ユーザーは現在のPDBのデータ・ディクショナリ情報のみを表示できます。PDBに接続されているアプリケーションには、データ・ディクショナリが非CDBの場合と同様に表示されます。ただし、現在のコンテナがルートである場合、共通ユーザーは、ルートとこのユーザーが権限を持つPDBのメタデータを確認するために、CDB_ビューに問い合わせることができます。

ノート:



ルート・コンテナから問合せされる際、CDB_ビューおよびV\$_ビューはデータをAL32UTF8文字セットに暗黙的に変換します。AL32UTF8に変換される際に文字セットで文字を表すためにバイト数がさらに必要な場合、そしてビューの列幅が特定のPDBのデータを収容できない場合、データの切捨てが可能です。

次の表に、CDB_ビューの問合せを含むシナリオを示します。各行は、前の行のアクションの後に発生するアクションについて説明します。

表2-2 CDB_ビューの問合せ

操作	説明
SQL> CONNECT SYSTEM Enter password: ***** Connected.	SYSTEMユーザーは、CDBのすべてのコンテナに対して共通で、ルートに接続します(「CDBの共通ユーザー」 を参照)。
SQL> SELECT COUNT(*) FROM CDB_USERS WHERE CON_ID=1; COUNT(*) ----- 41	SYSTEMはCDB_USERSに対して、CDBの共通ユーザー数を取得するように問い合わせます。出力は、41の共通ユーザーが存在することを示しています。

操作	説明
<pre>SQL> SELECT COUNT(DISTINCT(CON_ID)) FROM CDB_USERS; COUNT(DISTINCT(CON_ID)) ----- ----- 4</pre>	SYSTEM は CDB_USERS に対して、CDB の個別のコンテナ数を確認するように問い合わせます。
<pre>SQL> CONNECT SYSTEM@hrdb Enter password: ***** Connected.</pre>	SYSTEM ユーザーは、今度は hrpdb という PDB に接続します。
<pre>SQL> SELECT COUNT(*) FROM CDB_USERS; COUNT(*) -- ----- 45</pre>	SYSTEM は CDB_USERS に対して問い合わせます。出力は、45 の共通ユーザーが存在することを示しています。SYSTEM はルートに接続していないため、CDB_USERS ビューには DBA_USERS と同じ出力が表示されます。DBA_USERS には現行コンテナ内のユーザーのみが表示されるため、45 と表示されます。

関連項目:

コンテナ・データ・オブジェクトについてさらに学習するには、[「ビュー内のCDBおよびコンテナの情報について」](#)を参照してください

親トピック: [CDBのデータ・ディクショナリ・アーキテクチャ](#)

CDBのデータ・ディクショナリ・ストレージ

CDB全体のメタデータを格納するデータ・ディクショナリは、システム表領域にのみ格納されます。

特定のPDBのメタデータを格納するデータ・ディクショナリは、このPDB専用の自己完結型表領域に格納されます。PDB表領域には、アプリケーション・バックエンドのデータとメタデータの両方が格納されます。このため、データ・ディクショナリ表の各セットは、それぞれの専用の表領域のセットに格納されます。

関連項目:

- データ・ディクショナリの概要は、[『Oracle Database概要』](#)を参照してください
- [「CDBの表領域およびデータベース・ファイルの概要」](#)

親トピック: [CDBのデータ・ディクショナリ・アーキテクチャ](#)

現在のコンテナ

指定されたセッションで、現在のコンテナは、セッションが実行されているコンテナです。現在のコンテナはCDBルート、アプリケーション・ルートまたはPDBになります。

各セッションには、任意の時点で現在のコンテナがそれぞれ1つのみ含まれます。各コンテナのデータ・ディクショナリは個別のため、Oracle Databaseでは、名前の解決と権限の認証に、現在のコンテナのデータ・ディクショナリを使用します。

関連項目:

親トピック: [CDBのコンテナの概要](#)

コンテナ間操作

コンテナ間操作は、1回で複数のコンテナに影響を及ぼすDDL文またはDML文です。

CDBルートまたはアプリケーション・ルートのいずれかに接続されている共通ユーザーのみが、コンテナ間操作を実行できます。コンテナ間操作は次の対象に影響を及ぼします。

- CDB自体
- CDB内の複数のコンテナ
- 複数のコンテナで表される共通ユーザーや共通ロールなどの複数の現象
- DDL文またはDML文を発行しているユーザーが現在接続されていないコンテナ

コンテナ間DDL操作の例として、ユーザーSYSTEMが別の共通ユーザーに共通に権限を付与したり([「CDBで共通に付与されるロールおよび権限」](#)を参照)、ALTER DATABASE . . . RECOVER文をCDB全体に適用します。

CDBルートまたはアプリケーション・ルートのいずれかに接続しているユーザーは、1つのDML文を実行することでコンテナ内の複数のPDBで表またはビューを変更できます。データベースでは、DML文で指定されたCON_ID列の値からターゲットPDBが推測されます。CON_IDが指定されていない場合、データベースではALTER PLUGGABLE DATABASE CONTAINERS DEFAULT TARGET文で指定されたCONTAINERS_DEFAULT_TARGETプロパティが使用されます。

例2-3 1つのDML文における複数のPDBの更新

この例での目的は、sh. sales表でcountry_name列を値USAに設定することです。この表はコンテナID 7および8の2つの別個のPDBに存在します。両方のPDBがsaas_sales_acというアプリケーション・コンテナにあります。管理者としてアプリケーション・ルートに接続し、次のように更新できます。

```
CONNECT sales_admin@saas_sales_ac
Password: *****

UPDATE CONTAINERS(sh. sales) sal
  SET sal.country_name = 'USA'
  WHERE sal.CON_ID IN (7,8);
```

前述のUPDATE文で、salはCONTAINERS (sh. sales)の別名です。

関連項目:

- [「CDBの共通ユーザー」](#)
- [「CDBの管理」](#)

親トピック: [CDBのコンテナの概要](#)

CDBの共通性の概要

CDBでは、すべてのユーザー、ロールまたはオブジェクトが共通またはローカルのいずれかになります。同様に、特権は共通に付

与されるか、ローカルに付与されるかのいずれかです。

- [CDBの共通性について](#)
CDBまたはアプリケーション・ルートで定義されている共通現象は、このルートに接続されているすべてのコンテナでも同じです。
- [CDBの共通ユーザーおよびローカル・ユーザーの概要](#)
ユーザー・アカウントがデータベースを定義するオブジェクトを所有する場合、このユーザー・アカウントは共通です。Oracle提供ではないユーザー・アカウントはローカルまたは共通のいずれかです。
- [CDBの共通ロールおよびローカル・ロールの概要](#)
ユーザー作成のロールは、ローカルまたは共通のいずれかです。共通ロールは、CDB自体または特定のアプリケーション・コンテナのいずれかに共通します。
- [CDBでの権限およびロール付与の概要](#)
非CDBと同様に、ロールと権限をCDB内のユーザーが付与することも、CDB内のユーザーに付与することもできます。ただし、CDB内のロールおよび権限は、ローカルに付与されたものか、共通に付与されたもののいずれかです。
- [CDBの共通オブジェクトおよびローカル・オブジェクトの概要](#)
共通オブジェクトは、CDBルートまたはアプリケーション・ルートのいずれかで定義されており、メタデータ・リンクまたはオブジェクト・リンクを使用して参照できます。ローカル・オブジェクトは、共通オブジェクトではないすべてのオブジェクトです。
- [共通監査構成の概要](#)
混合モードと統合監査のどちらの場合も、共通監査構成が表示され、すべてのPDBにわたって適用されます。
- [PDBロックダウン・プロファイルの概要](#)
PDBロックダウン・プロファイルは、PDBに接続されたユーザーが使用できる操作を制限する一連の機能に名前を付けたものです。たとえば、PDBロックダウン・プロファイルによって、ALTER SYSTEM文に伴う権限を無効化できます。

親トピック: [マルチテナント・アーキテクチャの概要](#)

CDBの共通性について

CDBまたはアプリケーション・ルートで定義されている共通現象は、このルートに接続されているすべてのコンテナでも同じです。

- [共通性の原則](#)
CDBでは、ある現象はシステム・コンテナ(CDB自体)内または特定のアプリケーション・コンテナ内のいずれにも共通します。
- [CDBのネームスペース](#)
CDBでは、すべてのオブジェクトのネームスペースはそのオブジェクトのコンテナをスコープとしています。

親トピック: [CDBの共通性の概要](#)

共通性の原則

CDBでは、ある現象はシステム・コンテナ(CDB自体)内または特定のアプリケーション・コンテナ内のいずれにも共通します。

たとえば、CDB\$ROOTへの接続中に共通ユーザー・アカウントを作成した場合、このユーザー・アカウントはCDB内のすべてのPDBおよびアプリケーション・ルートにも共通です。しかし、アプリケーション・ルートへの接続中にアプリケーション共通ユーザー・アカウントを作成した場合、このユーザー・アカウントはこのアプリケーション・コンテナ内のPDBのみに共通です。

CDB\$ROOTまたはアプリケーション・ルートのコンテキストにおける共通性の原則は、次のとおりです。

- 共通現象は、既存および将来のすべてのコンテナで同じになります。
したがって、CDBルートで定義された共通ユーザーはCDBルートに接続しているすべてのPDBで同じIDを持ち、アプリケーション・ルートで定義された共通ユーザーはこのアプリケーション・ルートに接続しているすべてのアプリケーションPDBで同じIDを持ちます。一方、ローカル現象は、厳密に既存の1つのコンテナをスコープとしています。

- 共通ユーザーのみが共通現象の存在を変更できます。

もっと正確に言えば、CDBルートまたはアプリケーション・ルートのいずれかにログインした共通ユーザーのみが、現在のテナンに共通のユーザー、ロールまたはオブジェクトの属性を作成、無効化または変更できます。

親トピック: [CDBの共通性について](#)

CDBのネームスペース

CDBでは、すべてのオブジェクトのネームスペースはそのオブジェクトのテナンをスコープとしています。

次の原則は、スコープ指定のルールをまとめたものです。

- アプリケーションの観点からすると、PDBは非CDBと区別できません。
- ローカル現象は、1つのテナンの内部で作成され、1つのテナンに制限されます。



ノート:

このトピックでは、「現象」という用語は「ユーザー・アカウント、ロールまたはデータベース・オブジェクト」を意味します。

- 共通現象は、CDBルートまたはアプリケーション・ルートで定義され、このルートに接続されている(または今後接続される)すべてのPDBに存在します。

前述の原則は、ローカル現象および共通現象に影響します。

ローカル現象

ローカル現象は、テナンの内部で一意的な名前を付ける必要がありますが、CDBのテナン全体ではその必要はありません。異なるテナンに含まれる同じ名前を持つローカル現象は、別個のものです。たとえば、あるPDBのローカル・ユーザーshは、別のPDBのローカル・ユーザーshと競合しません。

CDB\$ROOTの共通現象

CDB\$ROOTで定義された共通現象は、複数のテナンに存在し、それらのネームスペースのそれぞれで一意的である必要があります。たとえば、CDBルートにはSYSTEMやSYSなどの事前定義された共通ユーザーが含まれています。Oracle Databaseでは、ネームスペースを確実に分離するため、別のテナン内でSYSTEMユーザーを作成できません。

ネームスペースを確実に分離するため、CDBルートでユーザーが作成する共通現象の名前は、COMMON_USER_PREFIX初期化パラメータによって指定された値で始まる必要があります。デフォルトの接頭辞はc##またはC##です。ユーザーが作成する他の現象にc##およびC##で始まる名前を付けることはできません。たとえば、hrpdbでc##hrという名前のローカル・ユーザーを作成したり、CDBルートでhrという名前の共通ユーザーを作成することはできません。

アプリケーション共通現象

アプリケーション・テナン内で、ローカルおよびアプリケーション共通の現象の名前が競合することはできません。

- アプリケーション共通ユーザーおよびロール

CDB共通ユーザーに対する同じ原則がアプリケーション共通ユーザーにも適用されます。違いは、CDB共通ユーザーの場合、共通ユーザー接頭辞のデフォルト値はc##またはC##であるのに対し、アプリケーション・ルートでは、共通ユーザー接頭辞のデフォルト値は空の文字列であることです。

マルチテナン・アーキテクチャでは、アプリケーション・ルートからアプリケーションPDBを作成するか、シングルテナン・ア

アプリケーションをマルチテナント・アプリケーションに変換することが想定されています。

- アプリケーション共通オブジェクト

マルチテナント・アーキテクチャでは、アプリケーション・ルートでアプリケーション共通オブジェクトを作成することが想定されています。その後、アプリケーションPDBの内部でローカルにデータを追加します。ただし、Oracle DatabaseではアプリケーションPDB内でのローカル・テーブルの作成がサポートされています。この場合、ローカル表はアプリケーションPDB内のアプリケーション共通オブジェクトと同じネームスペースに存在します。

関連項目:

共通ユーザーおよびロールについてさらに学習するには、[Oracle Databaseセキュリティ・ガイド](#)を参照してください

親トピック: [CDBの共通性について](#)

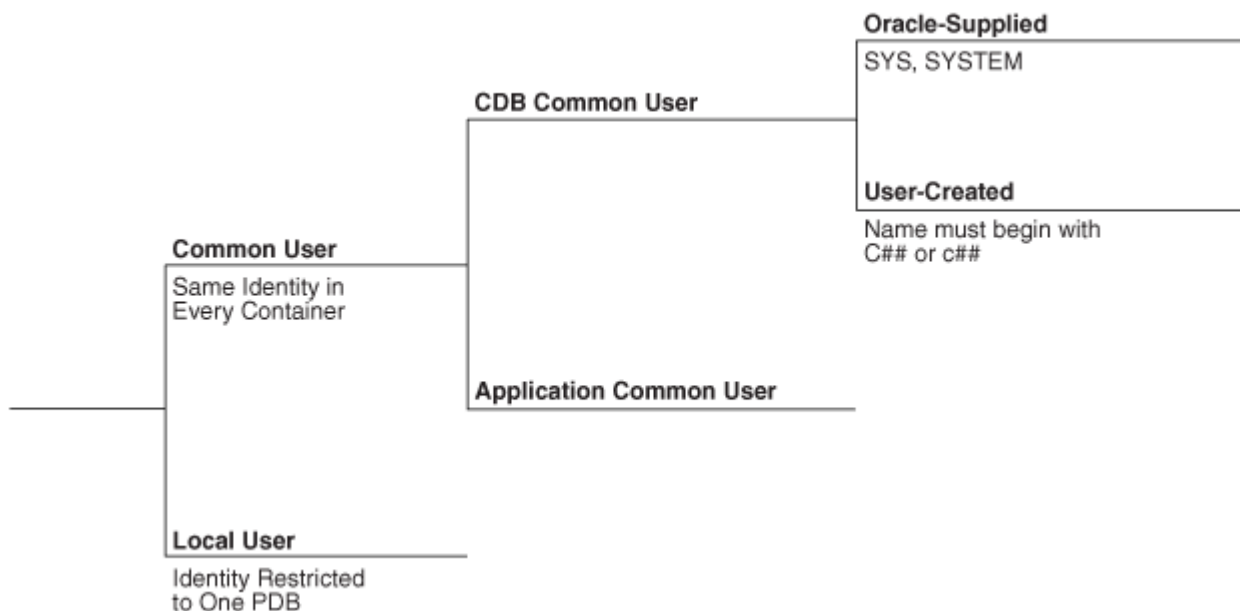
CDBの共通ユーザーおよびローカル・ユーザーの概要

ユーザー・アカウントがデータベースを定義するオブジェクトを所有する場合、このユーザー・アカウントは共通です。Oracle提供ではないユーザー・アカウントはローカルまたは共通のいずれかです。

CDB共通ユーザーは、[CDBルート](#)で作成される共通ユーザーです。[アプリケーション共通ユーザー](#)はアプリケーション・ルートで作成されるユーザーで、このアプリケーション・コンテナ内のみで共通です。

次の図は、CDBで使用可能なユーザー・アカウント・タイプを示しています。

図2-6 CDBのユーザー・アカウント



CDB共通ユーザーは、十分な権限を持つCDB内の任意のコンテナに接続できます。対照的に、アプリケーション共通ユーザーは(権限に応じて)そのユーザーが作成されたアプリケーション・ルートまたはこのアプリケーション・ルートに接続しているPDBのみに接続できます。

- [CDBの共通ユーザー](#)

システム・コンテナ(CDB)またはアプリケーション・コンテナのいずれかのコンテキスト内で、共通ユーザーは、ルートとこのコンテナ内の既存と将来のすべてのPDBにおいて同じIDを持つデータベース・ユーザーです。

- [CDBのローカル・ユーザー](#)

ローカル・ユーザーとは、共通ではなく、1つのPDBでのみ操作が可能なデータベース・ユーザーです。

関連項目:

共通ユーザーおよびローカル・ユーザーの概要は、[Oracle Databaseセキュリティ・ガイド](#)を参照してください

親トピック: [CDBの共通性の概要](#)

CDBの共通ユーザー

システム・コンテナ(CDB)またはアプリケーション・コンテナのいずれかのコンテキスト内で、共通ユーザーは、ルートとこのコンテナ内の既存と将来のすべてのPDBにおいて同じIDを持つデータベース・ユーザーです。

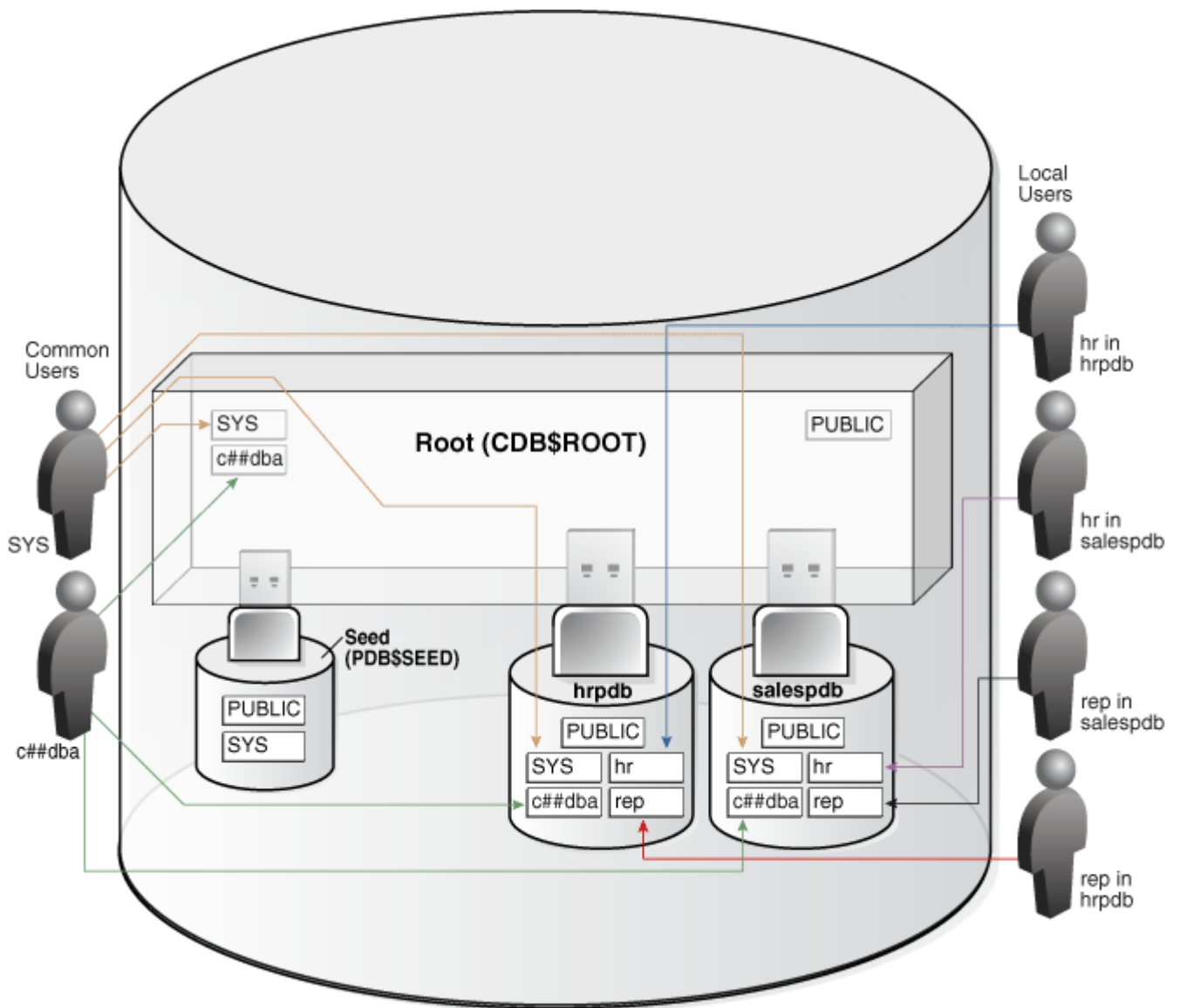
どの共通ユーザーも、そのコンテナのルートに接続し、ルート内および十分な権限を持つすべてのPDB内で操作を実行できます。一部の管理タスクは、共通ユーザーによって実行される必要があります。例として、PDBの作成やPDBの切断などがあります。

たとえば、SYSTEMはDBA権限を持つCDB共通ユーザーです。したがって、SYSTEMはCDBルートおよびデータベース内の任意のPDBに接続できます。saas_salesアプリケーション・コンテナでsaas_sales_admin共通ユーザーを作成する場合があります。この場合、saas_sales_adminユーザーが接続できるのはsaas_salesアプリケーション・ルートか、saas_salesアプリケーション・コンテナ内のアプリケーションPDBのみです。

すべての共通ユーザーは、Oracle提供またはユーザー作成のいずれかです。Oracle提供の共通ユーザーの例は、SYSやSYSTEMです。すべてのユーザー作成共通ユーザーは、CDB共通ユーザーまたはアプリケーション共通ユーザーのいずれかです。

[図2-7](#)に、2つのPDB(hrpdbとsalespdb)のユーザーとスキーマの例を示します。SYSとc##dbaは、CDB\$ROOT、hrpdbおよびsalespdbにスキーマを持つCDB共通ユーザーです。ローカル・ユーザーhrとrepは、hrpdbに存在します。ローカル・ユーザーhrとrepは、salespdbにも存在します。

図2-7 CDBのユーザーとスキーマ



共通ユーザーには、次の特徴があります。

- 共通ユーザーは、CREATE SESSION権限を持つどのコンテナ(CDB\$ROOTも含む)にもログインできます。
共通ユーザーは、すべてのコンテナで同じ権限を持つ必要はありません。たとえば、c##dbaユーザーは、hrpdbおよびルートでセッション作成の権限を持っていて、salespdbではセッション作成の権限を持っていないという場合があります。適切な権限を持つ共通ユーザーはコンテナの切替えができるため、ルートの共通ユーザーはPDBを管理できます。
- アプリケーション共通ユーザーには、その固有アプリケーション・コンテナ外のコンテナでのCREATE SESSION権限がありません。
したがって、アプリケーション共通ユーザーはその固有のアプリケーション・コンテナに制限されます。たとえば、saas_salesアプリケーションで作成されたアプリケーション共通ユーザーは、saas_salesアプリケーション・コンテナ内のアプリケーション・ルートおよびPDBにのみ接続できます。
- ユーザーが作成するCDB共通ユーザーの名前は、他のデータベース・ユーザーに対する命名規則に従う必要があります。また、COMMON_USER_PREFIX初期化パラメータによって指定された文字(デフォルトではc##またはC##)で始まる名前を付ける必要があります。Oracle提供の共通ユーザー名とユーザー作成のアプリケーション共通ユーザー名にはこの制限がありません。
ローカル・ユーザー名でc##またはC##の文字で始まる名前はありません。
- すべての共通ユーザーには、そのユーザーが作成されたコンテナ(システム・コンテナまたは特定のアプリケーション・コンテナのいずれか)内のすべてのPDBにおいて、一意の名前が付けられます。

CDB共通ユーザーはCDBルートで定義されますが、同じIDを持つどのPDBにも接続できる必要があります。アプリケーション共通ユーザーはアプリケーション・ルートに常駐しますが、同じIDを持つそのコンテナのどのアプリケーションPDBにも接続できます。

関連項目:

- 共通ユーザー・アカウントについて学習するには、[Oracle Databaseセキュリティ・ガイド](#)を参照してください
- COMMON_USER_PREFIXについて学習するには、[Oracle Databaseリファレンス](#)を参照してください

親トピック: [CDBの共通ユーザーおよびローカル・ユーザーの概要](#)

CDBのローカル・ユーザー

ローカル・ユーザーとは、共通ではなく、1つのPDBでのみ操作が可能なデータベース・ユーザーです。

ローカル・ユーザーには、次の特徴があります。

- ローカル・ユーザーは、PDBに固有であり、このPDB内でスキーマを所有できます。
[図2-7](#)では、hrpdb上のローカル・ユーザーhrがhrスキーマを所有しています。salespdbでは、ローカル・ユーザーrepがrepスキーマを、ローカル・ユーザーhrがhrスキーマを所有しています。
- ローカル・ユーザーは、オープンやクローズなどを含め、PDBを管理できます。
SYSDBA権限を持つ共通ユーザーは、ローカル・ユーザーにSYSDBA権限を付与できます。この場合、権限を付与されたユーザーはローカルのままとなります。
- あるPDBのローカル・ユーザーは、別のPDBまたはCDBルートにはログインできません。
たとえば、ローカル・ユーザーhrがhrpdbに接続する場合、hrは、データベース・リンクを使用せずに、salespdbデータベースに常駐するshスキーマのオブジェクトにアクセスすることはできません。同様に、ローカル・ユーザーshがsalespdb PDBに接続する場合、shは、データベース・リンクを使用せずに、hrpdbに存在するhrスキーマのオブジェクトにアクセスすることはできません。
- ローカル・ユーザーには、c##またはC##の文字で始まる名前を付けることはできません。
- ローカル・ユーザーの名前は、PDB内でのみ一意である必要があります。
ユーザー名とそのユーザーのスキーマが含まれているPDBにより、一意のローカル・ユーザーが決まります。[図2-7](#)では、repというローカル・ユーザーおよびスキーマがhrpdbに存在します。repという完全に独立したローカル・ユーザーとスキーマが、salespdb PDBに存在します。

次の表に、[図2-7](#)のCDBに関連したシナリオを示します。各行は、前の行のアクションの後に発生するアクションについて説明します。共通ユーザーSYSTEMが、2つのPDBでローカル・ユーザーを作成します。

表2-3 CDBのローカル・ユーザー

操作	説明
SQL> CONNECT SYSTEM@hrpdb Enter password: ***** Connected.	SYSTEM が、サービス名 hrpdb を使用して hrpdb コンテナに接続します。
SQL> CREATE USER rep IDENTIFIED BY password;	SYSTEM は、ここでローカル・ユーザー rep を作成し、このユーザー

操作	説明
User created. SQL> GRANT CREATE SESSION TO rep; Grant succeeded.	ザーにこの PDB での CREATE SESSION 権限を付与します。共通ユーザーを作成できるのはルートに接続された共通ユーザーのみであるため、このユーザーはローカルです。
SQL> CONNECT rep@salespdb Enter password: ***** ERROR: ORA-01017: invalid username/password; logon denied	rep ユーザーは、hrpdb に対するローカル・ユーザーで、salespdb への接続を試みます。rep は PDB salespdb に存在しないため、この試みは失敗します。この動作は、非 CDB の動作に似ています。ある非 CDB のユーザー・アカウントは、別の非 CDB のユーザー・アカウントとは無関係です。
SQL> CONNECT SYSTEM@salespdb Enter password: ***** Connected.	SYSTEM は、サービス名 salespdb を使用して salespdb コンテナに接続します。
SQL> CREATE USER rep IDENTIFIED BY password; User created. SQL> GRANT CREATE SESSION TO rep; Grant succeeded.	SYSTEM は、salespdb でローカル・ユーザー rep を作成し、このユーザーにこの PDB での CREATE SESSION 権限を付与します。ローカル・ユーザー名はその PDB でのみ一意である必要があるため、rep というユーザーは salespdb と hrpdb の両方に存在できます。
SQL> CONNECT rep@salespdb Enter password: ***** Connected.	rep ユーザーは、正常に salespdb にログインします。

関連項目:

ローカル・ユーザー・アカウントについて学習するには、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBの共通ユーザーおよびローカル・ユーザーの概要](#)

CDBの共通ロールおよびローカル・ロールの概要

ユーザー作成のロールは、ローカルまたは共通のいずれかです。共通ロールは、CDB自体または特定のアプリケーション・コンテナのいずれかに共通します。

Oracle提供ロールは、例として事前定義DBAロールのように、すべて共通です。Oracle提供のスクリプトでは、Oracle提供のユーザーおよびロールに付与されるすべての権限またはロールは共通に付与されますが、例外は、システム権限が共通ロール PUBLICに対してローカルに付与される場合です。

- [CDBの共通ロール](#)
共通ロールは、CDBルートまたはアプリケーション・ルートのいずれかに存在し、ルート・コンテナ(CDBまたはアプリケーション・コンテナのいずれか)内のあらゆるPDBに適用されます。
- [CDBのローカル・ロール](#)
ローカル・ロールは、非CDBのロールが非CDBにのみ存在するのと同様に、単一のPDBにのみ存在します。

関連項目:

[「CDBでのPUBLICへの付与」](#)

親トピック: [CDBの共通性の概要](#)

CDBの共通ロール

共通ロールは、CDBルートまたはアプリケーション・ルートのいずれかに存在し、ルート・コンテナ(CDBまたはアプリケーション・コンテナのいずれか)内のあらゆるPDBに適用されます。

共通ロールは、コンテナ間の操作に便利で、共通ユーザーがすべてのPDBでロールを持つことを保証します。すべての共通ユーザーは、次のいずれかのタイプになります。

- Oracle提供

DBAやPUBLICなどのOracle提供ロールはすべて、CDBに共通です。

- ユーザー作成

共通ロールを作成するには、CDBルートまたはアプリケーション・ルートのいずれか(これによって、ロールが共通となるコンテナが決まります)でCREATE ROLE ... CONTAINER=ALLを実行します。標準の命名規則が適用されます。また、CDB共通ロールには、COMMON_USER_PREFIX初期化パラメータによって指定された文字(デフォルトではc##またはC##)で始まる名前を付ける必要があります。

ロールのスコープは、そのロールが定義されているルートのスコープです。CDB\$ROOTでロールを定義すると、そのスコープはCDB全体になります。アプリケーション・ルート内でロールを定義すると、そのスコープはアプリケーション・コンテナになります。

関連項目:

- [「コンテナ間操作」](#)
- 共通ロールの管理方法を学習するには、[Oracle Databaseセキュリティ・ガイド](#)を参照してください
- CREATE ROLE文について学習するには、[Oracle Database SQL言語リファレンス](#)を参照してください

親トピック: [CDBの共通ロールおよびローカル・ロールの概要](#)

CDBのローカル・ロール

非CDBのロールが非CDBにのみ存在するのと同様に、ローカル・ロールは単一のPDBにのみ存在します。

ローカル・ロールには、そのロールが存在するコンテナで適用されるロールおよび権限のみを含めることができます。たとえば、hrpdbでローカル・ロールpdbadminを作成した場合、このロールのスコープはこのPDBに制限されます。

同じCDB、または同じアプリケーション・コンテナのPDBには、同じ名前のローカル・ロールが含まれる場合があります。たとえば、ユーザー作成のロールpdbadminは、hrpdbとsalespdbのどちらにも存在する可能性があります。しかし、これらのロールは、別々の非CDBにあるかのように、互いにまったく関係ありません。

関連項目:

ローカル・ロールの管理方法を学習するには、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBの共通ロールおよびローカル・ロールの概要](#)

CDBでの権限およびロール付与の概要

非CDBと同様に、ロールと権限をCDB内のユーザーが付与することも、CDB内のユーザーに付与することもできます。ただし、CDB内のロールおよび権限は、ローカルに付与されたものか、共通に付与されたもののいずれかです。

ローカルに付与された権限またはロールは、それが付与されたPDBでのみ実行可能です。共通に付与された権限またはロールは、付与された対象のコンテナ(CDBまたはアプリケーション・コンテナ)の既存および将来のどのPDBにおいても実行可能です。

ユーザーおよびロールは、共通の場合もローカルの場合もあります。ただし、権限は、それ自体は共通でもローカルでもありません。ユーザーがCONTAINER=CURRENT句を使用してローカルに権限を付与した場合、権限受領者は現行コンテナ内でのみ実行可能な権限を持ちます。ユーザーがCDBルートまたはアプリケーション・ルートのいずれかに接続している場合、そしてこのユーザーがCONTAINER=ALL句を使用して共通に権限を付与した場合、権限受領者は現在のコンテナ内の既存または将来のPDBでこの権限を持ちます。

- [CDBにおける権限およびロールの付与の原則](#)

CDBでは、付与の行為は、ローカルであろうと共通であろうと、どれも特定のコンテナ内で起こります。コンテナはCDBルート、アプリケーション・ルートまたはPDBになります。

- [CDBでローカルに付与される権限およびロール](#)

ロールおよび権限は、受領者、付与者または付与されるロールがローカルか共通かに関係なく、ユーザーおよびロールにローカルに付与できます。

- [CDBで共通に付与されるロールおよび権限](#)

権限および共通ロールは、共通に付与される可能性があります。

- [CDBでのPUBLICへの付与](#)

CDBでは、PUBLICは共通ロールです。PDBでは、PUBLICにローカルに付与された権限により、すべてのローカルおよび共通ユーザー・アカウントはこのPDBでのみこれらの権限を実行できるようになります。

- [権限およびロールの付与: シナリオ](#)

このシナリオでは、SYSTEMは共通ユーザーc##dbaを作成し、hrpdbのhrスキーマで表を問い合わせるためにこのユーザー権限を付与しようとします。

関連項目:

共通権限の管理方法を学習するには、[Oracle Databaseセキュリティ・ガイド](#)を参照してください

親トピック: [CDBの共通性の概要](#)

CDBにおける権限およびロールの付与の原則

CDBでは、付与の行為は、ローカルか共通かに関係なく、すべてコンテナの内部で起こります。コンテナはCDBルート、アプリケーション・ルートまたはPDBになります。

現在のコンテナがCDBルートの場合、共通に付与することはCDBのすべてのコンテナに付与することを意味します。しかし、現在のコンテナがアプリケーション・ルートの場合、共通に付与することは現在のアプリケーション・コンテナのすべてのPDBに付与することを意味します。

付与の基本原則は、次のとおりです。

- 共通およびローカルのどちらの現象も、ローカルに付与および受領できます。
- 共通に付与または受領できるのは、共通の現象のみです。

ローカルのユーザー、ロールおよび権限は、特定のPDBに制限されます。したがって、ローカル・ユーザーは、共通のロールおよび権限を付与することはできず、ローカルのロールおよび権限は共通に付与されることはありません。

次の各項では、前述の原則の意味を説明します。

親トピック: [CDBでの権限およびロール付与の概要](#)

CDBでローカルに付与される権限およびロール

ロールおよび権限は、受領者、付与者または付与されるロールがローカルか共通かに関係なく、ユーザーおよびロールにローカルに付与できます。

次の表に、ローカルに付与されたロールおよび権限の有効な可能性を示します。

表2-4 ローカルな付与

現象	ローカルな付与が可能	ローカルな受領が可能	ローカルに付与されたロールまたは権限の受領が可能
共通ユーザー	はい	該当なし	はい
ローカル・ユーザー	はい	該当なし	はい
共通ロール	該当なし	はい 脚注 1	はい
ローカル・ロール	該当なし	はい 脚注 2	はい
権限	該当なし	はい	該当なし

脚注1

このロールでの権限は、権限がロールにローカルに付与されたか共通に付与されたかに関係なく、被付与者にとってロールが付与されたコンテナでのみ使用可能です。

脚注2

このロールでの権限は、被付与者にとってロールが付与され作成されたコンテナでのみ使用可能です。

- [ロールまたは権限がローカルに付与される条件](#)
ロールまたは権限をローカルに付与するには、CONTAINER=CURRENT句(デフォルト)を含むGRANT文を使用します。
- [ローカルに付与されるロールおよび権限](#)
ユーザーまたはロールは、ローカルに付与された権限である可能性があります(CONTAINER=CURRENT)。

親トピック: [CDBでの権限およびロール付与の概要](#)

ロールまたは権限がローカルに付与される条件

ロールまたは権限をローカルに付与するには、CONTAINER=CURRENT句(デフォルト)を含むGRANT文を使用します。

具体的には、次の条件を満たした場合にのみ、ロールまたは権限がローカルに付与されます。

- 付与者に、指定したロールまたは権限を付与するために必要な権限がある。

システム・ロールまたは権限の場合、付与者には付与するロールまたは権限に対するADMIN OPTIONが必要です。オブジェクト権限の場合、付与者には付与する権限に対するGRANT OPTIONが必要です。

- 付与は1つのコンテナに対してのみ適用される。

GRANT文には、権限またはロールがローカルに付与されることを示すCONTAINER=CURRENT句がデフォルトで含まれています。

例2-4 ローカルでの権限の付与

この例で、SYSTEMおよびc##hr_adminは両方とも共通ユーザーです。この例ではSYSTEM(管理者権限を持つ)としてhrpdbに接続し、employees表での読取り権限をローカルにc##hr_adminに付与します。この付与は、他のPDB内ではなくhrpdb内のc##hr_adminのみに適用されます。

```
CONNECT SYSTEM@hrpdb
Enter password: password
Connected.

GRANT READ ON employees TO c##hr_admin CONTAINER=CURRENT;
```

関連項目:

ローカルなロールおよび権限の付与についてさらに学習するには、[Oracle Databaseセキュリティ・ガイド](#)を参照してください

親トピック: [CDBでローカルに付与される権限およびロール](#)

ローカルに付与されるロールおよび権限

ユーザーまたはロールは、ローカルに付与された権限である可能性があります(CONTAINER=CURRENT)。

たとえば、hrpdbのローカルまたは共通ユーザーに対してローカルに付与されたREAD ANY TABLE権限は、このPDB内のこのユーザーに対してのみ適用されます。同様に、非CDBのユーザーhrに付与されたREAD ANY TABLE権限は、別の非CDBに存在するhrユーザーの権限には何の関係もありません。

ユーザーまたはロールは、ローカルに付与されたロールである可能性があります(CONTAINER=CURRENT)。表2-4に示すように、共通ロールは、ローカルに付与される権限を受領する可能性があります。たとえば、共通ロールc##dbaは、hrpdbでREAD ANY TABLE権限をローカルに付与される可能性があります。c##cdb共通ロールがローカルに付与されると、そのロールの権限は、ロールの付与先のコンテナでのみ適用されます。この例では、c##cdbaロールを持つ共通ユーザーは、権限がこのロールに対してhrpdbでローカルに付与されたものであるため、この権限をhrpdb以外のどのPDBでも実行する権利はありません。

関連項目:

CDBでロールおよび権限を付与する方法を学習するには、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBでローカルに付与される権限およびロール](#)

CDBで共通に付与されるロールおよび権限

権限および共通ロールは、共通に付与される可能性があります。

ユーザー・アカウントまたはロールには、被付与者と付与者がどちらも共通である場合にのみ、ロールおよび権限を共通に付与できます。ロールを共通に付与する場合、そのロール自体が共通である必要があります。次の表に、共通の付与の可能性を示します。

表2-5 共通の付与

現象	共通の付与が可能	共通の受領が可能	共通に付与されたロールおよび権限の受領が可能
共通ユーザー・アカウント	はい	該当なし	はい
ローカル・ユーザー・アカウント	いいえ	該当なし	いいえ
共通ロール	該当なし	はい 脚注3	はい
ローカル・ロール	該当なし	いいえ	いいえ
権限	該当なし	はい	該当なし

脚注3

共通ロールに共通に付与された権限は、すべてのコンテナで被付与者が使用できます。さらに、共通ロールに対してローカルに付与された権限はどれも、その権限が共通ロールに付与されたコンテナでのみ、被付与者が使用できます。

- [付与が共通になる条件](#)
CONTAINER=ALL句は、権限またはロールが共通に付与されることを指定します。
- [共通に付与されるロールおよび権限](#)
共通ユーザー・アカウントまたはロールには、権限を共通に付与できます(CONTAINER=ALL)。

関連項目:

共通の付与についてさらに学習するには、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBでの権限およびロール付与の概要](#)

付与が共通になる条件

CONTAINER=ALL句は、権限またはロールが共通に付与されることを指定します。

ロールまたは権限は、次の条件を満たしたとき、共通に付与されます。

- 付与者は、共通ユーザーである。
付与を実行するユーザーは、CDB自体または特定のアプリケーション・コンテナのいずれかに共通します。
- 被付与者は、共通ユーザーまたは共通ロールである。
付与の受領者は、CDB自体または特定のアプリケーション・コンテナのいずれかに共通します。

- 付与者に、指定したロールまたは権限を付与するために必要な権限がある。
システム・ロールまたは権限の場合、付与者には付与するロールまたは権限に対するADMIN OPTIONが必要です。オブジェクト権限の場合、付与者には付与する権限に対するGRANT OPTIONが必要です。
- 付与は、付与が発生したコンテナ(CDBまたはアプリケーション・コンテナのいずれか)内のすべてのPDBに適用されます。
GRANT文には、権限またはロールが共通に付与されることを指定するCONTAINER=ALL句が含まれます。
- ロールが付与される場合、それは共通であり、オブジェクト権限が付与される場合は、権限が付与される対象のオブジェクトが共通であることが必要。

例2-5 共通での権限の付与

この例で、SYSTEMおよびc##hr_adminは両方とも共通ユーザーです。SYSTEMはCDBルートに接続して、CREATE ANY TABLE権限をc##hr_adminに共通に付与します。この場合、c##hr_adminはCDBのどのPDBにも表を作成できるようになります。

```
CONNECT SYSTEM@root
Enter password: password
Connected.

GRANT CREATE ANY TABLE TO c##hr_admin CONTAINER=ALL;
```

関連項目:

共通権限の付与方法を学習するには、[Oracle Databaseセキュリティ・ガイド](#)を参照

親トピック: [CDBで共通に付与されるロールおよび権限](#)

共通に付与されるロールおよび権限

共通ユーザー・アカウントまたはロールには、権限を共通に付与できます(CONTAINER=ALL)。

CDBルートまたはアプリケーション・ルートのいずれかのコンテキスト内で、現在のコンテナ内の既存および将来のすべてのPDBにおいて、権限がこの共通ユーザー・アカウントまたはロールに付与されます。たとえば、SYSTEMがCDBルートに接続して、SELECT ANY TABLE権限をCDB共通ユーザー・アカウントc##dbaに共通に付与した場合、ユーザーc##dbaはCDBのすべてのPDBでこの権限を保持します。共通に付与されたロールまたは権限をローカルに取り消すことはできません。

ユーザーまたはロールは、共通に付与された共通ロールを受領する可能性があります。[表2-5](#)の脚注で言及されているように、共通ロールは、ローカルに付与された権限を受領できます。したがって、共通ユーザーには共通ロールを付与でき、このロールにはローカルに付与された権限が含まれる可能性があります。

たとえば、共通ロールc##adminは、hrpdbに対してローカルなSELECT ANY TABLE権限を付与される可能性があります。共通ロールでローカルに付与された権限は、その権限が付与されたコンテナでのみ適用されます。したがって、c##adminロールを持つ共通ユーザーには、salespdbまたはhrpdb以外のどのPDBでも、hrpdbに含まれる権限を実行する権利はありません。

関連項目:

CDBでロールおよび権限を付与する方法を学習するには、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBで共通に付与されるロールおよび権限](#)

CDBでのPUBLICへの付与

CDBでは、PUBLICは共通ロールです。PDBでは、PUBLICにローカルに付与された権限により、すべてのローカルおよび共通ユーザー・アカウントはこのPDBでのみこれらの権限を実行できるようになります。

Oracle提供のユーザーおよびロールに付与される権限およびロールはいずれも共通に付与されますが、PUBLICに付与されるシステム権限のみはローカルに付与されます。この例外は、Oracle Databaseにデフォルトで含まれている一部の付与 (SYS. UTL_FILEに対するEXECUTEなど)を取り消すことができるように設けられています。

ローカル・ユーザー・アカウントhrがhrpdbに存在するとします。このユーザーは、hr. employeesでのSELECT権限を、PUBLICに対してローカルに付与します。hrpdbの共通およびローカル・ユーザーは、PUBLICに付与された権限を実行できます。salespdbまたはその他のPDBのユーザー・アカウントには、hrpdbでhr. employeesに問い合わせる権限はありません。

PUBLICに対して共通に付与され権限により、すべてのローカル・ユーザーは、それぞれのPDBで付与された権限を実行でき、すべての共通ユーザーは、アクセス権限を持つPDBでこの権限を実行できます。ユーザーは、PUBLICに対しては、権限およびロールを共通に付与しないことをお勧めします。

関連項目:

マルチテナント環境でPUBLICロールが機能する仕組みを学習するには、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBでの権限およびロール付与の概要](#)

権限およびロールの付与: シナリオ

このシナリオでは、SYSTEMは共通ユーザーc##dbaを作成し、hrpdbのhrスキーマで表を問い合わせるためにこのユーザー権限を付与しようとしています。

このシナリオは、CONTAINER句がロールおよび権限の付与にどのような影響を与えるかを示しています。最初の列には、CDB\$ROOTでの操作が示されています。2番目の列には、hrpdbでの操作が示されています。

表2-6 CDBのロールおよび権限の付与

t	CDB\$ROOTでの操作	hrpdbでの操作	説明
t1	SQL> CONNECT SYSTEM@root Enter password: ***** Connected.	該当なし	共通ユーザーSYSTEM は、ルート・コンテナに接続します。
t2	SQL> CREATE USER c##dba IDENTIFIED BY password CONTAINER=ALL;	該当なし	SYSTEM は、共通ユーザーc##dbaを作成します。 CONTAINER=ALL 句により、ユーザーが共通ユーザーになります。
t3	SQL> GRANT CREATE SESSION TO c##dba;	該当なし	SYSTEM は、CREATE SESSION システム権限を c##dba に付与します。 CONTAINER=ALL 句がないた

t	CDB\$ROOTでの操作	hrpdbでの操作	説明
t4	SQL> CREATE ROLE c##admin CONTAINER=ALL;	該当なし	め、この権限はローカルに付与され、したがって現行のコンテナであるルートに対してのみ適用されます。
t5	SQL> GRANT SELECT ANY TABLE TO c##admin; Grant succeeded.	該当なし	SYSTEM は、SELECT ANY TABLE 権限を c##admin ロールに付与します。CONTAINER=ALL 句がないため、権限はルートに対してローカルになります。したがって、この共通ロールには、ルートでのみ実行可能な権限が含まれます。
t6	SQL> GRANT c##admin TO c##dba; SQL> EXIT;	該当なし	SYSTEM は、c##admin ロールを c##dba に付与します。CONTAINER=ALL 句がないため、ロールは、共通ロールであっても、現行のコンテナに対してのみ適用されます。c##dba が PDB に接続すると、c##dba にこのロールはありません。
t7	該当なし	SQL> CONNECT c##dba@hrpdb Enter password: ***** ERROR: ORA-01045: user c##dba lacks CREATE SESSION privilege; logon denied	c##dba は、t3 での付与がルートに対してローカルであったため、hrpdb への接続に失敗します。
t8	該当なし	SQL> CONNECT SYSTEM@hrpdb Enter password: ***** Connected.	SYSTEM は、hrpdb に接続します。
t9	該当なし	SQL> GRANT CONNECT, RESOURCE TO c##dba;	SYSTEM は、CONNECT ロールおよび RESOURCE ロールを共通

t	CDB\$ROOTでの操作	hrpdbでの操作	説明
		Grant succeeded. SQL> EXIT	ユーザーc##dba に付与します。CONTAINER=ALL 句がないため、付与は hrpdb に対してローカルです。
t10	該当なし	SQL> CONNECT c##dba@hrpdb Enter password: ***** Connected.	共通ユーザーc##dba は、hrpdb に接続します。
t11	該当なし	SQL> SELECT COUNT(*) FROM hr.employees; select * from hr.employees * ERROR at line 1: ORA-00942: table or view does not exist	c##dba は hrpdb 内の表に対して SELECT 権限がないため、hr.employees の問合せからは、まだエラーが戻されません。t5 でローカルに付与された SELECT ANY TABLE 権限は、ルートに限定されるため、hrpdb には適用されません。
t12	SQL> CONNECT SYSTEM@root Enter password: ***** Connected.	該当なし	共通ユーザーSYSTEM は、ルート・コンテナに接続します。
t13	SQL> GRANT SELECT ANY TABLE TO c##admin CONTAINER=ALL; Grant succeeded.	該当なし	SYSTEM は、SELECT ANY TABLE 権限を c##admin ロールに付与します。CONTAINER=ALL が存在するため、この権限は共通に付与されます。
t14	該当なし	SQL> SELECT COUNT(*) FROM hr.employees; select * from hr.employees * ERROR at line 1: ORA-00942: table or view does not exist	hr.employees の問合せでは、まだエラーが戻されます。これは、t6 で c##admin 共通ロールが、ルートでのみ c##dba に付与されたためです。
t15	SQL> GRANT c##admin TO c##dba CONTAINER=ALL; Grant succeeded.	該当なし	SYSTEM は、CONTAINER=ALL を指定して、c##admin という共通ロールを c##dba に付与します。これでユーザーc##dba は、ルートのみでなく、すべてのコンテナでこのロールを持つこと

t	CDB\$ROOTでの操作	hrpdbでの操作	説明
			になります。
t17	該当なし	SQL> SELECT COUNT (*) FROM hr. employees; COUNT (*) ----- 107	問合せに成功します。

関連項目:

共通ロールおよびローカル・ロールの管理方法を学習するには、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBでの権限およびロール付与の概要](#)

CDBの共通オブジェクトおよびローカル・オブジェクトの概要

共通オブジェクトは、CDBルートまたはアプリケーション・ルートのいずれかで定義されており、メタデータ・リンクまたはオブジェクト・リンクを使用して参照できます。ローカル・オブジェクトは、共通オブジェクトではないすべてのオブジェクトです。

データベースによって提供される共通オブジェクトは、CDB\$ROOTで定義されており、変更できません。Oracle Databaseでは、CDB\$ROOTでの共通オブジェクトの作成はサポートされていません。

アプリケーション・ルートでは、表、ビュー、PL/SQLおよびJavaプログラム・ユニット、順序などのほとんどのスキーマ・オブジェクトを共通オブジェクトとして作成できます。オブジェクトがアプリケーション・ルートに存在する場合、それは[アプリケーション共通オブジェクト](#)と呼ばれます。

ローカル・ユーザーは共通オブジェクトを所有できます。また、共通ユーザーはローカル・オブジェクトを所有できますが、これはオブジェクトがデータリンクまたはメタデータリンクされてなく、なおかつメタデータ・リンクでもデータ・リンクでもない場合のみです。

関連項目:

- [「アプリケーション共通オブジェクト」](#)
- [「アプリケーション共通オブジェクトについて」](#)
- 共通オブジェクトの権限管理についてさらに学習するには、[Oracle Databaseセキュリティ・ガイド](#)を参照してください

親トピック: [CDBの共通性の概要](#)

共通監査構成の概要

混合モードと統合監査のどちらの場合も、共通監査構成が表示され、すべてのPDBにわたって実行されます。

監査構成は、ローカルまたは共通のいずれかです。ユーザーやロールなど、その他のローカルまたは共通の現象に適用されるスコーピング・ルールは、すべて監査構成に適用されます。



ノート:

監査初期化パラメータは、CDB レベルに存在し、各 PDB には存在しません。

PDBでは、次の監査オプションがサポートされます。

- オブジェクト監査

オブジェクト監査とは、特定のオブジェクトに対する監査構成のことを指します。共通監査構成に含めることができるのは、共通オブジェクトのみです。ローカル監査構成には、共通オブジェクトを含めることはできません。

- 監査ポリシー

監査ポリシーは、ローカルまたは共通のいずれかです。

- ローカル監査ポリシー

ローカル監査ポリシーは、1つのPDBに適用されます。ローカル監査ポリシーは、このPDBのみのローカルおよび共通ユーザーに対して実行できます。ローカル監査ポリシーをすべてのコンテナにわたって実行しようとすると、エラーが発生します。

いかなる場合でも、ローカル監査ポリシーの実行は、ローカル監査フレームワークの一部です。

- 共通監査ポリシー

共通監査ポリシーは、すべてのコンテナに対して適用されます。このポリシーには、アクション、システム権限、共通ロールおよび共通オブジェクトのみを含めることができます。共通監査ポリシーは、共通ユーザーに対してのみ適用できます。ローカル・ユーザーに対する共通監査ポリシーをすべてのコンテナにわたって実行しようとすると、エラーが発生します。

共通監査構成は、ルートSYSスキーマに格納されています。ローカル監査構成は、適用対象のPDBのSYSスキーマに格納されています。

監査証跡は、関連PDBのSYSまたはAUDSYSスキーマに格納されています。オペレーティング・システムおよびPDBのXML監査証跡は、AUDIT_FILE_DEST初期化パラメータで指定されたディレクトリのサブディレクトリに格納されています。

関連項目:

- データベース監査の詳細は、[『Oracle Database概要』](#)を参照してください。
- 共通監査構成について学習するには、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBの共通性の概要](#)

PDBロックダウン・プロファイルの概要

PDBロックダウン・プロファイルは、PDBに接続されたユーザーが使用できる操作を制限する一連の機能に名前を付けたものです。たとえば、PDBロックダウン・プロファイルによって、ALTER SYSTEM文に伴う権限を無効化できます。

PDBがIDを共有する際、権限が上昇する可能性があります。たとえば、ネットワーク・レベルや、PDBが共通オブジェクトにアクセスする、またはデータベース・リンク経由で接続する際にIDを共有できます。セキュリティを向上するため、CDB管理者はアクセスを区画化することで、ユーザーがPDBで実行可能な操作を制限する場合があります。

ユースケースとしては、高、中および低レベルのロックダウン・プロファイルの作成があります。高いレベルではアクセスが大幅に制限されますが、低いレベルではアクセスが有効になります。

次のタイプのアクセスを制限できます。

- ネットワーク・アクセス

たとえば、UTL_HTTPやUTL_MAILへのアクセスを制限します。

- 共通ユーザーおよび共通オブジェクトのアクセス

たとえば、PDBのローカル・ユーザーが共通ユーザーを介してプロキシを使用する操作や、共通スキーマのオブジェクトにアクセスする操作を制限します。

- オペレーティング・システム・アクセス

たとえば、PL/SQLパッケージUTL_FILEやDBMS_FILE_TRANSFERへのアクセスを制限します。

- 接続

たとえば、共通ユーザーによるPDBへの接続を制限したり、SYSOPER管理権限を持つローカル・ユーザーによる制限モードでオープンしたPDBへの接続を制限できます。

- 管理機能

たとえば、ALTER SYSTEM、ALTER SESSIONおよびALTER DATABASEの使用を制限できます。

- データベース・オプション

たとえば、ロックダウン・プロファイルを使用して、Oracleのパーティション化やOracle Databaseアドバンスト・キューイングなどのデータベース・オプションへのアクセスを無効にできます。

CDBルートまたはアプリケーション・ルートにログインするときに、次のオプション句をサポートしているCREATE LOCKDOWN PROFILE文を発行して、ロックダウン・プロファイルを作成します。

- FROM static_base_profileは、既存のプロファイルの値を使用して、新しいロックダウン・プロファイルを作成します。既存のプロファイルへの以降の変更は、新しいプロファイルには影響しません。
- INCLUDING dynamic_base_profileは、既存のプロファイルの値を使用して新しいロックダウン・プロファイルを作成しますが、この新しいロックダウン・プロファイルは、基本プロファイルを構成するDISABLE STATEMENTルール、および基本プロファイルへの以降の変更を継承するという点が異なります。

文を発行するユーザーは、現在のコンテナでCREATE LOCKDOWN PROFILEシステム権限を持っている必要があります。制約を追加および削除するには、ALTER LOCKDOWN PROFILE文を使用します。ユーザーはCDBルートまたはアプリケーション・ルートでALTER文を発行する必要があり、現在のコンテナでALTER LOCKDOWN PROFILEシステム権限を持っている必要があります。

ロックダウン・プロファイルは、PDB_LOCKDOWN初期化パラメータを使用して指定します。このパラメータによって、PDBロックダウン・プロファイルが指定のPDBに適用されるかどうかが決まります。このパラメータは、次のレベルで設定できます。

- PDB

プロファイルは、設定されるPDBにのみ適用されます。

- アプリケーション・コンテナ

プロファイルは、アプリケーション・コンテナ内のすべてのアプリケーションPDBに適用されます。値を変更できるのは、アプリケーション共通のSYSDBA権限または共通のALTER SYSTEM権限を持つアプリケーション共通ユーザー、あるいは共通のSYSDBA権限または共通のALTER SYSTEM権限を持つCDB共通ユーザーのみです。

- CDB

プロファイルはすべてのPDBに適用されます。共通のSYSDBA権限または共通のALTER SYSTEM権限を持つ共通ユーザーは、特定のPDBに対するCDB全体の設定をオーバーライドできます。

PDBのPDB_LOCKDOWNパラメータが、このPDBのコンテナ(CDBまたはアプリケーション・コンテナ)とは異なるロックダウン・プロファイルの名前に設定されている場合は、一連のルールによって制限の間の相互作用が制御されます。

例2-6 PDBロックダウン・プロファイルの作成

この例では、CREATE LOCKDOWN PROFILE権限を持つ共通ユーザーとしてCDBルートに接続します。ALTER SYSTEM FLUSH SHARED POOL以外のすべてのALTER SYSTEM文を無効にする、mediumというプロファイルを作成します。

```
CREATE LOCKDOWN PROFILE medium;  
ALTER LOCKDOWN PROFILE medium DISABLE STATEMENT=(' ALTER SYSTEM');  
ALTER LOCKDOWN PROFILE medium ENABLE STATEMENT=(' ALTER SYSTEM') CLAUSE=(' FLUSH SHARED POOL');
```

同じ共通ユーザーとして、このプロファイルを必要とする各PDBに接続し、ALTER SYSTEMを使用してPDB_LOCKDOWN初期化パラメータをmediumに設定します。たとえば、hrpdbについてPDB_LOCKDOWNをmediumに設定できますが、salespdbには設定できません。

次の例では、mediumからmedium2プロファイルを作成します。

```
CREATE LOCKDOWN PROFILE medium2 FROM medium;
```

ノート:



- PDB ロックダウン・プロファイルについてさらに学習するには、[「セキュリティを強化するための PDB ユーザーの制限について」](#)を参照してください
- PDB ロックダウン・プロファイルの作成、有効化および削除方法を学習するには、[Oracle Database セキュリティ・ガイド](#)を参照してください

親トピック: [CDBの共通性の概要](#)

アプリケーション・コンテナ内のアプリケーションの概要

アプリケーション・コンテナ内で、**アプリケーション**はアプリケーション・ルートに格納される共通データおよびメタデータの名前付けおよびバージョンングされたセットです。

このアプリケーション・コンテナのコンテキストでは、「アプリケーション」という用語は「マスター・アプリケーション定義」を意味します。たとえば、表、ビュー、パッケージなどの定義をアプリケーションに含めることができます。

- [アプリケーション・コンテナについて](#)
アプリケーション・コンテナはオプションのユーザー作成CDBコンポーネントで、1つ以上のアプリケーション・バックエンドのデータおよびメタデータが格納されます。CDBにはゼロ以上のアプリケーション・コンテナが含まれます。
- [アプリケーション共通オブジェクト](#)
アプリケーション共通オブジェクトは、アプリケーション・ルートのアプリケーション内で作成される共通オブジェクトです。共通オブジェクトはデータリンクされているか、メタデータリンクされているかのいずれかです。
- [アプリケーション・メンテナンス](#)
このコンテキストにおける**アプリケーション・メンテナンス**は、アプリケーションのインストール、アンインストール、アップグレードまたはパッチ適用を指します。
- [既存のアプリケーションの移行](#)
PDBにインストールされているアプリケーションを、アプリケーション・ルートまたはアプリケーションPDBのいずれかに移行できます。

- [暗黙的に作成されるアプリケーション](#)
ユーザー作成アプリケーション以外に、アプリケーション・コンテナには暗黙的に作成されるアプリケーションも含まれます。
- [アプリケーションの同期化](#)
アプリケーションPDB内で、同期化は、アプリケーション・ルートでの最新のバージョンおよびパッチへのユーザー始動のアプリケーションの更新です。
- [コンテナ・マップ](#)
コンテナ・マップを使用することにより、アプリケーション・ルートに接続したセッションで発行したSQL文が、SQL文に使用された条件の値に応じて適切なPDBにルーティングされるようになります。

関連項目:

- アプリケーション共通オブジェクトについて学習するには、「[CDBの共通オブジェクトおよびローカル・オブジェクトの概要](#)」を参照してください
- [「アプリケーション・コンテナおよびシードの作成と削除」](#)
- [「アプリケーション・コンテナの管理」](#)

親トピック: [マルチテナント・アーキテクチャの概要](#)

アプリケーション・コンテナについて

アプリケーション・コンテナはオプションのユーザー作成CDBコンポーネントで、1つ以上のアプリケーション・バックエンドのデータおよびメタデータが格納されます。CDBにはゼロ以上のアプリケーション・コンテナが含まれます。

たとえば、1つのアプリケーション・コンテナ内で複数の販売関連PDBを作成し、これらのPDBが一連の共通表および表定義で構成されるアプリケーション・バックエンドを共有することができます。複数のHR関連PDBを、その独自の共通表および表定義とともに別のアプリケーション・コンテナ内で格納できます。

AS APPLICATION CONTAINER句を使用したCREATE PLUGGABLE DATABASE文によってアプリケーション・コンテナのアプリケーション・ルートが作成され、これによって暗黙的にアプリケーション・コンテナ自体が作成されます。アプリケーション・コンテナを最初に作成した時点では、PDBは含まれていません。アプリケーションPDBを作成するには、アプリケーション・ルートに接続してCREATE PLUGGABLE DATABASE文を実行する必要があります。

CREATE PLUGGABLE DATABASE文で、例としてsaas_sales_acなどのコンテナ名(アプリケーション・ルート名と同じ)を指定する必要があります。アプリケーション・コンテナ名は、CDB内と、そのインスタンスが特定のリスナーを介してアクセスされるすべてのCDBの範囲内で一意にする必要があります。すべてのアプリケーション・コンテナに、アプリケーション・コンテナと同じ名前のデフォルト・サービスがあります。

- [アプリケーション・コンテナの目的](#)
アプリケーション・コンテナはある意味で、CDB内のアプリケーション固有CDBとして機能します。アプリケーション・コンテナにはCDB自身のように複数のPDBが含まれ、これらのPDBはメタデータおよびデータを共有できます。
- [アプリケーション・ルート](#)
アプリケーション・コンテナには、コンテナ内のアプリケーションPDBの親であるアプリケーション・ルートが1つあります。
- [アプリケーションPDB](#)
アプリケーションPDBはアプリケーション・コンテナ内に存在するPDBです。CDB内のPDBはすべて、ゼロまたは1つのアプリケーション・コンテナに存在します。
- [アプリケーション・シード](#)
アプリケーション・シードはアプリケーション・コンテナ内のオプションのユーザー作成PDBです。アプリケーション・コンテナ

にはゼロまたは1つのアプリケーション・シードが含まれます。

関連項目:

- [「アプリケーション・コンテナ内のアプリケーションの概要」](#)
- [「アプリケーション・コンテナおよびシードの作成と削除」](#)
- [「アプリケーション・コンテナの管理」](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションの概要](#)

アプリケーション・コンテナの目的

アプリケーション・コンテナはある意味で、CDB内のアプリケーション固有CDBとして機能します。アプリケーション・コンテナにはCDB自身のように複数のPDBが含まれ、これらのPDBはメタデータおよびデータを共有できます。

アプリケーション・ルートによってアプリケーションPDBが、[アプリケーション](#)(このコンテキストでは共通メタデータおよびデータの名前付けおよびバージョンングされたセット)を共有できます。一般的なアプリケーションではアプリケーション共通ユーザー、メタデータリンク共通オブジェクト、およびデータリンク共通オブジェクトがインストールされます。

- [アプリケーション・コンテナの主な利点](#)
アプリケーション・コンテナにより、個々のアプリケーションを別々のPDBに格納することの利点が提供されます。
- [アプリケーション・コンテナのユースケース: SaaS](#)
SaaSデプロイメントでは、個々が別の顧客用で、メタデータおよびデータを共有する複数のアプリケーションPDBを使用できます。
- [アプリケーション・コンテナのユースケース: 論理データ・ウェアハウス](#)
顧客は複数のアプリケーションPDBを使用してデータの主権の問題に対応できます。

親トピック: [アプリケーション・コンテナについて](#)

アプリケーション・コンテナの主な利点

アプリケーション・コンテナにより、個々のアプリケーションを別々のPDBに格納することの利点が提供されます。

- アプリケーション・ルートは、すべてのアプリケーションPDBが共有できるメタデータおよびデータを格納します。
たとえば、すべてのアプリケーションPDBは中央にある1つの表(デフォルトのアプリケーション・ロールをリストする表など)のデータを共有できます。また、すべてのPDBはPDB固有行の追加先である表定義を共有できます。
- マスター・アプリケーション定義は、各PDBで個別のコピーを保持するのではなく、アプリケーション・ルートで保持されます。
アプリケーション・ルートでアプリケーションをアップグレードすると、その変更はすべてのアプリケーションPDBに自動的に伝播されます。アプリケーション・バックエンドには、[データリンク共通オブジェクト](#) app_rolesが含まれている可能性があります。これは、admin、manager、sales_repなどのデフォルト・ロールをリストする表です。アプリケーションPDBに接続したユーザーは、この表を問い合わせることができます。
- アプリケーション・コンテナには、アプリケーション・シード、アプリケーションPDBおよびプロキシPDB (他のCDB内のPDBを参照する)を含めることができます。
- 新規のアプリケーションPDBを[アプリケーション・シード](#)から迅速に作成できます。
- アプリケーション・コンテナですべてのPDBについてレポートするビューを問合せできます。
- アプリケーション・ルートへの接続中に、CONTAINERS関数を使用して、複数のPDB内のオブジェクトに対してDMLを実

行できます。

たとえば、あらゆるアプリケーションPDBにproducts表が存在する場合、アプリケーション・ルートに接続して、単一のSELECT文を使用してすべてのアプリケーションPDB内の製品を問合せできます。

- PDBをアプリケーション・ルートから切断してから、新しいOracle Databaseリリースのアプリケーション・ルートにそのPDBを接続できます。したがって、PDBはOracle Databaseのアップグレードに役立ちます。

関連項目:

[「アプリケーション・コンテナ内のアプリケーションの概要」](#)

親トピック: [アプリケーション・コンテナの目的](#)

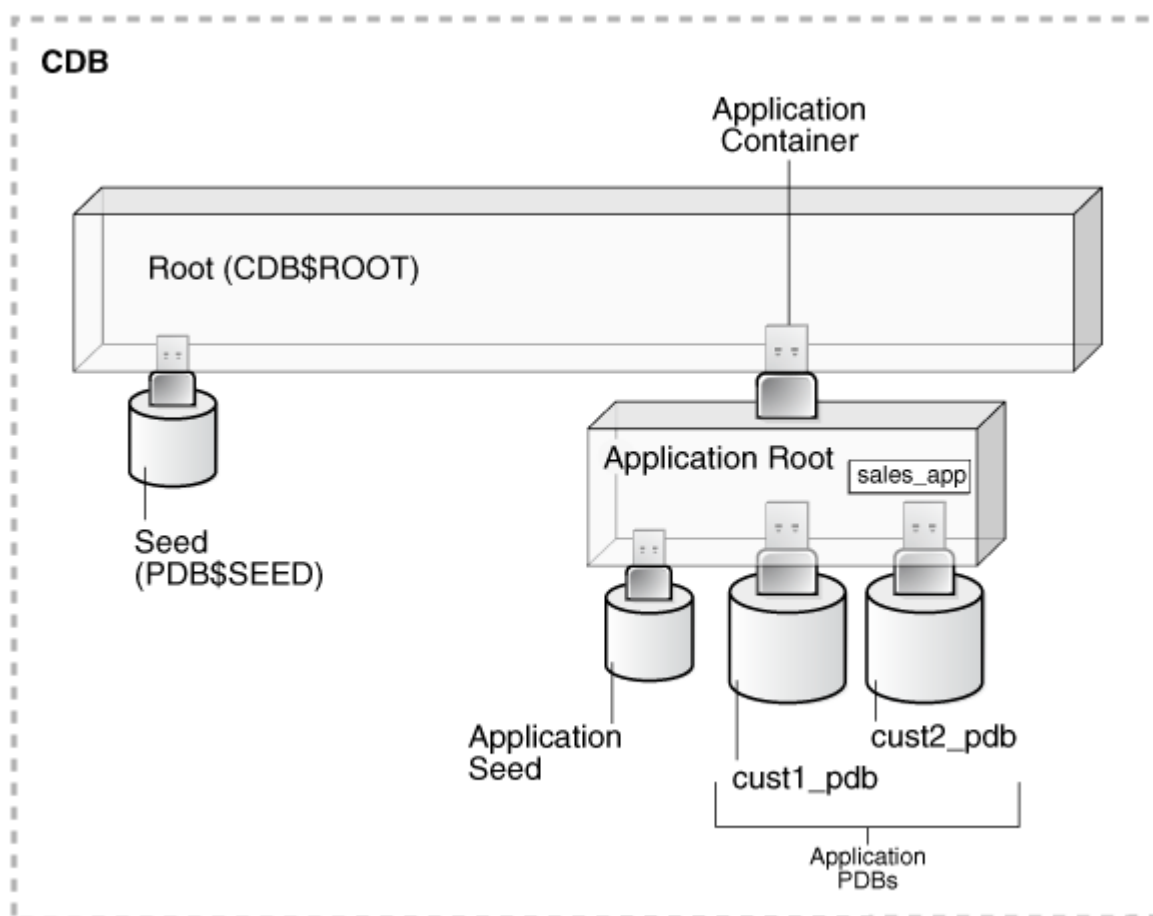
アプリケーション・コンテナのユースケース: SaaS

SaaSデプロイメントでは、個々が別の顧客用で、メタデータおよびデータを共有する複数のアプリケーションPDBを使用できます。

純粋なSaaS環境では、マスター・アプリケーション定義はアプリケーション・ルート内に存在しますが、顧客固有のデータはその固有のアプリケーションPDB内に存在します。たとえば、sales_appはアプリケーション・ルート内のアプリケーション・モデルです。

cust1_pdbというアプリケーションPDBには顧客1用の販売データのみが含まれる一方、cust2_pdbというアプリケーションPDBには顧客2用の販売データのみが含まれます。接続、切断、クローニングおよびその他のPDBレベルの操作を、個々の顧客PDBで使用できます。

図2-8 SaaSのユースケース



純粋なSaaS構成には次のメリットがあります。

- パフォーマンス

- セキュリティ
- 複数の顧客のサポート

各顧客のデータは、その固有のコンテナ内に存在しますが、多くの顧客をまとめて管理できるように統合されます。このモデルにより、多くの要素を1つにまとめて管理する規模の経済性が、DBAのみでなくアプリケーション管理者にも適用されます。

親トピック: [アプリケーション・コンテナの目的](#)

アプリケーション・コンテナのユースケース: 論理データ・ウェアハウス

顧客は複数のアプリケーションPDBを使用してデータの主権の問題に対応できます。

サンプル・ユースケースでは、ある会社が各会計四半期に固有のデータを別個のPDBに配置しています。たとえば、sales_acというアプリケーション・コンテナにq1_2016_pdb、q2_2016_pdb、q3_2016_pdbおよびq4_2016_pdbが含まれています。個々のトランザクションは、関連する四半期に対応するPDB内で定義します。1年間の業績を集計するレポートを生成するには、CONTAINERS () 句を使用して4つのPDB間で集計します。

この論理ウェアハウス設計の利点は次のとおりです。

- 1つのPDBに固有のデータに対するETLが他のPDBに影響しません。
- 実行計画が、実際のデータ分布に基づいているため、より効率的です。

親トピック: [アプリケーション・コンテナの目的](#)

アプリケーション・ルート

アプリケーション・コンテナには、コンテナ内のアプリケーションPDBの親であるアプリケーション・ルートが1つだけあります。

アプリケーション・ルートになるためのプロパティは作成時に確立され、変更することはできません。アプリケーション・ルートが属すコンテナはCDBルートのみです。アプリケーション・ルートはある面ではCDBルートに類似し、別の面ではPDBに類似しています。

- CDBルートと同様に、アプリケーション・ルートは接続されているPDBの親コンテナとして機能します。アプリケーション・ルートへの接続中に、共通ユーザーおよび権限の管理、アプリケーションPDBの作成、コンテナの切替え、およびアプリケーション・コンテナのすべてのPDBに適用されるDDLの発行などができます。
- PDBと同様に、CREATE PLUGGABLE DATABASE文でアプリケーション・ルートを作成し、それをALTER PLUGGABLE DATABASEで変更し、その可用性をSTARTUPおよびSHUTDOWNで変更することができます。アプリケーション・ルートの接続、切断および削除にはDDLを使用できます。アプリケーション・ルートには固有のサービス名があり、ユーザーはPDBに接続するのと同じ方法でアプリケーション・ルートに接続できます。

アプリケーション・ルートは、アプリケーション共通オブジェクトと呼ばれるユーザー作成共通オブジェクトを格納できるため、CDBルートおよび標準PDBの両方と異なります。アプリケーション共通オブジェクトはアプリケーション・ルートに接続されたアプリケーションPDBにとってアクセス可能です。アプリケーション共通オブジェクトはCDBルート、他のアプリケーション・ルート、またはアプリケーション・ルートに属さないPDBには表示されません。

関連項目:

- [「アプリケーション・コンテナおよびシードの作成と削除」](#)
- [「アプリケーション・コンテナの管理」](#)

例2-7 アプリケーション・ルートの作成

この例では、管理共通ユーザーc##systemとしてCDBルートにログインします。saas_sales_acというアプリケーション・コンテナを作成し、コンテナと同じ名前を持つアプリケーション・ルートをオープンします。

```
-- Create the application container called saas_sales_ac
CREATE PLUGGABLE DATABASE saas_sales_ac AS APPLICATION CONTAINER
  ADMIN USER saas_sales_ac_adm IDENTIFIED BY manager;

-- Open the application root
ALTER PLUGGABLE DATABASE saas_sales_ac OPEN;
```

現在のコンテナをsaas_sales_acに設定し、このコンテナがアプリケーション・ルートであることを確認します。

```
-- Set the current container to saas_sales_ac
ALTER SESSION SET CONTAINER = saas_sales_ac;
```

```
COL NAME FORMAT a15
COL ROOT FORMAT a4
SELECT CON_ID, NAME, APPLICATION_ROOT AS ROOT,
       APPLICATION_PDB AS PDB,
FROM   V$CONTAINERS;
```

CON_ID	NAME	ROOT	PDB
3	SAAS_SALES_AC	YES	NO

親トピック: [アプリケーション・コンテナについて](#)

アプリケーションPDB

アプリケーションPDBはアプリケーション・コンテナ内に存在するPDBです。CDB内のPDBはすべて、ゼロまたは1つのアプリケーション・コンテナに存在します。

たとえば、saas_sales_acアプリケーション・コンテナで複数の顧客がサポートされ、個々の顧客アプリケーションがそのデータを別々のPDBに格納する場合があります。cust1_sales_pdbおよびcust2_sales_pdbのアプリケーションPDBがsaas_sales_acに存在する場合、これらは他のアプリケーション・コンテナには属していません(ただし、PDBは必然的にCDBルートにも属します)。

アプリケーション・ルートへの接続中にCREATE PLUGGABLE DATABASEを実行して、アプリケーションPDBを作成します。シードからアプリケーションPDBを作成するか、PDBをクローニングするか、または切断されたPDBを接続することができます。CDBルートに接続しているPDBと同様に、アプリケーションPDBをクローニング、切断または削除できます。ただし、アプリケーションPDBは常にアプリケーション・ルートに属する必要があります。

関連項目:

[「アプリケーション・コンテナおよびシードの作成と削除」](#)

親トピック: [アプリケーション・コンテナについて](#)

アプリケーション・シード

アプリケーション・シードはアプリケーション・コンテナ内のオプションのユーザー作成PDBです。アプリケーション・コンテナにはゼロまたは1つのアプリケーション・シードが含まれます。

アプリケーション・シードを使用すると、アプリケーションPDBを簡単に作成できます。これはアプリケーション・コンテナ内で、

PDB\$SEEDがCDB内で果たす役割と同じ役割を果たします。

アプリケーション・シード名は常にapplication_container_name\$SEEDとなり、application_container_nameはアプリケーション・コンテナの名前を表します。たとえば、CREATE PDB ... AS SEED文を使用して、saas_sales_acアプリケーション・コンテナにsaas_sales_ac\$SEEDを作成します。

関連項目:

[「アプリケーション・シードの作成および削除」](#)

親トピック: [アプリケーション・コンテナについて](#)

アプリケーション共通オブジェクト

アプリケーション共通オブジェクトは、アプリケーション・ルートのアプリケーション内で作成される共通オブジェクトです。共通オブジェクトはデータリンクされているか、メタデータリンクされているかのいずれかです。

[データリンク共通オブジェクト](#)の場合、アプリケーションPDBは単一のデータのセットを共有します。たとえば、saas_sales_acアプリケーション・コンテナのアプリケーションがsaas_sales_appという名前で、バージョンが1.0であり、データリンクされたusa_zipcodes表を含むとします。この場合、行はアプリケーション・ルート内の表に1回のみ格納されますが、すべてのアプリケーションPDBで表示されます。

[メタデータリンク共通オブジェクト](#)の場合、アプリケーションPDBはメタデータのみを共有しますが、異なるデータのセットが含まれます。たとえば、メタデータリンクされたproducts表はあらゆるアプリケーションPDBで同じ定義を持ちますが、行自体はPDBに固有です。cust1pdbというアプリケーションPDBには書籍を格納するproducts表があり、cust2pdbというアプリケーションPDBには自動車部品を格納するproducts表がある場合があります。

- [アプリケーション共通オブジェクトの作成](#)
共通オブジェクトを作成するには、アプリケーション・ルートに接続してから、共有属性を指定するCREATE文を実行します。
- [メタデータリンク・アプリケーション共通オブジェクト](#)
メタデータ・リンクは、アプリケーション・コンテナ内のすべてのPDBによって共有される共通メタデータの参照および権限付与をサポートするディクショナリ・オブジェクトです。
- [データリンク・アプリケーション共通オブジェクト](#)
データリンク・オブジェクトは、そのメタデータおよびデータがアプリケーション・ルートに存在し、このアプリケーション・コンテナのすべてのアプリケーションPDBからアクセスできるオブジェクトです。
- [拡張データリンク・アプリケーション・オブジェクト](#)
拡張データリンク・アプリケーション・オブジェクトは、データリンク・オブジェクトとメタデータリンク・オブジェクトを合成したものです。

関連項目:

- 共通オブジェクトについて学習するには、[「CDBの共通オブジェクトおよびローカル・オブジェクトの概要」](#)を参照
- [「アプリケーション共通オブジェクトについて」](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションの概要](#)

アプリケーション共通オブジェクトの作成

共通オブジェクトを作成するには、アプリケーション・ルートに接続してから、共有属性を指定するCREATE文を実行します。

アプリケーション共通オブジェクトを作成または変更できるのは、アプリケーションのインストール、アップグレードまたはパッチ適用の一環である場合のみです。共有は次の方法で指定できます。

- **DEFAULT_SHARING**初期化パラメータ

設定は、ルートで作成されたサポート対象タイプのすべてのデータベース・オブジェクトに対するデフォルト共有属性です。

- **SHARING**句

この句は、CREATE文自体で指定します。SHARING句がSQL文に含まれる場合、これはDEFAULT_SHARING初期化パラメータで指定された値よりも優先されます。指定できる値は、METADATA、DATA、EXTENDED DATAおよびNONEです。

次の表は、アプリケーション共通オブジェクトのタイプと、データおよびメタデータの格納場所を示しています。

表2-7 アプリケーション共通オブジェクト

オブジェクト・タイプ	SHARING値	メタデータ記憶域	データ記憶域
データリンク	DATA	アプリケーション・ルート	アプリケーション・ルート
拡張データリンク	EXTENDED DATA	アプリケーション・ルート	アプリケーション・ルートおよびアプリケーション PDB
メタデータリンク	METADATA	アプリケーション・ルート	アプリケーション PDB

関連項目:

- [「アプリケーション共通オブジェクトの作成」](#)
- 共通オブジェクトの権限を管理する方法を学習するには、[Oracle Databaseセキュリティ・ガイド](#)を参照してください

親トピック: [アプリケーション共通オブジェクト](#)

メタデータリンク・アプリケーション共通オブジェクト

メタデータ・リンクは、アプリケーション・コンテナ内のすべてのPDBによって共有される共通メタデータの参照および権限付与をサポートするディクショナリ・オブジェクトです。

METADATA値をSHARING句またはDEFAULT_SHARING初期化パラメータのいずれかで指定すると、[メタデータリンク共通オブジェクト](#)と呼ばれるオブジェクトのメタデータへのリンクが指定されます。オブジェクトのメタデータは、アプリケーション・ルートに1回のみ格納されます。

表、ビュー、およびコード・オブジェクト(PL/SQLプロシージャなど)はメタデータを共有できます。このコンテキストでの「メタデータ」には、列定義、制約、トリガーおよびコードが含まれています。たとえば、sales_mltがメタデータリンクされた共通表の場合、すべてのアプリケーションPDBがメタデータ・リンクによってこの表の同じ定義(これはアプリケーション・ルートに格納されています)にアクセスします。sales_mltの行はすべてのアプリケーションPDBで異なりますが、列定義は同じです。

一般に、アプリケーションのほとんどのオブジェクトはメタデータリンクされます。したがって、1つのマスター・アプリケーション定義を保持するのみで済みます。この方法により、複数のアプリケーションPDB内のアプリケーションの管理が集中化されます。

例2-8 メタデータリンク共通オブジェクトの作成

この例では、SYSTEMユーザーがsaas_sales_acアプリケーション・コンテナにログインします。SYSTEMは、saas_sales_appという名前のアプリケーションのバージョン1.0をインストールします(「[アプリケーション・メンテナンス](#)」を参照)。このアプリケーションは、saas_sales_admという共通ユーザー・アカウントを作成します。スキーマには、sales_mltというメタデータリンクされた共通表が含まれています。

```
-- Begin the install of saas_sales_app
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app BEGIN INSTALL '1.0';

-- Create the tablespace for the app
CREATE TABLESPACE saas_sales_tbs DATAFILE SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE 200M;

-- Create the user account saas_sales_adm, which will own the app
CREATE USER saas_sales_adm IDENTIFIED BY ***** CONTAINER=ALL;

-- Grant necessary privileges to this user account
GRANT CREATE SESSION, DBA TO saas_sales_adm;

-- Makes the tablespace that you just created the default for saas_sales_adm
ALTER USER saas_sales_adm DEFAULT TABLESPACE saas_sales_tbs;

-- Now connect as the application owner
CONNECT saas_sales_adm/*****@saas_sales_ac

-- Create a metadata-linked table
CREATE TABLE saas_sales_adm.sales_mlt SHARING=METADATA
(YEAR      NUMBER(4),
 REGION    VARCHAR2(10),
 QUARTER   VARCHAR2(4),
 REVENUE   NUMBER);

-- End the application installation
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app END INSTALL '1.0';
```

ALTER PLUGGABLE DATABASE APPLICATION ... SYNC文を使用して、同じマスター・アプリケーション定義を使用するようにアプリケーションPDBを同期化できます。このように、あらゆるアプリケーションPDBがsaas_sales_adm.sales_mlt共通表へのメタデータ・リンクを持ちます。cust1_pdbというPDB内でsales_mltを更新する中間層コードはcust1_pdbでこの表に行を追加し、cust2_pdbでsales_mltを更新する中間層コードはcust2_pdbでこの表のコピーに行を追加します。アプリケーション・ルートに格納される表メタデータのみが共有されます。

ノート:



- [「アプリケーション共通オブジェクトについて」](#)
- 共通に付与されるオブジェクト権限の使用方法についてさらに学習するには、[Oracle Database セキュリティ・ガイド](#)を参照してください

- [メタデータ・リンク](#)

メタデータリンク・アプリケーション共通オブジェクトの場合、オブジェクトのメタデータはアプリケーション・ルートに1回格納されます。メタデータ・リンクは、共有しているメタデータと同じオブジェクト・タイプのディクショナリ・オブジェクトです。

親トピック: [アプリケーション共通オブジェクト](#)

メタデータ・リンク

メタデータリンク・アプリケーション共通オブジェクトの場合、オブジェクトのメタデータはアプリケーション・ルートに1回格納されます。メタデータ・リンクは、共有しているメタデータと同じオブジェクト・タイプのディクショナリ・オブジェクトです。

メタデータ・リンクの説明は、それが作成されたPDBのデータ・ディクショナリに格納されます。メタデータ・リンクは、アプリケーション共通ユーザーによって所有されている必要があります。CDBルートまたはアプリケーション・ルートで、その作成者によって所有される共通オブジェクトのメタデータを共有するには、メタデータ・リンクのみを使用できます。

データ・リンクと異なり、メタデータ・リンクは共通データのみ依存します。たとえば、アプリケーションでローカル表dow_close_ltおよびnasdaq_close_ltがアプリケーション・ルートに含まれる場合、共通ユーザーはこれらのオブジェクトへのメタデータ・リンクを作成できません。しかし、sales_mltというアプリケーション共通表はメタデータリンクできます。

権限を持つ共通ユーザーが表に列を追加するなどしてsales_mltのメタデータを変更した場合、この変更はメタデータ・リンクに伝播されます。アプリケーションPDBユーザーはメタデータ・リンクのメタデータを変更しない場合があります。たとえば、cust1_pdbというアプリケーションPDBを管理するDBAは、このPDBのみのsales_mltに列を追加できません。このようなメタデータの変更はアプリケーション・ルートのみで行われるためです。

関連項目:

[「アプリケーション共通オブジェクトについて」](#)

親トピック: [メタデータリンク・アプリケーション共通オブジェクト](#)

データリンク・アプリケーション共通オブジェクト

データリンク・オブジェクトは、そのメタデータおよびデータがアプリケーション・ルートに存在し、このアプリケーション・コンテナのすべてのアプリケーションPDBからアクセスできるオブジェクトです。

DATA値をSHARING句またはDEFAULT_SHARING初期化パラメータのいずれかで指定すると、[データリンク共通オブジェクト](#)と呼ばれる共通オブジェクトへのリンクが指定されます。データ・ウェアハウスのディメンション表はほとんどの場合、データリンク共通表の適切な候補です。

データ・リンクは、ほとんどシノニムのように機能するディクショナリ・オブジェクトです。たとえば、countriesがアプリケーション共通表の場合、すべてのアプリケーションPDBがデータ・リンクによってこの表の同じコピーにアクセスします。ある行がこの表に追加されると、この行はすべてのアプリケーションPDBで表示可能です。

データ・リンクは、アプリケーション共通ユーザーによって所有されている必要があります。リンクはそれが示しているオブジェクトからオブジェクト・タイプを継承します。データ・リンクの説明は、それが作成されたPDBのデータ・ディクショナリに格納されます。たとえば、アプリケーション・コンテナに10個のアプリケーションPDBが含まれ、すべてのPDBにcountriesアプリケーション共通表へのリンクが含まれる場合、10個のPDBすべてにこのリンクのディクショナリ定義が含まれます。

例2-9 データリンク・オブジェクトの作成

この例では、SYSTEMがsaas_sales_acアプリケーション・コンテナに接続します。SYSTEMは、saas_sales_appという名前のアプリケーションをバージョン1.0から2.0にアップグレードします。このアプリケーション・アップグレードは、共通ユーザーsaas_sales_admとしてコンテナにログインし、countries_dltという名前のデータリンク表を作成して、その表に行を挿入します。

```
-- Begin an upgrade of the application
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app BEGIN UPGRADE '1.0' to '2.0';

-- Connect as application owner to application root
CONNECT saas_sales_adm/manager@saas_sales_ac
```

```

-- Create data-linked table named countries_dlt
CREATE TABLE countries_dlt SHARING=DATA
(country_id NUMBER,
country_name VARCHAR2(20));

-- Insert records into countries_dlt
INSERT INTO countries_dlt VALUES(1, 'USA');
INSERT INTO countries_dlt VALUES(44, 'UK');
INSERT INTO countries_dlt VALUES(86, 'China');
INSERT INTO countries_dlt VALUES(91, 'India');

-- End application upgrade
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app END UPGRADE TO '2.0';

```

ALTER PLUGGABLE DATABASE APPLICATION ... SYNC文を使用して、アプリケーションPDBをアプリケーション・ルートと同期化します(「[アプリケーションの同期化](#)」を参照)。このようにして、同期化されたあらゆるアプリケーションPDBに saas_sales_adm.countries_dltデータリンク表へのデータ・リンクが作成されます。

ノート:



[「アプリケーション共通オブジェクトについて」](#)

親トピック: [アプリケーション共通オブジェクト](#)

拡張データリンク・アプリケーション・オブジェクト

拡張データリンク・アプリケーション・オブジェクトは、データリンク・オブジェクトとメタデータリンク・オブジェクトを合成したものです。

拡張データリンク・オブジェクトでは、アプリケーション・ルートに格納されたデータはすべてのアプリケーションPDBに共通で、すべてのPDBがこのデータにアクセスできます。ただし、アプリケーション・ルートの共通データを共有すると同時に、各アプリケーションPDBは独自のPDB固有データを作成できます。このように、PDBはその固有データで共通データを補完します。

たとえば、販売アプリケーションでいくつかのアプリケーションPDBをサポートする場合があります。すべてのアプリケーションPDBで米国の郵便番号が必要です。この場合、アプリケーション・ルートでzipcodes_edt拡張データリンク表を作成できます。アプリケーション・ルートが米国の郵便番号を格納するため、すべてのアプリケーションPDBがこのデータにアクセスできます。ただし、1つのアプリケーションPDBでは米国およびカナダの郵便番号が必要です。このアプリケーションPDBは、カナダの郵便番号をアプリケーション・ルートではなくアプリケーションPDB内の拡張データリンク・オブジェクトに格納できます。

拡張データリンク・オブジェクトを作成するには、アプリケーション・ルートに接続して、CREATE文にSHARING=EXTENDED DATAキーワードを指定します。

例2-10 拡張データ・オブジェクトの作成

この例では、SYSTEMがsaas_sales_acアプリケーション・コンテナに接続し、(「[例2-8](#)」で作成した) saas_sales_appという名前のアプリケーションをバージョン2.0から3.0にアップグレードします。このアプリケーションは、共通ユーザーsaas_sales_admとしてコンテナにログインし、zipcodes_edtという名前の拡張データリンク表を作成して、その表に行を挿入します。

```

-- Begin an upgrade of the app
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app BEGIN UPGRADE '2.0' to '3.0';

-- Connect as app owner to app root
CONNECT saas_sales_adm/manager@saas_sales_ac

```

```

-- Create a common-data table named zipcodes_edt
CREATE TABLE zipcodes_edt SHARING=EXTENDED DATA
(code          VARCHAR2(5),
 country_id   NUMBER,
 region      VARCHAR2(10));

-- Load rows into zipcodes_edt
INSERT INTO zipcodes_edt VALUES (' 08820', ' 1', ' East');
INSERT INTO zipcodes_edt VALUES (' 10005', ' 1', ' East');
INSERT INTO zipcodes_edt VALUES (' 44332', ' 1', ' North');
INSERT INTO zipcodes_edt VALUES (' 94065', ' 1', ' West');
INSERT INTO zipcodes_edt VALUES (' 73301', ' 1', ' South');
COMMIT;

-- End app upgrade
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app END UPGRADE TO ' 3.0';

```

ALTER PLUGGABLE DATABASE APPLICATION ... SYNC文を使用して、アプリケーションPDBをアプリケーションと同期化します([「アプリケーションの同期化」を参照](#))。このようにして、同期化されたあらゆるアプリケーションPDBに saas_sales_adm.zipcodes_edtデータリンク表へのデータ・リンクが作成されます。これらのPDBに接続するアプリケーションは、アプリケーションのアップグレード中にzipcodes_edtに挿入された郵便番号を参照できますが、この表に独自の郵便番号を挿入することもできます。

ノート:



[「アプリケーション共通オブジェクトについて」](#)

親トピック: [アプリケーション共通オブジェクト](#)

アプリケーション・メンテナンス

このコンテキストにおける**アプリケーション・メンテナンス**は、アプリケーションのインストール、アンインストール、アップグレードまたはパッチ適用を指します。

アプリケーションには名前およびバージョン番号が必要です。このプロパティの組合せによって、実行可能なメンテナンス作業が決まります。すべてのメンテナンス作業で、次のステップを実行します。

1. ALTER PLUGGABLE DATABASE ... APPLICATION文をBEGIN INSTALL句、BEGIN UPGRADE句またはBEGIN PATCH句とともに実行することで開始します。
2. 文を実行してアプリケーションを変更します。
3. ALTER PLUGGABLE DATABASE ... APPLICATION文をEND INSTALL句、END UPGRADE句またはEND PATCH句とともに実行することで終了します。

アプリケーションの発展に伴って、アプリケーション・コンテナではバージョンおよびパッチの変更がすべて保守されます。

ノート:



[「アプリケーションの管理について」](#)

- [アプリケーション・メンテナンスについて](#)

ALTER PLUGGABLE DATABASE APPLICATION文を使用して、アプリケーションのインストール、アップグレードおよびパッチ適用操作を実行します。

- [アプリケーションのインストール](#)

アプリケーションのインストールは、マスター・アプリケーション定義の初期作成操作です。通常のインストールではユーザー・アカウント、表およびPL/SQLパッケージが作成されます。

- [アプリケーションのアップグレード](#)

アプリケーションのアップグレードはインストールされたアプリケーションへの大規模な変更です。

- [アプリケーション・パッチ](#)

アプリケーション・パッチは、アプリケーションの小規模な変更です。

親トピック: [アプリケーション・コンテナ内のアプリケーションの概要](#)

アプリケーション・メンテナンスについて

ALTER PLUGGABLE DATABASE APPLICATION文を使用して、アプリケーションのインストール、アップグレードおよびパッチ適用操作を実行します。

アプリケーション・メンテナンスの基本的なステップは、次のとおりです。

1. アプリケーション・ルートにログインします。
2. アプリケーション・ルートでALTER PLUGGABLE DATABASE APPLICATION ... BEGIN文を使用して操作を開始します。
3. アプリケーション・メンテナンスの文を実行します。
4. ALTER PLUGGABLE DATABASE APPLICATION ... END文を使用して操作を終了します。

スクリプト、SQL文またはGUIツールを使用してメンテナンスを実行します。

関連項目:

[「アプリケーションの管理について」](#)

親トピック: [アプリケーション・メンテナンス](#)

アプリケーションのインストール

アプリケーションのインストールは、マスター・アプリケーション定義の初期作成操作です。通常のインストールではユーザー・アカウント、表およびPL/SQLパッケージが作成されます。

アプリケーションをインストールするには、ALTER PLUGGABLE DATABASE APPLICATION文に次の項目を指定します。

- アプリケーション名
- アプリケーションのバージョン番号

例2-11 アプリケーションのインストール

この例では、saas_sales_acという名前のアプリケーション・コンテナにログインしていると仮定します。例ではバージョン1.0のsaas_sales_appというアプリケーションがインストールされます。バージョンを数値ではなく文字列で指定することに注意してください。このアプリケーションは、saas_sales_admという名前のアプリケーション共通ユーザーを作成し、必要な権限を付与してから、このユーザーとしてアプリケーション・ルートに接続します。このユーザーは、sales_mltという名前のメタデータリンク表を作成します。

```

-- Begin the install of saas_sales_app
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app BEGIN INSTALL '1.0';

-- Create the tablespace for the app
CREATE TABLESPACE saas_sales_tbs DATAFILE SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE 200M;

-- Create the user account saas_sales_adm, which will own the application
CREATE USER saas_sales_adm IDENTIFIED BY manager CONTAINER=ALL;

-- Grant necessary privileges to this user account
GRANT CREATE SESSION, DBA TO saas_sales_adm;

-- Make the tablespace that you just created the default for saas_sales_adm
ALTER USER saas_sales_adm DEFAULT TABLESPACE saas_sales_tbs;

-- Now connect as the application owner
CONNECT saas_sales_adm/manager@saas_sales_ac

-- Create a metadata-linked table
CREATE TABLE saas_sales_adm.sales_mlt SHARING=METADATA
(YEAR      NUMBER(4),
 REGION    VARCHAR2(10),
 QUARTER   VARCHAR2(4),
 REVENUE   NUMBER);

-- End the application installation
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app END INSTALL '1.0';

```

[PDBの同期化](#)は、ユーザーがアプリケーション・ルートのアプリケーションを使用して開始するアプリケーションPDBの更新です。saas_sales_appアプリケーションを使用してアプリケーションPDBを同期化した後、各アプリケーションPDBにproducts_mltと呼ばれる空の表が格納されます。アプリケーションは、アプリケーションPDBに接続してから、この表にPDB固有の行を挿入できます。

関連項目:

- [「アプリケーションの同期化」](#)
- [「自動伝播によるアプリケーション・コンテナ内のアプリケーションのインストール」](#)

親トピック: [アプリケーション・メンテナンス](#)

アプリケーションのアップグレード

アプリケーションのアップグレードはインストールされたアプリケーションへの大規模な変更です。

通常、アップグレードではアプリケーションの物理アーキテクチャが変更されます。たとえば、アップグレードによって新規のユーザー・アカウント、表およびパッケージが追加されたり、既存のオブジェクトの定義が変更されます。

アプリケーションをアップグレードするには、ALTER PLUGGABLE DATABASE APPLICATION文で次の項目を指定する必要があります。

- アプリケーション名
- 古いアプリケーションのバージョン番号
- 新規のアプリケーションのバージョン番号

例2-12 自動による手法を使用したアプリケーションのアップグレード

この例では、管理者としてアプリケーション・ルートに接続し、アプリケーションsaas_sales_appをバージョン1.0からバージョン2.0にアップグレードします。アップグレードにより、countries_dltという名前のデータリンク表が作成され、その表に行が追加されます。また、zipcodes_edtという名前の拡張データリンク表が作成され、その表に行が追加されます。

```
-- Begin an upgrade of the app
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app
  BEGIN UPGRADE '1.0' to '2.0';

-- Connect as app owner to app root
CONNECT saas_sales_adm/manager@saas_sales_ac

-- Create data-linked table named countries_dlt
CREATE TABLE countries_dlt SHARING=DATA
(country_id  NUMBER,
 country_name VARCHAR2(20));

-- Insert records into countries_dlt
INSERT INTO countries_dlt VALUES(1, 'USA');
INSERT INTO countries_dlt VALUES(44, 'UK');
INSERT INTO countries_dlt VALUES(86, 'China');
INSERT INTO countries_dlt VALUES(91, 'India');

-- Create an extended data-linked table named zipcodes_edt
CREATE TABLE zipcodes_edt SHARING=EXTENDED DATA
(code        VARCHAR2(5),
 country_id  NUMBER,
 region     VARCHAR2(10));

-- Load rows into zipcodes_edt
INSERT INTO zipcodes_edt VALUES ('08820', '1', 'East');
INSERT INTO zipcodes_edt VALUES ('10005', '1', 'East');
INSERT INTO zipcodes_edt VALUES ('44332', '1', 'North');
INSERT INTO zipcodes_edt VALUES ('94065', '1', 'West');
INSERT INTO zipcodes_edt VALUES ('73301', '1', 'South');
COMMIT;

-- End app upgrade
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app END UPGRADE TO '2.0';
```

- [アプリケーションのアップグレードの動作](#)
アプリケーションのアップグレード中に、アプリケーションは引き続き使用可能です。この可用性を可能にするため、Oracle Databaseではアプリケーション・ルートがクローニングされます。
- [異なるバージョンのアプリケーション](#)
異なるアプリケーションPDBが、異なるバージョンのアプリケーションを使用する場合があります。

関連項目:

[「アプリケーション・コンテナ内のアプリケーションのアップグレード」](#)

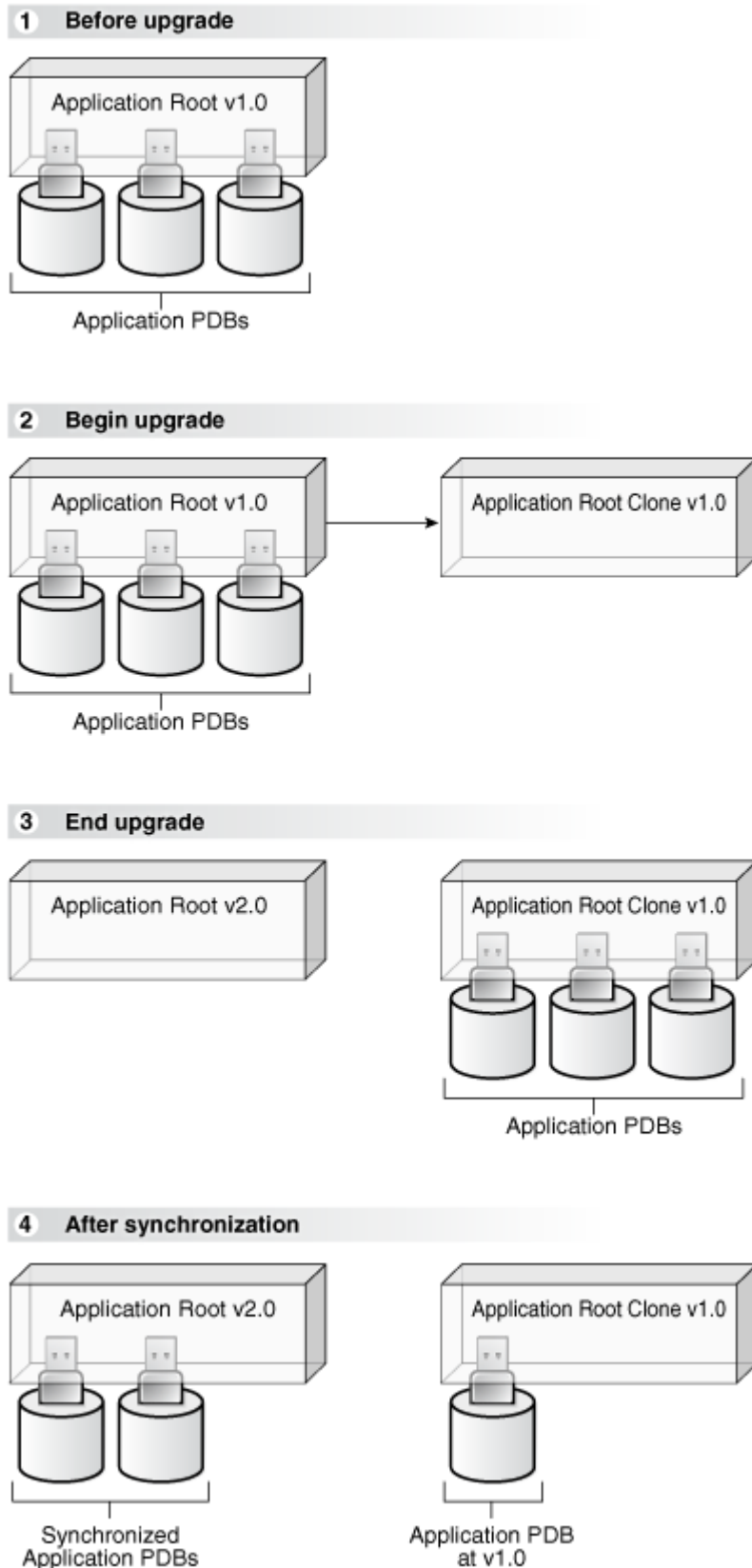
親トピック: [アプリケーション・メンテナンス](#)

アプリケーションのアップグレードの動作

アプリケーションのアップグレード中に、アプリケーションは引き続き使用可能です。この可用性を可能にするため、Oracle Databaseではアプリケーション・ルートがクローニングされます。

次の図で、アプリケーションのアップグレード・プロセスの概要を示します。

図2-9 アプリケーションのアップグレード



アップグレードは次のように実行されます。

1. 初期の状態で、アプリケーション・ルートには特定のバージョンのアプリケーションがあります。

2. ユーザーはALTER PLUGGABLE DATABASE APPLICATION BEGIN UPGRADE文を実行してから、アプリケーションのアップグレード文を発行します。

アップグレード中にデータベースでは自動的に次の処理が実行されます。

- アプリケーション・ルートをクローニングします
たとえば、saas_sales_appアプリケーションがアプリケーション・ルートでバージョン1.0である場合、クローンもバージョン1.0になります。
- アプリケーションPDBをアプリケーション・ルート・クローンに示します
クローンは読取り専用モードです。アプリケーションはアプリケーションPDBに対して引き続き使用可能です。

3. ユーザーはALTER PLUGGABLE DATABASE APPLICATION END UPGRADE文を実行します。

この段階で、アプリケーションPDBは依然としてアプリケーション・ルート・クローンを示しており、元のアプリケーション・ルートが新規バージョンになります。たとえば、saas_sales_appアプリケーションがアプリケーション・ルートでバージョン1.0である場合、アップグレードによってこれがバージョン2.0になります。ただし、アプリケーション・ルート・クローンはバージョン1.0のままです。

4. オプションで、SYNC句とともにALTER PLUGGABLE DATABASE APPLICATION文を発行することで、ユーザーはアプリケーションPDBをアップグレード済のアプリケーション・ルートと同期化します。

たとえば、同期後に、一部のアプリケーションPDBはバージョン2.0のアプリケーション・ルートに接続されます。ただし、アプリケーション・ルート・クローンでは、バージョン1.0のままであることが必要なアプリケーションPDBや、バージョン1.0のアプリケーション・ルートに接続される新規のアプリケーションPDBが引き続きサポートされます。

関連項目:

- [「アプリケーションの同期化」](#)
- [「アプリケーション・コンテナ内のアプリケーションのアップグレード」](#)

親トピック: [アプリケーションのアップグレード](#)

異なるバージョンのアプリケーション

異なるアプリケーションPDBが、異なるバージョンのアプリケーションを使用する場合があります。

たとえば、あるアプリケーションPDBにsaas_sales_appのバージョン1.0があるとします。同じアプリケーション・コンテナの別のアプリケーションPDBには、このアプリケーションのバージョン2.0があります。

ユースケースとしては、異なる顧客に提供されたSaaSアプリケーションがあります。個々の顧客に各自のアプリケーションPDBがある場合、中にはアプリケーションをアップグレードするまでより長期間待機する顧客がいます。この場合、あるアプリケーションPDBでは最新バージョンのアプリケーションが使用されますが、他のアプリケーションPDBでは古いバージョンが使用される可能性があります。

関連項目:

異なるバージョンのアプリケーションについてさらに学習するには、[「アプリケーション・コンテナ内のアプリケーションのアップグレード」](#)を参照してください

親トピック: [アプリケーションのアップグレード](#)

アプリケーション・パッチ

アプリケーション・パッチは、アプリケーションの小規模な変更です。

アプリケーションのパッチ適用の一般的な例として、バグ修正やセキュリティ・パッチがあります。パッチの内部では、新しいファンクションおよびパッケージが許可されます。

一般に、破壊的な操作は許可されません。たとえば、DROP文や、列の削除またはデータ型の変更を行うALTER TABLE文をパッチに含めることはできません。

Oracle Databaseのパッチ適用プロセスでOracle Databaseパッチで許可される操作の種類が制限されるのと同様、アプリケーションのパッチ適用プロセスではアプリケーション・パッチで許可される操作が制限されます。ある修正に「アプリケーション・パッチで操作がサポートされていません」というエラーを引き起こす操作が含まれる場合は、かわりに[アプリケーションのアップグレード](#)を実行してください。

ノート:



他のアプリケーションのパッチ適用またはアップグレードが進行中の間は、アプリケーションのパッチ適用ができません。

アプリケーションにパッチを適用するには、ALTER PLUGGABLE DATABASE APPLICATION文にアプリケーション名とパッチ番号を指定します。オプションで、アプリケーション最小バージョンを指定できます。

例2-13 自動による手法を使用したアプリケーションのパッチ適用

この例では、SYSTEMがアプリケーション・ルートにログインし、アプリケーションsaas_sales_appにバージョン1.0以上のパッチを適用します。パッチ101は、saas_sales_admとしてアプリケーション・コンテナにログインし、get_total_revenueという名前のメタデータリンクされたPL/SQLファンクションを作成します。

```
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app BEGIN PATCH 101 MINIMUM VERSION '1.0';

-- Connect to the saas_sales_ac container as saas_sales_adm, who owns the application
CONNECT saas_sales_adm/*****@saas_sales_ac

-- Now install the get_total_revenue() function
CREATE FUNCTION get_total_revenue SHARING=METADATA (p_year IN NUMBER)
RETURN SYS_REFCURSOR
AS
c1_cursor SYS_REFCURSOR;
BEGIN
OPEN c1_cursor FOR
  SELECT a.year, sum(a.revenue)
  FROM containers(sales_data) a
  WHERE a.year = p_year
  GROUP BY a.year;
RETURN c1_cursor;
END;
/

-- End the patch
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app END PATCH 101;
```

関連項目:

[「アプリケーション・コンテナ内のアプリケーションへのパッチ適用」](#)

親トピック: [アプリケーション・メンテナンス](#)

既存のアプリケーションの移行

PDBにインストールされているアプリケーションを、アプリケーション・ルートまたはアプリケーションPDBのいずれかに移行できます。

既存のアプリケーションを移行する一般的な理由として、次のものがあります。

- インストール・プログラムを使用するアプリケーション
一部のアプリケーションは、スクリプトではなくインストール・プログラムを使用します。この場合は、新しいアプリケーション・ルートでインストール・プログラムを実行し、DBMS_PDB_ALTER_SHARINGパッケージを使用してオブジェクトを適切な共有モード(METADATA、DATAまたはEXTENDED DATA)に設定します。この変更はルートからアプリケーションPDBに自動的に伝播します。Oracle Databaseによってインストールの文ログが作成されるため、前のアプリケーション・バージョンのPDBはアプリケーション・ルートに接続できます。
- 各PDBで別個に定義されたアプリケーション
一部のアプリケーションは各PDBで定義されますが、アプリケーション・コンテナが存在しません。この場合は、インストール・スクリプトを更新して適切な共有モードを設定できます。アプリケーション・ルートを作成し、このルートでマスター・アプリケーション定義を作成します。既存のPDBをアプリケーションPDBとして採用するには、そのPDBをアプリケーション・ルートに接続し、共通定義への参照を使用して定義全体を置き換えるSQLスクリプトを実行します。

たとえば、Oracle Database 12c CDBに接続中のPDBにインストールされているアプリケーションを、Oracle Database 18c CDBのアプリケーション・コンテナに移行できます。

関連項目:

- 既存アプリケーションの移行方法を学習するには、「[アプリケーションの管理について](#)」を参照してください。
- DBMS_PDB_ALTER_SHARINGパッケージについてさらに学習するには、『[Oracle Database PL/SQLパッケージおよびタイプ・リファレンス](#)』を参照してください

親トピック: [アプリケーション・コンテナ内のアプリケーションの概要](#)

暗黙的に作成されるアプリケーション

ユーザー作成アプリケーション以外に、アプリケーション・コンテナには暗黙的に作成されるアプリケーションも含まれます。

ALTER PLUGGABLE DATABASE BEGIN文を最初に使用せずに、アプリケーション共通ユーザー操作がCONTAINER=ALL句によって発行されると、アプリケーション・ルートでアプリケーションが暗黙的に作成されます。

アプリケーション共通ユーザー操作には、CREATE USER文による共通ユーザーの作成や、ALTER USER文による共通ユーザーの変更などの操作が含まれます。データベースによって暗黙的アプリケーションにAPP\$guid (guidはアプリケーション・ルートのグローバル一意ID)という名前が自動的に付けられます。暗黙的アプリケーションは、アプリケーション・ルートが初めて開かれたときに作成されます。

関連項目:

暗黙的に作成されるアプリケーションについてさらに学習するには、[「アプリケーションPDB内のアプリケーションの同期」](#)を参照してください

親トピック: [アプリケーション・コンテナ内のアプリケーションの概要](#)

アプリケーションの同期化

アプリケーションPDB内で、同期化は、アプリケーション・ルートでの最新のバージョンおよびパッチへのユーザー始動のアプリケーションの更新です。

アプリケーション・ルート内でアプリケーションがインストール、アップグレード、パッチ適用またはアンインストールされた場合、その変更は、自動的にアプリケーションPDBに伝播されません。PDBを手動で同期する必要があります。アプリケーションPDBに接続している場合は、ALTER PLUGGABLE DATABASE APPLICATION ... SYNCを発行することで1つ以上のアプリケーションを同期できます。

- [単一のアプリケーションの同期](#)
SYNCの前に1つのアプリケーション名を指定した場合は、指定されたアプリケーションのみがデータベースによって同期されます。
- [複数のアプリケーションの同期](#)
ALLキーワードを指定することで、複数のアプリケーションを同期できます。

親トピック: [アプリケーション・コンテナ内のアプリケーションの概要](#)

単一のアプリケーションの同期

SYNCの前にアプリケーション名を指定した場合は、指定されたアプリケーションのみがデータベースによって同期されます。

次の文は、アプリケーションPDBで実行され、apexappとそのアプリケーションPDBを同期します。

```
ALTER PLUGGABLE DATABASE APPLICATION apexapp SYNC;
```

SYNC TO PATCH patchnum句を使用すると、アプリケーションを特定のパッチ番号に同期できます。次の文は、saas_sales_appというアプリケーションを、アプリケーションPDB内のパッチ100に同期します。

```
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC TO PATCH 100;
```

アプリケーションを特定のアプリケーション・バージョンに同期するには、SYNC TO versionを使用します。次の文は、saas_sales_appというアプリケーションを、アプリケーションPDB内のバージョン2.0に同期します。

```
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC TO '2.0';
```

親トピック: [アプリケーションの同期化](#)

複数のアプリケーションの同期

ALLキーワードを指定することで、複数のアプリケーションを同期できます。

ALL SYNCを指定した場合は、データベースにより、暗黙的に作成されたアプリケーションも含め、すべてのアプリケーションが同期されます。ALLではSYNC TO PATCH patchno句およびSYNC TO version句はサポートされていないことに注意してください。次の文は、すべてのアプリケーションを同期します。

同期中の、アプリケーションのBEGINおよびENDブロックのリプレイ順序は、取得順序と同じです。

関連項目:

[「アプリケーションPDB内のアプリケーションの同期」](#)

親トピック: [アプリケーションの同期化](#)

コンテナ・マップ

コンテナ・マップを使用することにより、アプリケーション・ルートに接続したセッションで発行したSQL文が、SQL文に使用された条件の値に応じて適切なPDBにルーティングされるようになります。

マップ表はメタデータリンクされた共通表内の列を指定し、パーティションを使用して異なるアプリケーションPDBを異なる列値と関連付けます。このように、データが表レベルで物理的にパーティション化されていないときに、コンテナ・マップはPDBレベルでのデータのパーティション化を有効にします。

コンテナ・マップを使用するための主なコンポーネントは次のとおりです。

- **メタデータリンク表**

この表はコンテナ・マップを使用して問合せするためのものです。たとえば、各アプリケーションPDBに異なるデータを格納する、countries_mltというメタデータリンク表を作成する場合があります。amer_pdb内のcountries_mlt.cname列には北アメリカの国名が格納され、euro_pdb内のcountries_mlt.cname列にはヨーロッパの国名が格納され、asia_pdb内のcountries_mlt.cname列にはアジアの国名が格納されます。

- **マップ表**

アプリケーション・ルートで、リスト、ハッシュまたはレンジ別にパーティション化された単一系列のマップ表を作成します。マップ表によって、コンテナ・マップで有効化されているパーティション化計画を使用して、メタデータリンク表が問合せできるようになります。マップ・オブジェクト表内のパーティションの名前は、アプリケーション・コンテナ内のアプリケーションPDBの名前と一致する必要があります。

たとえば、pdb_map_tblというマップ表がcname列でリスト別にパーティション化します。amer_pdb、euro_pdbおよびasia_pdbというパーティションがアプリケーションPDBの名前に対応します。各パーティションの値は、たとえばPARTITION amer_pdb VALUES ('US', 'MEXICO', 'CANADA')のように、国の名前です。

Oracle Database 18c以降は、CONTAINERS() 問合せでマップを使用するために、マップ表のパーティション化列がメタデータリンク表の列と一致している必要はありません。表sh.salesがコンテナ・マップpdb_map_tblに対して有効で、cnameがマップ表のパーティション化列であるとします。sh.salesにcname列が含まれていない場合でも、マップ表は次の問合せを適切なPDBにルーティングします: SELECT * FROM CONTAINERS(sh.sales) WHERE cname = 'US' ORDER BY time_id。

- **コンテナ・マップ**

コンテナ・マップは、マップ表を指定するデータベース・プロパティです。プロパティを設定するには、アプリケーション・ルートに接続してALTER PLUGGABLE DATABASE SET CONTAINER_MAP=map_table文を実行します。map_tableはマップ表の名前を表します。

例2-14 メタデータリンク表、マップ表およびコンテナ・マップの作成: パート1

この例では、アプリケーション管理者としてアプリケーション・ルートにログインします。アプリケーション・コンテナに、amer_pdb、euro_pdbおよびasia_pdbという3つのアプリケーションPDBがあると想定します。各アプリケーションPDBは異なる地域の国名を格納します。oe.countries_mltというメタデータリンク表に、国名が格納されるcname列があります。このパーティション化計画のため、リスト別のパーティションを使用して、各地域のパーティションを作成するsalesadm.pdb_map_tblと言う名前のマップ・オブジェクトを作成します。国名によって地域が決まります。

```
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app BEGIN INSTALL '1.0';

-- Create the metadata-linked table.
CREATE TABLE oe.countries_mlt SHARING=METADATA (
  region  VARCHAR2(30),
  cname   VARCHAR2(30));

-- Create the partitioned map table, which is list partitioned on the
-- cname column. The names of the partitions are the names of the
-- application PDBs.
CREATE TABLE salesadm.pdb_map_tbl (cname VARCHAR2(30) NOT NULL)
PARTITION BY LIST (cname) (
  PARTITION amer_pdb VALUES ('US', 'MEXICO', 'CANADA'),
  PARTITION euro_pdb VALUES ('UK', 'FRANCE', 'GERMANY'),
  PARTITION asia_pdb VALUES ('INDIA', 'CHINA', 'JAPAN'));

-- Set the CONTAINER_MAP database property to the map object.
ALTER PLUGGABLE DATABASE SET CONTAINER_MAP='salesadm.pdb_map_tbl';

-- Enable the container map for the metadata-linked table to be queried.
ALTER TABLE oe.countries_mlt ENABLE CONTAINER_MAP;

-- Ensure that the table to be queried is enabled for the
-- CONTAINERS clause.
ALTER TABLE oe.countries_mlt ENABLE CONTAINERS_DEFAULT;

-- End the application installation.
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app END INSTALL '1.0';
```

ノート:



パーティション化構文を使用してコンテナ・マップを作成しても、データベースではパーティション化機能を使用しません。コンテナ・マップの定義に Oracle パーティション化は必要ありません。

前述のスクリプトでは、ALTER TABLE oe.countries_mlt ENABLE CONTAINERS_DEFAULT文によって、アプリケーション・ルートで発行される問合せおよびDML文がデータベース・オブジェクトに対してデフォルトでCONTAINERS()句を使用する必要があることを指定します。

例2-15 アプリケーションの同期化と、データの追加: パート2

この例は前の例からの続きです。アプリケーション・ルートへの接続中に、現在のコンテナを各PDBに順番に切り替えて、saas_sales_appアプリケーションを同期化し、PDB固有データをoe.countries_mlt表に追加します。

```
ALTER SESSION SET CONTAINER=amer_pdb;
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;
INSERT INTO oe.countries_mlt VALUES ('AMER', 'US');
INSERT INTO oe.countries_mlt VALUES ('AMER', 'MEXICO');
INSERT INTO oe.countries_mlt VALUES ('AMER', 'CANADA');
```



```

COMMIT;

ALTER SESSION SET CONTAINER=euro_pdb;
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;
INSERT INTO oe.countries_mlt VALUES ('EURO','UK');
INSERT INTO oe.countries_mlt VALUES ('EURO','FRANCE');
INSERT INTO oe.countries_mlt VALUES ('EURO','GERMANY');
COMMIT;

ALTER SESSION SET CONTAINER=asia_pdb;
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;
INSERT INTO oe.countries_mlt VALUES ('ASIA','INDIA');
INSERT INTO oe.countries_mlt VALUES ('ASIA','CHINA');
INSERT INTO oe.countries_mlt VALUES ('ASIA','JAPAN');
COMMIT;

```

例2-16 メタデータリンク表の問合せ: パート3

この例は前の例からの続きです。アプリケーション・ルートへ接続して、WHERE句で別の国を指定しながらoe.countries_mltを複数回、問合せします。問合せではoe.countries_mlt.region列から正しい値が返されます。

```

ALTER SESSION SET CONTAINER=saas_sales_ac;

SELECT region FROM oe.countries_mlt WHERE cname='MEXICO';

REGION
-----
AMER

SELECT region FROM oe.countries_mlt WHERE cname='GERMANY';

REGION
-----
EURO

SELECT region FROM oe.countries_mlt WHERE cname='JAPAN';

REGION
-----
ASIA

```

関連項目:

[「コンテナ・マップを使用したPDBでのパーティション化」](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションの概要](#)

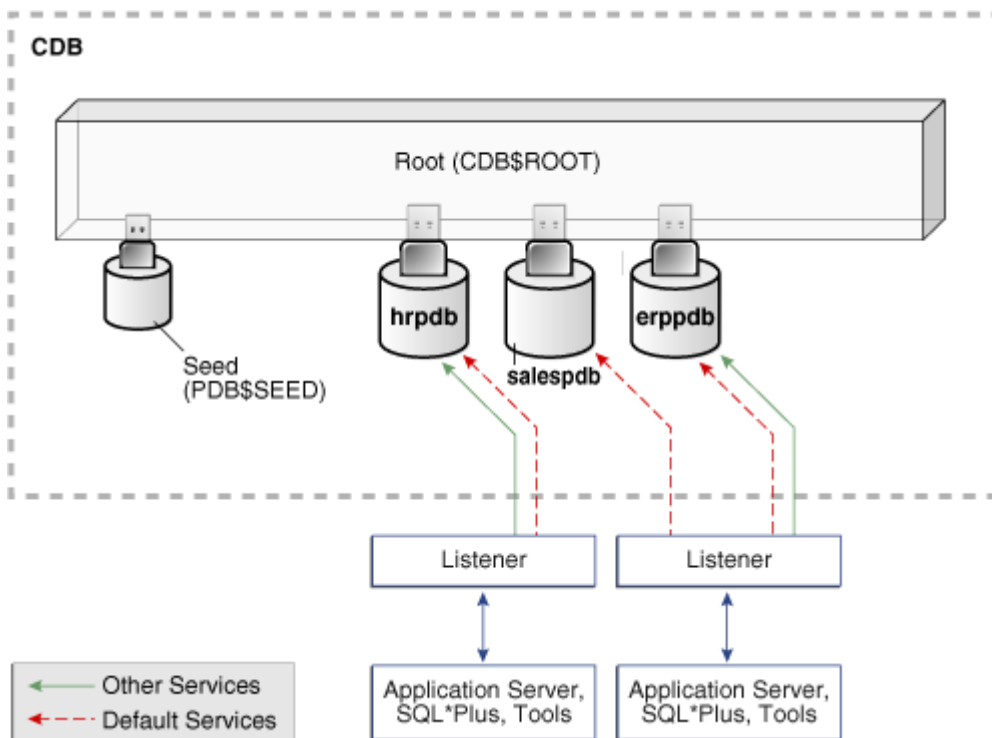
CDBのサービスの概要

クライアントは、サービスを使用してPDBまたはアプリケーション・ルートに接続する必要があります。

サービス名を使用した接続により、PDBまたはアプリケーション・ルートでは新規のセッションが開始されます。フォアグラウンド・プロセス、およびその結果のセッションは、ライフタイムの各時点で、一意に定義された現在のコンテナがあります。

次の図に、2つの異なるリスナーを使用してPDBに接続する2つのクライアントを示します。

図2-10 CDBのサービス



- [CDBのサービス作成](#)
CREATE PLUGGABLE DATABASE文を実行してPDBを作成すると、データベースではCDB内のサービスが自動的に作成されて起動されます。
- [CDBのコンテナへの接続](#)
一般に、CDB管理者はPDBをプロビジョニングして様々なコンテナに接続するための適切な権限が必要です。CDB管理者は共通ユーザーです。

関連項目:

[「PDBのサービスの管理」](#)

親トピック: [マルチテナント・アーキテクチャの概要](#)

CDBのサービス作成

CREATE PLUGGABLE DATABASE文を実行してPDBを作成すると、データベースではCDB内のサービスが自動的に作成されて起動されます。

このデフォルト・サービスには、PDBをサービスの最初の現行コンテナとして識別するプロパティがあります。このプロパティは、DBA_SERVICES. PDB列に表示されます。

- [CDBのデフォルト・サービス](#)
デフォルト・サービスの名前はPDBと同じになります。PDB名は、CDB内で一意となる有効なサービス名である必要があります。
- [CDBの非デフォルト・サービス](#)
各PDBについて、CDBごとに上限10,000までの追加サービスを作成できます。各追加サービスでは、そのPDBが最初の現行コンテナとして表示されます。

関連項目:

[「PDBのサービスの管理」](#)

親トピック: [CDBのサービスの概要](#)

CDBのデフォルト・サービス

デフォルト・サービスの名前はPDBと同じになります。PDB名は、CDB内で一意となる有効なサービス名である必要があります。アプリケーション・コンテナを作成する際、AS APPLICATION CONTAINER句の指定が必要になると、Oracle Databaseではアプリケーション・ルートの新規のデフォルト・サービスが自動的に作成されます。サービスはアプリケーション・コンテナと同じ名前を持ちます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。同様に、すべてのアプリケーションPDBには固有のデフォルト・サービス名があり、アプリケーション・シードPDBには固有のデフォルト・サービス名があります。

例2-17 デフォルト・サービスを使用したPDBへの切替え

この例では、PDBと同じ名前を持つデフォルト・サービスを使用して、PDB名salespdbに切り替えます。

```
ALTER SESSION SET CONTAINER = salespdb;
```

関連項目:

- サービス名の詳細は、『[Oracle Database概要](#)』を参照してください
- [「PDBのサービスの管理」](#)

親トピック: [CDBのサービスの作成](#)

CDBの非デフォルト・サービス

各PDBについて、CDBごとに上限10,000までの追加サービスを作成できます。各追加サービスでは、そのPDBが最初の現行コンテナとして表示されます。

[図2-10](#)では、erppdbとhrpdbにデフォルト以外のサービスが存在します。非CDBで使用するのと同じ方法で、サービスを作成、維持および削除します。

たとえば、[図2-10](#)では、hrpdbというPDBには、hrpdbというデフォルトのサービスがあります。デフォルト・サービスは削除できません。

ALTER SESSION SET CONTAINERを使用してコンテナに切り替えると、コンテナのデフォルト・サービスがセッションで使用されます。オプションで、SERVICE = service_nameを指定することでコンテナに別のサービスを使用できます。service_nameはサービスの名前を表します。特定のサービスを使用することで、セッションがそのサービス属性および機能(サービス・メトリック、ロード・バランシング、Resource Manager設定など)を利用できるようにする場合があります。

例2-18 非デフォルト・サービスを使用したPDBへの切替え

この例では、hrpdbのデフォルト・サービスによってすべてのサービス属性および機能(サービス・メトリック、FAN、ロード・バランシング、Oracle Database Resource Manager、トランザクション・ガード、アプリケーション・コンティニューイティなど)がサポートされるとはかぎりません。非デフォルト・サービスには次のように切り替えます。

```
ALTER SESSION SET CONTAINER = hrpdb SERVICE = hrpdb_full;
```

CDBのコンテナへの接続

一般に、CDB管理者はPDBをプロビジョニングして様々なコンテナに接続するための適切な特権を持つ必要があります。CDB管理者は共通ユーザーです。

CDB管理者は、次のいずれかの方法を使用できます。

- PDBまたはアプリケーション・ルートに直接接続します。
ユーザーにはそのコンテナでのCREATE SESSION権限が必要です。
- 接続プーリングおよび高度なCDB管理の両方に役立つALTER SESSION SET CONTAINER文を使用して、コンテナの切替えを行います。構文はALTER SESSION SET CONTAINER = container_name [SERVICE = service_name]です。
たとえば、CDB管理者は、あるセッションのルートに接続してから、同じセッションのPDBに切り替えることができます。この場合、ユーザーにはそのコンテナでのSET CONTAINERシステム権限が必要です。

次の表に、[図2-10](#)のCDBに関連したシナリオを示します。各行は、前の行のアクションの後に発生するアクションについて説明します。共通ユーザーSYSTEMは、現行コンテナの名前と、CDB内のPDBの名前を問い合せます。

表2-8 CDBのサービス

操作	説明
SQL> CONNECT SYSTEM@prod Enter password: ***** Connected.	SYSTEM ユーザーは、CDB のすべてのコンテナに対して共通で、prod というサービスを使用してルートに接続します。
SQL> SHOW CON_NAME CON_NAME ----- CDB\$ROOT	SYSTEM は、ユーザーが現在接続されているコンテナの名前をリストするために、SQL*Plus コマンド SHOW CON_NAME を使用します。CDB\$ROOT は、ルート・コンテナの名前です。
SQL> SELECT NAME, PDB FROM V\$SERVICES 2 ORDER BY PDB, NAME; NAME PDB ----- --- SYS\$BACKGROUND CDB\$ROOT SYS\$USERS CDB\$ROOT prod.example.com CDB\$ROOT erppdb.example.com ERPPDB erp.example.com ERPPDB hr.example.com HRPDB hrpdb.example.com HRPDB salespdb.example.com SALESPDB 8 rows selected.	V\$SERVICES の問合せは、PDB 名と一致するサービス名を持つ 3 つの PDB が存在することを示します。hrpdb と erppdb の両方に追加サービスが 1 つあります。
SQL> ALTER SESSION SET CONTAINER = hrpdb; Session altered.	SYSTEM は、ALTER SESSION を使用して hrpdb に接続します。
SQL> SELECT SYS_CONTEXT 2 ('USERENV', 'CON_NAME') 3 AS CUR_CONTAINER FROM DUAL; CUR_CONTAINER ----- HRPDB	問合せにより、現行コンテナが hrpdb であることを確認します。

関連項目:

- [「PDBへの接続」](#)
- ALTER SESSION SET CONTAINERの構文およびセマンティクスについては、[『Oracle Database SQL言語リファレンス』](#)

親トピック: [CDBのサービスの概要](#)

CDBの表領域およびデータベース・ファイルの概要

CDBの構造は非CDBと同じですが、各PDBおよびアプリケーション・ルートに独自の表領域のセット(独自のSYSTEM、SYSAUXおよびUNDO表領域を含む)が存在する点が異なります。

CDBには次のファイルが含まれています。

- 1つの制御ファイル
- 1つのオンラインREDOログ
- 1つ以上のUNDO表領域

適切な権限を持ち、現行コンテナがルートとなっている共通ユーザーのみが、UNDO表領域を作成できます。どの時点でも、CDBは次のいずれかのUNDOモードにあります。

- ローカルUNDOモード

この場合は、各PDBに固有のUNDO表領域があります。CDBがローカルUNDOモードを使用している場合は、データベースによってすべてのPDBにUNDO表領域が自動的に作成されます。ローカルUNDOはPDBのホット・クローンを実行する機能などの利点を備えており、PDBの再配置を高速化します。また、ローカルUNDOでは分離レベルが提供され、迅速な切断およびポイント・イン・タイム・リカバリ操作が可能です。

ローカルUNDO表領域は、Oracle Real Application Clusters (RAC)クラスタ内のPDBが開いているノードごとに必要です。たとえば、PDBを2ノードのクラスタから4ノードのクラスタに移動した場合や、すべてのノードでPDBが開いている場合は、必要な追加のUNDO表領域がデータベースによって自動的に作成されます。PDBを元に戻した場合は、冗長なUNDO表領域を削除できます。



ノート:

デフォルトでは、Database Configuration Assistant (DBCA)によってローカル UNDO が有効になった状態で新しい CDB が作成されます。

- 共有UNDOモード

単一インスタンスCDBでは、アクティブなUNDO表領域が1つのみ存在します。Oracle RAC CDBの場合、それぞれのインスタンスについてアクティブなUNDO表領域が1つ存在します。すべてのUNDO表領域は、すべてのコンテナのデータ・ディクショナリおよび関連ビューで表示されます。

UNDOモードはCDB全体に適用されます。つまり、すべてのコンテナが共有UNDOを使用するか、すべてのコンテナがローカルUNDOを使用するかのどちらかです。CDBでUNDOモードを切り替えることは可能で、これにはデータベースの再起動が伴います。

- コンテナごとのSYSTEM表領域およびSYSAUX表領域

CDBと非CDBの第一の物理的違いは、SYSTEMとSYSAUXのデータファイルです。非CDBには、SYSTEM表領域とSYSAUX表領域がそれぞれ1つのみ含まれます。これに対して、CDB内のCDBルート、各[アプリケーション・ルート](#)および

各PDBには、それぞれ固有のSYSTEMおよびSYSAUX表領域があります。各コンテナには、コンテナに常駐するオブジェクトを記述した独自のディクショナリ表のセットも含まれます。

- 0以上のユーザー作成表領域

一般的なユースケースでは、各PDBに独自の非システム表領域のセットがあります。これらの表領域には、PDB内のユーザー定義のスキーマおよびオブジェクトのデータが含まれています。

PDB内では、永続表領域と一時表領域を非CDBで管理する場合と同じ方法で管理します。CREATE PLUGGABLE DATABASEまたはALTER PLUGGABLE DATABASE文でSTORAGE句を使用して、PDBのデータファイルが使用する記憶域の量を制限することもできます。

PDB内のデータ・ディクショナリの記憶域は、移植可能です。PDBはCDBから切断し、異なるCDBに接続できます。

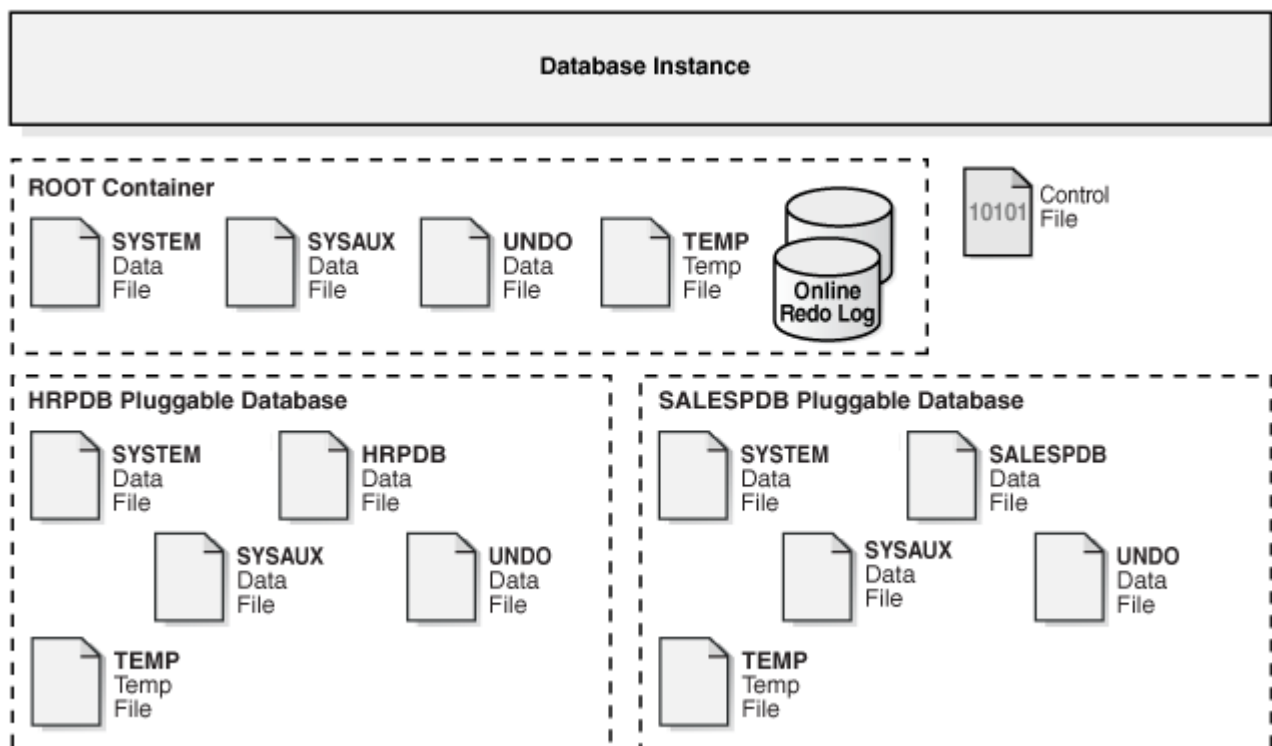
- コンテナごとの一連の一時ファイル

デフォルトの一時表領域は、CDBルートに1つ存在し、[アプリケーション・ルート](#)、[アプリケーションPDB](#)およびPDBごとに1つ存在します。

例2-19 ローカルUNDOモードのCDB

この例は、2つのPDB(hrpdbとsalespdb)を持つCDBの物理記憶域アーキテクチャの様子を示します。この例では、データベースはローカルUNDOモードを使用するため、CDBルート、hrpdbおよびsalespdb内にUNDOデータ・ファイルがあります。

図2-11 ローカルUNDOモードのCDBの物理アーキテクチャ



関連項目:

- [「CDBのデータ・ディクショナリ・アーキテクチャ」](#)
- CDBの作成後の状態について学習するには、[「CDBの作成後」](#)を参照してください

親トピック: [マルチテナント・アーキテクチャの概要](#)

CDBの可用性の概要

非CDBについて存在する多くの可用性機能は、CDB内の個々のPDBについても存在します。

- [CDBのバックアップおよびリカバリの概要](#)
RMANおよびOracle Enterprise Manager Cloud Controlによって、マルチテナント環境でのバックアップとリカバリが完全にサポートされます。
- [CDBでのフラッシュバックPDBの概要](#)
SQLまたはRecovery ManagerでFLASHBACK PLUGGABLE DATABASEコマンドを使用してPDBを巻き戻すことができます。このコマンドは、非CDBのFLASHBACK DATABASEに似ています。

親トピック: [マルチテナント・アーキテクチャの概要](#)

CDBのバックアップおよびリカバリの概要

RMANおよびOracle Enterprise Manager Cloud Controlによって、マルチテナント環境でのバックアップとリカバリが完全にサポートされます。

CDB全体、rootのみまたは1つ以上のPDBをバックアップおよびリカバリできます。また、PDB内の個々の表領域とデータファイルをバックアップおよびリカバリすることができます。

リカバリの観点では、ルートとすべてのPDBを別個にバックアップすることは、CDB全体をバックアップするのと同様です。主な違いは、入力する必要があるRMANコマンドの数とリカバリの時間です。CDB全体のリカバリにかかる時間は、CDBルートとすべてのPDBをリカバリする場合より短くなります。

他の開いているPDBの操作に影響を与えずに、1つ以上のPDBの完全リカバリを実行できます。RMANは、PDBレベルのPoint-in-Timeリカバリもサポートします。手順は、非CDBのPoint-in-Timeリカバリの手順と類似しています。

関連項目:

CDBの作成後の状態について学習するには、[Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド](#)を参照してください

親トピック: [CDBの可用性の概要](#)

CDBでのフラッシュバックPDBの概要

SQLまたはRecovery ManagerでFLASHBACK PLUGGABLE DATABASEコマンドを使用してPDBを巻き戻すことができます。このコマンドは、非CDBのFLASHBACK DATABASEに似ています。

フラッシュバックPDBはデータ破損、広範囲のユーザー・エラー、およびREDO破損から個々のPDBを保護します。操作ではCDB内の他のPDBのデータは巻き戻されません。

Oracle Database 12cリリース2 (12.2)よりも前のリリースでは、ルートへの接続時のみ、リストア・ポイント(SCNの別名)を作成できました。現在は、CREATE RESTORE POINT ... FOR PLUGGABLE DATABASEを使用して[PDBリストア・ポイント](#)(指定されたPDB内のみで使用可能)を作成できます。CDBリストア・ポイントと同様に、PDBリストア・ポイントも通常または保証付きのいずれかです。保証付きリストア・ポイントは、制御ファイルからエージ・アウトされず、明示的に削除する必要があります。ルートに接続しており、FOR PLUGGABLE DATABASE句を指定しない場合は、すべてのPDBで使用可能な[CDBリストア・ポイント](#)を作成します。

特殊なPDBリストア・ポイントのタイプは[クリーン・リストア・ポイント](#)で、これはPDBが閉じている場合のみ作成できます。共有

UNDOがあるPDBでは、PDBをクリーン・リストア・ポイントまで巻き戻すことによって、データベースの整合性が維持され、パフォーマンスが向上します。パフォーマンスを低下させる可能性がある自動インフラストラクチャがデータベースによって使用されなくなります。

関連項目:

FLASHBACK PLUGGABLE DATABASEの使用について学習するには、[Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド](#)を参照

親トピック: [CDBの可用性の概要](#)

CDBでのOracle Resource Managerの概要

Oracle Resource Manager (Resource Manager)を使用して、CDBリソース・プランを作成し、PDBにリソースを割り当てるための初期化パラメータを設定できます。

非CDBでは、リソース・マネージャを使用して、システム・リソースおよびデータベース・リソースについて競合する複数のワークロードを管理できます。したがって、CDBでは複数のPDB内の複数のワークロードがシステム・リソースおよびCDBリソースについて競合する場合があります。

CDBでは、リソース・マネージャにより、CDBとPDBの2つのレベルでリソースを管理できます。

CDBリソース・プラン

CDBリソース・プランにより、そのリソース・プラン・ディレクティブ(ディレクティブ)のセットに従って、リソースがそのPDBに割り当てられます。CDBリソース・プランとそのディレクティブは親子関係にあります。各リソース・プラン・ディレクティブは、PDBのセットまたは個々のPDBのいずれかを参照します。

パフォーマンス・プロファイルは、PDBのセットに対するシステム・リソースの共有を指定します。PDBパフォーマンス・プロファイルにより、個々のPDBではなくプロファイルのリソース・マネージャ・ディレクティブを指定することで、大量のPDBのリソースを管理できます。

ディレクティブにより、CPUとパラレル実行サーバーの割当てが制御されます。ディレクティブにより、PDBまたはPDBパフォーマンス・プロファイルごとに指定する共有値に基づいてPDBへのリソースの割当てを制御できます。共有値が高いほど、保証されるリソースは増加します。PDBおよびPDBパフォーマンス・プロファイルでは、CPUおよびパラレル・サーバーの使用率制限を設定することもできます。

CDBリソース・プランを作成するには、DBMS_RESOURCE_MANAGER PL/SQLパッケージのCREATE_CDB_PLANプロシージャを使用し、RESOURCE_MANAGER_PLANパラメータを使用してCDBリソース・プランを設定します。CDBリソース・プランのディレクティブを作成するには、CREATE_CDB_PLAN_DIRECTIVEプロシージャを使用します。

PDBリソース・プラン

CDBリソース・プランにより、システム・リソースの一部がPDBに割り当てられます。PDBリソース・プランにより、PDB内でのこの部分の割当て方法が決定します。

PDBリソース・プランは、非CDBのリソース・プランを作成する場合と同じ方法で作成します。プランを作成するには、DBMS_RESOURCE_MANAGERパッケージ内のプロシージャを使用します。

PDBリソース・プランを作成するには、DBMS_RESOURCE_MANAGER PL/SQLパッケージのCREATE_PLANプロシージャを使用し、RESOURCE_MANAGER_PLANパラメータを使用してPDBリソース・プランを設定します。PDBリソース・プランのディレクティブを作成するには、CREATE_PLAN_DIRECTIVEプロシージャを使用します。

PDBレベルのメモリー制御

CDBでは、PDBがSGAまたはPGAメモリーについて競合する可能性があります。複数の初期化パラメータでPDBのメモリー使用(メモリーの保証またはメモリーの制限)を制御できます。PDBを現在のコンテナとして次の初期化パラメータを設定する場合、パラメータは現在のPDBのメモリー使用を制御します。

重要なパラメータの例として、次のものがあります。

- SGA_MIN_SIZEは、PDBの保証される最小のSGAサイズを設定します。
- SGA_TARGETは、PDBでいつでも使用できる最大のSGAを指定します。
- PGA_AGGREGATE_LIMITは、PDBでいつでも使用できる最大のPGAを設定します。

関連項目:

初期化パラメータの設定の詳細は、[PDBのメモリー関連の初期化パラメータ](#)を参照してください

PDBレベルのI/O制御

集中的なディスクI/Oはパフォーマンス低下の原因となることがあります。不適切に設計されたSQL、大量トランザクションでの索引や表のスキャンなど、複数の要因によりディスクI/Oが過剰になることがあります。1つのPDBが過剰なディスクI/Oを生成している場合は、同じCDB内の他のPDBのパフォーマンスが低下することがあります。

非エンジニアド・システムでは、特定のPDBによって生成されるI/Oを制御するために、次の初期化パラメータのどちらかまたは両方を使用します。

- MAX_IOPSは、毎秒のI/O操作の数を制限します。
- MAX_MBPSは、毎秒のI/O操作のMB数を制限します。

エンジニアド・システムでは、I/Oリソース管理を使用してPDBのI/Oを管理します。

関連項目:

- データベース・リソース・マネージャの詳細は、[『Oracle Database概要』](#)を参照してください
- [「PDBに対するOracle Resource Managerの使用」](#)
- DB_CACHE_SIZEおよびその他の初期化パラメータについてさらに学習するには、[Oracle Databaseリファレンス](#)を参照してください
- DBMS_RESOURCE_MANAGERパッケージについてさらに学習するには、[Oracle Database PL/SQLパッケージおよびタイプ・リファレンス](#)を参照してください
- I/Oリソース管理についてさらに学習するには、[Oracle Exadata Storage Server Softwareユーザーズ・ガイド](#)を参照してください

親トピック: [マルチテナント・アーキテクチャの概要](#)

第II部 マルチテナント環境の作成および構成

マルチテナント環境を作成および構成できます。

- [マルチテナント環境の構成と管理の概要](#)

マルチテナント環境の構成と管理に関連する基本的な概念を理解します。

- [CDBの作成および構成](#)

マルチテナント・コンテナ・データベース(CDB)の作成と構成には、CDBの作成、計画、オプションでEM Expressの構成などのタスクが含まれます。

3 マルチテナント環境の構成と管理の概要

マルチテナント環境の構成と管理に関する基本的な概念を理解します。

- [マルチテナント環境の構成と管理について](#)
Oracle Multitenantオプションを使用してマルチテナント環境を構成および管理できます。
- [マルチテナント環境の前提条件](#)
マルチテナント環境の前提条件を満たす必要があります。
- [マルチテナント環境のタスクおよびツール](#)
マルチテナント環境で実行する一般的なタスクおよびタスクを完了するために使用するツールがあります。

親トピック: [マルチテナント環境の作成および構成](#)

マルチテナント環境の構成と管理について

Oracle Multitenantオプションを使用してマルチテナント環境を構成および管理できます。

マルチテナント環境を構成および管理する前に、特定の前提条件を満たす必要があります。そのためには、いくつかの共通タスクを実行し、一連のツールを使用してこれらのタスクを完了します。

マルチテナント・アーキテクチャを使用すると、Oracle Databaseを、ユーザーが作成した1つ以上のプラグابل・データベース (PDB)を含むマルチテナント・コンテナ・データベース(CDB)として機能させることができます。PDBは、Oracle Netクライアントに非CDBとして表示されるスキーマ、スキーマ・オブジェクトおよび非スキーマ・オブジェクトのポータブル・コレクションです。Oracle Database 12cより前のOracle Databaseはすべて非CDBでした。

- [共通ユーザーとローカル・ユーザー](#)
共通ユーザーは、ルート、および既存と将来のすべてのPDBにおいて同じIDを持つユーザーです。
- [CDB管理とPDB管理の業務の分離](#)
一部のデータベース管理者はCDB全体を管理し、他のデータベース管理者は個々のPDBを管理します。

関連項目:

- [「マルチテナント・アーキテクチャの紹介」](#)
- [「マルチテナント・アーキテクチャの概要」](#)

親トピック: [マルチテナント環境の構成と管理の概要](#)

共通ユーザーとローカル・ユーザー

共通ユーザーは、ルート、および既存と将来のすべてのPDBにおいて同じIDを持つユーザーです。

共通ユーザーは、ルートおよびCREATE SESSION権限が付与されているすべてのコンテナにログインできます。共通ユーザーが実行できる操作は、その共通ユーザーに付与されている権限によって異なります。PDBの作成やPDBの切断などの一部の管理タスクは、共通ユーザーが実行する必要があります。

CDBでは、ローカル・ユーザーもサポートされています。ローカル・ユーザーは、1つのPDBにのみ存在するユーザーです。

関連項目:

- [「CDBの共通性の概要」](#)
- 共通ユーザーおよびローカル・ユーザーの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください

親トピック: [マルチテナント環境の構成と管理について](#)

CDB管理とPDB管理の業務の分離

一部のデータベース管理者はCDB全体を管理し、他のデータベース管理者は個々のPDBを管理します。

CDB全体を管理するデータベース管理者は共通ユーザーとしてCDBに接続し、CDB全体とルートの属性、およびPDBの一部の属性を管理します。たとえば、これらのデータベース管理者は、PDBの作成、切断、接続および削除を実行できます。また、一時表領域やルートのデフォルト表領域を指定したり、PDBのオープン・モードを変更することもできます。

データベース管理者は、ローカルPDB管理者として特定のPDBに接続し、データベース管理者が非CDBで実行する管理タスクのサブセットをそのPDBで実行できます。サブセットには、PDBが[アプリケーション](#)をサポートするために必要なタスクが含まれています。たとえば、PDB内の表領域およびスキーマの管理、そのPDBの記憶域/パラメータの指定、現在のPDBのオープン・モードの変更、PDBレベルの初期化パラメータの設定などのタスクがあります。

親トピック: [マルチテナント環境の構成と管理について](#)

マルチテナント環境の前提条件

マルチテナント環境の前提条件を満たす必要があります。

マルチテナント環境を作成して使用する前に、次の最小前提条件を満たしている必要があります。

- Oracle Database 12c以降のリリースをインストールするか、そのリリースにアップグレードする必要があります。
Oracle MultitenantはOracle Database 11g以前のリリースではサポートされていません。
インストールには、オペレーティング・システム固有の各環境変数の設定、およびソフトウェアとデータベース・ファイルのディレクトリ構造の設定が含まれます。
- データベースの互換性レベルは、12.0.0以上に設定する必要があります。
- Oracle Databaseインスタンスを起動するために十分なメモリーが使用可能であること。
CDBに必要なメモリーは、その各コンテナのワークロードおよびコンテナの数にあわせてサイズ指定します。
- Oracle Databaseを実行するコンピュータ上で、設計したPDBのための十分なディスク記憶域が使用可能であること。
Oracle RAC環境では、十分な共有記憶域が使用可能である必要があります。
CDBに必要なディスク記憶域は、そのCDBに含められるすべてのPDBの領域要件の合計です。

前提条件は、使用しているオペレーティング・システム固有の[『Oracle Databaseインストール・ガイド』](#)または[『Oracle Grid Infrastructureインストールおよびアップグレード・ガイド』](#)に説明されています。Oracle Universal Installerを使用すると、表示される手順に従ってインストールでき、環境変数、ディレクトリ構造および認可の設定に関するヘルプが表示されます。

関連項目:

- 使用しているオペレーティング・システム固有の[『Oracle Databaseインストール・ガイド』](#)を参照してください
- データベースの互換性レベルの詳細は、[『Oracle Databaseアップグレード・ガイド』](#)を参照してください

親トピック: [マルチテナント環境の構成と管理の概要](#)

マルチテナント環境のタスクおよびツール

マルチテナント環境で実行する一般的なタスクおよびタスクを完了するために使用するツールがあります。

- [マルチテナント環境のタスク](#)
マルチテナント環境を使用すると、複数の目標を達成できます。一般的なタスクを実行することによって、マルチテナント環境を構成および使用できます。
- [マルチテナント環境のツール](#)
様々なツールを使用して、マルチテナント環境を構成および管理できます。

親トピック: [マルチテナント環境の構成と管理の概要](#)

マルチテナント環境のタスク

マルチテナント環境を使用すると、複数の目標を達成できます。一般的なタスクを実行することによって、マルチテナント環境を構成および使用できます。

これらの目標は、[「マルチテナント・アーキテクチャの利点」](#)で説明されています。そのためには、次の一般的なタスクを実行する必要があります。

タスク 1 マルチテナント環境のプラン

データベースを作成および構成するには、綿密な計画が必要です。CDB には特別な注意が必要です。たとえば、CDB を計画するときは、次のことを考慮する必要があります。

- 各 CDB に接続する PDB の数
- 計画している CDB のサポートに必要なリソース
- CDB 全体に対してまとめて実行されるか、個別の PDB でローカルに実行されるコンテナ管理ポリシー
- アプリケーション・コンテナとアプリケーション PDB、CDB と PDB、またはその両方の組合せで構成されるコンテナ・データベース・トポロジ

CDB の計画の詳細は、[「CDB 作成計画」](#)を参照してください。

タスク 2 1 つ以上の CDB の作成

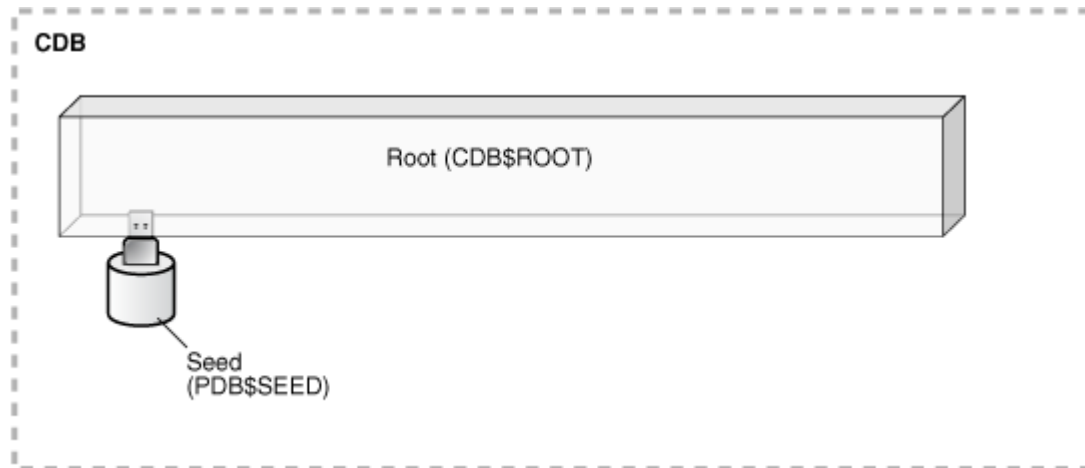
必要な計画を完了すると、Database Configuration Assistant (DBCA)または CREATE DATABASE 文を使用して、1 つ以上の CDB を作成できます。いずれの場合も、各 CDB の構成の詳細を指定する必要があります。

CDB の作成の詳細は、[「DBCA を使用した CDB 作成について」](#)および[「CDB の作成」](#)を参照して

ください。

CDB は、作成されると、[図 3-1](#) に示すようにルートおよび PDB\$SEED で構成されます。CDB ルートには Oracle によって保守されるオブジェクトおよびデータ構造のみが含まれ、PDB\$SEED はクローニングのための汎用的なシード・データベースです。

図3-1 新しく作成されたCDB

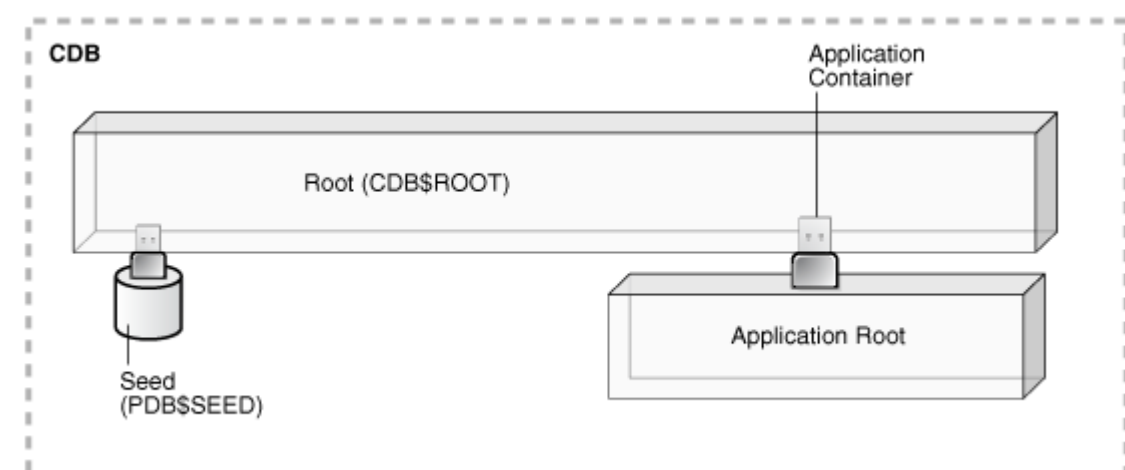


タスク 3 オプションでのアプリケーション・コンテナの作成

アプリケーション・コンテナは、アプリケーション・ルートとそれに関連付けられたアプリケーション PDB から構成される CDB のオプション・コンポーネントです。アプリケーション・コンテナには、1 つ以上のアプリケーションのデータが格納されています。

次の図は、1 つの空のアプリケーション・コンテナがある CDB を示しています。

図3-2 アプリケーション・コンテナ



[「アプリケーション・コンテナ内のアプリケーションの概要」](#)を参照してください。

タスク 4 PDB の作成、接続および切断

PDB にはユーザー・データが含まれています。CDB の作成後、PDB を作成し、切断された PDB をその CDB に接続し、必要な場合はいつでも PDB をその CDB から切断できます。CDB から PDB

を切断して、この PDB を異なる CDB に接続できます。PDB のワークロードをあるサーバーから別のサーバーに移動する必要がある場合など、PDB をある CDB から別の CDB に移動することがあります。

PDB の作成、PDB の接続および PDB の切断の詳細は、「[PDB およびアプリケーション・コンテナの作成および削除](#)」を参照してください。

図 3-3 に、複数の PDB が含まれる CDB を示します。

図3-3 PDBが含まれるCDB

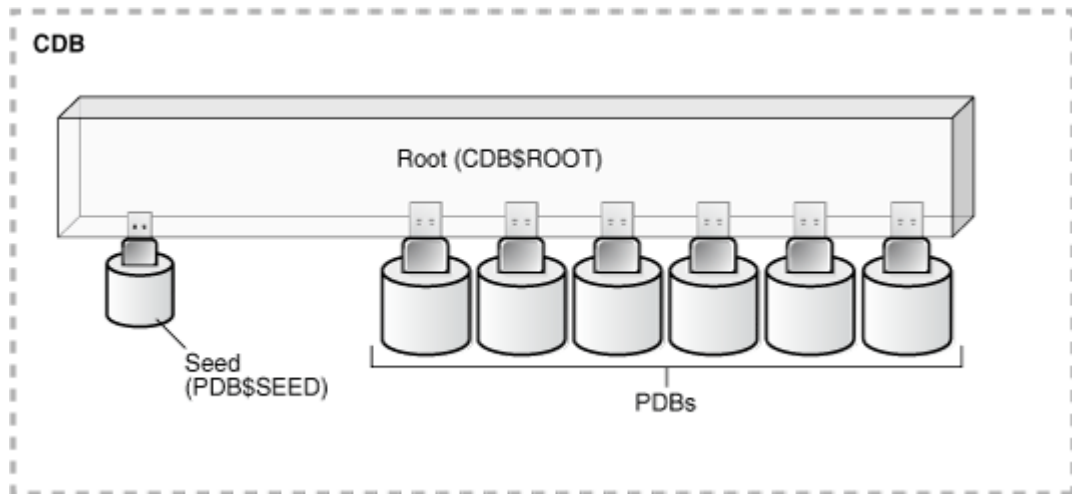
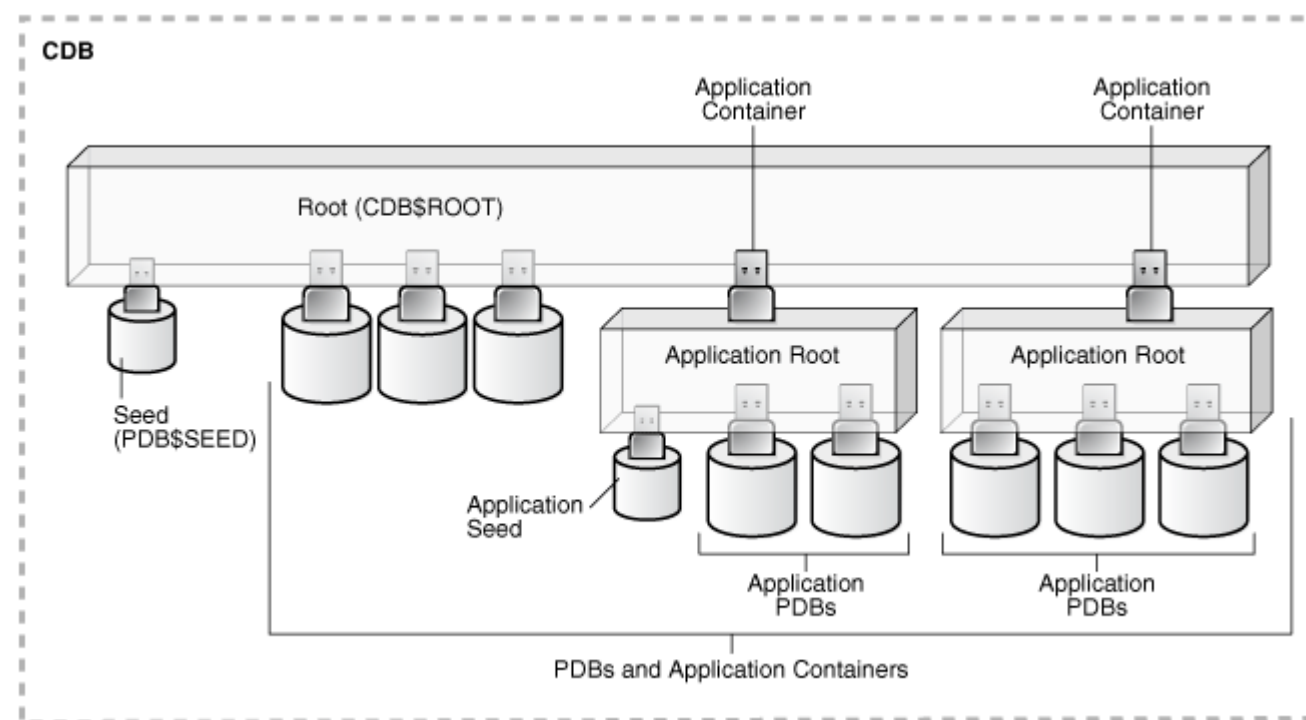


図 3-4 は、PDB、アプリケーション・コンテナおよびアプリケーション PDB が含まれる CDB を示しています。

図3-4 PDB、アプリケーション・コンテナおよびアプリケーションPDBが含まれるCDB



タスク 5 CDB およびアプリケーション・コンテナの管理および監視

CDB の管理および監視には、CDB 全体、ルート、および PDB の一部の属性の管理が含まれます。CDB と非 CDB で同じ管理タスクもあれば、異なる管理タスクもあります。

アプリケーション・コンテナの管理および監視は、CDB の管理および監視と似ていますが、アクションはアプリケーション・ルートとアプリケーション・コンテナに属するアプリケーション PDB にのみ影響します。

類似のタスクおよび異なるタスクの詳細は、[「CDB の作成後」](#)を参照してください。「[CDB の管理](#)」および「[CDB および PDB の監視](#)」も参照してください。

Oracle Resource Manager を使用して、CDB 内でホストされている PDB 間でリソースを割り当てて管理したり、PDB 内のユーザー・プロセス間でリソース使用量を割り当てて管理できます。[「PDB に対する Oracle Resource Manager の使用」](#)を参照してください。

Oracle Scheduler を使用して、CDB 内のジョブ、および個々の PDB 内のジョブをスケジュールすることもできます。[「CDB での Oracle Scheduler の使用」](#)を参照してください。

タスク 6 PDB およびアプリケーション PDB の管理および監視

PDB またはアプリケーション PDB の管理および監視は、非 CDB の管理および監視に似ていますが、異なる点もあります。「[PDB の管理](#)」および「[CDB および PDB の監視](#)」を参照してください。

親トピック: [マルチテナント環境のタスクおよびツール](#)

マルチテナント環境のツール

様々なツールを使用して、マルチテナント環境を構成および管理できます。

表3-1 マルチテナント環境のツール

ツール	説明	関連項目
SQL*Plus	SQL*Plus とは、CDB および PDB を作成、管理および監視できるコマンドライン・ツールです。SQL 文およびオラクル社が提供する PL/SQL パッケージを使用して、SQL*Plus でこれらのタスクを実行します。	SQL*Plus ユーザーズ・ガイドおよびリファレンス
Oracle Database Configuration Assistant(DBCA)	DBCA は、CDB の作成および複製が可能なグラフィカル・ユーザー・インタフェースを備えたユーティリティです。また、PDB を作成、再配置、クローニング、接続および切断することもできます。	Oracle Database 2 日でデータベース管理者 、 Oracle Database インストール・ガイド および DBCA オンライン・ヘルプ

ツール	説明	関連項目
Oracle Enterprise Manager Cloud Control	Cloud Control とは、CDB およびその PDB の管理および監視に使用できるグラフィカル・ユーザー・インターフェイスが含まれたシステム管理ツールです。	Cloud Control オンライン・ヘルプ
Oracle SQL Developer	<p>Oracle SQL Developer は、CDB の構成、PDB の作成、PDB の接続と切断、PDB の状態の変更、Oracle Cloud への PDB のクローニング、PDB のホット・クローニング/リフレッシュ、アプリケーション・ルート間での PDB の再配置などを行うことのできるグラフィカル・ユーザー・インターフェイスのあるクライアント・アプリケーションです。</p> <p>また、Oracle SQL Developer には、リソース管理、記憶域、セキュリティ、構成および CDB 内のコンテナとプラグブル・データベースに関するパフォーマンス・メトリックのレポートのためのグラフィカル・インターフェイスがあります。</p>	Oracle SQL Developer ユーザーズ・ガイド
サーバー制御(SRVCTL)ユーティリティ	SRVCTL ユーティリティを使用して、PDB のサービスを作成および管理できます。	「PDB のサービスの管理」
EM Express	EM Express は、グラフィカル・ユーザー・インターフェイスを持つ管理および監視ツールであり、Oracle Database に同梱されています。これは、CDB、ホストされる個別の PDB、またはその両方のために構成できます。このツールは PDB を管理するために使用することが意図されており、PDB のコンテキストで、アプリケーション DBA としてアプリケーション開発を管理および監視するために使用します。	Oracle Database 2 日でデータベース管理者
Oracle Multitenant Self-Service Provisioning アプリケーション	このアプリケーションを使用すると、PDB のセルフサービス・プロビジョニングが可能になります。CDB 管理者は、このセルフサービス・アプリケーションへのアクセスを制御し、PDB に対する割当てを管理しま	http://www.oracle.com/goto/multitenant アプリケーションにアクセスするには、「ダウンロード」タブをクリックして、Oracle

ツール	説明	関連項目
	す。	Multitenent のダウンロード・セクションで Oracle Pluggable Database Self-Service Provisioning アプリケーションを選択します。

親トピック: [マルチテナント環境のタスクおよびツール](#)

4 CDBの作成および構成

マルチテナント・コンテナ・データベース(CDB)の作成と構成には、CDBの計画、作成、オプションでEM Expressの構成などのタスクが含まれます。

- [CDBの作成の概要](#)
マルチテナント・コンテナ・データベース(CDB)の作成手順は、非CDBの作成手順に似ています。
- [CDB作成計画](#)
CDB作成では、複数のオペレーティング・システム・ファイルがCDBとして連携して動作するように準備します。
- [CDBの作成](#)
CDBは、DBCAを使用するか、CREATE DATABASE SQL文を手動で発行して作成できます。
- [CDB用EM Expressの構成](#)
CDBの場合は、グローバルHTTPSポートを設定して、ルートおよび各PDB用にOracle Enterprise Manager Database Express (EM Express)を構成するか、CDB内のコンテナごとに異なるポートを設定できます。
- [CDBの作成後](#)
作成後のCDBは、ルートとPDBシードで構成されます。

親トピック: [マルチテナント環境の作成および構成](#)

CDBの作成について

マルチテナント・コンテナ・データベース(CDB)の作成手順は、非CDBの作成手順に似ています。

非CDBの作成手順は、『[Oracle Database管理者ガイド](#)』で説明されています。CDBを作成する前に、このマニュアルで説明されている概念およびタスクを理解する必要があります。

この章では、CDBを作成する場合の特別な考慮事項について説明します。また、この章では、『[Oracle Database管理者ガイド](#)』にある非CDBの作成手順とCDBの作成手順の違いについても説明します。

『[CDB作成計画](#)』に示すいくつかのガイドラインを使用してCDBを計画した後、Oracle Databaseソフトウェアのインストール中またはインストール後に、CDBを作成できます。インストール後にCDBを作成する一般的な理由は、次のとおりです。

- Oracle Universal Installer (OUI)を使用してソフトウェアのみをインストールし、CDBを作成しなかったため。
- 既存のCDBまたは既存の非CDBと同じホストに別のCDBを作成するため。この場合、この章では、新しいCDBで既存のデータベースと同じOracleホームが使用されると想定します。OUIを再度実行して、新しいOracleホームにCDBを作成することもできます。

CDBを作成する場合、具体的には次の方法があります。

- Database Configuration Assistant (DBCA)グラフィカル・ツールを使用
[「DBCAを使用したCDB作成について」](#)を参照してください。
- CREATE DATABASE SQL文を使用
[「CDBの作成」](#)を参照してください。

親トピック: [CDBの作成および構成](#)

CDB作成計画

CDB作成では、複数のオペレーティング・システム・ファイルがCDBとして連携して動作するように準備します。



ノート:

CDBを計画する前に、[「マルチテナント・アーキテクチャの紹介」](#)でCDBとPDBの概念に関する情報を確認します。

- [CDBの構成方法の決定](#)
CDBの作成を準備するには、調査と綿密な計画が必要です。
- [CDB作成の前提条件](#)
新規CDBを作成する前に、前提条件を満たす必要があります。

親トピック: [CDBの作成および構成](#)

CDBの構成方法の決定

CDBの作成を準備するには、調査と綿密な計画が必要です。

最初のステップとして、CDBで現在サポートされていないOracle Database機能のリストが記載されている『[Oracle Database!リリース・ノート](#)』を確認してください。これらの機能のうち1つ以上を使用する必要がある場合、非CDBを作成します。

次の各項では、CDBに適用される推奨アクションおよび考慮事項について説明します。

- [PDBの計画](#)
プラグブル・データベース(PDB)の表および索引を計画し、それらに必要な領域を見積ります。
- [物理レイアウトの計画](#)
CDBを構成する基礎となるオペレーティング・システム・ファイルのレイアウトを計画します。
- [初期化パラメータの管理方法の理解](#)
初期化パラメータ・ファイルに含まれる可能性がある初期化パラメータを学習します。
- [文字セットの選択](#)
CDBの文字セットを選択する必要があります。
- [CDBでサポートするタイム・ゾーンの決定](#)
CDBでサポートする必要があるタイム・ゾーンを検討します。
- [データベースおよびREDOログ・ブロック・サイズの選択](#)
CDBの標準データベース・ブロック・サイズを選択します。
- [SYSTEMおよびSYSAUX表領域の計画](#)
CDBルートおよび各PDBには、個別のSYSAUXおよびSYSTEM表領域が存在します。
- [一時表領域の計画](#)
デフォルトの一時表領域を使用するように計画します。
- [UNDOモードの選択](#)
UNDO表領域を使用してUNDOデータを管理するように計画します。
- [アプリケーションのサービスの計画](#)
アプリケーションのニーズを満たすために必要なデータベース・サービスを計画します。
- [CDBの起動と停止方法の理解](#)
データベース・インスタンスの起動と停止、CDBのマウントとオープン of の原理およびオプションを学習します。
- [Oracle RACの計画](#)
Oracle RACを使用する場合は、Oracle RAC環境を計画します。

親トピック: [CDB作成計画](#)

PDBの計画

プラグブル・データベース(PDB)の表および索引を計画し、それらに必要な領域を見積ります。

CDBでは、ほとんどのユーザー・データがPDBに存在します。ルートには、ユーザー・データは含まれないか、または最小限のユーザー・データが含まれます。CDBに含められるPDBを計画します。CDBのディスク記憶域要件は、Oracle Databaseインストールに必要な領域に、CDBに含められるPDBの領域要件の合計を加えたものです。

MAX_PDBS初期化パラメータは、CDBルートまたはアプリケーション・ルートで作成できるPDBの合計数の制限を指定します。MAX_PDBSのデフォルト値および最大値は、Oracle Databaseオフリングによって異なります。異なるエディションとサービスでサポートされる機能の詳細は、『[Oracle Databaseライセンス情報ユーザー・マニュアル](#)』を参照してください。

CDBにアプリケーション・コンテナを作成することもできます。アプリケーション・コンテナは、1つ以上のアプリケーションのデータを格納するアプリケーションPDBのコレクションです。また、アプリケーション・コンテナではユーザーが作成するアプリケーション共通オブジェクトがサポートされ、アプリケーション・コンテナ内のアプリケーションPDBで共有できます。

関連項目:

- [「PDBおよびアプリケーション・コンテナの作成および削除」](#)
- [「アプリケーション・コンテナ内のアプリケーションの概要」](#)
- [データベース構造と記憶域](#)および[スキーマ・オブジェクト](#)についてさらに学習するには、『*Oracle Database管理者ガイド*』を参照してください
- MAX_PDBSについてさらに学習するには、[Oracle Databaseリファレンス](#)を参照してください

親トピック: [CDBの構成方法の決定](#)

物理レイアウトの計画

CDBを構成する基礎となるオペレーティング・システム・ファイルのレイアウトを計画します。

CDBルート、PDB\$SEED、各PDB、各アプリケーション・ルートおよび各アプリケーションPDBについて個別のデータファイルが存在します。

単一インスタンスCDBにREDOログが1つ存在するか、またはOracle Real Application Clusters (Oracle RAC) CDBのインスタンスごとにREDOログが1つ存在します。また、Oracle RACでは、すべてのデータファイルおよびREDOログ・ファイルが共有記憶域に存在する必要があります。

関連項目:

- Oracle Managed Filesの使用の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください
- [Oracle Automatic Storage Management管理者ガイド](#)
- [Oracle Databaseパフォーマンス・チューニング・ガイド](#)
- 『[Oracle Databaseバックアップおよびリカバリ・アドバンスト・ユーザーズ・ガイド](#)』
- Oracle RACの記憶域の構成の詳細は、『[Oracle Grid Infrastructureインストレーションおよびアップグレード・ガイド](#)』を参照してください。
- 該当のOracle Databaseインストレーション・ガイドなど、使用しているオペレーティング・システム固有のOracleマニユ

アル。

親トピック: [CDBの構成方法の決定](#)

初期化パラメータの管理方法の学習

初期化パラメータ・ファイルに含まれる可能性がある初期化パラメータについて理解します。

CDBを作成する前に、サーバー・パラメータ・ファイル(SPFIL)E)の概念および操作について理解しておいてください。SPFILEファイルを使用すると、初期化パラメータをサーバー側のバイナリ・ファイルに永続的に格納して管理できます。

CDBでは、1つのSPFILEまたは1つのテキスト形式の初期化パラメータ・ファイル(PFILE)が使用されます。ルートに設定した初期化パラメータの値は、PDBで継承することが可能です。ALTER SYSTEM文を使用して、PDBの一部の初期化パラメータを設定できます。

SPFILEを操作する場合、CDBルートは現在のコンテナである必要があります。SPFILEを作成または変更するユーザーは、SYSDBA、SYSOPERまたはSYSBACKUP管理権限を持つ共通ユーザーである必要があります。AS SYSDBA、AS SYSOPERまたはAS SYSBACKUPとしてそれぞれ接続して権限を行使する必要があります。

次の初期化パラメータが重要です。

- CDBを作成するには、ENABLE_PLUGGABLE_DATABASE初期化パラメータがTRUEに設定されている必要があります。
- CDBルートグローバル・データベース名を作成するには、DB_NAMEとDB_DOMAINの両方の初期化パラメータを設定します。ルートグローバル・データベース名は、CDBのグローバル・データベース名です。PDBのグローバル・データベース名は、PDB名およびDB_DOMAIN初期化パラメータによって定義されます。

関連項目:

- [「現在のコンテナについて」](#)
- [「ALTER SYSTEMを使用したCDBの変更」](#)
- [「PDBの変更可能な初期化パラメータをリスト」](#)
- スキーマ・オブジェクトの詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- グローバル・データベース名の決定の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- [『Oracle Databaseリファレンス』](#)

親トピック: [CDBの構成方法の決定](#)

文字セットの選択

CDBの文字セットを選択する必要があります。

CDBのデータベースの文字セットを選択する場合、CDBに統合(接続)するデータベースの現在の文字セットを考慮する必要があります。CDBデータベースの文字セットにはAL32UTF8、CDBの各国語文字セットにはAL16UTF6をお勧めします。これらは最も柔軟性が高いためです。

CDBルート文字セットがAL32UTF8の場合、CDBに接続されているPDBはCDBルートと異なる文字セットを保持できます。PDB\$SEEDから作成されたPDBは、そのシードからAL32UTF8文字セットを継承しますが、PDBは別の文字セットに移行できます。ルート文字セットがAL32UTF8でない場合、CDBルート内のすべてのPDBはそのルートの文字セットを使用します。

ノート:



Oracle Multitenant では、あるコンテナの LOB に対する異なる文字セットのコンテナによるデータ・リンク、拡張データ・リンクまたは CONTAINERS () 句を使用したアクセスはサポートされません。たとえば、CDB ルートと salespdb の文字セットが異なる場合、CDB ルートで実行される CONTAINERS () 問合せは salespdb の LOB にアクセスできません。

非CDBをCDBに移行する場合、最初に非CDBをAL32UTF8に移行するのが最適です。Oracle Database Migration Assistant for Unicode (DMU)を使用すると、非CDBをAL32UTF8に移行できます。CDBの作成後に、DMUを使用してCDBの文字セットを移行することはできません。

関連項目:

[Oracle Databaseグローバル化・サポートガイド](#)

親トピック: [CDBの構成方法の決定](#)

サポートするタイム・ゾーンの決定

CDBでサポートする必要があるタイム・ゾーンを検討します。

CDB全体(すべてのPDBを含む)についてタイム・ゾーンを設定できます。また、PDBごとに個々にタイム・ゾーンを設定することもできます。

関連項目:

データベースのタイム・ゾーンとタイム・ゾーン・ファイルの指定の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBの構成方法の決定](#)

データベース・ブロック・サイズおよびREDOログ・ブロック・サイズの選択

CDBの標準データベース・ブロック・サイズを選択します。

このサイズはCDBの作成時にDB_BLOCK_SIZE初期化パラメータによって指定しますが、CDBの作成後は変更できません。標準ブロック・サイズはCDB全体に適用されます。

セクター・サイズが4KBのディスク上にオンラインREDOログ・ファイルを格納する場合は、オンラインREDOログのブロック・サイズを手動で指定する必要があるかどうかを確認します。

- データベース・ブロック・サイズの指定の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- REDOログ・ファイルのブロック・サイズの計画の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBの構成方法の決定](#)

SYSTEMおよびSYSAUX表領域の作成

CDBルートおよび各PDBには、個別のSYSAUXおよびSYSTEM表領域が存在します。

SYSAUX表領域の適切な初期サイズを決定する必要があります。また、SYSTEM以外のユーザーに対してはデフォルトの表領域を使用し、データベース・オブジェクトがSYSTEM表領域に誤って保存されないようにします。CDBルートおよび各PDBに、個別のデフォルト表領域を指定できます。

関連項目:

- SYSAUX表領域の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- デフォルト永続表領域の作成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- [「CDBルートへの接続時のコンテナの変更について」](#)

親トピック: [CDBの構成方法の決定](#)

一時表領域の計画

デフォルト一時表領域の使用を計画します。

デフォルトの一時表領域は、CDB内のすべてのコンテナに存在します。したがって、CDBルート、すべてのPDB、アプリケーションルートおよびアプリケーションPDBには、独自のデフォルト一時表領域があります。

Oracle Databaseは、再帰的SQL専用の共有一時表領域を使用します。ホストされるPDBテナントは、この表領域を直接使用しません。

関連項目:

- [「CDBルートへの接続時のコンテナの変更について」](#)
- デフォルト一時表領域の作成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBの構成方法の決定](#)

UNDOモードの選択

UNDOデータを管理するために、UNDO表領域を使用します。

CDBは異なるUNDOモードで実行できます。CDB全体で1つのアクティブなUNDO領域を持つか、CDB内のコンテナごとに別々のUNDO表領域を持つように、CDBを構成できます。CDBの作成時にUNDOモードを指定でき、CDBの作成後にUNDOモードを変更できます。

CDB全体で1つのアクティブなUNDO表領域を持つことを選択した場合は、共有UNDOが使用され、ローカルUNDOが無効になります。この構成では、単一インスタンスCDBには、アクティブなUNDO表領域が1つ存在します。ローカルUNDOが有効にされている場合は、単一インスタンス構成の各コンテナに1つのUNDO表領域があります。Oracle RACのCDBの場合、PDBがオープンされる各ノードで、各PDBに1つのUNDO表領域があります。共有UNDOを使用する場合は、適切な権限を持ち、現在のコンテナがルートである共通ユーザーのみが、UNDO表領域を作成できます。

ベスト・プラクティスはCDBでローカルUNDOを使用することです。共有UNDOは、主にアップグレードおよび移行のためにのみサポートされます。共有UNDOと比べると、ローカルUNDOにはわずかなオーバーヘッドがありますが、ローカルUNDOが持つ利点

によって、ほとんどの環境でより適したUNDOとなります。ローカルUNDOによって切断操作およびポイント・イン・タイム・リカバリが速くなり、一部の機能(PDBの再配置など)が必要となります。デフォルトでは、DBCAはローカルUNDOを有効にして新しいCDBを作成します。

CDBでは、UNDO_MANAGEMENT初期化パラメータをAUTOに設定する必要があり、UNDOデータの管理にUNDO表領域が必要です。

ローカルUNDOが有効になっていない場合、現在のコンテナがルートであると、UNDO表領域は静的データ・ディクショナリ・ビューおよび動的パフォーマンス(V\$)ビューに表示されます。現在のコンテナがPDBである場合、UNDO表領域は動的パフォーマンス・ビューにのみ表示されます。

また、ローカルUNDOが無効になっている場合、Oracle Databaseでは、現在のコンテナがPDBであると、UNDO表領域とロールバック・セグメントの操作が警告なしで無視されます。

関連項目:

- [「ALTER DATABASEを使用したCDBのUNDOモードの設定」](#)
- [「現在のコンテナについて」](#)
- UNDO管理の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBの構成方法の決定](#)

アプリケーションのサービスの計画

アプリケーションのニーズを満たすために必要なデータベース・サービスを計画します。

ルートおよび各PDBには、複数のサービスが必要な場合があります。ルートまたは個々のPDBに、サービスを作成できます。

データベース・サービスには、オプションのPDBプロパティがあります。サービスを作成し、作成したサービスをPDBプロパティを指定して特定のPDBに関連付けることができます。nullのPDBプロパティが指定されたサービスは、CDBルートに関連付けられます。

提供されるPL/SQLパッケージDBMS_SERVICEを使用してサービスを作成し、それらをPDBに関連付けることもできます。CREATE_SERVICEプロシージャを実行すると、サービスが現在のコンテナに関連付けられます。

SRVCTLユーティリティ、Oracle Enterprise Manager Cloud Controlおよび提供されているDBMS_SERVICE PL/SQLパッケージを使用して、サービスを管理できます。

PDBを作成すると、PDBの新しいデフォルト・サービスが自動的に作成されます。サービスには、PDBと同じ名前が付いています。このサービスは、SRVCTLユーティリティでは管理できません。ただし、ユーザー定義サービスを作成し、アプリケーションに合わせてカスタマイズできます。

関連項目:

- [「PDBのサービスの管理」](#)
- [データベース・サービスおよび単一インスタンス・データベースでのSRVCTLユーティリティの使用](#)について学習するには、[『Oracle Database管理者ガイド』](#)を参照してください
- Oracle RACデータベースでのSRVCTLユーティリティの使用の詳細は、[『Oracle Real Application Clusters管理およびデプロイメント・ガイド』](#)を参照してください。

親トピック: [CDBの構成方法の決定](#)

CDBの起動と停止方法の学習

データベース・インスタンスの起動と停止、CDBのマウントとオープンの原理およびオプションについて理解します。

CDBでは、CDBルートおよびすべてのコンテナで単一のデータベース・インスタンスが共有されますが、Oracle RACを使用する場合は、複数の同時インスタンスが共有されます。CDB全体を起動して停止すると、ホストされるPDBの状態を判別できます。CDBがオープンされている場合は、CDBまたはPDBのコンテキストでALTER PLUGGABLE DATABASE文を使用して、ホストされるPDBをオープンまたはクローズすることによって、PDBのオープン・モードを制御できます。下位互換性を維持するために、ALTER DATABASE OPEN文の実行時にPDBが現在のコンテナである場合、この文はサポートされます。PDBが現在のコンテナである場合は、SQL*PlusのSTARTUPコマンドおよびSHUTDOWNコマンドを使用することもできます。ただし、SQL*PlusのSTARTUP MOUNTコマンドはCDBのみの操作であり、PDBが現在のコンテナである場合は使用できません。

関連項目:

- [「PDBのオープン・モードの変更」](#)
- [「ALTER PLUGGABLE DATABASE文を使用したPDBの変更」](#)
- データベースの起動の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBの構成方法の決定](#)

Oracle RACの計画

Oracle RACを使用する場合は、Oracle RAC環境を計画します。

Oracle RACドキュメントでは、Oracle RAC環境でのCDBの特別な考慮事項について説明しています。Oracle RAC環境でのCDB作成の詳細は、使用しているプラットフォーム固有のOracle RACインストレーション・ガイドを参照してください。

関連項目:

[Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)

親トピック: [CDBの構成方法の決定](#)

CDB作成の前提条件

新規CDBを作成する前に、前提条件を満たす必要があります。

新しいCDBを作成するには、次の前提条件を満たす必要があります。

- [「マルチテナント環境の前提条件」](#)で説明する前提条件を満たしていることを確認してください。
- Oracle Databaseインスタンスを起動するために十分なメモリーが使用可能であること。
CDBに必要なメモリーは、その各コンテナのワークロードおよびコンテナの数にあわせてサイズ指定します。
- Oracle Databaseを実行するコンピュータ上で、設計したPDBのための十分なディスク記憶域が使用可能であること。
Oracle RAC環境では、十分な共有記憶域が使用可能である必要があります。

CDBに必要なディスク記憶域は、そのCDBに含まれるすべてのPDBの領域要件の合計です。

前提条件は、使用しているオペレーティング・システム固有の『[Oracle Databaseインストール・ガイド](#)』または『[Oracle Grid Infrastructureインストールおよびアップグレード・ガイド](#)』に説明されています。Oracle Universal Installerを使用すると、表示される手順に従ってインストールでき、環境変数、ディレクトリ構造および認可の設定に関するヘルプが表示されます。

親トピック: [CDB作成計画](#)

CDBの作成

CDBは、DBCAを使用するか、CREATE DATABASE SQL文を手動で発行して作成できます。

ノート:



Database Configuration Assistant (DBCA)テンプレート・デプロイメントの使用はほとんど自動化されており、DBCAが完了するとCDBが使用可能状態になるので、CREATE DATABASE SQL文ではなくこの方法でCDBを作成することをお勧めします。

- [DBCAを使用したCDB作成について](#)

CDBは、Database Configuration Assistant (DBCA)を使用して作成することをお勧めします。

- [SQL文を使用したCDB作成について](#)

CREATE DATABASE SQL文を使用したCDBの作成は、非CDBの作成に似ています。

- [CREATE DATABASE文を使用したCDBの作成](#)

CREATE DATABASE文を使用してCDBを作成する場合は、実行可能なCDBを作成する前に他の処理を実行する必要があります。

- [CREATE DATABASE文を使用したCDBの作成: 例](#)

次の例では、newcdbという名前のCDBを作成します。

親トピック: [CDBの作成および構成](#)

DBCAを使用したCDB作成について

CDBは、Database Configuration Assistant (DBCA)を使用して作成することをお勧めします。

DBCAには、代替方法より優れた次の利点があります。

- 作成の大部分は自動化されています。
- DBCAを使用すると、CDBを作成するときに、CDB内のPDB数を指定できます。
- DBCAが完了すると、CDBは使用可能となります。
- CDBの作成後は、DBCAを使用して、次の処理を実行できます。
 - ローカルPDBのクローニング
 - PDBの接続と切断
 - CDBの複製(サイレント・モードのみ)

選択したインストール・タイプによっては、Oracle Universal Installer (OUI)でDBCAを起動できます。また、Oracle

Databaseをインストール後は、スタンドアロン・ツールとして、いつでもDBCAを起動できます。

DBCAを使用すると、次のいずれかのモードでCDBを作成できます。

- 対話型モード

このモードには、CDBを作成して構成するためのグラフィカル・インタフェースおよびガイド付きワークフローが用意されています。

- 非対話型モード(サイレント・モードとも呼ばれる)

このモードでは、クローニングに適したカスタマイズされたPDBシード・データベースを使用した、事前構成されたCDBテンプレート・デプロイメントをスクリプト記述できます。DBCAをサイレント・モードで実行するには、コマンドライン引数またはレスポンス・ファイル、あるいはその両方を指定します。

関連項目:

- DBCAを使用したデータベースの作成方法を学習するには、[『Oracle Database管理者ガイド』](#)を参照
- [Oracle Database 2日でデータベース管理者](#)
- DBCAオンライン・ヘルプ

親トピック: [CDBの作成](#)

SQL文を使用したCDB作成について

CREATE DATABASE SQL文を使用したCDBの作成は、非CDBの作成に似ています。

この項では、CDBを作成するための追加要件について説明します。CREATE DATABASEを使用してCDBを作成する場合は、次を実行する必要があります。

- PDBの有効化
- CDBルート・ファイルの名前と場所の指定
- PDB\$SEEDファイルの名前と場所の指定

ノート:



DBCAを使用しないで手でCDBを作成する場合は、CREATE DATABASE SQL文を使用します。

- [PDBの有効化について](#)
CDBを作成するには、CREATE DATABASE文にENABLE PLUGGABLE DATABASE句を含める必要があります。
- [CDBルートおよびPDB\\$SEEDのファイルの名前と場所について](#)
CDBを作成するには、Oracle Databaseで、CDBルートおよびPDB\$SEEDのファイルの名前と場所が認識されている必要があります。
- [PDB\\$SEEDのデータファイル属性について](#)
PDBシード(PDB\$SEED)は、新しいコンテナを作成するためのテンプレートとして使用できます。
- [CDBのUNDOモードについて](#)
共有UNDOがデフォルトです。undo_mode_clauseをENABLE PLUGGABLE DATABASE句に使用することで、CDB

のUNDOモードを指定できます。

関連項目:

CDBに含まれるファイルの詳細は、『[Oracle Database概要](#)』を参照してください

親トピック: [CDBの作成](#)

PDBの有効化について

CDBを作成するには、CREATE DATABASE文にENABLE PLUGGABLE DATABASE句が含まれている必要があります。

この句はCREATE DATABASE文に次のような影響を与えます。

- ENABLE PLUGGABLE DATABASEが含まれている場合
この文では、ルートおよびPDB\$SEEDを含むCDBを作成します。CDBを非CDBに変更することはできません。
- ENABLE PLUGGABLE DATABASEが含まれていない場合
新しく作成されたデータベースは非CDBで、これはCDBルートまたはPDB\$SEEDが含まれていないことを意味します。非CDBにPDBを含めることはできません。

親トピック: [SQL文を使用したCDB作成について](#)

CDBルートおよびPDB\$SEEDのファイルの名前と場所について

CDBを作成するには、Oracle Databaseで、CDBルートおよびPDB\$SEEDのファイルの名前と場所が認識されている必要があります。

CREATE DATABASE文が正常に完了した後は、PDB\$SEEDとそのファイルを使用して新しいPDBを作成できます。作成後のPDBシードを変更することはできません。

PDB\$SEEDのファイルの名前と場所は、次のいずれかの方法で指定する必要があります。

1. CREATE DATABASEのENABLE PLUGGABLE DATABASE SEED FILE_NAME_CONVERT句
2. Oracle Managed Files
3. PDB_FILE_NAME_CONVERT初期化パラメータ

複数の手法を使用している場合、CREATE DATABASE文では、リストの優先順位に従っていずれかの手法が使用されます。たとえば、すべての手法を使用している場合、CREATE DATABASE文では、ENABLE PLUGGABLE DATABASE SEED FILE_NAME_CONVERT句での指定のみが使用されます。これは、この句がリストの最初にあるためです。

- [SEED FILE_NAME_CONVERT句](#)
CREATE DATABASE文のSEED FILE_NAME_CONVERT句は、CDBルートのファイル名を使用してPDB\$SEEDファイル名の生成方法を指定します。
- [Oracle Managed Files](#)
Oracle Managed Filesを有効化すると、PDB\$SEEDのファイルの名前と場所を指定できます。
- [PDB_FILE_NAME_CONVERT初期化パラメータ](#)
PDB_FILE_NAME_CONVERT初期化パラメータを使用すると、シードのファイルの名前と場所を指定できます。

関連項目:

親トピック: [SQL文を使用したCDB作成について](#)

SEED FILE_NAME_CONVERT句

CREATE DATABASE文のSEED FILE_NAME_CONVERT句は、CDBルートOfFile名を使用してPDB\$SEEDのファイル名の生成方法を指定します。

この句を使用して、次のいずれかのオプションを指定できます。

- 1つ以上のファイル名パターンおよび置換ファイル名パターンを次の形式で指定します。

```
'string1' , 'string2' , 'string3' , 'string4' , ...
```

string1ファイル名パターンがstring2ファイル名パターンによって置き換えられ、string3ファイル名パターンがstring4ファイル名パターンによって置き換えられます。ファイル名パターン文字列と置換ファイル名パターン文字列のペアを必要な数だけ使用できます。

奇数の文字列を指定した(最終文字列に対応する置換文字列が存在しない)場合、エラーが返されます。1つのファイル名またはディレクトリと一致する複数のパターン/置換文字列を指定しないでください。

Oracle Managed Filesで管理されているファイルまたはディレクトリと一致するファイル名のパターンは指定できません。

- ファイル名を変換する必要がない場合は、NONEを指定します。SEED FILE_NAME_CONVERT句を省略することは、NONEを指定することと同じです。

例4-1 SEED FILE_NAME_CONVERT句

このSEED FILE_NAME_CONVERT句では、/oracle/dbsディレクトリ内のファイル名を使用して、/oracle/pdbseed/ディレクトリ内のPDB\$SEEDのファイルに対するファイル名を生成します。

```
SEED FILE_NAME_CONVERT = (' /oracle/dbs/' , ' /oracle/pdbseed/' )
```

関連項目:

SEED FILE_NAME_CONVERT句の構文は、[『Oracle Database SQL言語リファレンス』](#)を参照してください

親トピック: [CDBルートおよびPDB\\$SEEDのファイルの名前と場所について](#)

Oracle Managed Files

Oracle Managed Filesを有効化すると、PDB\$SEEDのファイルの名前と場所を指定できます。

関連項目:

[Oracle Database管理者ガイド](#)

親トピック: [CDBルートおよびPDB\\$SEEDのファイルの名前と場所について](#)

PDB_FILE_NAME_CONVERT初期化パラメータ

PDB_FILE_NAME_CONVERT初期化パラメータを使用すると、シードのファイルの名前と場所を指定できます。

この手法を使用するには、CDBを作成するとき、PDB_FILE_NAME_CONVERT初期化パラメータが初期化パラメータ・ファイルに含まれるようにします。

この初期化パラメータで指定するファイル名パターンをOracle Managed Filesによって管理されているファイルまたはディレクトリと一致させることはできません。

関連項目:

[『Oracle Databaseリファレンス』](#)

親トピック: [CDBルートおよびPDB\\$SEEDのファイルの名前と場所について](#)

PDB\$SEEDのデータファイル属性について

PDBシード(PDB\$SEED)は、新しいコンテナを作成するためのテンプレートとして使用できます。

CDBルートのSYSTEM表領域およびSYSAUX表領域のデータファイル属性は、PDBシードに適さない場合があります。この場合は、`tablespace_datafile`句を使用して、PDBシードのデータファイルに異なる属性を指定できます。これらの句を使用して、PDBシード内のSYSTEM表領域とSYSAUX表領域を構成しているすべてのデータファイルに対して属性を指定します。値が指定されていないすべての属性には、ルートから継承された値が使用されます。

`tablespace_datafile`句の構文は、データファイルの名前と場所およびREUSE属性を除き、データファイル指定の構文と同じです。[「CDBルートおよびPDB\\$SEEDのファイルの名前と場所について」](#)で説明されている、PDBシードのデータファイルの名前と場所を指定するためのどの方法でも、`tablespace_datafile`句を使用できます。

`tablespace_datafile`句では、PDBシードのデータファイルの名前と場所は指定しません。代わりに、PDBシード内のSYSTEMおよびSYSAUXデータファイルについて、ルートとは異なる属性を指定します。SIZEが表領域の`tablespace_datafile`句で指定されていない場合、表領域のデータファイル・サイズは、対応するルートのデータファイル・サイズの事前定義された割合に設定されます。

例4-2 tablespace_datafile句の使用

次のCREATE DATABASE句により、ルート内のSYSTEMおよびSYSAUX表領域を構成するデータファイルの名前、場所および属性が指定されると想定します。

```
DATAFILE '/u01/app/oracle/oradata/newcdb/system01.dbf'  
  SIZE 325M REUSE  
SYSAUX DATAFILE '/u01/app/oracle/oradata/newcdb/sysaux01.dbf'  
  SIZE 325M REUSE
```

次の`tablespace_datafile`句を使用して、これらのデータファイルに異なる属性を指定できます。

```
SEED  
  SYSTEM DATAFILES  
    SIZE 125M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED  
  SYSAUX DATAFILES  
    SIZE 100M
```

この例では、PDBシードのSYSTEM表領域およびSYSAUX表領域のデータファイルは、ルートのデータファイルからREUSE属性を継承します。ただし、PDBシードの次のデータファイル属性は、ルートの属性とは異なります。

- SYSTEM表領域のデータファイルは、PDBシードの場合は125MB、ルートの場合は325MBです。
- AUTOEXTENDは、PDBシードのSYSTEMデータファイルに対して有効で、ルートのSYSTEMデータファイルに対してデフォルト

で無効です。

- SYSAUX表領域のデータファイルは、PDBシードの場合は100MB、ルートの場合は325MBです。

関連項目:

データファイル指定の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください

親トピック: [SQL文を使用したCDB作成について](#)

CDBのUNDOモードについて

共有UNDOがデフォルトです。undo_mode_clauseをENABLE PLUGGABLE DATABASE句に使用することで、CDBのUNDOモードを指定できます。

undo_mode_clauseでは、CDBのUNDOモードがローカルと共有のどちらであるかを指定します。ローカルUNDOモードは、CDB内のすべてのコンテナがローカルUNDOを使用することを意味します。CDBにローカルUNDOモードを構成するには、LOCAL UNDO ONを指定します。

共有UNDOモードは、単一インスタンスCDBに1つのアクティブなUNDO表領域があるか、Oracle RAC CDBの場合はインスタンスごとに1つのアクティブなUNDO表領域があることを意味します。CDBの共有UNDOモードを構成するには、undo_mode_clauseを指定しないか、またはLOCAL UNDO OFFを指定します。

親トピック: [SQL文を使用したCDB作成について](#)

CREATE DATABASE文を使用したCDBの作成

CREATE DATABASE文を使用してCDBを作成する場合は、実行可能なCDBを作成する前に他の処理を実行する必要があります。

この処理には、データ・ディクショナリ表のビューの作成、標準PL/SQLパッケージのルートでのインストールなどがあります。これらの処理は、提供されたcatcdb.sqlスクリプトを実行することで実行され、これによって、CDBに必要なすべてのコンポーネントがインストールされます。

前提条件

次の前提条件に注意してください。

- この項の手順が適用できるのは、単一インスタンス・データベースのインストールの場合のみです。Oracle RAC CDBの作成手順は、使用しているプラットフォーム固有のOracle Real Application Clusters (Oracle RAC)インストール・ガイドを参照してください。
- Oracle ASMを使用してディスク記憶域を管理している場合は、これらのステップを実行する前に、Oracle ASMインスタンスを起動してディスク・グループを構成する必要があります。
- ENABLE_PLUGGABLE_DATABASE初期化パラメータがtrueに設定されている必要があります。

CREATE DATABASE文を使用してCDBを作成するには:

1. [『Oracle Database管理者ガイド』](#)にあるトピック「CREATE DATABASE文を使用したデータベースの作成」のステップ1 - 8を完了します。

CDBでは、DB_NAME初期化パラメータを使用してルートの名前を指定します。また、SIDはルートの名前に設定するのが一般的です。この名前の最大文字数は8文字です。詳細は、[『Oracle Databaseリファレンス』](#)のDB_NAME初期化

パラメータの説明を参照してください。

2. CREATE DATABASE文を使用して、新しいCDBを作成します。

[「Oracle Managed Filesを使用したCDBの作成: 例」](#)および[「Oracle Managed Filesを使用しないCDBの作成」](#)にある例を参照してください。

ヒント:



CREATE DATABASE 文が失敗した場合、および『[Oracle Database 管理者ガイド](#)』にあるトピック「CREATE DATABASE 文を使用したデータベースの作成」のステップ 7 が未完了の場合は、予期しない方法で初期化パラメータが設定されている、このインスタンス用の既存のサーバー・パラメータ・ファイル(SPFIL)がないことを確認します。たとえば、すべての制御ファイルへの完全パスが SPFILE に設定されている場合は、それらの制御ファイルが存在しないと、CREATE DATABASE 文が失敗します。不要な SPFILE を削除した後は、必ずインスタンスを停止して(STARTUP NOMOUNT を指定して)再起動します。

3. catcdb.sql SQLスクリプトを実行します。

SQL*Plusに次を入力し、スクリプトを実行します。

```
@?/rdbms/admin/catcdb.sql
```

4. スクリプトによりプロンプトが表示されたら、パラメータ1のログ・ファイル・ディレクトリおよびパラメータ2のログ・ファイル名を入力します。

次の例の場合は、最初のプロンプトに対して/tmpを、2番目のプロンプトに対してcreate_cdb.logを入力します。

```
SQL> host perl -I &&rdbms_admin &&rdbms_admin_catcdb --logDirectory &&1 --logFilename &&2
Enter value for 1: /tmp
Enter value for 2: create_cdb.log
```

5. スクリプトによりプロンプトが表示されたら、必要な他の情報を入力します。

たとえば、スクリプトによって、管理者パスワードや一時表領域名の入力を求めるプロンプトが表示されます。

```
Enter new password for SYS: *****
Enter new password for SYSTEM: *****
Enter temporary tablespace name: TEMP
```

6. catcdb.sqlが完了した後は、『[Oracle Database管理者ガイド](#)』にあるトピック「CREATE DATABASE文を使用したデータベースの作成」のステップ12 - 14を実行します。

関連項目:

- [CREATE DATABASE](#)句および[Oracle Managed Files](#)についてさらに学習するには、『[Oracle Database 管理者ガイド](#)』を参照してください
- Oracle RACの詳細は、『[Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)』を参照してください
- [Oracle Automatic Storage Management管理者ガイド](#)
- CREATE DATABASE文に対する句およびパラメータ値についてさらに学習するには、『[Oracle Database SQL言語リ](#)

親トピック: [CDBの作成](#)

CREATE DATABASE文を使用したCDBの作成: 例

次の例では、newcdbという名前のCDBを作成します。

これらの例は、『[Oracle Database管理者ガイド](#)』にあるトピック「CREATE DATABASE文を使用したデータベースの作成」のステップ1 - 8を完了していることを前提としています。

- [Oracle Managed Filesを使用しないCDBの作成](#)
次の文では、newcdbという名前のCDBを作成します。この名前は、初期化パラメータ・ファイルのDB_NAMEパラメータと同じにする必要があります。
- [Oracle Managed Filesを使用したCDBの作成: 例](#)
ここでは、より簡単なCREATE DATABASE文を使用できる、Oracle Managed Filesを使用したCDBの作成例を示します。

親トピック: [CDBの作成](#)

Oracle Managed Filesを使用しないCDBの作成

次の文では、newcdbという名前のCDBが作成されます。この名前は、初期化パラメータ・ファイルのDB_NAMEパラメータと同じにする必要があります。

想定

この例では、次のことを想定しています。

- 初期化パラメータ・ファイルのCONTROL_FILESパラメータには、制御ファイルの数と場所が指定されています。
- ディレクトリ/u01/app/oracle/oradata/newcdbが存在します。
- ディレクトリ/u01/app/oracle/oradata/pdbseedが存在します。
- ディレクトリ/u01/logs/myおよび/u02/logs/myが存在します。

この例には、ルートおよびPDBシードのあるCDBを作成するENABLE PLUGGABLE DATABASE句が含まれています。また、この例には、PDBシードのファイルの名前と場所を指定するSEED FILE_NAME_CONVERT句も含まれています。この例には、SYSTEM表領域およびSYSAUX表領域に対して、ルートのデータファイルとは異なるPDBシードのデータファイル属性を指定するtablespace_datafile句も含まれています。この例には、CDBのUNDOモードがローカルであることを指定するundo_mode_clauseが含まれます。

```
CREATE DATABASE newcdb
  USER SYS IDENTIFIED BY sys_password
  USER SYSTEM IDENTIFIED BY system_password
  LOGFILE GROUP 1 ('/u01/logs/my/redo01a.log', '/u02/logs/my/redo01b.log')
    SIZE 100M BLOCKSIZE 512,
  GROUP 2 ('/u01/logs/my/redo02a.log', '/u02/logs/my/redo02b.log')
    SIZE 100M BLOCKSIZE 512,
  GROUP 3 ('/u01/logs/my/redo03a.log', '/u02/logs/my/redo03b.log')
    SIZE 100M BLOCKSIZE 512
  MAXLOGHISTORY 1
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 1024
```



```

CHARACTER SET AL32UTF8
NATIONAL CHARACTER SET AL16UTF16
EXTENT MANAGEMENT LOCAL
DATAFILE '/u01/app/oracle/oradata/newcdb/system01.dbf'
  SIZE 700M REUSE AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED
SYSAUX DATAFILE '/u01/app/oracle/oradata/newcdb/sysaux01.dbf'
  SIZE 550M REUSE AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED
DEFAULT TABLESPACE deftbs
  DATAFILE '/u01/app/oracle/oradata/newcdb/deflbs01.dbf'
  SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED
DEFAULT TEMPORARY TABLESPACE tempts1
  TEMPFILE '/u01/app/oracle/oradata/newcdb/temp01.dbf'
  SIZE 20M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE UNLIMITED
UNDO TABLESPACE undotbs1
  DATAFILE '/u01/app/oracle/oradata/newcdb/undotbs01.dbf'
  SIZE 200M REUSE AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED
ENABLE PLUGGABLE DATABASE
SEED
FILE_NAME_CONVERT = ('/u01/app/oracle/oradata/newcdb/',
                    '/u01/app/oracle/oradata/pdbseed/')
SYSTEM DATAFILES SIZE 125M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
SYSAUX DATAFILES SIZE 100M
USER_DATA TABLESPACE userlbs
  DATAFILE '/u01/app/oracle/oradata/pdbseed/userlbs01.dbf'
  SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED
LOCAL UNDO ON;

```

次の特性を持つCDBが作成されます。

- CDBはnewcdbという名前です。グローバル・データベース名はnewcdb.us.example.comであり、ドメイン部分(us.example.com)は、初期化パラメータ・ファイルから取得されます。グローバル・データベース名の決定の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- CONTROL_FILES初期化パラメータ(CDB作成前に初期化パラメータ・ファイルに設定)で指定された3つの制御ファイルが作成されます。初期化パラメータ・ファイルのサンプルについては[『Oracle Database管理者ガイド』](#)を参照し、制御ファイル指定の詳細は[『Oracle Database管理者ガイド』](#)を参照してください。
- ユーザー・アカウントSYSとSYSTEMのパスワードが指定した値に設定されます。パスワードでは、大/小文字が区別されます。SYSおよびSYSTEMのパスワードを指定する2つの句は、このリリースのOracle Databaseではオプションです。ただし、いずれか一方の句を指定した場合は、もう一方の句も指定する必要があります。これらの句の使用方法の詳細、ユーザーSYSおよびSYSTEMに対するパスワードの指定方法の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- 新しいCDBには、LOGFILE句で指定されているように、オンラインREDOログ・ファイルのグループが3つあり、各グループには2つのメンバーが含まれます。MAXLOGFILES、MAXLOGMEMBERSおよびMAXLOGHISTORYは、REDOログの制限を定義します。REDOログ・ファイル数の選択の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。REDOログのブロック・サイズは、ディスクの物理セクターと同じサイズである512バイトに設定されています。ブロック・サイズを物理セクター・サイズと同じ(デフォルト)にする場合には、BLOCKSIZE句はオプションです。通常のセクター・サイズおよびブロック・サイズは512です。BLOCKSIZEに設定可能な値は512、1024および4096です。新しいディスクでセクター・サイズが4KBの場合には、オプションでBLOCKSIZEを4096に指定します。REDOログ・ファイルのブロック・サイズの計画の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- MAXDATAFILESでは、CDBでオープンできるデータファイルの最大数を指定します。この数は、制御ファイルの初期サイズに影響を及ぼします。CDBの場合は、CDBルート・ファイルに加えて、すべてのコンテナのデータファイルの合計数を考慮した大きい数にMAXDATAFILESを設定します。

ノート:

CDB の作成時に、制限をいくつか設定できます。これらの制限には、オペレーティング・システムの制限によって制限され、その影響を受けるものがあります。たとえば、MAXDATAFILES を設定すると、Oracle Database は、初期の CDB にデータファイルが 1 つしかなくても、制御ファイルに MAXDATAFILES 個のファイル名を格納できるだけの領域を割り当てます。ただし、制御ファイルの最大サイズは制限されており、オペレーティング・システムによって異なるので、CREATE DATABASE 文のパラメータをすべて理論的な最大値で設定できるとはかぎりません。

CDB 作成時に制限を設定する方法の詳細は、[『Oracle Database SQL 言語リファレンス』](#)および使用しているオペレーティング・システム固有の Oracle マニュアルを参照してください。

- このCDBにデータを格納する際は、AL32UTF8文字セットが使用されます。
- NATIONAL CHARACTER SETとしてAL16UTF16文字セットが指定され、特にNCHAR、NCLOBまたはNVARCHAR2として定義された列にデータを格納する際に使用されます。
- DATAFILE句で指定したとおりに、SYSTEM表領域(オペレーティング・システム・ファイル /u01/app/oracle/oradata/newcdb/system01.dbf,から構成される)が作成されます。指定した名前のファイルがすでに存在する場合は上書きされます。
- SYSTEM表領域はローカル管理の表領域として作成されます。ローカル管理SYSTEM表領域の作成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- SYSAUX DATAFILE句で指定したとおりに、SYSAUX表領域(オペレーティング・システム・ファイル /u01/app/oracle/oradata/newcdb/sysaux01.dbfから構成される)が作成されます。SYSAUX表領域の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- DEFAULT TABLESPACE句により、指定した名前で、このCDB用のデフォルト永続表領域が作成されます。
- DEFAULT TEMPORARY TABLESPACE句は、このCDBのルートのデフォルト一時表領域を作成し、名前を付けます。デフォルトの一時表領域の作成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- UNDO TABLESPACE句により、指定した名前で、このCDB用のUNDOデータの格納に使用されるUNDO表領域が作成されます。CDBでは、UNDOデータの管理にUNDO表領域が必要であり、UNDO_MANAGEMENT初期化パラメータをAUTOに設定する必要があります。このパラメータを省略すると、デフォルトでAUTOに設定されます。UNDO表領域の作成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- このCREATE DATABASE文ではARCHIVELOG句が指定されていないため、初期状態ではREDOログ・ファイルはアーカイブされません。これはCDB作成時の慣例です。後でALTER DATABASE文を使用して、ARCHIVELOGモードに切り替えることができます。newcdbのアーカイブに関連する初期化パラメータ・ファイル内の初期化パラメータは、LOG_ARCHIVE_DEST_1およびLOG_ARCHIVE_FORMATです。アーカイブREDOログ・ファイルの管理の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- ENABLE PLUGGABLE DATABASE句では、ルートおよびPDBシードを含むCDBを作成します。
- FILE_NAME_CONVERT句およびtablespace_datafile句にはSEEDが必要です。
- FILE_NAME_CONVERT句では、/u01/app/oracle/oradata/newcdbディレクトリ内にあるファイル名を使用して、/u01/app/oracle/oradata/pdbseedディレクトリ内にあるPDBシードのファイルに対してファイル名を生成します。

- SYSTEM DATAFILES句では、ルートとの属性とは異なるPDBシードのSYSTEM表領域データファイルの属性を指定します。
- SYSAUX DATAFILES句では、ルートとの属性とは異なるPDBシードのSYSAUX表領域データファイルの属性を指定します。
- USER_DATA TABLESPACE句では、ユーザー・データおよびデータベース・オプション(Oracle XML DBなど)を格納するためのPDBシードの表領域を作成して名前を付けます。PDBシードを使用して作成されたPDBには、この表領域とそのデータファイルが含まれます。この句で指定された表領域およびデータファイルは、ルートでは使用されません。
- LOCAL UNDO ON句はCDBのUNDOモードをローカルに設定します。これは、CDB内の各コンテナがローカルUNDOを使用することを意味します。

CDBがローカルUNDOモードで作成された場合、PDBシードにはUNDO表領域が含まれるため、そのPDBシードから作成された新しいPDBにはUNDO表領域が設定されます。PDBが接続またはリモートPDBのクローニングによって作成され、ソースPDBが共通UNDOモードであった場合、PDBが最初にオープンされるときに、PDBにUNDO表領域が自動的に作成されます。

ノート:

- CREATE DATABASE 文で使用しているディレクトリがすべて存在することを確認します。CREATE DATABASE 文ではディレクトリは作成されません。
- Oracle Managed Files を使用していない場合は、すべての TABLESPACE 句に DATAFILE 句または TEMPFILE 句を指定する必要があります。
- CDB の作成に失敗した場合は、アラート・ログを参照すると、失敗の原因とその対処措置を判別できます。アラート・ログを表示する方法の詳細は、『[Oracle Database 管理者ガイド](#)』を参照してください。プロセス番号を含むエラー・メッセージが表示される場合は、そのプロセスのトレース・ファイルを調べます。そのプロセス番号がトレース・ファイル名に含まれているトレース・ファイルを探します。詳細は、[Oracle Database 管理者ガイド](#)を参照してください。
- CREATE DATABASE 文を失敗後に再発行するには、最初にインスタンスを停止してから、前の CREATE DATABASE 文で作成されたファイルを削除する必要があります。

親トピック: [CREATE DATABASE文を使用したCDBの作成](#)

Oracle Managed Filesを使用したCDBの作成: 例

ここでは、より簡単なCREATE DATABASE文を使用できる、Oracle Managed Filesを使用したCDBの作成例を示します。

Oracle Managed Filesを使用するには、初期化パラメータDB_CREATE_FILE_DESTを設定する必要があります。このパラメータは、CDBで作成されて自動的に名前が付けられる様々なCDBファイル用のベース・ディレクトリを定義します。

次の文は、初期化パラメータ・ファイルにこのパラメータを設定する例を示しています。

```
DB_CREATE_FILE_DEST=' /u01/app/oracle/oradata'
```

この例では、Oracle ASM記憶域のパラメータを設定します。

```
DB_CREATE_FILE_DEST = +data
```

この例には、ルートおよびPDBシードのあるCDBを作成するENABLE PLUGGABLE DATABASE句が含まれています。この例では、

Oracle Managed FilesによってPDBシードのファイルの名前と場所が指定されるため、SEED FILE_NAME_CONVERT句は含まれていません。ただし、この例には、SYSTEM表領域およびSYS_AUX表領域に対して、ルートの子データファイルとは異なるPDBシードの子データファイル属性を指定するtablespace_datafile句が含まれています。

Oracle Managed Filesおよび次のCREATE DATABASE文を使用すると、SYSTEM表領域とSYS_AUX表領域および文に指定されている追加の表領域がCDBによって作成され、すべてのデータファイル、制御ファイルおよびREDOログ・ファイルのデフォルト・サイズとプロパティが選択されます。この方法で設定されるこれらのプロパティおよび他のデフォルトCDBプロパティは本番環境には適さない場合があるため、生成された構成を調べて必要に応じて変更することをお勧めします。

```
CREATE DATABASE newcdb
USER SYS IDENTIFIED BY sys_password
USER SYSTEM IDENTIFIED BY system_password
EXTENT MANAGEMENT LOCAL
DEFAULT TABLESPACE users
DEFAULT TEMPORARY TABLESPACE temp
UNDO TABLESPACE undotbs1
ENABLE PLUGGABLE DATABASE
  SEED
  SYSTEM DATAFILES SIZE 125M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
  SYS_AUX DATAFILES SIZE 100M;
```

次の特性を持つCDBが作成されます。

- CDBはnewcdbという名前です。グローバル・データベース名はnewcdb.us.example.comであり、ドメイン部分(us.example.com)は、初期化パラメータ・ファイルから取得されます。グローバル・データベース名の決定の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- ユーザー・アカウントSYSとSYSTEMのパスワードが指定した値に設定されます。パスワードでは、大/小文字が区別されます。SYSおよびSYSTEMのパスワードを指定する2つの句は、このリリースのOracle Databaseではオプションです。ただし、いずれか一方の句を指定した場合は、もう一方の句も指定する必要があります。これらの句の使用の詳細、ユーザーSYSおよびSYSTEMに対するパスワードの指定方法の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- DEFAULT TABLESPACE句により、指定した名前でのこのCDB用のデフォルト永続表領域が作成されます。
- DEFAULT TEMPORARY TABLESPACE句は、このCDBのルートの子デフォルト一時表領域を作成し、名前を付けます。デフォルトの子一時表領域の作成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- UNDO TABLESPACE句により、指定した名前でのこのCDB用のUNDOデータの格納に使用されるUNDO表領域が作成されます。CDBでは、UNDOデータの管理にUNDO表領域が必要であり、UNDO_MANAGEMENT初期化パラメータをAUTOに設定する必要があります。このパラメータを省略すると、デフォルトでAUTOに設定されます。UNDO表領域の作成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- このCREATE DATABASE文ではARCHIVELOG句が指定されていないため、初期状態ではREDOログ・ファイルはアーカイブされません。これはCDB作成時の慣例です。後でALTER DATABASE文を使用して、ARCHIVELOGモードに切り替えることができます。newcdbのアーカイブに関連する初期化パラメータ・ファイル内の初期化パラメータは、LOG_ARCHIVE_DEST_1およびLOG_ARCHIVE_FORMATです。アーカイブREDOログ・ファイルの管理の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- ENABLE PLUGGABLE DATABASE句では、ルートおよびPDBシードを含むCDBを作成します。
- tablespace_datafile句にはSEEDが必要です。
- SYSTEM DATAFILES句では、ルートの子属性とは異なるPDBシードのSYSTEM表領域データファイルの属性を指定します。
- SYS_AUX DATAFILES句では、ルートの子属性とは異なるPDBシードのSYS_AUX表領域データファイルの属性を指定します。

CDB用EM Expressの構成

CDBの場合、グローバルHTTPSポートを設定して、ルートおよび各PDB用にOracle Enterprise Manager Database Express (EM Express)を構成できます。あるいは、CDB内のコンテナごとに異なるポートを設定できます。

CDB用にEM Expressを構成するには:

グローバル・ポートを設定できます。設定すると、CDBにHTTPSポートを使用してEM ExpressでCDB内のすべてのPDBに接続できるようになります。あるいは、CDB内のコンテナごとに異なるHTTPSポートを設定できます。

1. SQL*Plusで、CDBのコンテナにアクセスします。

ユーザーには共通のSYSDBA管理権限が必要であり、接続時にAS SYSDBAを使用してこの権限を行使する必要があります。コンテナは、ルートまたはPDBのいずれでもかまいません。

[「CDB内のコンテナへのアクセスについて」](#)を参照してください。

2. 次のいずれかの方法を使用して、ポートを設定します。

- グローバル・ポートを設定するには、CDB\$ROOTに接続し、次のSQL文を発行してCDB用にグローバル・ポートを構成します。

```
EXEC DBMS_XDB_CONFIG.SETGLOBALPORTENABLED(TRUE);
```

- 現在のコンテナにHTTPSポートを設定するには、次のプロシージャを実行します。

```
exec DBMS_XDB_CONFIG.SETHTTPSPORT(https_port_number);
```

https_port_numberを適切なHTTPSポート番号と置き換えます。

3. EM Expressには、次のいずれかの方法でアクセスします。

- グローバル・ポートを使用するには、PDBを含むCDBを構成したときにDatabase Configuration Assistant (DBCA)によって提供されたEM Express URLを入力します。EM Expressログイン画面が表示されたら、管理者資格証明を指定し、接続するPDBの名前を「コンテナ名」フィールドに入力します。
- コンテナ用のHTTPSポートのURLは、次のとおりです。

```
https://database_hostname:https_port_number/em/
```

database_hostnameをデータベース・インスタンスが実行されているコンピュータのホスト名と置き換え、https_port_numberを適切なHTTPSポート番号と置き換えます。

ルートに接続されている場合、EM ExpressにはCDB全体に該当するデータが表示され、アクションが有効になります。PDBに接続されている場合、EM Expressには、PDBのみに該当するデータが表示され、アクションが有効になります。

ノート:



ポート 1521 にリスナーが構成されていない場合、EM Express 用にこのポートを手動で構成する必要があります。手順は、[『Oracle Database 2 日でデータベース管理者』](#)を参照してください。

関連項目:

EM Expressの詳細は、『[Oracle Database 2日でデータベース管理者](#)』を参照してください

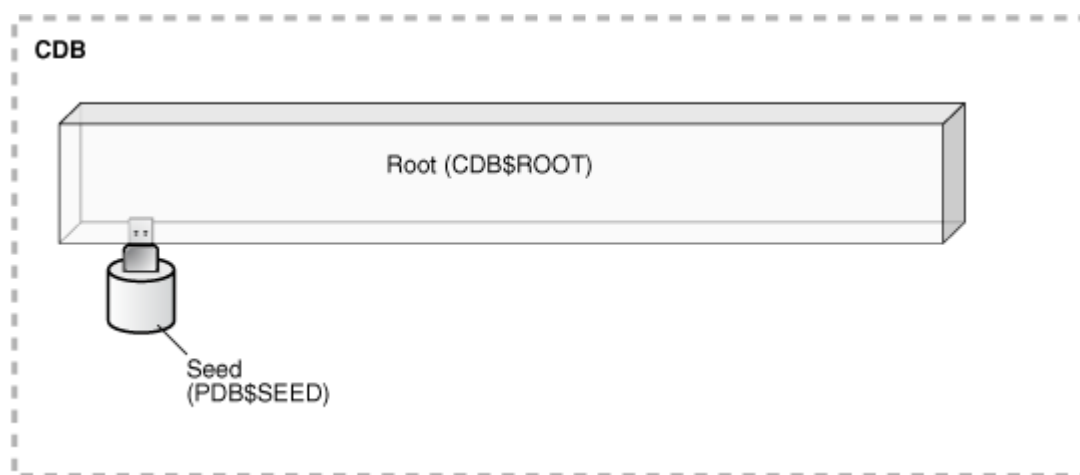
親トピック: [CDBの作成および構成](#)

CDBの作成後

作成後のCDBは、ルートとPDBシードで構成されます。

ルートには、システムにより指定されたメタデータおよびPDBを管理できる共通ユーザーが含まれています。PDBシードは、新しいPDBの作成に使用できるテンプレートです。次の図に、新規に作成されたCDBを示します。

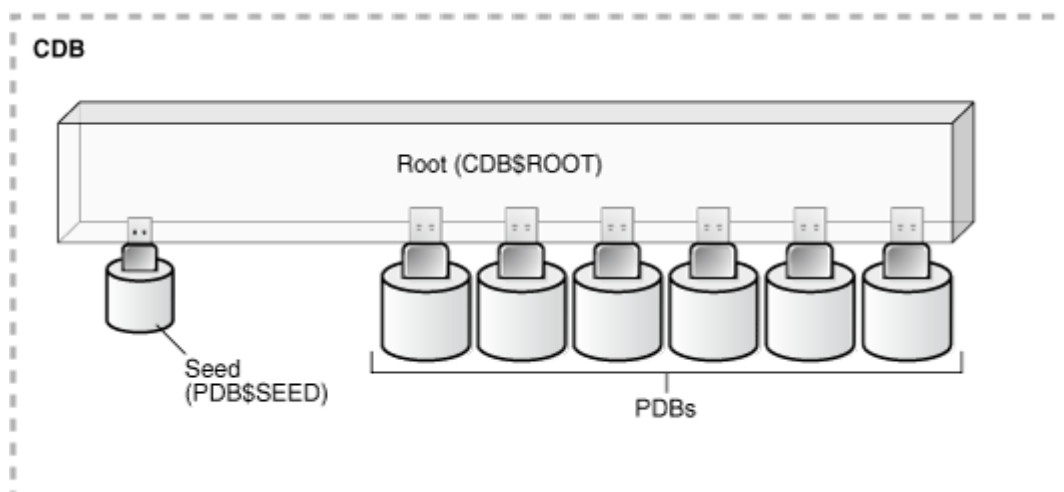
図4-1 新しく作成されたCDB



CDBにおいて、ルートには最小限のユーザー・データが含まれるか、またはユーザー・データは含まれていません。ユーザー・データはPDBに存在します。したがって、CDBの作成後、最初に行うタスクの1つは、ユーザー・データを含むPDBを追加することです。

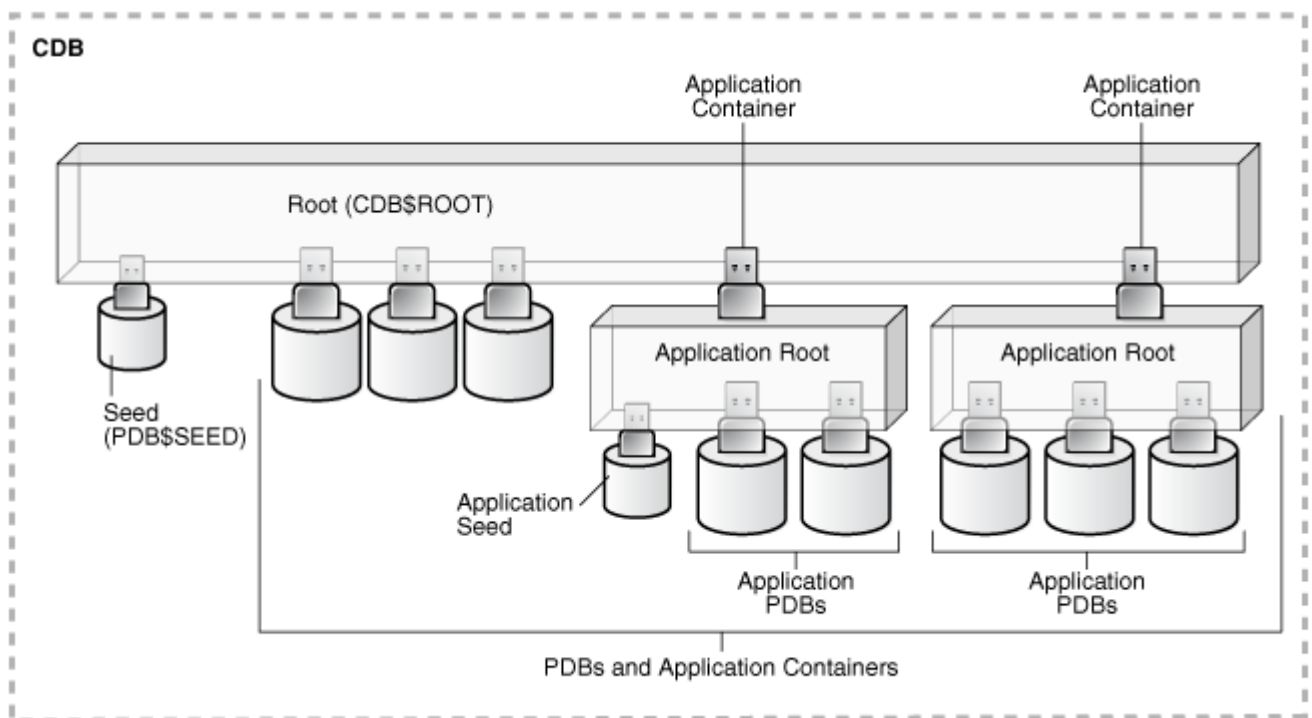
次の図は、PDBが含まれるCDBを示しています。

図4-2 PDBが含まれるCDB



1つ以上のアプリケーション・コンテナを作成するオプションがあります。アプリケーション・コンテナは、アプリケーション・ルートおよびアプリケーションPDBから構成され、1つ以上のアプリケーションのデータを格納します。アプリケーション・コンテナはアプリケーション共通オブジェクトを格納でき、アプリケーション・コンテナ内のアプリケーションPDBで共有できるユーザー・データを含みます。アプリケーション・コンテナ内のアプリケーションPDBを迅速に作成するためのアプリケーション・シードを含むこともできます。

図4-3 CDB内のアプリケーション・コンテナ



PDBとアプリケーション・コンテナをCDBに追加すると、CDBの物理構造は非CDBの物理構造とよく似たものになります。CDBには次のファイルが含まれています。

- 1つの制御ファイル
- 単一インスタンスCDBに1つのアクティブなオンラインREDOログ、またはOracle RAC CDBのインスタンスごとに1つのアクティブなオンラインREDOログ

- 一時ファイルのセット

CDBのルート用に1つ、各PDB、各アプリケーション・ルート、各アプリケーションPDB用に1つのデフォルト一時表領域があります。

- システム・データファイルのセット

CDBと非CDBとの主な物理的違いは、データファイルにあります。非CDBに含まれるシステム・データファイルは1セットのみです。これに対して、CDBには、CDB内のコンテナごとに1セットのシステム・データファイル、およびPDB、アプリケーション・ルート、アプリケーションPDBごとに1セットのデータファイルが含まれます。これに加えて、CDBにはコンテナごとに1セットのユーザー作成のデータファイルが含まれています。CDBがローカルUNDOモードである場合、各コンテナにも固有のUNDO表領域および関連付けられているデータファイルがあります。

- ユーザーが作成したデータファイルのセット

各PDBには、システム以外のデータファイルの独自のセットが含まれます。これらのデータファイルには、PDBのユーザー定義スキーマおよびデータベース・オブジェクトが含まれます。

CDBのバックアップおよびリカバリには、Recovery Manager (RMAN)を使用することをお勧めします。PDBポイントインタイム・リカバリ(PDB PITR)は、RMANを使用して実行する必要があります。デフォルトでは、RMANはCDBの制御ファイルの自動バックアップをオンにします。CDBの制御ファイルの自動バックアップを有効にして、PDB PITRでデータファイルの追加や削除を取り消すことができるようにすることをお勧めします。

CDBと非CDBの物理構造は似ているため、ほとんどの管理タスクはCDBと非CDBとで違いはありません。ただし、CDBに固有の管理タスクもあります。次の章では、これらのタスクについて説明します。

- [「PDBおよびアプリケーション・コンテナの作成および削除」](#)

この章では、次のタスクについて説明しています。

- PDBシードを使用したPDBの作成
- 既存のPDBまたは非CDBのクローニングによるPDBの作成
- 再配置によるPDBの作成
- プロキシPDBとしてのPDBの作成
- PDBの接続
- PDBの切断
- PDBの削除

- [「アプリケーション・コンテナおよびシードの作成と削除」](#)

この章では、次のタスクについて説明しています。

- アプリケーション・コンテナの作成
- アプリケーション・シードの作成
- アプリケーション・コンテナの切断
- アプリケーション・シードの切断
- アプリケーション・コンテナの削除
- アプリケーション・シードの削除

- [「CDBの管理」](#)

この章では、次のタスクについて説明しています。

- コンテナへの接続
- コンテナへの切替え
- CDBの変更
- ルートの変更
- PDBのオープン・モードの変更
- CDBでのDDL文の実行
- CDBインスタンスの停止

- [「PDBの管理」](#)

この章では、次のタスクについて説明しています。

- PDBへの接続
- PDBの変更
- PDBに関連付けられたサービスの管理

- [「アプリケーション・コンテナの管理」](#)

この章では、次のタスクについて説明しています。

- アプリケーション・コンテナ内のアプリケーションの管理(アプリケーションのインストール、アップグレードおよびパッチ適用を含む)
- アプリケーション共通オブジェクトの管理

- アプリケーション・コンテナ内のコンテナに対するDML文の発行
- コンテナ・マップを使用したPDBでのパーティション化
- [「CDBおよびPDBの監視」](#)

この章では、次のタスクについて説明しています。

 - CDBおよびそのPDBを監視するためのビューの問合せ
 - CDBおよびそのPDBに関する情報を提供するサンプル問合せの実行
- [「PDBに対するOracle Resource Managerの使用」](#)

この章では、次のタスクについて説明しています。

 - CDBにおけるリソース・プランの作成
 - CDBにおけるリソース・プランの管理
- [「CDBでのOracle Schedulerの使用」](#)

この章では、次の項目について説明しています。

 - CDBにおけるDBMS_SCHEDULERの起動
 - CDBにおけるジョブ・コーディネータおよびスレーブ・プロセス
 - DBMS_JOBの使用
 - PDBをクローズするプロセス
 - 新規ビューおよび変更されたビュー

関連項目:

- マルチテナント・アーキテクチャ・ドキュメント・ロードマップについては、[『Oracle Database概要』](#)を参照してください
- RMANの詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド』](#)を参照してください

親トピック: [CDBの作成および構成](#)

第III部 PDBおよびアプリケーション・コンテナの作成および削除

様々な手法を使用して、PDB、アプリケーション・コンテナおよびアプリケーション・シードを作成できます。

たとえば、PDBを最初から作成したり、既存のPDBや非CDBをクローニングしたり、切断されたPDBを接続することができます。PDBをCDBから削除することもできます。



ノート:

この部のタスクは、SQL*Plus または Oracle SQL Developer を使用して実行できます。

- [PDB作成の概要](#)
CDBは、複数のPDB作成方法をサポートしています。
- [PDBを最初から作成](#)
CREATE PLUGGABLE DATABASE文により、PDBシード(PDB\$SEED)のファイルを使用して、CDBにPDBを作成します。
- [PDBまたは非CDBのクローニング](#)
ローカルPDB、リモートPDBまたは非CDBをクローニングすることで、PDBを作成できます。
- [PDBの再配置](#)
PDBを異なるCDBまたはアプリケーション・コンテナに移動できます。
- [切断されたPDBの接続](#)
切断されたPDBをCDBに接続することで、PDBを作成できます。
- [プロキシPDBとしてのPDBの作成](#)
リモートCDB内のPDBを参照することで、PDBをプロキシPDBとして作成できます。
- [PDBの削除](#)
接続されたPDBは、切断、削除または再配置することでCDBから削除できます。
- [アプリケーション・コンテナおよびシードの作成と削除](#)
いくつかの異なる方法でアプリケーション・コンテナおよびアプリケーション・シードを作成できます。アプリケーション・コンテナをCDBから削除したり、アプリケーション・コンテナからアプリケーション・シードを削除することもできます。

関連トピック

- [マルチテナント環境のツール](#)

5 PDB作成の概要

CDBは、複数のPDB作成方法をサポートしています。

作成されたPDBでは、メタデータとシステム提供オブジェクトへの内部リンクをはじめとする完全なデータ・ディクショナリが、CDBルートに自動的に含まれます。すべてのPDBを単一のルート([CDBルート](#)または[アプリケーション・ルート](#))から定義する必要があります。

各PDBにはグローバル一意識別子(GUID)があります。PDB GUIDは、PDBのファイルを格納するディレクトリ(Oracle Managed Filesディレクトリおよび非Oracle Managed Filesディレクトリの両方を含む)の名前を生成するために主に使用されます。

- [PDBを作成する手法](#)
様々な手法でPDBを作成できますが、すべてCREATE PLUGGABLE DATABASE文が必要です。
- [PDB記憶域](#)
どのような方法を選択してPDBを作成する場合も、データを格納する表領域およびファイルを決定する必要があります。
- [サービス名変換](#)
PDBの作成には、データベース・サービス名の変更を管理する重要な側面があります。
- [PDB作成のための句の概要](#)
CREATE PLUGGABLE DATABASE文を使用してPDBを作成する場合は、様々な要因に基づいて多様な句を使用できます。
- [PDB作成の一般的な前提条件](#)
PDBを作成する前に、特定の前提条件を満たす必要があります。

親トピック: [PDBおよびアプリケーション・コンテナの作成および削除](#)

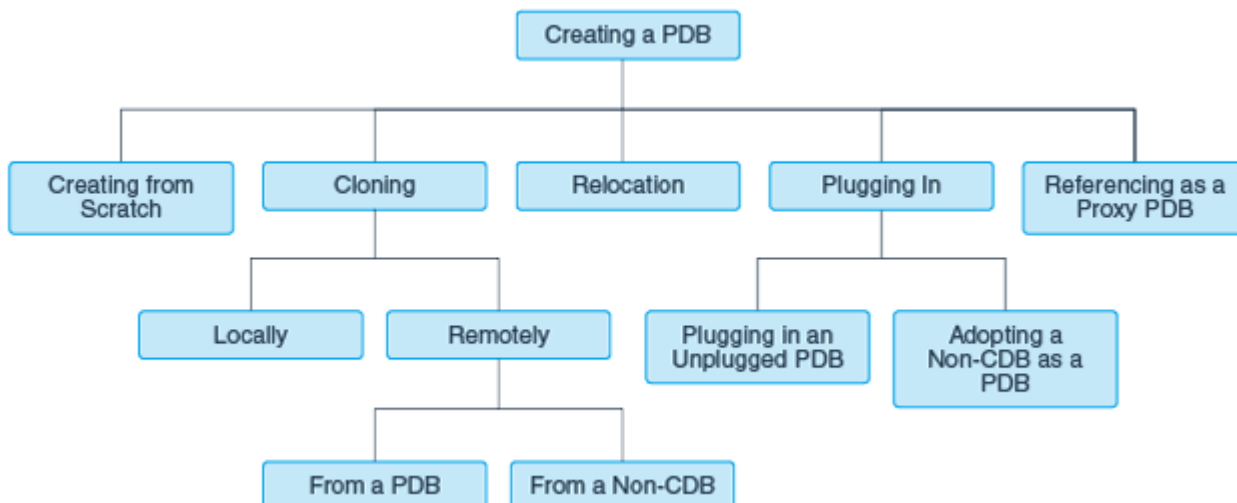
PDBを作成する手法

様々な手法でPDBを作成できますが、すべてCREATE PLUGGABLE DATABASE文が必要です。

PDBの作成は、PDBをCDBまたはアプリケーション・コンテナに関連付けるプロセスです。

次の図に、PDB作成のオプションを示します。

図5-1 PDBを作成するためのオプション



次の表は、作成方法を示しています。このマニュアルには説明されていないその他の手法として、Recovery ManagerのDUPLICATEコマンドを使用して、あるCDBから別のCDBにPDBをコピーする方法があります。

表5-1 PDBを作成する手法

手法	説明	詳細情報
PDB を最初から作成します	PDB シードまたはアプリケーション・シードのファイルを使用して、PDB を CDB に作成します。この手法では、シードに関連付けられているファイルを新しい場所にコピーし、コピーしたファイルを新しい PDB に関連付けます。これは、デフォルトの作成メカニズムです。他の手法は、ソース・データベース (PDB または非 CDB) または XML のいずれかが必要です。	「PDB を最初から作成」
既存の PDB または非 CDB をクローニングします	ソース PDB または非 CDB をクローニングして PDB を作成します。ソースは、ローカル CDB 内の PDB、リモート CDB 内の PDB、ローカルまたはリモート・アプリケーション・コンテナ内の PDB または非 CDB にすることができます。この手法では、ソースに関連付けられているファイルを新しい場所にコピーし、コピーしたファイルを新しい PDB に関連付けます。	「PDB または非 CDB のクローニング」
PDB を異なる CDB に再配置します	ある CDB から別の CDB に再配置することで PDB を作成します。この手法は、PDB に関連付けられているファイルを新しい場所に移動します。	「PDB の再配置」
切断された PDB を CDB に接続します	PDB を記述している XML メタデータ・ファイルを使用して PDB を作成し、これらのファイルを PDB に関連付けて PDB を CDB に接続します。	「切断された PDB の接続」
PDB をプロキシ PDB として参照します	データベース・リンクで別の PDB を参照することで、PDB をプロキシ PDB として作成します。参照先 PDB は、プロキシ PDB と同じ CDB または別の CDB に存在できます。	「プロキシ PDB としての PDB の作成」
非 CDB から PDB を作成し、PDB を CDB に接続します	PDB に非 CDB を適用して、PDB を作成します。DBMS_PDB パッケージを使用して、切断されている PDB を、Oracle Database 12c の非 CDB から作成できます。次に、切断されている PDB を CDB に接続できます。	「非 CDB から PDB を作成するためのオプション」

PDBを異なるCDBに接続する必要がある場合は、そのPDBを切断します。PDBがなくなっただけの場合は、そのPDBを切断または削除できます。切断されたPDBは、CDBに接続されるまで使用できません。

- [現在のコンテナおよびPDB作成](#)

CREATE PLUGGABLE DATABASE文を使用してPDB、アプリケーション・コンテナ、アプリケーション・シードおよびアプリケーションPDBを作成できます。

- [非CDBからPDBを作成するためのオプション](#)
非CDBをPDBに移動するための複数のオプションがあります。

関連項目:

- [「アプリケーション・コンテナの作成および削除」](#)
- [「CDBからのPDBの切断」](#)
- [「PDBの削除」](#)
- DUPLICATEコマンドを使用してPDBを複製する方法を学習するには、[『Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド』](#)を参照してください
- CREATE PLUGGABLE DATABASE文の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください

親トピック: [PDB作成の概要](#)

現在のコンテナおよびPDB作成

CREATE PLUGGABLE DATABASE文を使用してPDB、アプリケーション・コンテナ、アプリケーション・シードおよびアプリケーションPDBを作成できます。

PDBを作成すると、現在のコンテナ(CDBルートまたはアプリケーション・ルート)によってPDBの関連付けが決定されます。PDBおよびアプリケーションPDBを作成するSQL文は同じです。たとえば、CREATE PLUGGABLE DATABASE文をCDBルートで実行した場合、PDBはCDBルートに属します。CREATE PLUGGABLE DATABASE文をアプリケーション・ルートで実行した場合、アプリケーションPDBはアプリケーション・ルートに属します。

CDBルートが現在のコンテナであるときに、アプリケーション・ルートを作成するには、AS APPLICATION CONTAINER句を指定したCREATE PLUGGABLE DATABASE文を実行します。PDBをアプリケーション・コンテナにクローニング、再配置または接続する場合は、PDBのアプリケーション名およびバージョンが、アプリケーション・コンテナのアプリケーション名およびバージョンと一致する必要があります。

親トピック: [PDBを作成する手法](#)

非CDBからPDBを作成するためのオプション

非CDBをPDBに移動するには複数のオプションがあります。

このタスクは、次の方法で実行できます。

- 非CDBをクローニングします
これは、非CDBを使用してPDBを作成する最も簡単な方法ですが、非CDBのファイルを新しい場所にコピーする必要があります。

CDBと非CDBの両方でOracle Database 12cリリース1 (12.1.0.2)以上が実行されている必要があります。現在の非CDBでOracle Database 12cリリース1 (12.1.0.2)よりも前のOracle Databaseリリースを使用している場合、この手法を使用するには、非CDBを最新のリリースにアップグレードする必要があります。
- DBMS_PDBパッケージを使用してXMLメタデータ・ファイルを生成します

XMLメタデータ・ファイルで非CDBのデータベース・ファイルを記述しているため、その非CDBをCDBに接続できます。

この方法は、非CDBをクローニングしてPDBを作成するよりもステップが多くなりますが、場合によっては、非CDBファイルを移動せずに非CDBを使用してPDBを作成することが可能になります。

この手法を使用するには、非CDBがOracle Database 12c以降で実行されている必要があります。現在の非CDBでOracle Database 12cより前のリリースを使用している場合は、新しいリリースにアップグレードする必要があります。

- 非CDBからデータをエクスポートし、Oracle Data Pumpを使用して、PDBにインポートします

インポート時に、ユーザー名の後にPDBの接続識別子を指定します。たとえば、PDBの接続識別子がhrpdbである場合は、Oracle Data Pumpインポート・ユーティリティを実行するとき、次のように入力します。

```
impdp user_name@hrpdb ...
```

非CDBのOracle DatabaseリリースがOracle Database 11g リリース2 (11.2.0.3)以上である場合、完全なトランスポータブル・エクスポート/インポートを使用して、データを移動できます。Oracle Database 11g リリース2 (11.2.0.3)またはOracle Database 11g以上のデータベースからOracle Database 12c以降に非CDBをトランスポートする場合、VERSIONデータ・ポンプ・エクスポート・パラメータを12.0.0.0以上に設定する必要があります。

非CDBのOracle DatabaseリリースがOracle Database 11g リリース2 (11.2.0.3)より前である場合、トランスポータブル表領域を使用してデータを移動するか、または完全なデータベース・エクスポート/インポートを実行できません。

- GoldenGateを使用してデータを非CDBからPDBにレプリケートします

PDBが非CDBに追いついている場合は、そのPDBにフェイルオーバーします。

Oracle GoldenGateのドキュメントを参照してください。

関連項目:

- [「PDBまたは非CDBのクローニング」](#)
- [PDBとしての非CDBの適用](#)
- アップグレードについては、『[Oracle Databaseアップグレード・ガイド](#)』を参照してください
- データのトランスポートの詳細は、『[Oracle Database管理者ガイド](#)』を参照してください

親トピック: [PDBを作成する手法](#)

PDB記憶域

どのような方法を選択してPDBを作成する場合も、データを格納する表領域およびファイルを決定する必要があります。

- [ストレージ制限](#)
CREATE PLUGGABLE DATABASE文のオプションのSTORAGE句では、PDBに対してストレージ制限を指定します。
- [デフォルト表領域](#)
CREATE PLUGGABLE DATABASE文のDEFAULT TABLESPACE句では、新しいPDBのデフォルト表領域を指定します。
- [ユーザー表領域](#)
CREATE PLUGGABLE DATABASE文のUSER TABLESPACES句では、新しいPDBで使用できる表領域を指定します。

- [PDBファイルの場所](#)

CREATE PLUGGABLE DATABASE文で、新しいPDBで使用するファイルの場所を指定できます。

親トピック: [PDB作成の概要](#)

ストレージ制限

CREATE PLUGGABLE DATABASE文のオプションのSTORAGE句では、PDBに対してストレージ制限を指定します。

STORAGE句では、次の制限を指定します。

- PDBに属するすべての表領域が使用できる記憶域の量
MAXSIZEおよびsize句を使用して制限を指定するか、またはMAXSIZEをUNLIMITEDに設定して制限がないことを指定します。
- PDB内の統合監査OSスピルオーバー(.bin形式)ファイルに使用できるストレージ量
MAX_AUDIT_SIZEおよびsize句を使用して制限を指定するか、またはMAX_AUDIT_SIZEをUNLIMITEDに設定して制限がないことを指定します。
- PDBが使用できる自動診断リポジトリ(ADR)の診断(トレース・ファイルおよびインシデント・ダンプ)の量
MAX_DIAG_SIZEおよびsize句を使用して制限を指定するか、またはMAX_DIAG_SIZEをUNLIMITEDに設定して制限がないことを指定します。

STORAGE UNLIMITEDが設定されている場合、またはSTORAGE句がない場合は、PDBのストレージ制限はありません。

次に、STORAGE句の使用例を示します。

例5-1 ストレージ制限を指定するSTORAGE句

次のSTORAGE句では、PDBに属するすべての表領域が使用する記憶域が2GBを超えない必要があることを指定しています。

```
STORAGE (MAXSIZE 2G)
```

例5-2 無制限の記憶域を指定するSTORAGE句

次のSTORAGE句では、PDBに属するすべての表領域の記憶域に制限がないことを指定しています。

```
STORAGE (MAXSIZE UNLIMITED)
```

関連項目:

STORAGE句の構文は、[『Oracle Database SQL言語リファレンス』](#)を参照してください

親トピック: [PDB記憶域](#)

デフォルト表領域

CREATE PLUGGABLE DATABASE文のDEFAULT TABLESPACE句では、新しいPDBのデフォルト表領域を指定します。

Oracle Databaseによって、別の永続表領域が指定されていないSYSTEM以外のユーザーに対してデフォルト表領域が割り当てられます。

PDBをPDBシードまたはアプリケーション・シードから作成し、DEFAULT TABLESPACE句を指定する場合、Oracle Databaseはsmallfile表領域を作成し、それをPDBのデフォルト表領域として設定します。PDBのクローニングや切断されたPDBの接続

など、PDBシードまたはアプリケーション・シード以外の方法を使用してPDBを作成する場合、デフォルト表領域はソースPDBにすでに存在する表領域である必要があります。

例5-3 DEFAULT TABLESPACE句

```
DEFAULT TABLESPACE sales
```

親トピック: [PDB記憶域](#)

ユーザー表領域

CREATE PLUGGABLE DATABASE文のUSER_TABLESPACES句では、新しいPDBで使用できる表領域を指定します。

この句を使用すると、複数のスキーマのデータを異なるPDBに分けることができます。たとえば、非CDBをPDBに移動する際、非CDBに様々なアプリケーションでそれぞれサポートされている複数のスキーマがある場合は、この句を使用して、各スキーマに属するデータを別個のPDBに分離できます。この手法では、各スキーマが非CDBの個別の表領域を使用するものとします。

この句を使用して、次のいずれかのオプションを指定できます。

- 含める1つ以上の表領域をリストします。
- すべての表領域を含める場合はALL (デフォルト)を指定します。
- リストした表領域を除くすべての表領域を含める場合はALL EXCEPTを指定します。
- すべての表領域を除外する場合はNONEを指定します。
- ユーザー表領域の作成モードがオラクル社提供の表領域(SYSTEMやSYSAUXなど)の作成モードと異なる必要がある場合は、USER_TABLESPACES句に次のいずれかを指定します。
 - COPY: 表領域のファイルは新しい場所にコピーされます。
 - MOVE: 表領域のファイルは新しい場所に移動されます。
 - NOCOPY: 表領域のファイルはコピーまたは移動されません。
 - SNAPSHOT COPY: 表領域は記憶域スナップショットにクローニングされます。
 - NO DATA: 表領域のデータ・モデル定義はクローニングされますが、表領域のデータはクローニングされません。

CDBの互換性レベルが12.2.0以上の場合、この句によって除外される表領域は新しいPDBにオフラインで作成され、データファイルは関連付けられません。CDBの互換性レベルが12.2.0未満の場合、この句で除外される表領域は新しいPDBでオフラインとなり、このような表領域に属するデータファイルはすべて無名でオフラインとなります。

この句は、SYSTEM、SYSAUXまたはTEMPの各表領域には適用されません。これらの表領域は、この句の表領域リストに含めないでください。

次に、USER_TABLESPACES句の使用例を示します。

例5-4 1つの表領域が含まれるUSER_TABLESPACES句

PDBの作成元となる非CDBまたはPDBにはtbs1、tbs2およびtbs3の各表領域があるとします。このUSER_TABLESPACES句には、tbs2表領域は含まれますが、tbs1表領域およびtbs3表領域は含まれません。

```
USER_TABLESPACES=( ' tbs2' )
```

例5-5 表領域のリストが含まれるUSER_TABLESPACES句

PDBの作成元となる非CDBまたはPDBにはtbs1、tbs2、tbs3、tbs4およびtbs5の各表領域があるとします。このUSER_TABLESPACES句には、tbs1、tbs4およびtbs5の各表領域は含まれますが、tbs2表領域およびtbs3表領域は含まれま

せん。

```
USER_TABLESPACES=( ' tbs1' , ' tbs4' , ' tbs5' )
```

例5-6 リストした表領域を除くすべての表領域が含まれるUSER_TABLESPACES句

PDBの作成元となる非CDBまたはPDBにはtbs1、tbs2、tbs3、tbs4およびtbs5の各表領域があります。このUSER_TABLESPACES句には、tbs2表領域およびtbs3表領域は含まれますが、tbs1、tbs4およびtbs5の各表領域は含まれません。

```
USER_TABLESPACES=ALL EXCEPT( ' tbs1' , ' tbs4' , ' tbs5' )
```

例5-7 異なる作成モードのUSER_TABLESPACES

この例は、非CDBを接続し、非CDBからtbs3ユーザー表領域のみ含める完全なCREATE PLUGGABLE DATABASE文を示します。例では、オラクル社が提供する表領域(SYSTEMやSYSAUXなど)のファイルは新しい場所にコピーしますが、tbs3ユーザー表領域のファイルは移動します。

```
CREATE PLUGGABLE DATABASE ncdb USING '/disk1/oracle/ncdb.xml'  
COPY  
FILE_NAME_CONVERT = ( '/disk1/oracle/dbs/' , '/disk2/oracle/ncdb/' )  
USER_TABLESPACES=( ' tbs3' ) MOVE;
```

親トピック: [PDB記憶域](#)

PDBファイルの場所

CREATE PLUGGABLE DATABASE文で、新しいPDBで使用するファイルの場所を指定できます。

「ファイル名」という用語は、ファイルの名前と場所の両方を意味します。CREATE PLUGGABLE DATABASE文には、作成される新しいPDBのファイル名を示す次の句が含まれています。

- FILE_NAME_CONVERT句では、PDBが作成された後のPDBのファイル名を指定します。
ファイルが最終保存先にまだない場合にこの句を使用して、PDBの作成時にこれらのファイルをコピーまたは移動します。この句は、任意のCREATE PLUGGABLE DATABASE文で使用できます。
- CREATE_FILE_DEST句には、PDBのファイル用のデフォルトのOracle Managed Filesファイル・システム・ディレクトリまたはOracle ASMディスク・グループを指定します。
この句を使用して、CDB用にルートで指定したOracle Managed Filesのデフォルトの場所に関係なく、新しいPDBに対してOracle Managed Filesを有効にします。この句は、任意のCREATE PLUGGABLE DATABASE文で使用できます。

必要な場合は、この両方の句を同じCREATE PLUGGABLE DATABASE文で使用できます。さらに、次の初期化パラメータによって新しいPDBファイルの場所を制御できます。

- ルートに設定したDB_CREATE_FILE_DEST初期化パラメータ
この初期化パラメータでは、CDB用のOracle Managed Filesのデフォルトの場所を指定します。このパラメータをPDBで設定する場合、そのPDB用のOracle Managed Filesのデフォルトの場所を指定します。
- PDB_FILE_NAME_CONVERT初期化パラメータ
この初期化パラメータは、CREATE PLUGGABLE DATABASE文の処理時に、既存ファイルの名前と新しいファイルの名前をマップします。

次の表に、同じCREATE PLUGGABLE DATABASE文で両方の句を使用し、初期化パラメータをどちらも設定した場合の優先順

位を示します。各句と初期化パラメータについて、表ではCREATE PLUGGABLE DATABASE文で作成されたファイルがOracle Managed Filesを使用するかどうかを示します。

表5-2 ファイルの場所の句および初期化パラメータのサマリー

句または初期化パラメータ	優先順位	CREATE PLUGGABLE DATABASEによって作成されたファイルがOracle Managed Filesを使用するか。
FILE_NAME_CONVERT 句	1	いいえ
CREATE_FILE_DEST 句	2	はい
DB_CREATE_FILE_DEST 初期化パラメータ	3	はい
PDB_FILE_NAME_CONVERT 初期化パラメータ	4	いいえ

Oracle Managed Filesの使用について、表はCREATE PLUGGABLE DATABASE文で作成されるファイルにのみ適用されます。PDBが作成された後にPDBに対して作成されたファイルは、Oracle Managed Filesを使用する場合と使用しない場合があります。

また、CREATE PLUGGABLE DATABASE文でFILE_NAME_CONVERTとCREATE_FILE_DESTの両方を指定した場合、PDBの作成時、配置されるファイルにはFILE_NAME_CONVERT設定が使用され、PDBでのDB_CREATE_FILE_DEST初期化パラメータの設定にはCREATE_FILE_DEST設定が使用されます。この場合、PDBの作成後、Oracle Managed FilesによってPDBのファイルの場所が制御されます。



ノート:

PATH_PREFIX 句は、Oracle Managed Files によって作成されたファイルには影響しません。

- [FILE_NAME_CONVERT 句](#)
PDBでOracle Managed Filesを使用しない場合、CREATE PLUGGABLE DATABASE文のFILE_NAME_CONVERT句では、既存ファイルの名前を使用してファイル(データファイルなど)の名前を生成する方法を指定します。
- [CREATE_FILE_DEST 句](#)
CREATE PLUGGABLE DATABASE文のCREATE_FILE_DEST句は、PDBでOracle Managed Filesを有効にして、PDBファイル用のデフォルトのファイル・システム・ディレクトリまたはOracle ASMディスク・グループを指定します。
- [PDBファイルの場所に対する制限](#)
CREATE PLUGGABLE DATABASE文のPATH_PREFIX句は、PDBと関連付けられているすべてのディレクトリ・オブジェクト・パスが、指定したディレクトリまたはそのサブディレクトリに制限されるようにします。

関連項目:

[DB_CREATE_FILE_DEST](#)および[PDB_FILE_NAME_CONVERT](#)についてさらに学習するには、*Oracle Database*リファレンスを参照してください。

親トピック: [PDB記憶域](#)

FILE_NAME_CONVERT句

PDBでOracle Managed Filesを使用しない場合、CREATE PLUGGABLE DATABASE文のFILE_NAME_CONVERT句では、既存ファイルの名前を使用してファイル(データファイルなど)の名前を生成する方法を指定します。

この句を使用して、次のいずれかのオプションを指定できます。

- 1つ以上のファイル名パターンおよび置換ファイル名パターンを次の形式で指定します。

```
'string1' , 'string2' , 'string3' , 'string4' , ...
```

string1ファイル名パターンがstring2ファイル名パターンによって置き換えられ、string3ファイル名パターンがstring4ファイル名パターンによって置き換えられます。ファイル名パターン文字列と置換ファイル名パターン文字列のペアを必要な数だけ使用できます。

奇数の文字列を指定した(最終文字列に対応する置換文字列が存在しない)場合、エラーが返されます。1つのファイル名またはディレクトリと一致する複数のパターン/置換文字列を指定しないでください。

- PDBの作成時にいずれのファイルもコピーまたは移動しない場合はNONEを指定します。FILE_NAME_CONVERT句の省略は、NONEの指定と同じです。

FILE_NAME_CONVERT句は、任意のCREATE PLUGGABLE DATABASE文で使用できます。

FILE_NAME_CONVERT句がCREATE PLUGGABLE DATABASE文で指定されていない場合は、Oracle Managed FilesまたはPDB_FILE_NAME_CONVERT初期化パラメータのいずれかによって、ファイルの名前を生成する方法が指定されます。Oracle Managed FilesおよびPDB_FILE_NAME_CONVERT初期化パラメータの両方を使用する場合は、Oracle Managed Filesが優先されます。FILE_NAME_CONVERT句が指定されている場合はこれが優先されます。

FILE_NAME_CONVERT句で指定するファイル名パターンをOracle Managed Filesで管理されているファイルやディレクトリと一致させることはできません。

例5-8 FILE_NAME_CONVERT句

次のFILE_NAME_CONVERT句では、/oracle/dbsディレクトリ内のファイル名を使用して、/oracle/pdb5ディレクトリの新しいPDBのファイル名を生成します。

```
FILE_NAME_CONVERT = ('/oracle/dbs/' , '/oracle/pdb5/')
```

関連項目:

- [「例19-7」](#)
- [Oracle Database管理者ガイド](#)
- FILE_NAME_CONVERT句の構文は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。
- PDB_FILE_NAME_CONVERT初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください

親トピック: [PDBファイルの場所](#)

CREATE_FILE_DEST句

CREATE PLUGGABLE DATABASE文のCREATE_FILE_DEST句は、PDBでOracle Managed Filesを有効にして、PDBファイル用のデフォルトのファイル・システム・ディレクトリまたはOracle ASMディスク・グループを指定します。

PDBデータファイルおよび一時ファイルは、指定されたディレクトリおよびそのサブディレクトリに制限されます。ファイル・システム・

ディレクトリをデフォルトの場所としてこの句に指定する場合、そのディレクトリは存在している必要があります。また、CREATE PLUGGABLE DATABASE文を実行するユーザーには、指定ディレクトリでファイルを作成するために適切な権限が必要です。または、CDBルート(CDB\$ROOT)に存在するディレクトリ・オブジェクトの名前を指定できます。ディレクトリ・オブジェクトは、CREATE_FILE_DESTにより使用されるファイル・システム・ディレクトリを指します。

CDB用のOracle Managed Filesのデフォルトの場所がCDBルートに設定されている場合、CREATE_FILE_DEST設定によってCDBルートの設定が上書きされ、指定したCREATE_FILE_DEST設定がPDBに使用されます。

CREATE_FILE_DEST=NONEを指定すると、PDBに対してOracle Managed Filesは無効になります。

CREATE_FILE_DEST句をNONE以外の値に設定すると、DB_CREATE_FILE_DEST初期化パラメータはPDBでSCOPE=SPFILEと暗黙的に設定されます。

CDBルートでOracle Managed Filesを使用し、この句を指定しない場合、PDBはOracle Managed Filesのデフォルトの場所をCDBルートから継承します。

ノート:



この機能は、Oracle Database 12c リリース 1 (12.1.0.2)以降で使用可能です。

例5-9 CREATE_FILE_DEST句

このCREATE_FILE_DEST句では、新しいPDB用のデフォルトのOracle Managed Filesファイル・システム・ディレクトリとして /oracle/pdb2/を指定しています。

```
CREATE_FILE_DEST = '/oracle/pdb2/'
```

関連項目:

[Oracle Database管理者ガイド](#)

親トピック: [PDBファイルの場所](#)

PDBファイルの場所に対する制限

CREATE PLUGGABLE DATABASE文のPATH_PREFIX句は、PDBと関連付けられているすべてのディレクトリ・オブジェクト・パスが、指定したディレクトリまたはそのサブディレクトリに制限されるようにします。

また、この句により、PDBに関連付けられているファイル(CREATE PFILE文で作成されたファイル)は指定されたディレクトリ(PDBのOracle XMLリポジトリ)およびOracleウォレットのエクスポート・ディレクトリに制限されます。PDBのファイルが特定のディレクトリおよびそのサブディレクトリに確実に存在するようにする場合は、この句を使用します。

この句を使用して、次のいずれかのオプションを指定できます。

- PDBに関連付けられたすべてのファイル・パスのプリフィックスとして使用される絶対パス。
- CDBルート(CDB\$ROOT)に存在するディレクトリ・オブジェクトの名前。ディレクトリ・オブジェクトは、PATH_PREFIXに使用される絶対パスを指しています。
- NONE。ファイル・パスに制限がないことを示します。PATH_PREFIX句の省略は、NONEの指定と同じです。

PDBが作成された後は、そのPATH_PREFIX設定を変更できなくなります。

PATH_PREFIX句は、CREATE PLUGGABLE DATABASE文で使用できます。

例5-10 PATH_PREFIX句

このPATH_PREFIX句では、PDBと関連付けられているすべてのファイル・パスが/disk1/oracle/dbs/salespdb/ディレクトリに制限されるようにします。

```
PATH_PREFIX = '/disk1/oracle/dbs/salespdb/'
```

ファイル名が付加された際に正しい形式となるよう、パス名は指定してください。たとえば、UNIXシステムでは、パス名はフォワード・スラッシュ(/)で終わるようにします。

ノート:

- PDB に PATH_PREFIX 句を指定すると、既存のディレクトリ・オブジェクトが予期したとおりに動作しないことがあります。これは、PATH_PREFIX 文字列が PDB 内のすべてのローカル・ディレクトリ・オブジェクトの接頭辞として常に追加されるためです。
- PATH_PREFIX 句は、Oracle Managed Files によって作成されたファイルには影響しません。
- PATH_PREFIX 句は、ユーザーが作成したディレクトリ・オブジェクトにのみ適用されます。これは、Oracle が提供するディレクトリ・オブジェクトには適用されません。
- PATH_PREFIX 句は、データファイルまたは一時ファイルには適用されません。Oracle Managed Files を使用している場合は、CREATE_FILE_DEST 句を使用してデータファイルおよび一時ファイルの場所を制限します。

関連項目:

- [「マルチテナント環境について」](#)
- [「CDB内のコンテナに関する情報の表示」](#)

親トピック: [PDBファイルの場所](#)

サービス名変換

PDBの作成には、データベース・サービス名の変更を管理する重要な側面があります。

新しいPDBのサービス名がCDB内の既存のサービス名と競合する場合は、プラグイン違反が発生することがあります。CREATE PLUGGABLE DATABASE文のSERVICE_NAME_CONVERT句は、ソースPDBのサービス名に基づいて新しいPDBのユーザー定義サービスを名前変更します。この句を使用すると、サービス名を変更してプラグイン違反を回避できます。

この句を使用して、次のいずれかのオプションを指定できます。

- 次の形式の1つ以上のサービス名と置換サービス名:

```
'string1' , 'string2' , 'string3' , 'string4' , ...
```


string2サービス名はstring1サービス名を置換し、string4サービス名はstring3サービス名を置換します。サービス名と置換サービス名のペアは、必要に応じていくつでも使用できます。

奇数の文字列を指定した(最終文字列に対応する置換文字列が存在しない)場合、エラーが返されます。

- サービス名を変更する必要がない場合はNONE。SERVICE_NAME_CONVERT句の省略は、NONEの指定と同じです。

SERVICE_NAME_CONVERT句は、PDBシードからPDBを作成するCREATE PLUGGABLE DATABASE文を除く任意のCREATE PLUGGABLE DATABASE文で使用できます。PDBシードはユーザー定義サービスを持つことができません。ただし、アプリケーション・コンテナ内のアプリケーション・シードからアプリケーションPDBを作成するCREATE PLUGGABLE DATABASE文にはこの文を使用できます。



ノート:

この句は、PDB のデフォルト・サービスに適用されません。デフォルト・サービスの名前は PDB と同じになります。

例5-11 SERVICE_NAME_CONVERT句

このSERVICE_NAME_CONVERT句は、salesrepサービスの名前をsalespersonに変更します。

```
SERVICE_NAME_CONVERT = (' salesrep', ' salesperson')
```

関連項目:

[Oracle Database SQL言語リファレンス](#)

親トピック: [PDB作成の概要](#)

PDB作成のための句の概要

CREATE PLUGGABLE DATABASE文を使用してPDBを作成する場合は、様々な要因に基づいて多様な句を使用できます。1つの要因は、PDBを作成するために使用している手法です。一連の質問に回答することで、使用する句を判断できます。次の表では、様々な要因に基づいて指定するCREATE PLUGGABLE DATABASEの句について説明しています。

表5-3 PDBを作成する場合の句

質問	はい	いいえ	句を使用可能な場合
PDB のかわりにアプリケーション・コンテナを作成しますか。	AS APPLICATION CONTAINER 句を指定します。	AS APPLICATION CONTAINER 句を省略します。	CDB でのアプリケーション・コンテナの作成
同じ PDB を接続して作成した 1 つ以上の PDB を含む CDB に PDB を接続しますか。	AS CLONE 句を指定して、Oracle Database が一意の PDB DBID、GUID、および新しい PDB に予測されるその他の識別子を確実に生成するようにします。この PDB は、すべての識別子が一意であることを保証するため、切断	AS CLONE 句を省略します。	切断された PDB の接続

質問	はい	いいえ	句を使用可能な場合
	された PDB のクローンとして 接続されます。		
アプリケーション・コンテナにアプリケーション・シードを作成しますか。	AS SEED 句を指定します。	AS SEED 句を省略します。	アプリケーション・コンテナでのアプリケーション・シードの作成
CREATE_FILE_DEST 句を使用して、PDB ファイル用に Oracle Managed Files のデフォルトの場所を指定しますか。 PDB シードまたはアプリケーション・シードから PDB を作成する場合、ソース・ファイルは、シードに関連付けられたファイルです。	PDB のファイル用のデフォルトのファイル・システム・ディレクトリまたは Oracle ASM ディスク・グループを指定する CREATE_FILE_DEST 句を含めます。	CREATE_FILE_DEST 句を省略します。 次のいずれかの手法を使用して、ファイルのターゲットの場所を指定します。 ● FILE_NAME_CONVERT 句 ● CDB に対して Oracle Managed Files を有効にして、ターゲットの場所を識別できるようにします。 ● PDB_FILE_NAME_CONVERT 初期化パラメータでターゲットの場所を指定します。	PDB シードまたはアプリケーション・シードからの PDB の作成 PDB のクローニング PDB の再配置 切断された PDB の接続
		「PDB ファイルの場所」 を参照してください。	
PDB のデフォルト・パラメータ表領域を指定しますか。	適切な制限を使用して DEFAULT TABLESPACE 句を指定します。 この表領域には、Oracle Database によって、別の永続表領域が指定されていない SYSTEM 以外のユーザーが割り当てられます。 PDB を PDB シードまたはアプリケーション・シードから作成	DEFAULT TABLESPACE 句を省略します。 この句を指定しない場合、SYSTEM 以外のユーザーに対するデフォルトの永続表領域は SYSTEM 表領域です。SYSTEM 表領域を非 SYSTEM ユーザーに使用することはお勧めしません。	PDB シードまたはアプリケーション・シードからの PDB の作成 PDB のクローニング PDB の再配置 切断された PDB の接続

質問	はい	いいえ	句を使用可能な場合
	<p>する場合、Oracle Database は smallfile 表領域を作成し、それをデフォルト表領域として設定します。PDB シードまたはアプリケーション・シードからの作成以外の手法を使用する場合は、指定した表領域がソース PDB に存在する必要があります。</p>		
<p>FILE_NAME_CONVERT 句を使用して、ファイルのターゲットの場所を指定しますか。</p> <p>PDB シードまたはアプリケーション・シードから PDB を作成する場合、ソース・ファイルは、シードに関連付けられたファイルです。</p>	<p>ソース・ファイルの名前に基づいてファイルのターゲットの場所を指定する</p> <p>FILE_NAME_CONVERT 句を含めます。</p>	<p>FILE_NAME_CONVERT 句を省略します。</p> <p>次のいずれかの手法を使用して、ファイルのターゲットの場所を指定します。</p> <ul style="list-style-type: none"> ● CREATE_FILE_DEST 句 ● CDB に対して Oracle Managed Files を有効にして、ターゲットの場所を識別できるようにします。 ● PDB_FILE_NAME_CONVERT 初期化パラメータでターゲットの場所を指定します。 	<p>PDB シードまたはアプリケーション・シードからの PDB の作成</p> <p>PDB のクローニング</p> <p>PDB の再配置</p> <p>プロキシ PDB の作成 (SYSTEM および SYSAUX 表領域のデータファイルにのみ適用されます。)</p> <p>切断された PDB の接続</p>
		<p>「PDB ファイルの場所」を参照してください。</p>	
<p>PDB は依存するプロキシ PDB を持つ参照 PDB ですか？ リスナーのホスト名を変更しますか？</p>	<p>HOST 句を含め、作成される PDB のリスナーのホスト名を指定します</p> <p>たとえば、物理ホスト名およびデフォルトのポートのリスナー・ネットワークがあり、デフォルト以外のポート番号を使用した仮想ホスト名および仮想 IP</p>	<p>HOST 句を省略します。</p>	<p>PDB シードまたはアプリケーション・シードからの PDB の作成</p> <p>PDB のクローニング</p> <p>PDB の再配置</p>

質問	はい	いいえ	句を使用可能な場合
	アドレスのための 2 番目のリスナーが構成されている場合があります。		切断された PDB の接続
新しい PDB に表領域のロギング属性を指定しますか。	logging_clause を含めません。	logging_clause を省略します。	PDB シードまたはアプリケーション・シードからの PDB の作成 PDB のクローニング 切断された PDB の接続
ファイルを新しい場所にコピーまたは移動しますか。	COPY を指定して、ファイルを新しい場所にコピーします。COPY がデフォルトです。MOVE を指定して、ファイルを新しい場所に移動します。次のいずれかの手法を使用して、ターゲットの場所を指定します。 <ul style="list-style-type: none"> ● ソース・ファイルの名前に基づいてターゲットの場所を指定する FILE_NAME_CONVERT 句を含めます。 ● PDB のファイル用に Oracle Managed Files のデフォルトの場所を指定する CREATE_FILE_DEST 句を含めます。 ● Oracle Managed Files を有効にして、ターゲットの場所を識別できるようにします。 ● PDB_FILE_NAME_CONVERT 初期化パラメータでターゲットの場所を指定します。 	NOCOPY を指定します。	切断された PDB の接続

質問	はい	いいえ	句を使用可能な場合
	「PDB ファイルの場所」 を参照してください。		
ソース PDB のデータ・モデル定義をクローニングするが、ソース PDB のデータをクローニングしないことを指定しますか。	NO DATA 句を含めます。	NO DATA 句を省略します。	PDB のクローニング
複数のパラレル実行サーバーを使用して PDB の作成を並列化しますか。	CDB で並列度を選択するには、PARALLEL 句を含めるか省略します。 並列度を指定するには、PARALLEL 句で整数を指定します。たとえば、PARALLEL 4 を指定して、並列度 4 を指定します。	PARALLEL 0 または PARALLEL 1 を指定します。	PDB シードまたはアプリケーション・シードからの PDB の作成 PDB のクローニング
ディレクトリ・オブジェクト、PDB の Oracle XML リポジトリ、CREATE PFILE 文を使用して作成されたファイル、および Oracle ウォレットのエクスポート・ディレクトリのための PDB のファイル・パスを制限するために PATH_PREFIX 句を使用しますか？ PATH_PREFIX 句は、Oracle Managed Files によって作成されたファイルには影響しません。	絶対パスを指定する PATH_PREFIX 句を含めます。	PATH_PREFIX 句を NONE に設定するか、省略します。	PDB シードまたはアプリケーション・シードからの PDB の作成 PDB のクローニング PDB の再配置 切断された PDB の接続
PDB は依存するプロキシ PDB を持つ参照 PDB ですか？ リスナーのポート番号を 1521 以外の値に変更しますか？	PORT 句を含め、作成される PDB のリスナーのポート番号を指定します たとえば、物理ホスト名およびデフォルトのポートのリスナー・ネットワークがあり、デフォルト以外のポート番号を使用した仮想ホスト名および仮想 IP	PORT 句を省略します。	PDB シードまたはアプリケーション・シードからの PDB の作成 PDB のクローニング PDB の再配置

質問	はい	いいえ	句を使用可能な場合
	アドレスのための 2 番目のリスナーが構成されている場合があります。		切断された PDB の接続
ソース PDB からクローン PDB に変更を伝播するために PDB をリフレッシュできるようにしますか。	REFRESH MODE MANUAL または REFRESH MODE EVERY minutes 句を含めます。	REFRESH MODE 句を省略するか REFRESH MODE NONE 句を含めます。	PDB のクローニング
リフレッシュ可能 PDB は読取り専用モードでオープンする必要があります。			
Oracle の事前定義されているロールを PDB でローカルに PDB_DBA ロールに付与しますか。 PDB の新しい管理者には、PDB_DBA 共通ロールが PDB でローカルに付与されます。デフォルトでは、CREATE PLUGGABLE DATABASE 文は管理者またはロールにいずれの権限も付与しません。	ROLES 句を含めて、Oracle の事前定義されているロールを PDB_DBA ロールに付与することを指定します。指定されたロールは、PDB でローカルに PDB_DBA ロールに付与されます。CREATE PLUGGABLE DATABASE 文を実行するユーザーには、指定されたロールを割り当てる必要はありません。Oracle の事前定義されているロールの詳細は、 『Oracle Database セキュリティ・ガイド』 を参照してください。	ROLES 句を省略します。	PDB シードまたはアプリケーション・シードからの PDB の作成 プロキシ PDB の作成
SERVICE_NAME_CONVERT 句を使用して、ソース PDB のサービス名に基づいて新しい PDB のユーザー定義サービス名を変更しますか。	サービスの新しい名前とそれが置換するサービス名を指定する SERVICE_NAME_CONVERT 句を含めます。必要に応じて複数のサービス名と置換サービス名を指定します。	SERVICE_NAME_CONVERT 句を省略します。	PDB シードではなくアプリケーション・シードからの PDB の作成 PDB のクローニング PDB の再配置 プロキシ PDB の作成 (SYSTEM および SYSAUX 表領域のデータファイルにのみ適用されます。) 切断された PDB の接続

質問	はい	いいえ	句を使用可能な場合
<p>記憶域管理スナップショット (ALTER PLUGGABLE DATABASE SNAPSHOT で生成されたスナップショットではない) を使用して PDB をクローニングしますか。</p>	<p>記憶域管理スナップショットを使用して PDB をクローニングするには、SNAPSHOT COPY 句を指定します。SNAPSHOT COPY は、基礎となるファイル・システムで記憶域スナップショットがサポートされている場合にのみサポートされます。</p> <p>スナップショット・コピーは、ソース PDB のデータ・ファイルを完全にコピーする必要がないため、ほぼ一瞬で完了します。ただし、スナップショット・コピー PDB を CDB ルートまたはアプリケーション・ルートから切断することはできません。また、スナップショット・コピー PDB が存在する場合は、スナップショット・コピー PDB の基になっている記憶域スナップショットを削除できません。</p> <p>マテリアライズのプロセスでは、スナップショット・コピー PDB (スパース・ファイルを使用) が完全 PDB に変換されます。PDB をマテリアライズするには、ALTER PLUGGABLE DATABASE MATERIALIZE コマンドを実行します。</p>	<p>SNAPSHOT COPY 句を省略します。</p>	<p>PDB のクローニング</p>
<p>ALTER PLUGGABLE DATABASE SNAPSHOT を使用して PDB レベルのスナップショットを有効にしますか。</p>	<p>ALTER PLUGGABLE DATABASE SNAPSHOT コマンドに SNAPSHOT MODE 句を指定し、MANUAL または EVERY snapshot_interval [MINUTES HOURS] を指定します。</p>	<p>SNAPSHOT MODE 句を省略するか、SNAPSHOT MODE NONE を指定します。</p>	<p>PDB シードまたはアプリケーション・シードからの PDB の作成</p> <p>PDB のクローニング</p> <p>PDB の再配置</p> <p>プロキシ PDB の作成 (SYSTEM および SYSAUX 表領域のデータファイルにのみ適用)</p>

質問	はい	いいえ	句を使用可能な場合
			されます。) 切断された PDB の接続
単一ディレクトリ内のすべてのソース・ファイルが新しいファイル名を持ち、複数の SOURCE_FILE_NAME_CONVERT エントリが必要ですか。	ソース・ファイルへの完全絶対パスで SOURCE_FILE_DIRECTORY を指定します。	SOURCE_FILE_DIRECTORY 句を省略します。	XML ファイル・ディレクトリを使用して切断された PDB を接続します。 この句は、.pdb アーカイブ・ファイルによる切断された PDB の接続には適用されません。
XML ファイルの内容は、ソース・ファイルの場所を正確に記述していますか。	SOURCE_FILE_NAME_CONVERT 句を省略します。	SOURCE_FILE_NAME_CONVERT 句を使用して、ソース・ファイルの場所を指定します。	XML ファイル・ディレクトリを使用して切断された PDB を接続します。 この句は、.pdb アーカイブ・ファイルによる切断された PDB の接続には適用されません。
1 つ以上のスタンバイ CDB に新しい PDB を含めますか。	ALL、ALL EXCEPT またはスタンバイ CDB のリストを指定します。 リモート・クローンを作成する場合は、初期化パラメータ STANDBY_PDB_SOURCE_FILE_DBLINK を、ソース PDB データファイルを指すデータベース・リンクの名前に設定できます。データファイルは、ソース PDB が読み取り専用でオープンしている場合にのみコピーされます。	STANDBYS 句を省略するか、NONE を指定します。	PDB シードまたはアプリケーション・シードからの PDB の作成 PDB のクローニング PDB の再配置 切断された PDB の接続
PDB が使用できる記憶域の量を制限しますか。	適切な制限を使用して STORAGE 句を指定します。	STORAGE 句を省略するか、または STORAGE 句を使用して記憶域に制限がないことを指定します。	PDB シードまたはアプリケーション・シードからの PDB の作成 PDB のクローニング PDB の再配置

質問	はい	いいえ	句を使用可能な場合
			切断された PDB の接続
ターゲットの場所に一時ファイルがある場合、それを再利用しますか。	TEMPFILE REUSE 句を含めます。	TEMPFILE REUSE 句を省略します。 ターゲットの場所に新しい一時ファイルと同じ名前のファイルがないことを確認します。	PDB シードまたはアプリケーション・シードからの PDB の作成 PDB のクローニング PDB の再配置
			切断された PDB の接続
新しい PDB に含める表領域および新しい PDB から除外する表領域を指定しますか。	USER_TABLESPACES 句を含めて、新しい PDB に含める表領域を指定します。	USER_TABLESPACES 句を省略します。	切断された PDB の接続
切断された PDB を CDB に接続しますか。	USING filename 句を含めます。 Data Guard シナリオで PDB をプライマリ CDB に接続している場合は、STANDBY_PDB_SOURCE_FILE_DIRECTORY 初期化パラメータを、PDB をインスタンス化するためのソース・データファイルが含まれているスタンバイの場所に設定します。見つからない場合、スタンバイ・データベースは OMF の場所でファイルを探します。OMF の場所で見つからない場合は、データファイルを OMF の場所にコピーし、スタンバイ・データベースに対する REDO 適用を再起動します。	USING filename 句を省略します。	切断された PDB の接続
PDB スナップショットに基づいて新しい PDB を作成しますか。	USING SNAPSHOT 句を含めて、PDB スナップショット名、SCN またはタイムスタンプかを指定します。結果は、スタンドアロンの完全 PDB です。	USING SNAPSHOT 句を除外します。	PDB スナップショットのクローニング

質問	はい	いいえ	句を使用可能な場合
	<p>PDB スナップショットは PDB の Point-in-Time コピーです。スナップショットの作成中に、ソース PDB を読取り専用または読取り/書込みでオープンできます。PDB レベルのスナップショットを手動で作成するには、CREATE PLUGGABLE DATABASE (または ALTER PLUGGABLE DATABASE)の SNAPSHOT 句を指定します。EVERY interval 句を指定すると、PDB が、スナップショットを自動作成するように構成されます。</p> <p>ノート: PDB レベルのスナップショットは、記憶域管理スナップショットとは異なります。</p>		

ミラーを分割することで Oracle ASM に存在する PDB をクローニングしますか。	USING MIRROR COPY 句を含めて、ミラー・コピーの名前とソース PDB を指定します。	USING MIRROR COPY 句を省略します。	Oracle ASM 記憶域を使用する PDB のクローニング
---	--	----------------------------	---------------------------------

親トピック: [PDB作成の概要](#)

PDB作成の一般的な前提条件

PDBを作成する前に、特定の前提条件を満たす必要があります。

PDBを作成する前に、次の前提条件が満たされていることを確認してください。

表5-4 PDBを作成するための前提条件

前提条件	関連項目
CDB が存在していることが必要です。	「CDB の作成および構成」
CDB は、読取り/書込みモードである必要があります。	「PDB のオープン・モードの変更」
現行ユーザーは、現在のコンテナが CDB ルートまたはアプリケーション・コンテナである共通ユーザーである必要があります。	「CDB の共通ユーザー」
現行ユーザーには CREATE PLUGGABLE DATABASE システム権限	「共通に付与されるシステム権限の使用方法」

前提条件	関連項目
が必要です。	
各コンテナに対して一意のコンテナ名を決定する必要があります。各コンテナ名は、1 つの CDB 内で一意である必要があり、特定のリスナーからアクセスされるインスタンスを持つ CDB のスコープ内で一意である必要があります。	サービス名のルールを学習するには、 『Oracle Database Net Services リファレンス』 を参照してください
PDB 名は、PDB を CDB 内のその他の PDB と区別します。PDB 名のルールは、大文字と小文字が区別されるなど、サービス名のルールと同じです。	
フィジカル・スタンバイ・データベースがある Oracle Data Guard 構成に PDB を作成する場合は、PDB を作成する前に追加のタスクを完了する必要があります。	詳細は、 『Oracle Data Guard 概要および管理』 を参照してください
透過的データ暗号化で暗号化されたデータを含む PDB を作成している場合は、追加のタスクを完了する必要があります。	手順は、 『Oracle Database Advanced Security ガイド』 を参照してください
Database Vault に対応した PDB を作成している場合は、追加のタスクを完了する必要があります。	手順は、 『Oracle Database Vault 管理者ガイド』 を参照してください
非 CDB をクローニングして PDB を作成する場合に、ソースの非 CDB のバックアップを使用して新しい PDB をリカバリする必要があるときは、クローニングの前に DBMS_PDB.EXPORTMANBACKUP を実行する必要があります。ソース・データベースが読取り/書込みモードでオープンされている場合は、クローニングの前にプロシージャを最後のステップとして実行します。このプロシージャは、データ・ディクショナリ内のすべてのバックアップ・メタデータを取得します。	手順は、 『Oracle Database バックアップおよびリカバリ・ユーザズ・ガイド』 を参照してください
別の CDB に PDB を再配置する場合は、DBMS_PDB.EXPORTMANBACKUP を実行する必要はありません。PDB を切断すると、バックアップ・メタデータは自動的にエクスポートされます。	

関連項目:

- [『現在のコンテナについて』](#)
- DBMS_PDB.EXPORTMANBACKUPについてさらに学習するには、[『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)を参照してください

親トピック: [PDB作成の概要](#)

6 PDBを最初から作成

CREATE PLUGGABLE DATABASE文により、PDBシード(PDB\$SEED)のファイルを使用してCDBにPDBを作成します。

この文を使用すると、アプリケーション・シードまたはPDBシードのファイルを使用して、アプリケーション・コンテナにアプリケーションPDBを作成することもできます。

- [PDBを最初から作成する方法について](#)

CREATE PLUGGABLE DATABASE文により、PDBシードのファイルを使用して新しいPDBを作成したり、アプリケーション・シードまたはPDBシードのファイルからアプリケーションPDBを作成します。

- [PDBの作成](#)

CREATE PLUGGABLE DATABASE文を使用すると、PDBシードからPDBを作成したり、アプリケーション・シードまたはPDBシードからアプリケーションPDBを作成したりできます。

- [PDBの作成: 例](#)

次の例では、様々な要因に基づき、salespdbという名前の新しいPDBおよびsalesadmローカル管理者を作成します。

関連項目:

CREATE PLUGGABLE DATABASE文の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください

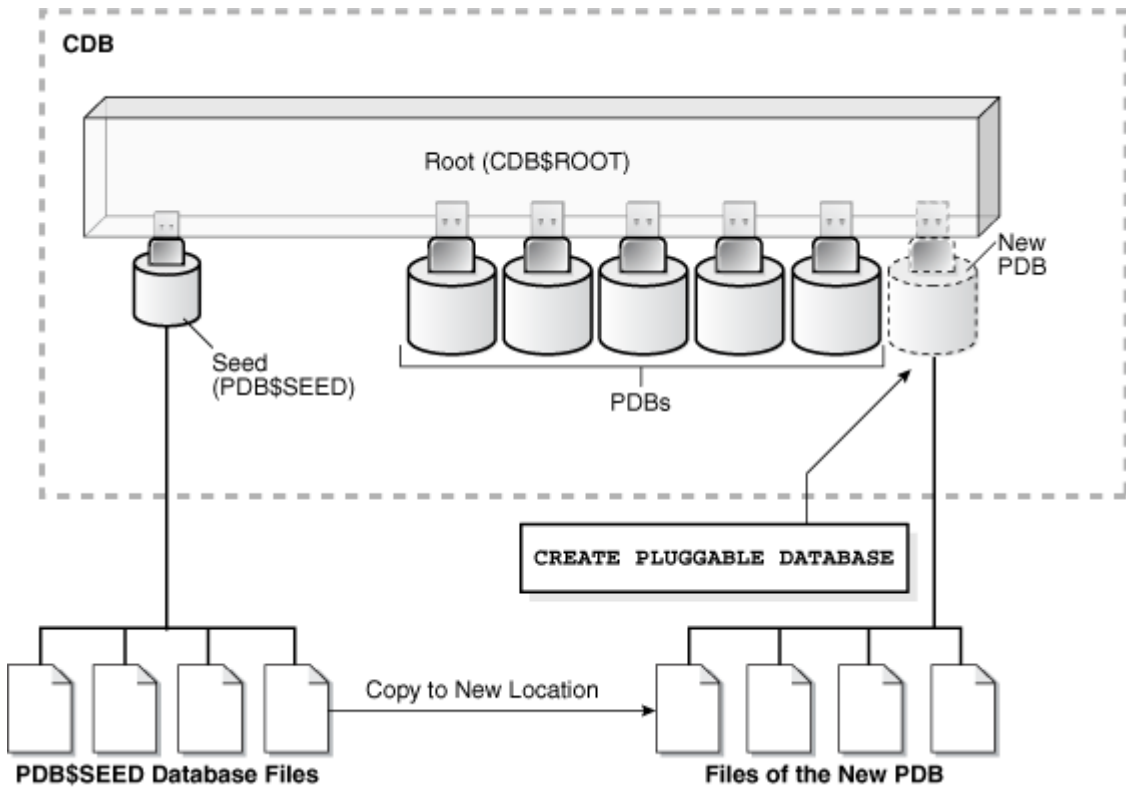
親トピック: [PDBおよびアプリケーション・コンテナの作成および削除](#)

PDBを最初から作成する方法について

CREATE PLUGGABLE DATABASE文により、PDBシードのファイルを使用して新しいPDBを作成したり、アプリケーション・シードまたはPDBシードのファイルからアプリケーションPDBを作成します。

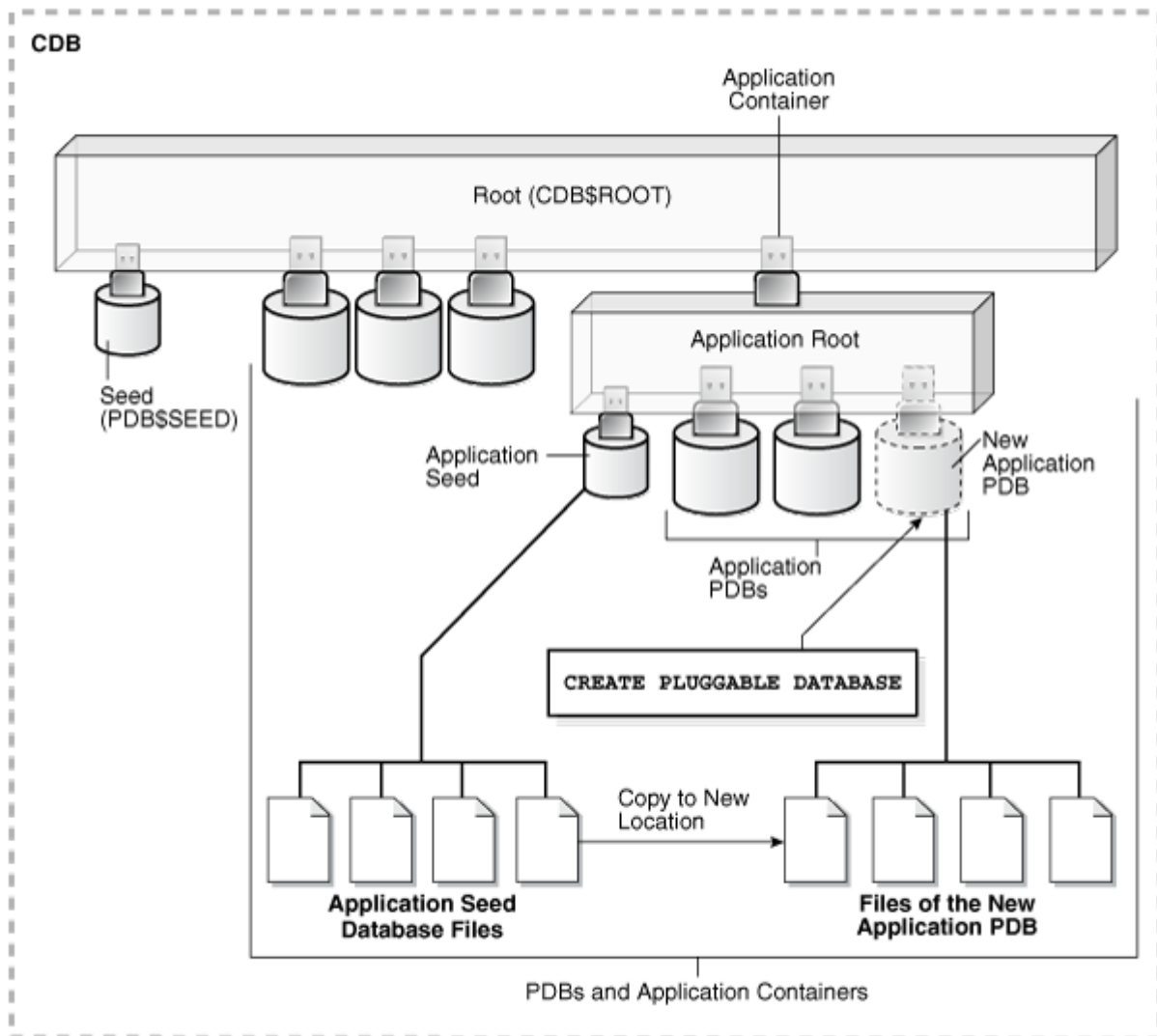
この文では、これらのファイルを新しい場所にコピーし、新しいPDBに関連付けます。次の図は、この手法でCDBルートを現在のコンテナとして使用して、新しいPDBをCDBに作成する方法を示しています。

図6-1 PDB\$SEEDファイルを使用したCDBルートでのPDBの作成



次の図は、この手法でアプリケーション・ルートを現在のコンテナとして使用して、新しいアプリケーションPDBをアプリケーション・コンテナに作成する方法を示しています。

図6-2 アプリケーション・シード・ファイルを使用したアプリケーション・ルートでのPDBの作成



関連項目:

アプリケーション・コンテナにアプリケーション・シードが含まれ、シードからアプリケーションPDBを作成するためにCREATE PLUGGABLE DATABASE文がアプリケーション・ルートで実行されている場合、アプリケーションPDBはそのアプリケーション・シードを使用して作成されます。ただし、アプリケーション・コンテナにアプリケーション・シードが含まれず、シードからアプリケーションPDBを作成するためにCREATE PLUGGABLE DATABASE文がアプリケーション・ルートで実行されている場合、アプリケーションPDBはPDBシード(PDB\$SEED)を使用して作成されます。

シードから新しいPDBまたはアプリケーションPDBを作成する場合、CREATE PLUGGABLE DATABASE文でPDBまたはアプリケーションPDBの管理者を指定する必要があります。この文では、管理者をPDBのローカル・ユーザーとして作成し、PDB_DBAロールを管理者にローカルに付与します。

PDBシードまたはアプリケーション・シードを使用してPDBを作成する前に、[表5-3](#)のシードからのPDBの作成に適用される質問に対処します。この表では、様々な要因に基づいて指定する必要があるCREATE PLUGGABLE DATABASEの句について説明しています。

関連項目:

- [「PDB記憶域」](#)
- [アプリケーションPDBの作成](#)

親トピック: [PDBを最初から作成](#)

PDBの作成

CREATE PLUGGABLE DATABASE文を使用すると、PDBシードからPDBを作成したり、アプリケーション・シードまたはPDBシードからアプリケーションPDBを作成したりできます。

前提条件

PDBシード(PDB\$SEED)からPDBを作成するか、アプリケーション・シードまたはPDBシードからアプリケーションPDBを作成する前に、[「PDB作成の一般的な前提条件」](#)で説明している前提条件を満たす必要があります。

PDBを作成するには:

1. SQL*Plusで、現在のコンテナがCDBルートまたはアプリケーション・ルートであることを確認します。
現在のコンテナがCDBルートの場合、PDBはPDBシードのファイルを使用してCDBに作成されます。
現在のコンテナがアプリケーション・ルートの場合、アプリケーションPDBはアプリケーション・シードのファイルを使用してアプリケーション・コンテナに作成されます。アプリケーション・コンテナにアプリケーション・シードがない場合、アプリケーションPDBはPDBシードのファイルを使用してアプリケーション・コンテナに作成されます。
2. CREATE PLUGGABLE DATABASE文を実行し、PDBのローカル管理者を指定します。他の句が必要な場合はそれらを指定します。
作成したPDBはマウント・モードになり、そのステータスはNEWです。PDBのオープン・モードは、V\$PDBSビューのOPEN_MODE列を問い合わせることで確認できます。CDB_PDBSまたはDBA_PDBSビューのSTATUS列を問い合わせ、PDBのステータスを表示できます。

PDBの新しいデフォルト・サービスが作成されます。サービスの名前はPDBと同じになり、このサービスを使用してPDBに

アクセスできます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。

3. 新しいPDBを読み取り/書き込みモードでオープンします。

Oracle Databaseで新しいPDBをCDBに統合するには、PDBを読み取り/書き込みモードでオープンする必要があります。PDBを読み取り専用モードでオープンしようとすると、エラーが返されます。PDBが読み取り/書き込みモードでオープンされると、そのステータスはNORMALになります。

4. PDBをバックアップします。

PDBは、バックアップしないとリカバリできません。

指定されたローカル管理者の名前を持つローカル・ユーザーが作成され、このユーザーにPDBでローカルにPDB_DBA共通ロールが付与されます。PDBの作成時にユーザーに管理者権限が付与されなかった場合は、SYSおよびSYSTEM共通ユーザーを使用してPDBを管理します。

ノート:



PDBの作成時にエラーが返された場合は、作成されたPDBがUNUSABLE状態である可能性があります。CDB_PDBSまたはDBA_PDBSビューを問い合わせるPDBの状態を確認したり、アラート・ログをチェックしてPDBの作成エラーについてさらに学習できます。使用不可のPDBは削除することのみができ、使用不可のPDBと同じ名前のPDBを作成できるようにするには、その前に削除する必要があります。

関連項目:

- [「CDB内のコンテナへのアクセスについて」](#)
- 詳細は、[「PDBのオープン・モードの変更」](#)を参照してください。
- PDBのバックアップの詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』](#)を参照してください。

親トピック: [PDBを最初から作成](#)

PDBの作成: 例

次の例では、様々な要因に基づき、salespdbという名前の新しいPDBおよびsalesadmローカル管理者を作成します。

salespdb PDBの作成に加えて、この文では、PDB_DBAロールをPDB管理者であるsalesadmに付与し、指定したOracleの事前定義済ロールをPDB_DBAロールにPDBでローカルに付与します。

各例で、新しいPDBが属するルートはCREATE PLUGGABLE DATABASE文の実行時の現在のコンテナによって決まります。

- 現在のコンテナがCDBルートの場合、新しいPDBはCDBルートに作成されます。
- 現在のコンテナがアプリケーション・コンテナ内のアプリケーション・ルートの場合、新しいPDBはアプリケーション・ルート内にアプリケーションPDBとして作成されます。
- [句を使用しないPDBの作成: 例](#)
この例は、PDBを作成する最も簡単な方法を示しています。

- [PDBの作成、およびPDB管理者へのOracleの事前定義されたロールの付与：例](#)
この例では、ROLESパラメータを使用して、事前定義済のロールを付与します。
- [複数の句を使用したPDBの作成：例](#)
この例では、STORAGE、DEFAULT TABLESPACE、PATH_PREFIXおよびFILE_NAME_CONVERTの各句を使用してPDBを作成します。

親トピック: [PDBを最初から作成](#)

句を使用しないPDBの作成：例

この例は、PDBを作成する最も簡単な方法を示しています。

この例では、次の要因を想定しています。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- PDBにはデフォルトの表領域は必要ありません。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

CDBに対してOracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。PDBシードまたはアプリケーション・シードに関連付けられているファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。
- Oracleの事前定義されたロールをPDB_DBAロールに付与する必要はありません。

次の文は、PDBを作成します。

```
CREATE PLUGGABLE DATABASE salespdb ADMIN USER salesadm IDENTIFIED BY pwd;
```

関連項目:

- Oracle Managed Filesの使用の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください
- PDB_FILE_NAME_CONVERT初期化パラメータの詳細は、『[Oracle Databaseリファレンス](#)』を参照してください
- パスワード選択のガイドラインは、『[Oracle Databaseセキュリティ・ガイド](#)』を参照してください

親トピック: [PDBの作成：例](#)

PDBの作成、およびPDB管理者へのOracleの事前定義されたロールの付与：例

この例では、ROLESパラメータを使用して、事前定義済のロールを付与します。

この例では、次の要因を想定しています。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- PDBにはデフォルトの表領域は必要ありません。
- PATH_PREFIX句は必要ありません。

- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。PDBシードまたはアプリケーション・シードに関連付けられているファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。
- PDB_DBAロールには、事前定義されたDBA Oracleロールがローカルで付与されている必要があります。

次の文は、PDBを作成します。

```
CREATE PLUGGABLE DATABASE salespdb
  ADMIN USER salesadm IDENTIFIED BY password
  ROLES= (DBA) ;
```

関連項目:

- Oracle Managed Filesの使用の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください
- PDB_FILE_NAME_CONVERT初期化パラメータの詳細は、『[Oracle Databaseリファレンス](#)』を参照してください
- パスワード選択のガイドラインは、『[Oracle Databaseセキュリティ・ガイド](#)』を参照してください

親トピック: [PDBの作成: 例](#)

複数の句を使用したPDBの作成: 例

この例では、STORAGE、DEFAULT TABLESPACE、PATH_PREFIXおよびFILE_NAME_CONVERTの各句を使用してPDBを作成します。

この例では、次の要因を想定しています。

- PDBにストレージ制限を強制する必要があります。したがって、STORAGE句が必要です。特に、PDBに属する表領域はいずれも2GBを超えることはできません。
- 非管理者ユーザーに対してそれぞれ異なる永続表領域を指定しない場合は、デフォルトの永続表領域が必要となります。特に、この例では次の特性を備えたsalesという名前のデフォルト永続表領域を作成します。
 - この表領域の単一データファイルはsales01.dbfであり、文では、このファイルが/disk1/oracle/dbs/salespdbディレクトリに作成されます。
 - SIZE句では、表領域の初期サイズが250MBであることを指定します。
 - AUTOEXTEND句により、ファイルを自動拡張できます。
- パス接頭辞をPDBのディレクトリ・オブジェクト・パスに追加する必要があります。したがって、PATH_PREFIX句が必要です。この例では、パス接頭辞/disk1/oracle/dbs/salespdb/をPDBのディレクトリ・オブジェクト・パスに追加しています。
- CREATE_FILE_DEST句が使用されず、Oracle Managed Filesが有効になっておらず、PDB_FILE_NAME_CONVERT初期化パラメータが設定されていません。したがって、FILE_NAME_CONVERT句が必要です。システムにおけるPDBシードまたはアプリケーション・シードのデータファイルの場所を指定します。この例では、Oracle Databaseは/disk1/oracle/dbs/pdbseedのファイルを/disk1/oracle/dbs/salespdbにコピーします。

- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。
- Oracleの事前定義されたロールをPDB_DBAロールに付与する必要はありません。

次の文は、PDBを作成します。

```
CREATE PLUGGABLE DATABASE salespdb
ADMIN USER salesadm IDENTIFIED BY password
STORAGE (MAXSIZE 2G)
DEFAULT TABLESPACE sales
DATAFILE '/disk1/oracle/dbs/salespdb/sales01.dbf' SIZE 250M
AUTOEXTEND ON
PATH_PREFIX = '/disk1/oracle/dbs/salespdb/'
FILE_NAME_CONVERT = ('/disk1/oracle/dbs/pdbseed/',
'/disk1/oracle/dbs/salespdb/');
```

関連項目:

- PDBシードまたはアプリケーション・シードのデータファイルの場所を表示する方法を学習するには、[「例19-7」](#)を参照してください
- DEFAULT TABLESPACE句の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください
- パスワード選択のガイドラインは、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください

親トピック: [PDBの作成: 例](#)

7 PDBまたは非CDBのクローニング

ローカルPDB、リモートPDBまたは非CDBをクローニングすることで、PDBを作成できます。

- [PDBまたは非CDBクローニングについて](#)
クローニングとは、ソースPDBまたは非CDBから新しいPDBを作成することです。
- [ローカルPDBのクローニング](#)
ローカルPDBのクローンは、CREATE PLUGGABLE DATABASE文を実行して、FROM文でローカルPDBを指定することで作成できます。
- [リモートPDBのクローニング](#)
ローカルPDBのクローンは、CREATE PLUGGABLE DATABASE文を実行し、FROM文でリモートPDBへのデータベースへのリンクを指定することで作成できます。
- [非CDBのクローニング](#)
非CDBのクローニング手順は、リモートPDBのクローニング手順と非常によく似ています。
- [リフレッシュ可能なクローンPDBについて](#)
CREATE PLUGGABLE DATABASE ... REFRESH MODE文は、ソースPDBをクローニングし、クローンをリフレッシュ可能に構成します。クローンPDBをリフレッシュすると、最後のREDOログの適用以降に蓄積されたREDOでクローンPDBが更新されます。
- [PDBスナップショットからのPDBのクローニング](#)
PDBスナップショットからPDBを作成するには、CREATE PLUGGABLE DATABASE ... USING SNAPSHOT文を実行します。
- [スナップショット・コピーPDBの作成およびマテリアライズ](#)
PDBは基礎となる記憶域のスナップショットからクローニングできます。PDBファイルはスパースですが、それらのファイルをマテリアライズしてスタンドアロンPDBを作成できます。
- [分割ミラーによるクローンPDBの作成](#)
Oracle ASMでは、分割ミラーとは親コピーからPoint-in-Timeメディア・コピーをデタッチするプロセスのことです。分割後、親コピーに対する更新は子コピーに影響しません。

親トピック: [PDBおよびアプリケーション・コンテナの作成および削除](#)

PDBまたは非CDBのクローニングについて

クローニングとは、ソースPDBまたは非CDBから新しいPDBを作成することです。

一般的なユースケースは開発テストです。PDBまたは非CDBの1つ以上のクローンを作成して、それらのクローンを分離して安全にテストできます。たとえば、新しいアプリケーションまたは変更されたアプリケーションを、本番PDBで使用する前に、クローニングPDBでテストできます。

- [クローニングの仕組み](#)
この手法では、ソースPDBまたは非CDBから新しいPDBを作成します。このプロセスでは、新しいPDBを自動的にCDBに接続します。
- [PDBクローニングのユーザー・インタフェース](#)
すべての形式のPDBクローニングに、CREATE PLUGGABLE DATABASE文を使用します。

関連項目:

暗号化データまたはキーストア・セットを持つソースのクローニングについて学習するには、[『Oracle Database Advanced』](#)

親トピック: [PDBまたは非CDBのクローニング](#)

クローニングの仕組み

この手法では、ソースPDBまたは非CDBから新しいPDBを作成します。このプロセスでは、新しいPDBを自動的にCDBに接続します。

この手法を使用するには、CREATE PLUGGABLE DATABASE文でソースを指定する必要があります。ソースは次のいずれかです。

- ローカルPDB
- リモートCDB内のPDB
- 非CDB

ターゲットPDBは、ソースPDBまたは非CDBのコピーです。このコピーのことをクローンPDBと呼びます。

CREATE PLUGGABLE DATABASE文では、ソースに関連付けられているファイルを新しい場所にコピーし、そのファイルをターゲットPDBに関連付けます。CDBがARCHIVELOGモードおよびローカルUNDOモードの場合、ソースPDBは読み取り/書き込みモードでオープンでき、クローニング・プロセス中に機能します。この手法は、**ホット・クローニング**と呼ばれます。

ノート:

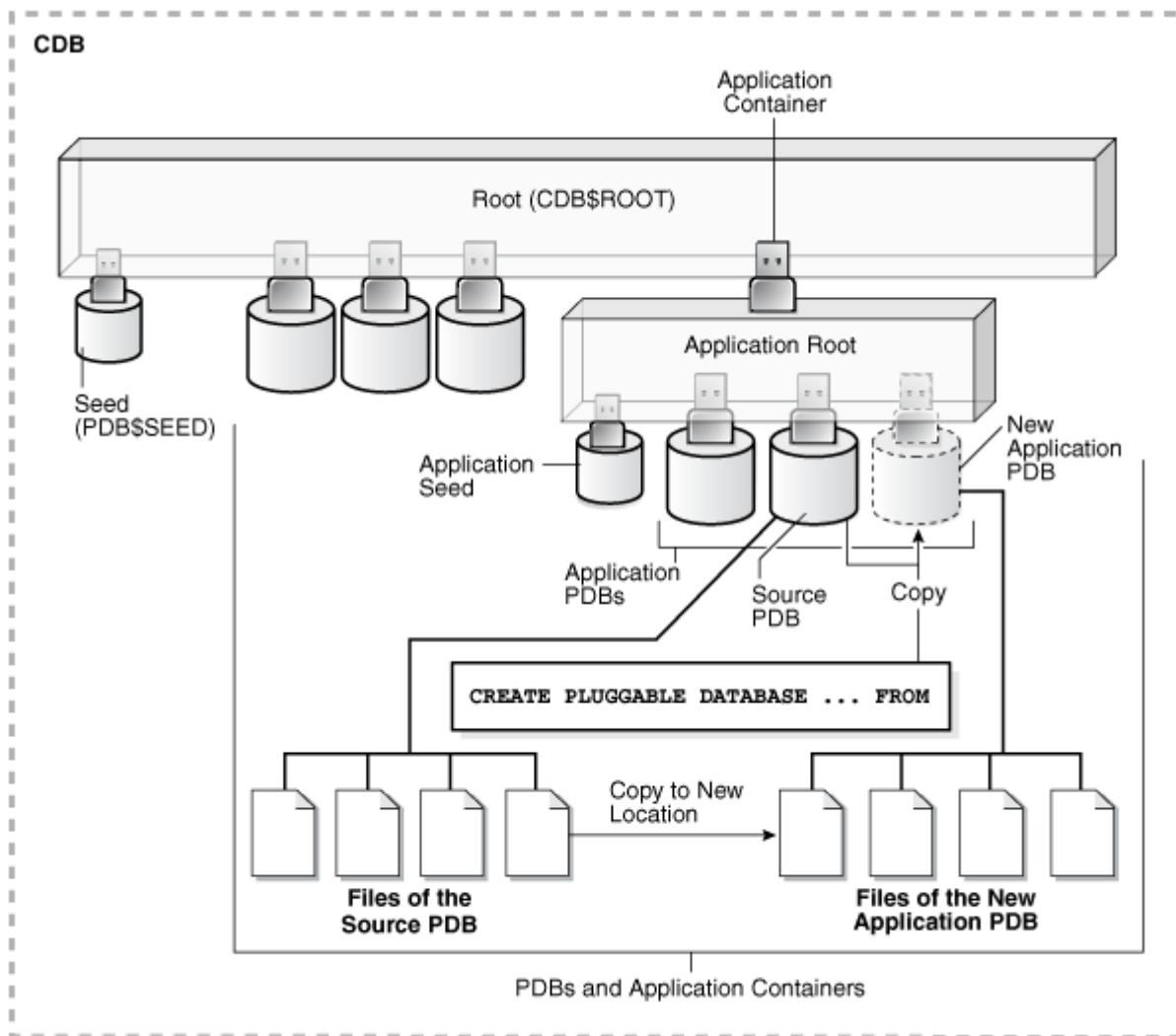


PDB をクローニングするときに、ソース・データベースに暗号化されたデータまたはキーストア・セットがある場合、CREATE PLUGGABLE DATABASE ... FROM SQL 文の keystore_password 句によって識別されるキーストアを含めることによって、キーストアのパスワードを提供する必要があります。このパスワードは、ソース・データベースが自動ログイン・ソフトウェア・キーストアを使用している場合であっても指定する必要があります。ソース・データベースに暗号化データまたはキーストアがあるかどうかを判別するには、DBA_ENCRYPTED_COLUMNS データ・ディクショナリ・ビューに問い合わせます。

すべてのクローニングのシナリオで、CREATE PLUGGABLE DATABASE文をアプリケーション・ルートで実行するときには、アプリケーション・コンテナ内にクローンPDBを作成します。ソースPDBのアプリケーション名およびバージョンは、アプリケーション・コンテナのアプリケーション名およびバージョンと一致する必要があります。

次の図に、この手法でローカル・ソース・アプリケーションPDBをクローニングすることで新しいアプリケーションPDBをアプリケーション・コンテナに作成する方法を示します。ソースPDBは、ローカルCDBルートに接続されたPDB、リモートCDBルートに接続されたPDBまたはリモート・アプリケーション・ルートに接続されたアプリケーションPDBにすることができます。

図7-1 アプリケーション・コンテナ内のPDBのクローニング



関連項目:

[「PDB記憶域」](#)

親トピック: [PDBまたは非CDBのクローニングについて](#)

PDBクローニングのユーザー・インタフェース

すべての形式のPDBクローニングに、CREATE PLUGGABLE DATABASE文を使用します。

クローニングの際には、FROM句でソースPDBを指定する必要があります。次の表に、最も重要な句のまとめを示します。

表7-1 PDBクローニングのためのCREATE PLUGGABLE DATABASEオプション

句	クローニング操作	関連項目
USING SNAPSHOT	PDB レベルのスナップショットからクローンを作成します(ALTER PLUGGABLE DATABASE SNAPSHOT)。PDB スナップショット名、SCN またはタイムスタンプを指定します。 ノート: USING SNAPSHOT 句と SNAPSHOT COPY 句の両方を使用して PDB スナップショットに基づいてスナップショット・コピーPDB を作成することはできないということに注意してください。ただし、USING SNAPSHOT を使用	PDB スナップショットからのクローン

句	クローニング操作	関連項目
	して PDB スナップショットに基づいてクローンを作成し、クローンから SNAPSHOT COPY PDB を作成できます。	
REFRESH MODE	リフレッシュ可能クローン PDB を作成します。	リフレッシュ可能クローン PDB
SNAPSHOT COPY	<p>記憶域管理スナップショットからスナップショット・コピー PDB を作成します (ALTER PLUGGABLE DATABASE SNAPSHOT ではない)。記憶域管理スナップショットは、特定のファイル・システムでのみサポートされています。</p> <p>スナップショット・コピー PDB には、ソース・データ・ファイルの完全なコピーは含まれていません。かわりに、基礎となるファイル・システムの記憶域レベルのスナップショットが作成され、そのスナップショットからクローン PDB が作成されます。</p> <p>標準のクローン PDB とは異なり、スナップショット・コピー PDB は記憶域スナップショットに依存しています。そのため、スナップショット・コピー PDB は、CDB ルートから切断することも、アプリケーション・ルートに接続することもできません。また、その PDB が基づいている記憶域スナップショットを削除することもできません。そのかわりに、スナップショット・コピー PDB をマテリアライズする必要があります。これにより、非スパース・ファイルによる完全 PDB に変換します。</p>	スナップショット・コピー PDB
USING MIRROR COPY	mirror_name で指定されている ASM 記憶域ミラーを分割することで、新しい PDB を作成します。準備済ミラー・コピーからは 1 つの PDB のみを分割できます。追加の分割を作成する場合は、新しいミラー・コピーを準備する必要があります。	「分割ミラーによるクローン PDB の作成」

関連項目:

- [「スナップショット・コピー PDB のマテリアライズ」](#)
- CREATE PLUGGABLE DATABASE 句についてさらに学習するには、[『Oracle Database SQL 言語リファレンス』](#)を参照

親トピック: [PDB または非 CDB のクローニングについて](#)

ローカル PDB のクローニング

ローカル PDB のクローンは、CREATE PLUGGABLE DATABASE 文を実行して、FROM 文でローカル PDB を指定することで作成できます。

- [ローカル PDB のクローニングについて](#)
クローニングの最も簡単な形式では、PDB を CDB から同じ CDB にコピーします。

- [ローカルPDBのクローニング: 基本的なステップ](#)

ローカルPDBのクローンは、CREATE PLUGGABLE DATABASEを実行して、FROM句でソースPDBを指定することで作成できます。

- [ローカルPDBのクローニング後](#)

ローカルPDBのクローニング後には、ユーザーと表領域に関する特定のルールが適用されます。

- [ローカルPDBのクローニング: 例](#)

次の例では、様々な要因に基づき、pdb1という名前のローカル・ソースPDBを、pdb2という名前のターゲットPDBにクローニングします。

親トピック: [PDBまたは非CDBのクローニング](#)

ローカルPDBのクローニングについて

クローニングの最も簡単な形式では、PDBをCDBから同じCDBにコピーします。

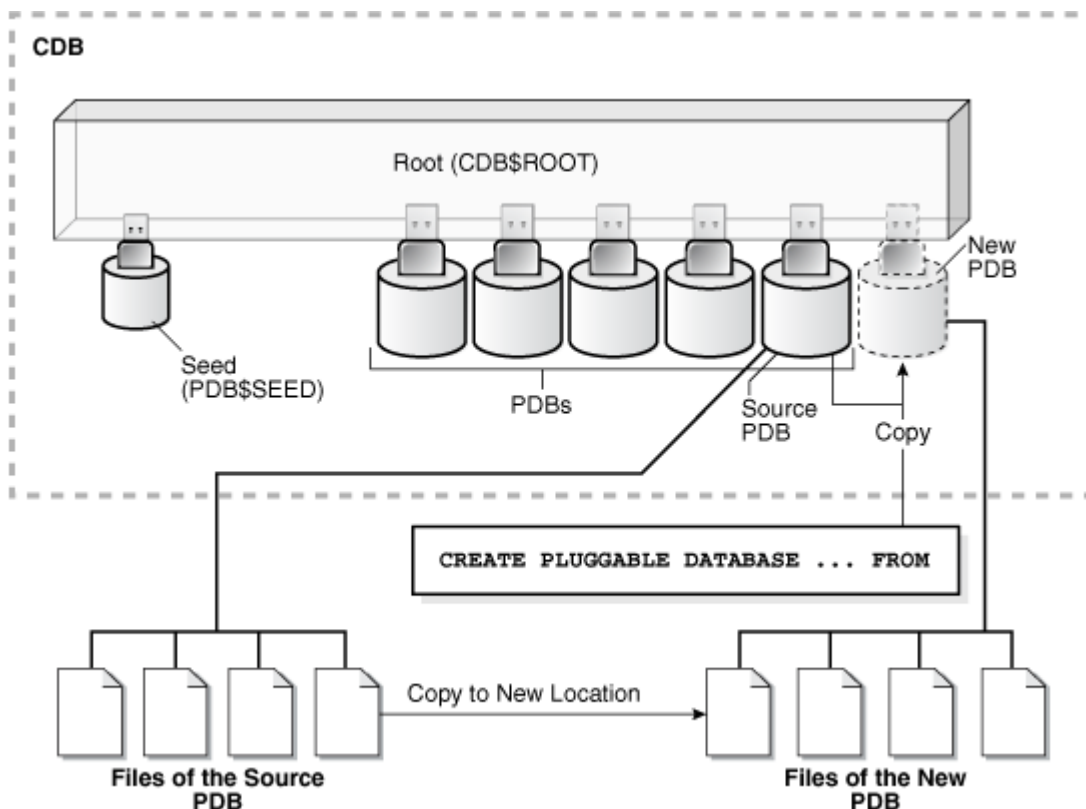
ノート:



PDBをPDBシード(PDB\$SEED)またはアプリケーション・シードから作成する場合に、CREATE PLUGGABLE DATABASE文にFROM句を含めることはできません。

次の図は、ローカルPDBをクローニングする方法を示しています。

図7-2 ローカルPDBのクローニング



PDBをクローニングする前に、[表5-3](#)のPDBのクローニングに適用される質問に対処します。この表では、様々な要因に基づいて指定するCREATE PLUGGABLE DATABASEの句について説明しています。

Oracle Database 18c以降では、DBCAを使用してローカルPDBをクローニングできます。

関連項目:

- [「現在のコンテナのIDまたは名前の判別」](#)
- シードからPDBを作成する方法を学習するには、[「PDBを最初から作成」](#)を参照してください
- [「DBCAを使用したローカルPDBのクローニング: 例」](#)
- [アプリケーションPDBの作成](#)

親トピック: [ローカルPDBのクローニング](#)

ローカルPDBのクローニング: 基本的なステップ

ローカルPDBのクローンは、CREATE PLUGGABLE DATABASEを実行して、FROM句でソースPDBを指定することで作成できます。

前提条件

次の前提条件を満たしている必要があります。

- [「PDB作成の一般的な前提条件」](#)で説明している前提条件を満たします。
- 現行ユーザーは、ルートおよびソースPDBの両方でCREATE PLUGGABLE DATABASEシステム権限を持っている必要があります。
- ソースPDBはクローズできません。
- CDBがローカルUNDOモードでない場合、ソースPDBは読み取り専用オープン・モードである必要があります。この要件は、CDBがローカルUNDOモードの場合は適用されません。
- CDBがARCHIVELOGモードでない場合、ソースPDBは読み取り専用オープン・モードである必要があります。この要件は、CDBがARCHIVELOGモードの場合は適用されません。
- アプリケーションPDBを作成している場合、アプリケーションPDBはアプリケーション・コンテナと同じ文字セットおよび各国語文字セットである必要があります。

CDBのデータベース文字セットがAL32UTF8の場合、アプリケーション・コンテナの文字セットおよび各国語文字セットはCDBと異なっても構いません。ただし、アプリケーション・コンテナ内のすべてのアプリケーションPDBには、アプリケーション・コンテナと同じ文字セットおよび各国語文字セットが必要です。

ノート:



REFRESH MODE 句を使用してローカル PDB のリフレッシュ可能なクローンを作成できるのは、データベース・リンクが同じ CDB にループ・バックしている場合のみです。

ローカルPDBをクローニングするには:

1. SQL*Plusで、現在のコンテナがCDBルートまたはアプリケーション・ルートであることを確認します。
現在のコンテナがCDBルートの場合、PDBはCDBに作成されます。現在のコンテナがアプリケーション・ルートの場合、アプリケーションPDBはアプリケーション・コンテナ内に作成されます。
2. CREATE PLUGGABLE DATABASE文を実行し、ソースPDBをFROM句で指定します。必要に応じて、他の句を指定します。

ローカルPDBのクローニング後、ソースPDBとターゲットPDBは同じCDBに存在します。新しいPDBはマウント・モードになっていて、そのステータスはNEWになります。PDBのオープン・モードは、V\$PDBSビューのOPEN_MODE列を問い合わせることで確認できます。CDB_PDBSまたはDBA_PDBSビューのSTATUS列を問い合わせ、PDBのステータスを表示できます。

PDBの新しいデフォルト・サービスが作成されます。サービスの名前はPDBと同じになり、このサービスを使用してPDBにアクセスできます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。

3. 新しいPDBを読取り/書込みモードでオープンします。

Oracle Databaseで新しいPDBをCDBに統合するには、PDBを読取り/書込みモードでオープンする必要があります。PDBを読取り専用モードでオープンしようとすると、エラーが返されます。PDBが読取り/書込みモードでオープンされると、そのステータスはNORMALになります。

4. 新しいPDBのバックアップを作成します。

PDBは、バックアップしないとリカバリできません。

ノート:



PDBの作成時にエラーが返された場合は、作成されたPDBがUNUSABLE状態である可能性があります。PDBの状態は、CDB_PDBSまたはDBA_PDBSビューを問い合わせることで確認できます。PDB作成のエラーについてさらに学習するには、アラート・ログを確認してください。使用不可のPDBのみを削除できます。また、使用不可のPDBを削除してからでないと、同じ名前のPDBを作成できません。

関連項目:

- [「現在のコンテナについて」](#)および[「CDB内のコンテナへのアクセスについて」](#)
- [「CDBのUNDOモードについて」](#)
- [「PDBのオープン・モードの変更」](#)
- PDBのバックアップ方法を学習するには、[『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』](#)を参照してください

親トピック: [ローカルPDBのクローニング](#)

ローカルPDBのクローニング後

ローカルPDBのクローニング後には、ユーザーと表領域に関する特定のルールが適用されます。

ソースPDBのデフォルト一時表領域を使用していた新しいPDBのユーザーは、新しいPDBのデフォルト一時表領域を使用します。PDB内のデフォルト以外の一時表領域を使用していたユーザーは、クローニングされたPDB内の同じローカル一時表領域を引き続き使用します。

関連項目:

[「CDB内表領域の管理について」](#)

親トピック: [ローカルPDBのクローニング](#)

ローカルPDBのクローニング: 例

次の例では、様々な要因に基づき、pdb1という名前のローカル・ソースPDBを、pdb2という名前のターゲットPDBにクローニングします。

各例で、新しいPDBが属するルートはCREATE PLUGGABLE DATABASE文の実行時の現在のコンテナによって決まります。

- 現在のコンテナがCDBルートの場合、データベースではCDBルートにPDBが作成されます。
- 現在のコンテナがアプリケーション・コンテナ内のアプリケーション・ルートの場合、データベースではアプリケーション・ルート内にアプリケーションPDBが作成されます。
- [句を使用しないローカルPDBのクローニング: 例](#)
この例では、PDBをクローニングする最も簡単な方法を示しています。
- [DBCAを使用したローカルPDBのクローニング: 例](#)
この例では、DBCAのサイレント・モードを使用してPDBをクローニングします。ホット・クローニングがサポートされます。
- [PATH_PREFIX句を使用したローカルPDBのクローニング: 例](#)
この例では、PATH_PREFIX、FILE_NAME_CONVERTおよびSERVICE_NAME_CONVERT句を使用してローカルPDBをクローニングする方法を説明しています。
- [STORAGE句を使用したローカルPDBのクローニング: 例](#)
この例では、FILE_NAME_CONVERT、STORAGEおよびSERVICE_NAME_CONVERT句を使用してローカルPDBをクローニングします。
- [NO DATA句を使用したローカルPDBのクローニング: 例](#)
この例では、PDBのデータ・モデル定義をクローニングしますが、PDBのデータはクローニングしません。

親トピック: [ローカルPDBのクローニング](#)

句を使用しないローカルPDBのクローニング: 例

この例では、PDBをクローニングする最も簡単な方法を示しています。

この例では、次の要因を想定しています。

- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。
Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。したがって、FILE_NAME_CONVERT句は必要ありません。これらのファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。
- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

次の文は、pdb1 PDBからpdb2 PDBをクローニングします。

```
CREATE PLUGGABLE DATABASE pdb2 FROM pdb1;
```

関連項目:

- Oracle Managed Filesの詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- PDB_FILE_NAME_CONVERT初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください

親トピック: [ローカルPDBのクローニング: 例](#)

DBCAを使用したローカルPDBのクローニング: 例

この例では、DBCAのサイレント・モードを使用してPDBをクローニングします。ホット・クローニングがサポートされます。

この例では、次の要因を想定しています。

- ソースCDBは、SIDがorclの単一インスタンス・データベースです。
- ソースPDBはpdb1です。クローニング操作中、pdb1はオープンのままにします。これは、ローカルUNDOおよびARCHIVELOGモードがCDBで有効になっていることを意味します。それ以外の場合、DBCAはクローン操作中にPDBをクローズし、確認を受け取った後に、ソースPDBを読取り専用モードでオープンします。
- 新しいPDBはpdb2です。
- DBCAを非対話型モードで実行しています。

次のコマンドは、pdb1 PDBからpdb2 PDBをクローニングします。

```
./dbca -silent
  -createpluggabledatabase
  -sourcedb orcl
  -createpdbfrom PDB
  -pdbName pdb2
  -sourcepdb pdb1
```

関連項目:

DBCAのコマンド・リファレンスは、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [ローカルPDBのクローニング: 例](#)

PATH_PREFIX句を使用したローカルPDBのクローニング: 例

この例では、PATH_PREFIX、FILE_NAME_CONVERTおよびSERVICE_NAME_CONVERT句を使用してローカルPDBをクローニングする方法を説明しています。

この例では、次の要因を想定しています。

- パス接頭辞をPDBのディレクトリ・パスに追加する必要があります。したがって、PATH_PREFIX句が必要です。この例では、パス接頭辞/disk2/oracle/pdb2をPDBのディレクトリ・オブジェクト・パスに追加しています。
- コピーしたファイルのターゲットの場所を指定するには、FILE_NAME_CONVERT句が必要です。この例では、ファイルは/disk1/oracle/pdb1から/disk2/oracle/pdb2にコピーされます。
CREATE_FILE_DEST句は使用されず、Oracle Managed FilesもPDB_FILE_NAME_CONVERT初期化パラメータも、コピーしたファイルのターゲットの場所の指定に使用されません。
PDBのデータファイルの場所を表示するには、[\[例19-7\]](#)の問合せを実行します。
- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。

- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。
- クローニングされるPDB (pdb1)には、カリフォルニア州の営業担当者およびオーダー入力担当者のための2つのユーザー定義のサービス(salesrep_caおよびorders_ca)があります。新しいサービスはオレゴン州の営業担当者およびオーダー入力担当者のためのものであり、クローニングされたPDB (pdb2)でサービス名はsalesrep_orおよびorders_orにそれぞれ名前変更されます。
- PDB内で今後作成される表領域は、デフォルトでNOLOGGING属性を使用して作成されます。この機能は、Oracle Database 12cリリース1 (12.1.0.2)以降で使用可能です。

次の文は、pdb1 PDBからpdb2 PDBをクローニングします。

```
CREATE PLUGGABLE DATABASE pdb2 FROM pdb1
  PATH_PREFIX = '/disk2/oracle/pdb2/'
  FILE_NAME_CONVERT = ('/disk1/oracle/pdb1/', '/disk2/oracle/pdb2/')
  SERVICE_NAME_CONVERT = ('salesrep_ca', 'salesrep_or', 'orders_ca', 'orders_or')
  NOLOGGING;
```

親トピック: [ローカルPDBのクローニング: 例](#)

STORAGE句を使用したローカルPDBのクローニング: 例

この例では、FILE_NAME_CONVERT、STORAGEおよびSERVICE_NAME_CONVERT句を使用してローカルPDBをクローニングします。

この例では、次の要因を想定しています。

- PATH_PREFIX句は必要ありません。
- コピーしたファイルのターゲットの場所を指定するには、FILE_NAME_CONVERT句が必要です。この例では、ファイルは/disk1/oracle/pdb1から/disk2/oracle/pdb2にコピーされます。

CREATE_FILE_DEST句は使用されず、Oracle Managed FilesもPDB_FILE_NAME_CONVERT初期化パラメータも、コピーしたファイルのターゲットの場所の指定に使用されません。

PDBのデータファイルの場所を表示するには、[例19-7](#)の問合せを実行します。
- PDBにストレージ制限を強制する必要があります。したがって、STORAGE句が必要です。特に、PDBに属する表領域はいずれも2GBを超えることはできません。
- ソースPDB (pdb1)には、カリフォルニア州の営業担当者およびオーダー入力担当者のための2つのユーザー定義のサービス(salesrep_caおよびorders_ca)があります。新しいサービスはオレゴン州の営業担当者およびオーダー入力担当者のためのものであり、クローニングされたPDB (pdb2)でサービス名はsalesrep_orおよびorders_orにそれぞれ名前変更されます。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

次の文は、pdb1 PDBからpdb2 PDBをクローニングします。

```
CREATE PLUGGABLE DATABASE pdb2 FROM pdb1
  FILE_NAME_CONVERT = ('/disk1/oracle/pdb1/', '/disk2/oracle/pdb2/')
  STORAGE (MAXSIZE 2G)
  SERVICE_NAME_CONVERT = ('salesrep_ca', 'salesrep_or', 'orders_ca', 'orders_or');
```

親トピック: [ローカルPDBのクローニング: 例](#)

NO DATA句を使用したローカルPDBのクローニング: 例

この例では、PDBのデータ・モデル定義をクローニングしますが、PDBのデータはクローニングしません。

この例では、次の要因を想定しています。

- NO DATA句は、目的がデータをクローニングせずにソースPDBのデータ・モデル定義をクローニングすることであるため必要です。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。したがって、FILE_NAME_CONVERT句は必要ありません。プロセスでは、Oracle Managed Files構成または初期化パラメータ設定に基づいて、ファイルが新しい場所にコピーされます。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

ソースPDB pdb1にデータが大量にあるとします。次のステップでは、操作の完了時にソースPDBのデータがクローンに含まれない動作の内容を示します。

1. ソースPDB pdb1を現在のコンテナとして使用し、データが大量にある表を問い合わせます。

```
SELECT COUNT(*) FROM tpch.lineitem;
```

COUNT(*)
60001215

表には、6000万超の行があります。

2. NO DATA句を指定して、ソースPDBをクローニングします。

```
CREATE PLUGGABLE DATABASE pdb2 FROM pdb1 NO DATA;
```

3. クローニングしたPDBをオープンします。

```
ALTER PLUGGABLE DATABASE pdb2 OPEN;
```

4. クローニングしたPDB pdb2を現在のコンテナとして使用し、ソースPDB内のデータが大量にある表を問い合わせます。

```
SELECT COUNT(*) FROM tpch.lineitem;
```

COUNT(*)
0

クローニングしたPDB内の表には行がありません。

親トピック: [ローカルPDBのクローニング: 例](#)

リモートPDBのクローニング

ローカルPDBのクローンは、CREATE PLUGGABLE DATABASE文を実行し、FROM文でリモートPDBへのデータベースへのリンクを指定することで作成できます。

- [リモートPDBのクローニングについて](#)
ソースがリモートCDB内のPDBである場合は、データベース・リンクを使用してPDBをローカルCDBにクローニングする必要があります。
- [リモートPDBのクローニング: 基本的なステップ](#)
PDBはリモートPDBをクローニングすることで作成できます。クローニング操作後、ソースとターゲットPDBは異なる場所にあります。
- [リモートPDBのクローニング後](#)
リモートPDBのクローニング後には、ユーザーおよび表領域に関する特定のルールが適用されます。
- [リモートPDBのクローニング: 例](#)
これらの例では、様々な要因に基づき、リモートPDBまたは非CDBをクローニングします。

親トピック: [PDBまたは非CDBのクローニング](#)

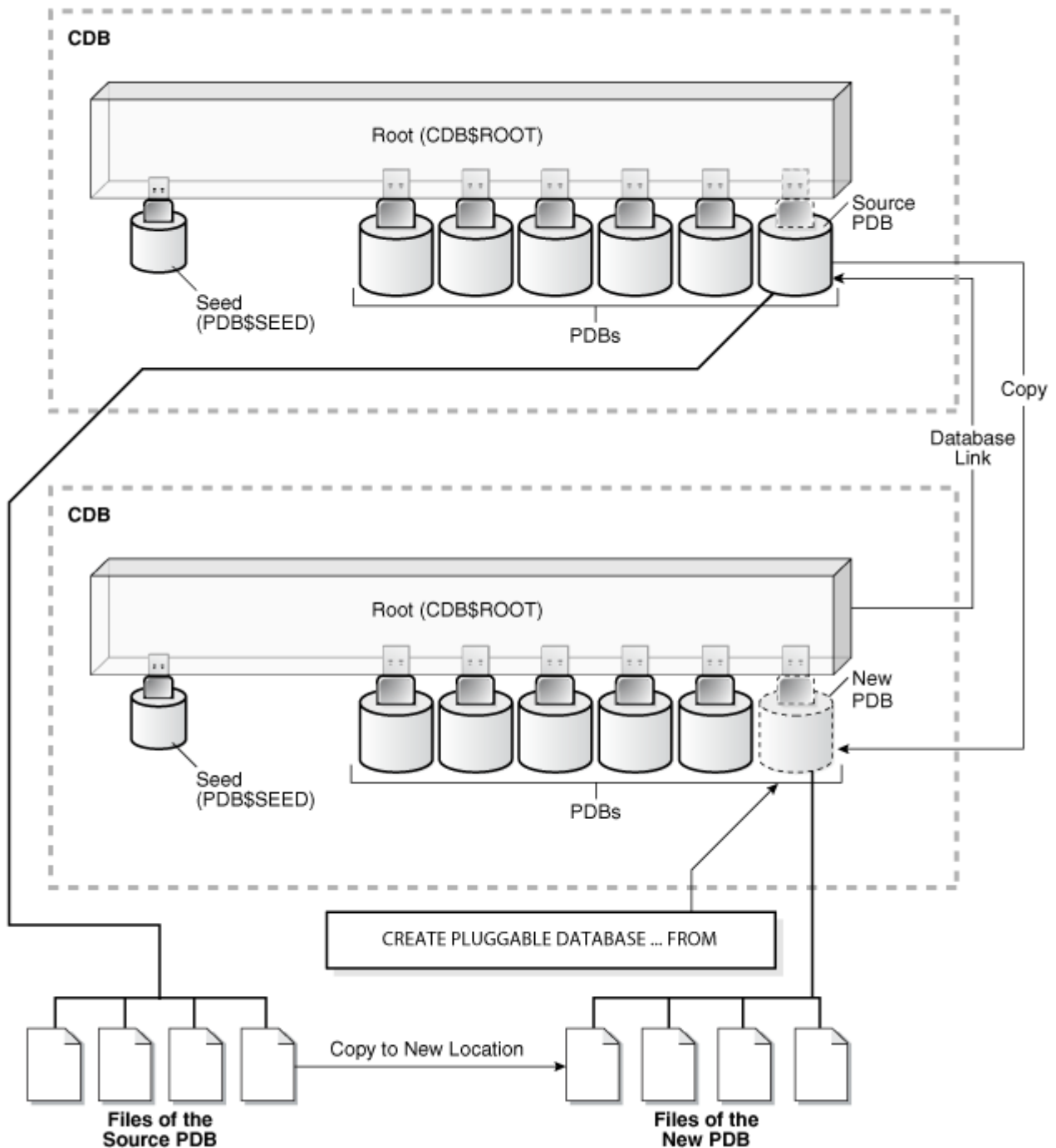
リモートPDBのクローニングについて

ソースがリモートCDB内のPDBである場合は、データベース・リンクを使用してPDBをローカルCDBにクローニングする必要があります。

データベース・リンクは、ローカルCDB (リモートCDBではない)に存在している必要があります。ローカルCDBのルートからCREATE PLUGGABLE DATABASE文を発行する場合は、クローニングされるPDBを含むリモートCDBへのデータベース・リンクをFROM句で指定する必要があります。データベース・リンクは、ローカルCDBから、リモートCDBのルートまたはリモート・ソースPDBのいずれかに接続します。

次の図は、ソースPDBがリモートである場合に、この手法で新規PDBがどのように作成されるかを示しています。

図7-3 リモートPDBのクローニングによるPDBの作成



Oracle Database 19c以降では、サイレント・モードでDBCAを使用してリモートPDBをクローニングできます。

親トピック: [リモートPDBのクローニング](#)

リモートPDBのクローニング: 基本的なステップ

PDBは、リモートPDBをクローニングすることで作成できます。クローニング操作後、ソースとターゲットPDBは異なる場所にあります。

一般的な前提条件

次の前提条件を満たしている必要があります。

- [「PDB作成の一般的な前提条件」](#)で説明している前提条件を満たします。
- 現行ユーザーは、ターゲットPDBが含まれるCDBのルートでCREATE PLUGGABLE DATABASEシステム権限を持っている必要があります。
- ソースとターゲットのプラットフォームは、次の要件を満たしている必要があります。

- endiannessが同じである必要があります。
- ソース・プラットフォームにインストールされているデータベース・オプションは、ターゲット・プラットフォームにインストールされているデータベース・オプションと同じか、サブセットである必要があります。
- アプリケーションPDBを作成している場合、ソースPDBのアプリケーション名およびバージョンは、ターゲット・アプリケーション・コンテナのアプリケーション名およびバージョンと一致する必要があります。

文字セットの前提条件

- PDBがクローニングされるCDBの文字セットがAL32UTF8でない場合、ソースとターゲットには互換性のある文字セットおよび各国語文字セットが必要です。PDBがクローニングされるCDBの文字セットがAL32UTF8である場合、この要件は適用されません。
- アプリケーションPDBを作成している場合、アプリケーションPDBはアプリケーション・コンテナと同じ文字セットおよび各国語文字セットである必要があります。

CDBのデータベース文字セットがAL32UTF8の場合、アプリケーション・コンテナの文字セットおよび各国語文字セットはCDBと異なっても構いません。ただし、アプリケーション・コンテナ内のすべてのアプリケーションPDBには、アプリケーション・コンテナと同じ文字セットおよび各国語文字セットが必要です。

ノート:



Oracle Multitenant では、あるコンテナの LOB に対する異なる文字セットのコンテナによるデータ・リンク、拡張データ・リンクまたは CONTAINERS () 句を使用したアクセスはサポートされません。たとえば、CDB ルートと salespdb の文字セットが異なる場合、CDB ルートで実行される CONTAINERS () 問合せは salespdb の LOB にアクセスできません。

ソースPDBのオープン・モードの前提条件

- ソースPDBはクローズできません。
- リモートCDBがローカルUNDOモードでない場合、ソースPDBは読取り専用モードでオープンしている必要があります。
[\[CDB UNDOモードについて\]](#)を参照してください。
- リモートCDBがARCHIVELOGモードでない場合、ソースPDBは読取り専用モードでオープンしている必要があります。
- リフレッシュ可能なPDBを作成している場合、ソースPDBはARCHIVELOGモードおよびローカルUNDOモードである必要があります。

データベース・リンクの前提条件

次の前提条件を満たしている必要があります。

- データベース・リンクで、宛先CDB (PDBのクローニング先のCDB)からソースCDB内のPDBへの接続が有効である必要があります。
- データベース・リンクでは、共通ユーザーとしてソースCDBのルートに、または共通ユーザーがローカル・ユーザーとしてソースPDBに接続できます。ソースPDBとしては、標準PDBまたはアプリケーションPDBのどちらかを使用できます。
- データベース・リンクで指定されたユーザー・アカウントには、次のいずれかの権限が必要です。
 - ソースPDBに共通またはローカルで付与されているCREATE PLUGGABLE DATABASE権限

- SYSOPER権限

- Oracle Data Guard環境でプライマリCDBへのPDBのリモート・クローンを実行している場合は、スタンバイCDBでSTANDBY_PDB_SOURCE_FILE_DBLINK初期化パラメータを設定します。このパラメータは、CREATE PLUGGABLE DATABASE ... FROM dblinkで使用するデータベース・リンクの名前を指定します。スタンバイCDBでは、ソースPDBが読取り専用モードでオープンしている場合のみ、データベース・リンクで参照されるソースPDBからデータファイルのコピーを試みます。それ以外の場合は、スタンバイCDBでデータ・ファイルをOracle Managed Filesの場所にコピーする必要があります。

リモートPDBをクローニングするには:

1. SQL*Plusで、現在のコンテナがターゲットCDBのルートまたはターゲット・アプリケーション・コンテナのアプリケーション・ルートであることを確認します。
2. CREATE PLUGGABLE DATABASE文を実行し、ソースPDBをFROM句で指定します。必要に応じて、他の句を指定します。

作成したPDBはマウント・モードになり、そのステータスはNEWです。PDBのオープン・モードは、V\$PDBSビューのOPEN_MODE列を問い合わせることで確認できます。CDB_PDBSまたはDBA_PDBSビューのSTATUS列を問い合わせ、PDBのステータスを表示できます。

PDBの新しいデフォルト・サービスが作成されます。サービスの名前はPDBと同じになり、このサービスを使用してPDBにアクセスできます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。

ノート:



PDBの作成時にエラーが返された場合は、作成されたPDBがUNUSABLE状態である可能性があります。CDB_PDBSまたはDBA_PDBSビューを問い合わせ、PDBの状態を確認したり、アラート・ログをチェックしてPDBの作成エラーについてさらに学習できます。使用不可のPDBは削除することのみができ、使用不可のPDBと同じ名前のPDBを作成できるようにするには、その前に削除する必要があります。

3. 新しいPDBを読取り/書込みモードでオープンします。

Oracle Databaseで新しいPDBをCDBに統合するには、PDBを読取り/書込みモードでオープンする必要があります。PDBを読取り専用モードでオープンしようとすると、エラーが返されます。PDBが読取り/書込みモードでオープンされると、そのステータスはNORMALになります。

4. PDBをバックアップします。

PDBは、バックアップしないとリカバリできません。

関連項目:

- [「PDBのリフレッシュ」](#)
- [「PDBのオープン・モードの変更」](#)
- PDBのバックアップの詳細は、『[Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド](#)』を参照してください
- Oracle Data Guard環境でのPDBの接続についてさらに学習するには、『[Oracle Data Guard概要および管理](#)』

を参照してください

- 文字セットの互換性に関する要件について学習するには、[『Oracle Databaseグローバルバージョン・サポート・ガイド』](#)を参照してください
- PDB_FILE_NAME_CONVERT初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください

親トピック: [リモートPDBのクローニング](#)

リモートPDBのクローニング後

リモートPDBのクローニング後には、ユーザーと表領域に関する特定のルールが適用されます。

リモートPDBのクローニング後、次の内容が適用されます。

- ソースPDBのデフォルト一時表領域を使用していた新しいPDBのユーザーは、新しいPDBのデフォルト一時表領域を使用します。PDB内のデフォルト以外の一時表領域を使用していたユーザーは、クローニングされたPDB内の同じローカル一時表領域を引き続き使用します。
- ソースCDBに存在していてターゲットCDBには存在しないユーザー作成の共通ユーザー・アカウントには、一般に権限は付与されていません。ただし、ターゲットCDBに共通ユーザー・アカウントが存在していて、PDBの共通ユーザー・アカウントと同じ名前である場合、後者は前者にリンクされ、ターゲットCDBの共通ユーザー・アカウントに付与された権限を持ちます。

クローニングまたは接続されたPDBにターゲットCDBに存在しない共通ユーザー・アカウントが存在し、このユーザーがPDB内のオブジェクトを所有していない場合、Oracle Databaseでは同期ステップ中にユーザーが削除されます。それ以外の場合、ユーザー・アカウントはターゲットPDBでロックされます。ロックされたアカウントに関して、次のオプションがあります。

- PDBをクローニングし、ルートに接続し、同じ名前で共通ユーザー・アカウントを作成します。PDBを読み取り/書き込みモードでオープンすると、ユーザー・アカウントに共通して付与されているロールや権限の相違点が解決され、アカウントをロック解除できます。ユーザー・アカウントに対してローカルに付与されている権限やロールは、このプロセス中も変更されません。
- PDBに新しいローカル・ユーザー・アカウントを作成し、データ・ポンプを使用して、ロックされたユーザーのデータを新しいローカル・ユーザーのスキーマにエクスポート/インポートします。
- ユーザー・アカウントはロックされたままにします。
- ユーザー・アカウントを削除します。

関連項目:

- [「CDB内表領域の管理について」](#)
- ローカル・ユーザーの作成の詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください
- CDBでのOracle Data Pumpの使用の詳細は、[『Oracle Databaseユーティリティ』](#)を参照してください

親トピック: [リモートPDBのクローニング](#)

リモートPDBのクローニング: 例

これらの例では、様々な要因に基づき、リモートPDBまたは非CDBをクローニングします。

各例で、新しいPDBが属するルートはCREATE PLUGGABLE DATABASE文の実行時の現在のコンテナによって決まります。

- 現在のコンテナがCDBルートの場合、新しいPDBはCDBルートに作成されます。
- 現在のコンテナがアプリケーション・コンテナ内のアプリケーション・ルートの場合、新しいPDBはアプリケーション・ルート内にアプリケーションPDBとして作成されます。
- [句を使用しないリモートPDBのクローニング: 例](#)
この例では、様々な要因に基づき、pdb1という名前のリモート・ソースPDBを、pdb2という名前のターゲットPDBにクローニングします。
- [DBCAを使用したリモートPDBのクローニング: 例](#)
この例では、DBCAを使用して、リモートCDBからローカルCDBにpdb1という名前のPDBをクローニングし、クローニング先でclonepdb1という名前に変更します。

親トピック: [リモートPDBのクローニング](#)

句を使用しないリモートPDBのクローニング: 例

次の例では、様々な要因に基づき、pdb1という名前のリモート・ソースPDBを、pdb2という名前のターゲットPDBにクローニングします。

この例では、次の要因を想定しています。

- リモートPDBへのデータベース・リンク名はpdb1_linkです。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。これらのファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

次の文は、pdb1リモートPDBからpdb2 PDBをクローニングします。

```
CREATE PLUGGABLE DATABASE pdb2 FROM pdb1@pdb1_link;
```

関連項目:

- Oracle Managed Filesの詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- PDB_FILE_NAME_CONVERT初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください

親トピック: [リモートPDBのクローニング: 例](#)

DBCAを使用したリモートPDBのクローニング: 例

この例では、DBCAを使用して、リモートCDBからローカルCDBにpdb1という名前のPDBをクローニングし、クローニング先で clonepdb1という名前に変更します。

前提条件

この例では、次のことを前提としています。

- ローカル・データベースのユーザーには、ルート・コンテナでのCREATE PLUGGABLE DATABASE権限が付与されている。
- リモートCDBはローカルUNDOモードになっている。
- リモートCDBとローカルCDBは、ARCHIVELOGモードになっている。
- データベース・リンクの接続先リモートCDBの共通ユーザーには、CREATE PLUGGABLE DATABASE、SESSIONおよびSYSOPER権限が付与されている。
- ローカルCDBとリモートCDBは、同じオプションでインストールされている。

想定

この例では、次のことを前提としています。

- クローンPDBを格納するCDBのホスト上でDBCAを実行しています。ローカルCDBの名前は、loccdb1です。
- リモート(ソース) CDBの名前はremcdb1で、ホストremcdb1hostに存在しています。リモートCDBのインスタンス名は、reminstです。
- クローニングされるPDBであるリモートPDBの名前は、rempdb1です。
- 共通ユーザーc##adminuser_remcdb1がremcdb1に存在しています。
- 管理ユーザーlocSYSには、loccdb1 (PDBのクローニング先のCDB)に対するSYSDBA権限が付与されています。
- 管理ユーザーremSYSにはremcdb1 (クローニングするPDBを格納しているCDB)に対するSYSDBA権限が付与されています。
- loccdb1へのクローニング後、PDBの名前はclonepdb1に変更します。

次のsilentコマンドでは、rempdb1をloccdb1にクローニングします。

```
./dbca -silent
-createPluggableDatabase
-createFromRemotePDB
-sourceDB remcdb1
-remotePDBName rempdb1
-remoteDBConnString remcdb1host:1521/reminst
-remoteDBSYSDBAUserName remSYS
  -remoteDBSYSDBAUserPassword remsyspwd
-dblinkUsername c##adminuser_remcdb1
  -dblinkUserPassword pwd4dblinkusr
-sysDBAUserName locSYS
  -sysDBAPassword locsyspwd
-pdbName clonepdb1
```

関連項目:

DBCAコマンドの構文およびセマンティクスは、[『Oracle Database管理者ガイド』](#)を参照

親トピック: [リモートPDBのクローニング: 例](#)

非CDBのクローニング

非CDBのクローニング手順は、リモートPDBのクローニング手順と非常によく似ています。

- [非CDBのクローニングについて](#)
ソースが非CDBである場合は、非CDBへのデータベース・リンクをFROM句で指定する必要があります。
- [非CDBのクローニング: 基本的なステップ](#)
PDBは、非CDBをクローニングすることで作成できます。
- [リモート非CDBのクローニング: 例](#)
この例では、様々な要因に基づき、mydbという名前のリモート・ソース非CDBを、pdb2という名前のターゲットPDBにクローニングして新しいPDBを作成します。

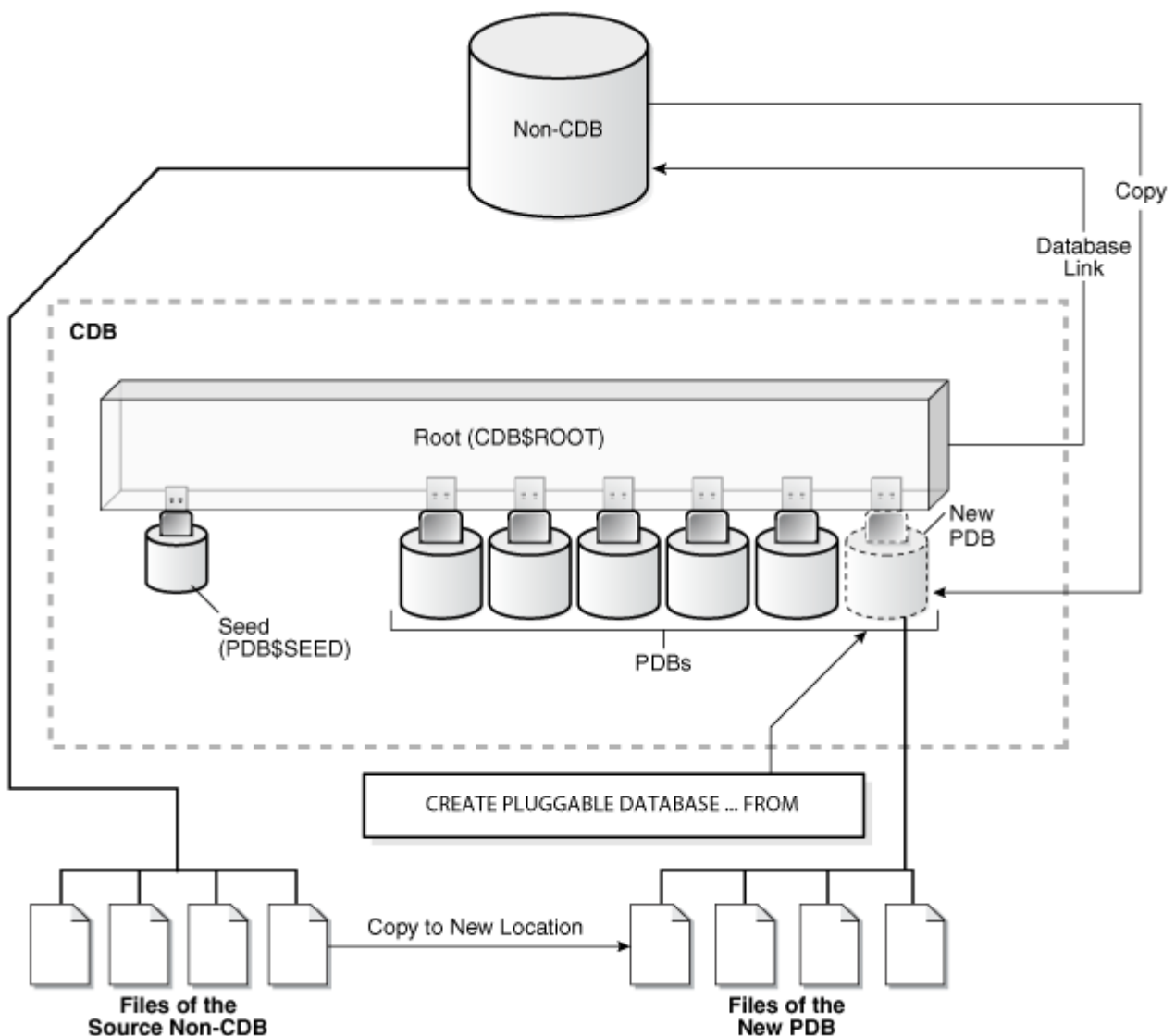
親トピック: [PDBまたは非CDBのクローニング](#)

非CDBのクローニングについて

ソースが非CDBである場合は、非CDBへのデータベース・リンクをFROM句で指定する必要があります。

次の図は、ソースがリモート非CDBである場合に、この手法で新しいPDBがどのように作成されるかを示しています。

図7-4 非CDBのクローニングによるPDBの作成



非CDBのクローニング: 基本的なステップ

PDBは、非CDBをクローニングすることで作成できます。

一般的な前提条件

次の前提条件を満たしている必要があります。

- [「PDB作成の一般的な前提条件」](#)で説明している前提条件を満たします。



ノート:

ソースの非 CDB のバックアップを使用して新しい PDB をリカバリできるようにするには、クローニングの前に DBMS_PDB.EXPORTMANBACKUP を使用する必要があります。

- 現行ユーザーは、ターゲットPDBが含まれるCDBのルートでCREATE PLUGGABLE DATABASEシステム権限を持っている必要があります。
- ソースとターゲットのプラットフォームは、次の要件を満たしている必要があります。
 - endiannessが同じである必要があります。
 - ソース・プラットフォームにインストールされているデータベース・オプションは、ターゲット・プラットフォームにインストールされているデータベース・オプションと同じか、サブセットである必要があります。
- CDBと非CDBでOracle Database 12cリリース1 (12.1.0.2)以上が実行されている必要があります。
- CDBと非CDBで同じOracle Databaseリリースが実行されている必要があります。
- 新しく作成したPDBのデータ・ブロック・サイズはCDBと一致している必要があります。
- 非CDBがNOARCHIVELOGモードになっている場合は、読取り専用モードでオープンする必要があります。非CDBがARCHIVELOGモードになっている場合は、読取り専用または読取り/書込みでオープンできます。

文字セットの前提条件

- CDBの文字セットがAL32UTF8でない場合、ソースとターゲットには互換性のある文字セットおよび各国語文字セットが必要です。PDBがクローニングされるCDBの文字セットがAL32UTF8である場合、この要件は適用されません。
- アプリケーションPDBを作成している場合、アプリケーションPDBはアプリケーション・コンテナと同じ文字セットおよび各国語文字セットである必要があります。

CDBのデータベース文字セットがAL32UTF8の場合、アプリケーション・コンテナの文字セットおよび各国語文字セットはCDBと異なっても構いません。ただし、アプリケーション・コンテナ内のすべてのアプリケーションPDBには、アプリケーション・コンテナと同じ文字セットおよび各国語文字セットが必要です。

ノート:



Oracle Multitenant では、あるコンテナの LOB に対する異なる文字セットのコンテナによるデータ・リンク、拡張データ・リンクまたは CONTAINERS () 句を使用したアクセスはサポートされません。たとえば、CDB ルートと salespdb の文字セットが異なる場合、CDB ルートで実行される CONTAINERS () 問合せは salespdb の LOB にアクセスでき

ません。

データベース・リンクの前提条件

次の前提条件を満たしている必要があります。

- データベース・リンクで、ターゲットCDBからソースCDBへの接続が有効である必要があります。データベース・リンクは、CDBのルート、アプリケーションPDB (ソースがアプリケーションPDBの場合)、またはPDBのいずれかに接続できます。
- データベース・リンクが接続するユーザーには、CREATE PLUGGABLE DATABASEシステム権限が必要です。
- データベース・リンクがソースPDBのCDBのルートに接続する場合、データベース・リンクが接続するユーザーは、共通ユーザーである必要があります。
- Oracle Data Guard環境でプライマリCDBへのPDBのリモート・クローンを実行している場合は、スタンバイCDBでSTANDBY_PDB_SOURCE_FILE_DBLINK初期化パラメータを設定します。このパラメータは、CREATE PLUGGABLE DATABASE ... FROM dblinkで使用するデータベース・リンクの名前を指定します。スタンバイCDBでは、ソースPDBが読取り専用モードでオープンしている場合のみ、データベース・リンクで参照されるソースPDBからデータファイルのコピーを試みます。それ以外の場合は、スタンバイCDBでデータ・ファイルをOracle Managed Filesの場所にコピーする必要があります。

リモート非CDBをクローニングするには:

1. SQL*Plusで、現在のコンテナがターゲットCDBのルートまたはターゲット・アプリケーション・コンテナのアプリケーション・ルートであることを確認します。
2. CREATE PLUGGABLE DATABASE文を実行し、FROM句でソース非CDBを指定します。必要に応じて、他の句を指定します。

作成したPDBはマウント・モードになり、そのステータスはNEWです。PDBのオープン・モードは、V\$PDBSビューのOPEN_MODE列を問い合わせることで確認できます。CDB_PDBSまたはDBA_PDBSビューのSTATUS列を問い合わせ、PDBのステータスを表示できます。

PDBの新しいデフォルト・サービスが作成されます。サービスの名前はPDBと同じになり、このサービスを使用してPDBにアクセスできます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。

ノート:

PDBの作成時にエラーが返された場合は、作成されたPDBがUNUSABLE状態である可能性があります。CDB_PDBSまたはDBA_PDBSビューを問い合わせ、PDBの状態を確認したり、アラート・ログをチェックしてPDBの作成エラーについてさらに学習できます。使用不可のPDBは削除することのみができ、使用不可のPDBと同じ名前のPDBを作成できるようにするには、その前に削除する必要があります。

3. ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sqlスクリプトを実行します。このスクリプトは、PDBを初めてオープンする前に実行する必要があります。

noncdb_to_pdb.sqlスクリプトを実行するには、次のステップを実行します。

- a. 新しく作成したPDBにコンテナを設定します。

現在のユーザーにはSYSDBA管理権限が必要であり、その権限は共通で付与されているか、またはPDBでローカルに付与されている必要があります。ユーザーは、接続時にAS SYSDBAを使用して権限を行使する必要があります。

b. noncdb_to_pdb.sql スクリプトを次のように実行します。

```
@$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

スクリプトによって、PDBのオープン、変更の実行、およびPDBのクローズが実行されます。

4. 新しいPDBを読み取り/書き込みモードでオープンします。

Oracle Databaseで新しいPDBをCDBに統合するには、PDBを読み取り/書き込みモードでオープンする必要があります。PDBを読み取り専用モードでオープンしようとすると、エラーが返されます。PDBが読み取り/書き込みモードでオープンされると、そのステータスはNORMALになります。

5. PDBをバックアップします。

PDBは、バックアップしないとリカバリできません。

関連項目:

- 非CDBからPDBをクローニングする詳細な例は、[My Oracle Supportノート1928653.1](#)を参照
- [「PDBのリフレッシュ」](#)
- [「PDBのオープン・モードの変更」](#)
- PDBのバックアップの詳細は、『[Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド](#)』を参照してください
- 文字セットの互換性に関する要件について学習するには、『[Oracle Databaseグローバルバージョン・サポート・ガイド](#)』を参照してください
- PDB_FILE_NAME_CONVERT初期化パラメータの詳細は、『[Oracle Databaseリファレンス](#)』を参照してください

親トピック: [非CDBのクローニング](#)

リモート非CDBのクローニング: 例

次の例では、様々な要因に基づき、mydbという名前のリモート・ソース非CDBを、pdb2という名前のターゲットPDBにクローニングして新しいPDBを作成します。

この例では、次の要因を想定しています。

- リモート非CDBへのデータベース・リンク名はmydb_linkです。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。これらのファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE

句は必要ありません。

次の文は、mydbという名前のリモート非CDBからpdb2 PDBを作成します。

```
CREATE PLUGGABLE DATABASE pdb2 FROM mydb@mydb_link;
```

ソース・データベースが非CDBである場合、非CDBの名前のかわりにNON\$CDBを使用できます。たとえば、次の文は前の例と同じです。

```
CREATE PLUGGABLE DATABASE pdb2 FROM NON$CDB@mydb_link;
```

関連項目:

Oracle Managed Filesの詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [非CDBのクローニング](#)

リフレッシュ可能なクローンPDBについて

CREATE PLUGGABLE DATABASE ... REFRESH MODE文は、ソースPDBをクローニングし、クローンをリフレッシュ可能に構成します。クローンPDBをリフレッシュすると、最後のREDOログの適用以降に蓄積されたREDOでクローンPDBが更新されます。

- [リフレッシュ可能なクローンPDBの目的](#)
本番PDBのクローニング操作には長い時間がかかる場合があります。
- [自動および手動リフレッシュ・モード](#)
設定した間隔で自動的にリフレッシュするようにクローンPDBを構成するか、またはALTER PLUGGABLE DATABASE REFRESH文により手動でリフレッシュできます。
- [リフレッシュ可能なクローンPDBの要件](#)
リフレッシュ可能なクローンPDBを作成するには、データベース・リンクが必要です。データベース・リンクは、同じCDBまたは別のCDBを指し示すことができます。
- [リフレッシュ可能なクローンPDBの作成: シナリオ](#)
このシナリオでは、pdb1という名前のリモートPDBからpdb1_ref_clnという名前のリフレッシュ可能なクローンを作成します。

親トピック: [PDBまたは非CDBのクローニング](#)

リフレッシュ可能なクローンPDBの目的

本番PDBのクローニング操作には長い時間がかかる場合があります。

システム上でのドラッグを避けるためPDBをあまりクローニングしないと、クローニングしたデータが古くなります。リフレッシュ可能なクローンPDBによってこの問題が解決します。リフレッシュ可能クローンPDBが古くなっている場合は、それをクローズしてから、最新のREDOでリフレッシュできます。リフレッシュ可能クローンPDBは、リフレッシュされていない場合は、読み取り専用でオープンできます。本番PDBの最終マスターのリフレッシュ可能クローンを保持し、PDBレベルのスナップショットを取得してから、開発およびテスト用にそれらのPDBスナップショットからクローンを作成するのが一般的な方法です。

ALTER PLUGGABLE DATABASE ... SWITCHOVER文を使用して、ソースとクローンPDBのロールを置き換えることができます。この機能は、次の場合に有効です。

- 計画済スイッチオーバー

ソースPDBをホストするCDBは、クローンPDBをホストするCDBよりもオーバーヘッドが多くなる可能性があります。ロード・バランシングをアーカイブするには、ロールを置き換えて、クローンを新しいソースPDBにし、ソースPDBを新しいクローンにできます。

- 計画外スイッチオーバー

ソースPDBで不測の失敗が発生することがあります。この場合、クローンPDBを新しいソースPDBにして、通常の操作を再開できます。

関連項目:

- [「リフレッシュ可能なクローンPDBの管理」](#)
- ALTER PLUGGABLE DATABASE ... SWITCHOVERについてさらに学習するには、[『Oracle Database SQL言語リファレンス』](#)を参照してください

親トピック: [リフレッシュ可能なクローンPDBについて](#)

自動および手動リフレッシュ・モード

設定した間隔で自動的にリフレッシュするようにクローンPDBを構成するか、またはALTER PLUGGABLE DATABASE REFRESH文により手動でリフレッシュできます。

REFRESH MODE句は、CREATE PLUGGABLE DATABASE ... FROM文でのみサポートされています。この句を使用して、次のいずれかのオプションを指定できます。

- デフォルトのREFRESH MODE NONEを指定して、リフレッシュ可能でないPDBを作成します。
ALTER PLUGGABLE DATABASE文にREFRESH MODE NONE句を含め、PDBを読み取り/書込みモードでオープンすることにより、リフレッシュ可能なクローンPDBを通常のPDBに変更できます。通常のPDBはリフレッシュ可能なクローンPDBに変更できません。リフレッシュ可能なクローンPDBが通常のPDBに変換された後は、リフレッシュ可能なクローンPDBに戻すことができません。
- REFRESH MODE MANUALを指定して、手動でリフレッシュする必要があるリフレッシュ可能PDBを作成します。
- REFRESH MODE EVERY number_of_minutes MINUTESを指定して、指定した時間(分単位)の経過後に自動的にリフレッシュされるリフレッシュ可能PDBを作成します。自動リフレッシュを使用するリフレッシュ可能PDBを手動でリフレッシュすることもできます。

ノート:

- リフレッシュ可能 PDB を作成する場合、PDB に REMOTE_RECOVERY_FILE_DEST 初期化パラメータを設定できます。この初期化パラメータは、ソース PDB がデータベース・リンク上で使用可能でない場合に、リフレッシュ操作中にアーカイブ・ログ・ファイルを読み取るディレクトリを指定します。
- ソース PDB に新しいデータファイルが作成された場合は、CDB で PDB_FILE_NAME_CONVERT 初期化パラメータを設定して、データファイルのパスをソース PDB からクローン PDB に変換する必要があります。
- 表領域暗号化アルゴリズムへの変更(AES128 から AES256 への変更など)は、ソース PDB でアルゴリズムが変更された後はリフレッシュ可能 PDB に適用されません。リフレッシュ可能 PDB を作成した後、そ

の表領域暗号化アルゴリズムを手動で更新する必要があります。

例7-1 自動リフレッシュを指定したREFRESH MODE句

このリフレッシュ・モード句では、リフレッシュ可能なPDBが2時間(120分)おきに自動的にリフレッシュされることを指定します。

```
REFRESH MODE EVERY 120 MINUTES
```

関連項目:

- [「リモートPDBのクローニング: 基本的なステップ」](#)
- [「PDBのリフレッシュ」](#)

親トピック: [リフレッシュ可能なクローンPDBについて](#)

リフレッシュ可能なクローンPDBの要件

リフレッシュ可能なクローンPDBを作成するには、データベース・リンクが必要です。データベース・リンクは、同じCDBまたは別のCDBを指し示すことができます。

リフレッシュ可能なクローンPDBは、次のいずれかの状態である必要があります。

- クローズ

リフレッシュ可能PDBは、リフレッシュの実行時にクローズする必要があります。自動リフレッシュが試行されたときにクローズされていない場合、リフレッシュは次のスケジュール済みリフレッシュまで延期されます。ユーザーが手動リフレッシュの実行を試行したときにクローズされていない場合は、エラーがレポートされます。

- 読取り専用モードでオープン

ソースPDBでは行われていない同期されていない変更がリフレッシュ可能なPDBで行われないように、リフレッシュ可能なPDBは読取り専用モードにしておく必要があります。リフレッシュ可能なPDBはクローン・マスターとして機能することが意図されているため、リフレッシュされた時点でソースPDBが正確に反映される必要があります。

親トピック: [リフレッシュ可能なクローンPDBについて](#)

リフレッシュ可能なクローンPDBの作成: シナリオ

このシナリオでは、pdb1という名前のリモートPDBからpdb1_ref_clnという名前のリフレッシュ可能なクローンを作成します。

クローンPDBはソースPDBのコピーです。クローンPDBを定期的にリフレッシュして、ソースPDBに対する変更内容で更新します。

想定

このシナリオでは、次の要因を想定しています。

- リモートPDBへのデータベース・リンク名はpdb1_linkです。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。これらのファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。
- リフレッシュ可能なクローンは、60分ごとに自動的にリフレッシュされます。

ノート:



リフレッシュ可能な PDB を作成するには、ソース PDB が ARCHIVELOG モードおよびローカル UNDO モードである必要があります。

リフレッシュ可能なクローンPDBを作成するには:

1. SQL*Plusで、現在のコンテナがCDBルートまたはアプリケーション・ルートであることを確認します。

現在のコンテナがCDBルートの場合、PDBはCDBに作成されます。現在のコンテナがアプリケーション・ルートの場合、アプリケーションPDBはアプリケーション・コンテナ内に作成されます。

2. CREATE PLUGGABLE DATABASE文を実行します。

次の文では、pdb1からpdb1_ref_clnを作成します。

```
CREATE PLUGGABLE DATABASE pdb1_ref_cln FROM pdb1@pdb1_link REFRESH MODE EVERY 60 MINUTES;
```

関連項目:

[「リフレッシュ可能なクローンPDBの管理」](#)

親トピック: [リフレッシュ可能なクローンPDBについて](#)

PDBスナップショットからのPDBのクローニング

PDBスナップショットからPDBを作成するには、CREATE PLUGGABLE DATABASE ... USING SNAPSHOT文を実行します。

- [PDBスナップショットからのPDBのクローニングについて](#)
PDBスナップショットはPDBのPoint-in-Timeコピーです。スナップショットの作成中に、ソースPDBを読み取り専用または読み取り/書き込みでオープンできます。PDBスナップショットからのクローンは、フルであり、スタンドアロンのPDBです。
- [PDBスナップショットからのPDBのクローニング: シナリオ](#)
このシナリオでは、CREATE PLUGGABLE DATABASE ... USING SNAPSHOTを実行することで、PDBスナップショットから新しいPDBを作成します。

親トピック: [PDBまたは非CDBのクローニング](#)

PDBスナップショットからのPDBのクローニングについて

PDBスナップショットはPDBのPoint-in-Timeコピーです。スナップショットの作成中に、ソースPDBを読み取り専用または読み取り/書き込みでオープンできます。PDBスナップショットからのクローンは、フルであり、スタンドアロンのPDBです。

親トピック: [PDBスナップショットからのPDBのクローニング](#)

PDBスナップショットからのPDBのクローニング: シナリオ

このシナリオでは、CREATE PLUGGABLE DATABASE ... USING SNAPSHOTを実行して、PDBスナップショットから新しいPDBを作成します。

想定

この例では、次の要因を想定しています。

- ソースPDB salespdbの8個の日次スナップショットを含むPDBスナップショット・カールセルが存在しています。スナップショットには作成時の曜日、日付および時刻にちなんだ名前が付けられています(例: pdb1_mon_2_1201、pdb1_tue_3_1201、pdb1_wed_4_1201など)。
- ソースのsalespdbが読み取り/書き込みモードであった場合は、すべてのスナップショットが作成されています。
- 新しいPDBはpdb1_wed_4_1201という名前のスナップショットのクローンで、これは先週の水曜日、その月の4日の午前12時1分を取得されたpdb1のスナップショットです。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。したがって、FILE_NAME_CONVERT句は必要ありません。これらのファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

PDBスナップショットからPDBをクローニングするには:

1. SQL*Plusで、現在のコンテナがCDBルートまたはアプリケーション・ルートであることを確認します。

現在のコンテナがCDBルートの場合、PDBはCDBに作成されます。現在のコンテナがアプリケーション・ルートの場合、アプリケーションPDBはアプリケーション・コンテナ内に作成されます。

2. CREATE PLUGGABLE DATABASE ... USING SNAPSHOT文を実行します。

次の文は、pdb1_wed_4_1201という名前のPDBのスナップショットからpdb1_copy PDBをクローニングします。

```
CREATE PLUGGABLE DATABASE pdb1_copy FROM pdb1
USING SNAPSHOT pdb1_wed_4_1201;
```

関連項目:

- [「自動PDBスナップショットの構成」](#)
- 各種エディションおよびサービスでサポートされる機能の詳細は、[『Oracle Databaseライセンス情報ユーザー・マニュアル』](#)を参照

親トピック: [PDBスナップショットからのPDBのクローニング](#)

スナップショット・コピーPDBの作成およびマテリアライズ

PDBは、基礎となる記憶域のスナップショットからクローニングできます。PDBファイルはスパースですが、それらのファイルをマテリアライズしてスタンドアロンPDBを作成できます。

- [スナップショット・コピーPDBについて](#)
スナップショット・コピーPDBは、CREATE PLUGGABLE DATABASE ... FROM ... SNAPSHOT COPY文を実行することで作成できます。ソースPDBは、FROM句で指定します。
- [スナップショット・コピーPDBの作成: シナリオ](#)
このシナリオでは、CREATE PLUGGABLE DATABASEでSNAPSHOT COPY句を指定することで、スナップショット・コピーPDBを作成します。
- [スナップショット・コピーPDBのマテリアライズ](#)
MATERIALIZE句を指定したALTER PLUGGABLE DATABASE文を実行して、スナップショット・コピーPDBをマテリアライズできます。スナップショット・コピーPDBをマテリアライズすると、すべてのデータ・ブロックがコピーされます。

親トピック: [PDBまたは非CDBのクローニング](#)

スナップショット・コピーPDBについて

スナップショット・コピーPDBは、CREATE PLUGGABLE DATABASE ... FROM ... SNAPSHOT COPY文を実行することで作成できます。ソースPDBは、FROM句で指定します。

スナップショット・コピーにより、クローンの作成に必要な時間が短縮されます。これは、ソース・データ・ファイルの完全なコピーが含まれていないためです。さらに、スナップショット・コピーPDBは、ソースPDBの領域よりもわずかな領域しか占有しません。

記憶域クローンには、ターゲットPDBのGUIDを使用して名前およびタグが付けられます。記憶域クローンのクローン・タグを表示するには、DBA_PDB_HISTORY.CLONETAG列を問い合わせます。

- [スナップショットのコピーPDBの記憶域要件](#)
CREATE PLUGGABLE DATABASE ... FROM srcpdb ... SNAPSHOT COPYを使用する場合は、ソースPDBのデータ・ファイルが、同じ記憶域タイプで存在している必要があります。
- [スナップショット・コピーPDBの制限](#)
スナップショット・コピーPDBの基になっている記憶域スナップショットは削除できません。

親トピック: [スナップショット・コピーPDBの作成およびマテリアライズ](#)

スナップショットのコピーPDBの記憶域要件

CREATE PLUGGABLE DATABASE ... FROM srcpdb ... SNAPSHOT COPYを使用する場合は、ソースPDBのデータ・ファイルが、同じ記憶域タイプで存在している必要があります。

CREATE PLUGGABLE DATABASE ... FROM ... SNAPSHOT COPYコマンドの動作は、次のルールに基づいて決まります。

1. ファイル・システムで記憶域管理スナップショットがサポートされている場合、スナップショット・コピーPDBは、基礎となるファイル・システムの記憶域レベル・コピーに基づいています。スナップショット・コピーPDBには、スパース・ファイルが含まれています。copy-on-writeテクノロジーを使用すると、ディスク上の記憶域がさらに必要になるのは変更されたブロックのみとなります。
2. ファイル・システムで記憶域スナップショットがサポートされていない場合、アルゴリズムは次のようになります。
 - ストレージ・システムでOracle Exadataスパース・ディスク・グループが使用されている場合は、Oracle Databaseによってスナップショット・コピーPDBが作成されます。ただし、そのスナップショット・コピーPDBの存続期間中は、ソースPDBが読取り専用のみである必要があります。

- ストレージ・システムでOracle Exadataスパース・ディスク・グループが使用されない場合、動作は次のようになります。
 - CLONEDB=trueの場合、ソースPDBファイルの基礎になるファイル・システムは、任意のローカル・ファイル・システム、ネットワーク・ファイル・システム(NFS)またはOracle ACFSなどのクラスタ化ファイル・システムになります。ネットワーク・ファイル・システムを使用する場合は、CDBに対してDirect NFSを有効にする必要があります。ファイル・システムはスパース・ファイルをサポートする必要があります。ほとんどのUNIXシステムは、これらの要件を満たしています。

CLONEDB=trueの場合、ソースPDBのオープン・モードには次のような影響があります。

- ソースPDBが読取り専用モードでオープンされている場合、Oracle Databaseでは、copy-on-writeテクノロジーを使用してスナップショット・コピーPDBが作成されます。スナップショット・コピーPDBには、完全コピーではなく、スパース・ファイルが含まれています。
- ソースPDBが読取り/書込みモードでオープンされていない場合は、Oracle Databaseによってエラーが発行されます。
- CLONEDB=falseの場合は、Oracle Databaseによってエラーが発行されます。

Direct NFSクライアントにより、Oracle Databaseは、オペレーティング・システム・カーネルのNFSクライアントを使用するかわりに、ネットワーク接続記憶域(NAS)デバイスに直接アクセスできます。ソースPDBのファイルがDirect NFSクライアント記憶域に格納されている場合は、次の追加の要件が満たされている必要があります。

- ソースPDBファイルはNFSボリューム上に存在する必要があります。
- 記憶域の資格証明は、透過的データ暗号化キースタアに格納されている必要があります。
- 記憶域ユーザーは、ソースPDBのファイルをホストするボリューム上でスナップショットを作成および破棄するために必要な権限を持っている必要があります。
- 資格証明は、ADMINISTER KEY MANAGEMENT ADD SECRET SQL文を使用してキースタアに格納されている必要があります。

次の例では、ソフトウェア・キースタアのOracle Databaseシークレットを構成しています。

```
ADMINISTER KEY MANAGEMENT
ADD SECRET 'secret' FOR CLIENT 'client_name'
USING TAG 'storage_user'
IDENTIFIED BY keystore_password WITH BACKUP;
```

構成内の各記憶域サーバーに対して個別のエントリを追加するには、この文を実行します。前の例では、次の値を指定する必要があります。

- secretは、記憶域パスワードです。
- client_nameは、記憶域サーバーです。LinuxまたはUNIXのプラットフォームでは、/etc/hostsに入力した名前または記憶域サーバーのIPアドレスです。
- tagは、記憶域サーバーに渡されるユーザー名です。
- keystore_passwordは、キースタアのパスワードです。



ノート:

スナップショット・コピーの動作および効率性はベンダー固有であり、ベンダーによって異なる場合があります。

関連項目:

- Oracle ACFSの詳細は、[Oracle Automatic Storage Management Cluster File System管理者ガイド](#)を参照してください。
- Direct NFSクライアントの詳細は、使用しているオペレーティング・システムの『[Oracle Grid Infrastructureインストール・およびアップグレード・ガイド](#)』を参照してください
- 透過的データ暗号化の詳細は、『[Oracle Database Advanced Securityガイド](#)』を参照してください。
- PDBのスナップショット・クローニングをサポートするプラットフォームの詳細は、[My Oracle Supportノート 1597027.1](#)を参照してください
- SNAPSHOT COPY句を使用して作成されたPDBクローンでのExadataのサポートの詳細は、『[Oracle Exadata System Softwareユーザーズ・ガイド](#)』を参照してください

親トピック: [スナップショット・コピーPDBについて](#)

スナップショット・コピーPDBの制限

スナップショット・コピーPDBの基になっている記憶域スナップショットは削除できません。

スナップショット・コピーPDBをCDBルートまたはアプリケーション・コンテナから切断することはできません。スナップショット・コピーPDBを切断しようとすると、エラーが発生します。ただし、スナップショット・コピーPDBのマテリアライズは可能です。マテリアライズによってスタンドアロンPDBにすると、削除できるようになります。

ノート:

USING SNAPSHOT 句を使用して作成された PDB と SNAPSHOT COPY 句を使用して作成された PDB とでは、プロパティが異なります。1 つの CREATE PLUGGABLE DATABASE コマンドで両方の句を指定することはできません。CREATE PLUGGABLE DATABASE ... FROM ... USING SNAPSHOT 句では、スタンドアロンの完全 PDB が作成されます。これには、マテリアライズは不要です。CREATE PLUGGABLE DATABASE ... FROM ... SNAPSHOT COPY 句では、スパーズ PDB が作成されます。これは、基になっている記憶域レベル・スナップショットを削除する場合にはマテリアライズする必要があります。

親トピック: [スナップショット・コピーPDBについて](#)

スナップショット・コピーPDBの作成: シナリオ

このシナリオでは、CREATE PLUGGABLE DATABASEでSNAPSHOT COPY句を指定することで、スナップショット・コピーPDBを作成します。

想定

このシナリオでは、次の要因を想定しています。

- 新しいスナップショット・コピーPDBは、pdb1というPDBから作成されます。
- 基礎となるファイル・システムでは、記憶域スナップショットがサポートされています。したがって、CLONEDB初期化パラメータを設定する必要はありません。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。したがって、FILE_NAME_CONVERT句は必要ありません。これらのファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

スナップショット・コピーPDBを作成するには:

1. SQL*Plusで、現在のコンテナがCDBルートまたはアプリケーション・ルートであることを確認します。

現在のコンテナがCDBルートの場合、PDBはCDBに作成されます。現在のコンテナがアプリケーション・ルートの場合、アプリケーションPDBはアプリケーション・コンテナ内に作成されます。

2. CREATE PLUGGABLE DATABASE ... SNAPSHOT COPY文を実行します。

次の文は、pdb1からpdb1_snap_copy PDBをクローニングします。

```
CREATE PLUGGABLE DATABASE pdb1_snap_copy FROM pdb1 SNAPSHOT COPY;
```

pdb1_snap_copyが存在するかぎり、pdb1_snap_copyの基になっている記憶域スナップショットは削除できません。

関連項目:

[「スナップショット・コピーPDBのマテリアライズ」](#)

親トピック: [スナップショット・コピーPDBの作成およびマテリアライズ](#)

スナップショット・コピーPDBのマテリアライズ

MATERIALIZE句を指定したALTER PLUGGABLE DATABASE文を実行して、スナップショット・コピーPDBをマテリアライズできます。スナップショット・コピーPDBをマテリアライズすると、すべてのデータ・ブロックがコピーされます。

スナップショット・コピーPDBをマテリアライズすると、スナップショット・コピーPDB(スパース・ファイルを使用する)は、完全PDB(スパース・ファイルを使用しない)に変換されます。マテリアライズされたPDBは、ソースPDBに依存しなくなり、削除や別のオープン・モードへの変更が可能です。

たとえば、pdb1_snap_copyがスナップショット・コピーPDBである場合は、ALTER PLUGGABLE DATABASE MATERIALIZEコマンドを実行することで、それをマテリアライズしてスタンドアロンPDBにできます。マテリアライズ後は、pdb1_snap_copyは記憶域レベルのスナップショットに依存しなくなり、削除できるようになります。

PDBスナップショットをマテリアライズするには:

1. SQL*Plusで、現在のコンテナがマテリアライズされるスナップショット・コピーPDBであることを確認します。
2. MATERIALIZE句を指定してALTER PLUGGABLE DATABASE文を実行します。

例7-2 スナップショット・コピーPDBのマテリアライズ

次のSQL文は、スナップショット・コピーPDBをマテリアライズします。

```
ALTER PLUGGABLE DATABASE MATERIALIZE;
```

関連項目:

- スナップショット・コピーPDBについてさらに学習するには、[「スナップショット・コピーPDBについて」](#)を参照してください。
- [「スナップショット・コピーPDBの作成: シナリオ」](#)
- すべてのスナップショット・クローンPDBを削除した後にソースPDBのデータ・ファイル権限を元に戻す方法を学習するには、[My Oracle Supportノート2627975.1](#)を参照してください。

親トピック: [スナップショット・コピーPDBの作成およびマテリアライズ](#)

分割ミラーによるクローンPDBの作成

Oracle ASMでは、分割ミラーは親コピーからPoint-in-Timeメディア・コピーをデタッチするプロセスです。分割後、親コピーに対する更新は子コピーに影響しません。

Oracle Database 18c以降では、親コピーは、記憶域ボリュームではなく、PDBであることが可能です。分割ミラーによるクローンPDBは親と同じメディア上に存在します。主要なユースケースでは、Oracle ASM環境でテストおよび開発のPDBを迅速にプロビジョニングします。

ノート:



分割ミラーによるクローン PDB には、Oracle ASM の柔軟性および拡張ディスク・グループが必要です。

ミラー・リフレッシュによって、親PDBからの変更で分割ミラーによるクローンPDBをリフレッシュします。実際には、この操作は、ミラー分割を削除してから新しいミラー分割を取得するのと同じです。

分割ミラーによるクローンPDBを削除するには、ALTER PLUGGABLE DATABASE ... DROP MIRROR COPYを入力します。

分割ミラーによるクローンPDBを作成するには:

1. SQL*Plusを起動し、CDBルートに接続します。
2. ソースPDBは、ALTER PLUGGABLE DATABASE ... PREPARE MIRROR COPY文を発行することで準備します。
3. CREATE PLUGGABLE DATABASE ... FROM ... USING MIRROR COPY文を発行して、ソースPDBからクローンPDBを作成します。
4. 必要に応じて、V\$ASM_DBCLONE_INFOビューを問い合せて、ソースPDB、クローニングされたPDB、およびそのファイル・グループ間の関係を確認します。

関連項目:

- 分割ミラーによるクローンPDBを作成または削除する方法を学習するには、[『Oracle Automatic Storage』](#)

[Management管理者ガイド](#)』を参照してください

- V\$ASM_DBCLONE_INFOについてさらに学習するには、[『Oracle Databaseリファレンス』](#)を参照してください

親トピック: [PDBまたは非CDBのクローニング](#)

8 PDBの再配置

PDBを異なるCDBまたはアプリケーション・コンテナに移動できます。

- [PDBの再配置について](#)
再配置の際にソースPDBを読取り/書き込みモードでオープンすると、すべての機能を利用できます。
- [PDBの再配置の目的](#)
この手法は、最小限の停止時間または停止時間なしでPDBを移動する最速の方法です。それ以外の手法では、ソースPDBを切断すると、PDBがターゲットCDBに接続されるまでPDBが停止している必要があります。
- [PDB再配置の仕組み](#)
操作では、PDBに関連付けられているファイルを新しい場所に移動し、PDBをターゲットCDBに追加して、そのPDBをオープンします。
- [PDBの再配置のユーザー・インタフェース](#)
SQL、DBCAユーティリティまたはフリート・パッチ適用およびプロビジョニング・ユーティリティを使用すると、コマンドラインでPDBを再配置できます。
- [CREATE PLUGGABLE DATABASEを使用したPDBの再配置](#)
CREATE PLUGGABLE DATABASE ... RELOCATE文により、PDBを別のコンテナに移動します。
- [PDBの再配置: 例](#)
この項の例では、SQLおよびDBCAを使用した再配置の方法を示します。

親トピック: [PDBおよびアプリケーション・コンテナの作成および削除](#)

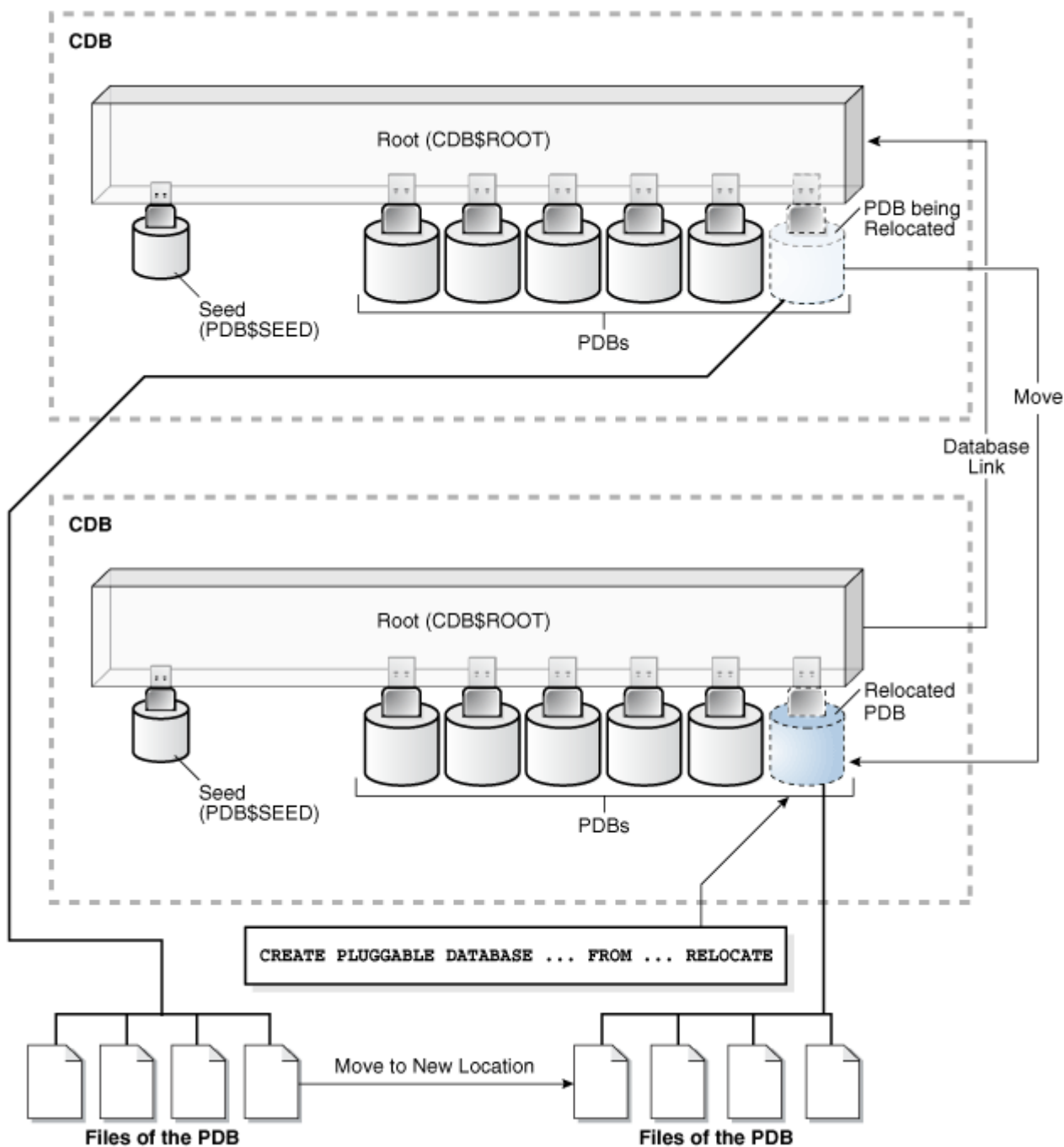
PDBの再配置について

再配置の際にソースPDBを読取り/書き込みモードでオープンすると、すべての機能を利用できます。

PDBの再配置では、ソースPDBがアクティブ・セッションでオープンされている間に、ソースPDBデータファイルのオンライン・ブロックレベル・コピー、REDOおよびUNDOを実行します。ALTER PLUGGABLE DATABASE OPEN文のためにターゲットPDBがオンラインになると、Oracle Databaseではアクティブ・セッションを終了し、ソースPDBをクローズします。

次の図は、新しい単一インスタンスCDBへの共通PDB(つまり、アプリケーションPDBでない)の再配置を示しています。ソースPDBはCDBルートに接続され、ターゲットPDBはCDBルートに接続されています。CREATE PLUGGABLE DATABASE ... RELOCATE文によって、データ・ブロック、UNDOブロックおよびREDOブロックが新しい場所にコピーされることに注意してください。データベース・リンクが必要です。

図8-1 ルート・コンテナへのPDBの再配置

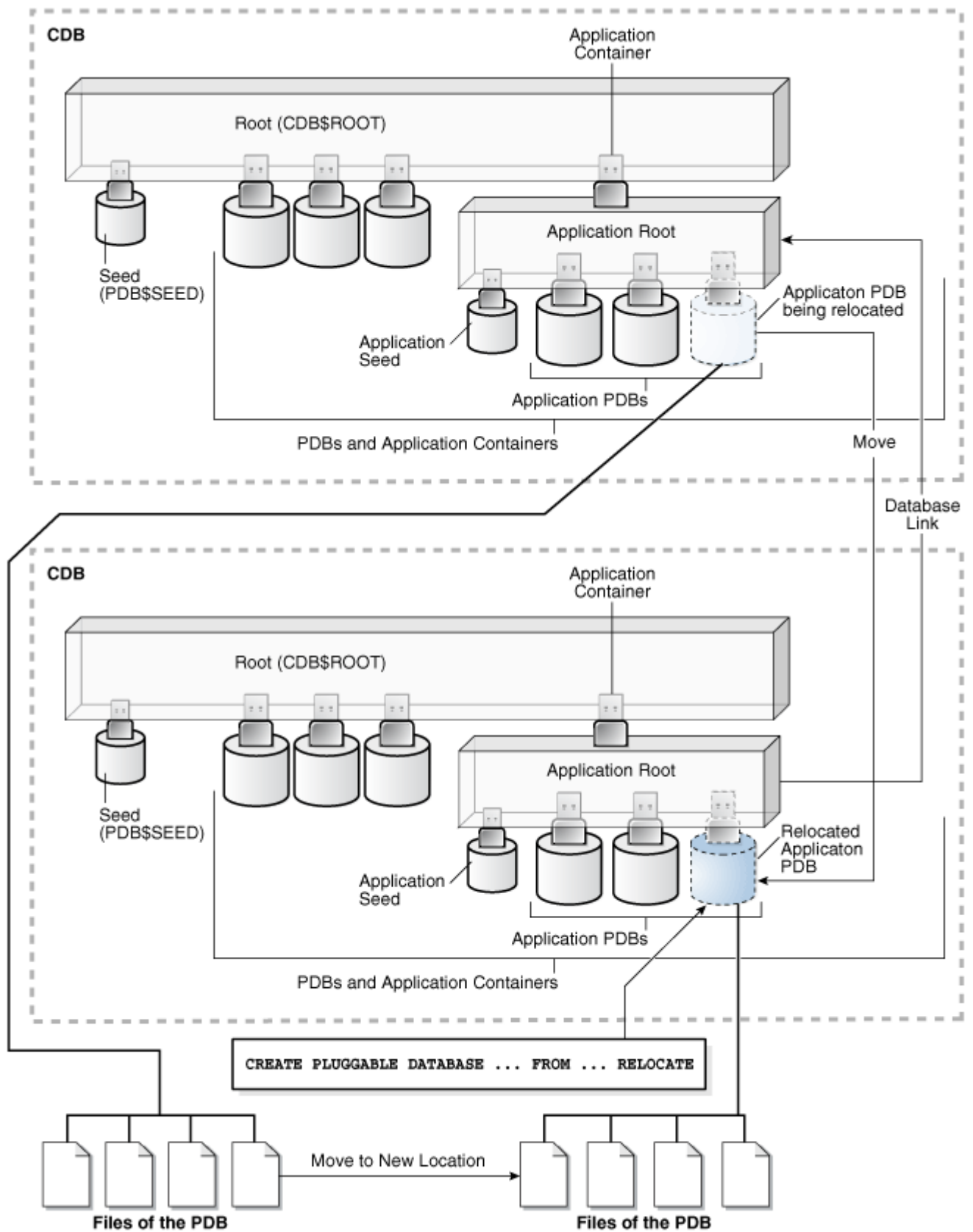


ターゲットPDBがアプリケーションPDBまたはアプリケーション・ルートである場合は、次のオプションがあります。

- PDBはアプリケーション・コンテナ内にアプリケーションPDBとして再配置できます。ターゲットPDBは同じCDBまたは別のCDB内のいずれにも指定できます。
- アプリケーションPDBは、あるアプリケーション・ルートから別のアプリケーション・ルートに再配置できます。ターゲットPDBは、別のCDB内にある必要があります。
- 空のアプリケーション・ルートのあるCDBから別のCDBに再配置できますが、そのアプリケーション・ルートにはホストされるアプリケーションPDBが存在しない必要があります。

次に図は、この手法で新しいアプリケーションPDBをアプリケーション・コンテナに作成する方法を示しています。

図8-2 アプリケーション・コンテナへのPDBの再配置



再配置されたPDBを初めてオープンすると、Oracle Databaseにより、ソースPDBのアクティブなセッションが排出され、再配置されたPDBサービスにクライアント接続がリダイレクトされます。再配置されたPDBをオープンすることで、元のソースPDBの停止が開始します。ソースおよび再配置されたPDBが同時にオープンすることはありません。

関連項目:

[「PDB記憶域」](#)

親トピック: [PDBの再配置](#)

PDBの再配置の目的

この手法は、最小限の停止時間または停止時間なしでPDBを移動する最速の方法です。それ以外の手法では、ソースPDBを切断すると、PDBがターゲットCDBに接続されるまでPDBが停止している必要があります。

データ・センター間で、またはオンプレミス環境からクラウド環境にPDBを移動する場合は、すべてのデータを物理的に移動する必要があります。大規模なPDBでは、このプロセスは非常に時間がかかり、SLAの可用性コンポーネントに違反する可能性があります。PDBの再配置は、停止が不要になります。アプリケーションをオフラインにしたり、アプリケーションを変更またはネットワーク接続文字列を変更せずに、PDBを再配置できます。

親トピック: [PDBの再配置](#)

PDB再配置の仕組み

操作では、PDBに関連付けられているファイルを新しい場所に移動し、PDBをターゲットCDBに追加して、そのPDBをオープンします。

- [PDBの再配置または停止時のサーバー・セッションの排出](#)
計画メンテナンスの重要な要件は、アプリケーションの動作が中断されないようにする、PDBのセッションの排出またはフェイルオーバーです。
- [PDBの再配置のステージ](#)
PDBの再配置の詳細は、リスナー・ネットワークによって異なります。

親トピック: [PDBの再配置](#)

PDBの再配置または停止時のサーバー・セッションの排出

計画メンテナンスの重要な要件は、アプリケーションの動作が中断されないようにする、PDBのセッションの排出またはフェイルオーバーです。

自動セッション・フェイルオーバー

データベースにおける一般的なセッション排出では、アクティブ・セッションは、タイマーに基づいて正常に終了できます。タイマーの期限が切れると、Oracle Databaseによってすべてのアクティブ・セッションが終了され、再配置されたPDBに再接続されます。

Oracle Database 19c以降では、計画メンテナンスの間に、データベースによって、セッションは排出期間内に排出する見込みがないと判断される場合があります。そのような場合は、データベースによってアプリケーション・コンティニューイティが起動され、セッションが自動的にフェイルオーバーされます。排出機能は、データベース・サービス・レベルおよびPDBレベルで呼び出されるすべてのメンテナンス操作(サービスの停止、サービスの再配置、PDBの再配置およびPDBの停止)に対して、デフォルトで有効になっています。

ノート:



アプリケーション・サーバーでプールのページプロパティが使用されている場合は、排出準備ができていないセッションが中断されるため、このプロパティを無効にします。

セッションの排出のルール

拡張可能な一連のルールを使用してルールを満たすまで続行され、データベース・セッションを排出するタイミングが決まります。ルールには次のものが含まれています。

- 標準アプリケーション・サーバーの有効性のテスト
- カスタムSQLの有効性のテスト
- リクエスト境界が使用され、リクエストがアクティブでないこと
- リクエスト境界が使用され、現在のリクエストが終了していること
- セッションにリカバリ可能なセッションの状態が1つ以上あり、フェイルオーバー時に再作成可能なこと

典型的なユースケースは、接続プールからの接続の借用時、プールへ接続の返却時、およびバッチ・コミット時に、接続をテストするアプリケーション・サーバーおよびプールされたアプリケーションです。セッションの排出時に、データベース接続テストが自動的に捕捉され、接続をクローズして、テストの失敗ステータスが返されます。失敗ステータスを受信した後、アプリケーション・レイヤーでは別の接続をリクエストできます。このように、アプリケーションが中断されることはありません。

Oracle RAC上のFANを使用したアプリケーション・コンティニューイティ

クライアントへの影響が最小限になる最適な構成にするために、Oracle RACデータベース上のFANを使用したアプリケーション・コンティニューイティを構成することを検討してください。Oracle Clusterwareでは、フリート・パッチ適用およびプロビジョニング機能により、PDBの再配置が自動化されます。Oracle RAC環境でのファイングレイン再配置の例は、PDBインスタンス間でのサービスの再配置です。Oracle RACとOracle Clusterwareでは、再配置中に、接続されているクライアントへの影響がさらに少なくなる上質な高可用性環境が提供されます。たとえば、共有記憶域では、データファイルをコピーする必要性が最小限に抑制または除去されます。アプリケーション・コンティニューイティのモードの1つである透過的アプリケーション・コンティニューイティは、Oracle Cloudでデフォルトで有効になっています。

ノート:



Oracle Clusterware 環境では、1 つの PDB を様々な CDB の間で再配置する場合は、SRVCTL を使用して非データベース・サービスを作成する必要があります。

関連項目:

[アプリケーション・コンティニューイティ](#)、[SRVCTL](#)および[フリート・パッチ適用とプロビジョニング](#)について学習するには、*Oracle Clusterware*管理およびデプロイメント・ガイドを参照してください

親トピック: [PDB再配置の仕組み](#)

PDBの再配置のステージ

PDBの再配置の詳細は、リスナー・ネットワークによって異なります。

- [共通のリスナー・ネットワークでのPDBの再配置](#)
ソースとターゲットの場所が共通のリスナー・ネットワークを共有している場合は、SQL*Netレイヤーがクライアント接続を暗黙的に転送するため、クライアント接続の転送は必要ありません。
- [独立したリスナー・ネットワークでのPDBの再配置](#)
独立したリスナーが相互登録を使用しない場合、ターゲットCDBとソースCDBのリスナーには、相互の情報またはそれぞれの公開されたサービスの情報がありません。

共通のリスナー・ネットワークでのPDBの再配置

ソースとターゲットの場所が共通のリスナー・ネットワークを共有している場合は、SQL*Netレイヤーがクライアント接続を暗黙的に転送するため、クライアント接続の転送は必要ありません。

AVAILABILITY NORMAL

リスナー・ネットワークが共通である場合は、CREATE PLUGGABLE DATABASE ... RELOCATEのAVAILABILITY NORMAL句を指定します。このオプションはデフォルトです。次の状況は、AVAILABILITY NORMALの一般的なユースケースを示しています。

- 共有リスナー

PDBの古い場所と新しい場所で同じリスナーを使用する場合、再配置が完了すると新しい接続は自動的にPDBの新しい場所にルーティングされます。この状況は、同じホストのCDB間での典型的な再配置です。この場合、PDBは新しい場所でリスナーに再登録されます。その他の接続処理は必要ありません。

- 相互登録リスナー

PDBが異なるリスナーを使用して、local_listenerおよびremote_listenerパラメータの構成を使用してそれぞれのリスナーの相互登録を利用する場合、再配置はシームレスです。PDBのサービスの可用性および場所が両方のリスナーに自動的に登録されます。この状況は、おそらくロード・バランシングを目的とした、データ・センター内のホスト間での典型的な再配置です。

共有リスナーおよび相互登録リスナーの環境では、すべてのデータベースからのサービスが共通のリスナー・ネットワークに公開されます。そのため、再配置されたPDBのサービスは共通のリスナー・ネットワークに即時に認識されます。サービス・ネームスペースの競合を回避するために、PDBのサービスの定義は、共通のリスナー・ネットワーク内で一意である必要があります。

共通のリスナー・ネットワークでの再配置のステージ

1. ユーザーがCREATE PLUGGABLE DATABASE ... RELOCATE AVAILABILITY NORMALを発行します。

このステップでは、元の場所からターゲットの場所にソースPDBのホット・クローンを実行します。ソースPDBは、SCNマーカーの暗黙的な開始時点でのデータファイル、UNDOブロックおよびREDOブロックをターゲットPDBにコピーします。

このステップが完了すると、このPDBに対して、トランザクション的に一貫性のある2つのコピー（ソース・コンテナ内のコピーとターゲット・コンテナ内のコピー）が存在することになります。操作中は、ソースPDBで処理が中断なく続行されます。ソースPDBに接続しているアプリケーションのユーザーまたはアプリケーションは、再配置が実行中であることを認識しません。

既存のすべてのアプリケーション接続およびこのステップの実行中に作成された新しい接続は、引き続きソースPDBに接続します。

2. ユーザーがALTER PLUGGABLE DATABASE OPENを発行します。

次の処理がバックグラウンドで実行されます。

- a. ターゲットPDBによって終了SCNのマーカーが暗黙的に設定され、暗黙的な終了SCNのマーカーを満たすためにメディア・リカバリの完了に必要なREDOまたはUNDOが適用されます。
- b. ターゲットPDBでメディア・リカバリが発生した場合は、ソースPDBでの排出アクティブ・セッションが開始されます。
- c. PDBサービスがリスナーに登録され、ターゲットCDBで使用できるようになります。
- d. ソースPDBがクローズされます。
- e. ターゲットPDBが読取り/書込みモードでオープンします。

このステップで、ターゲットCDBへのPDBの再配置が完了します。操作の終了時に、接続が、新しく再配置されたPDBを指すようになります。

PDBが読み取り/書き込みモードでオープンされると、そのステータスはNORMALになります。PDBを読み取り専用モードでオープンしようとすると、エラーが返されます。

関連項目:

- リスナーのリダイレクトの詳細は、『[Oracle Database Net Services管理者ガイド](#)』を参照してください
- 計画メンテナンス前にセッションを排出および移行するためのアプリケーション・コンティニューイティの使用についてさらに学習するには、『[Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)』を参照してください

親トピック: [PDBの再配置のステージ](#)

独立したリスナー・ネットワークでのPDBの再配置

独立したリスナーが相互登録を使用しない場合、ターゲットCDBとソースCDBのリスナーには、相互の情報またはそれぞれの公開されたサービスの情報がありません。

AVAILABILITY MAX

CREATE PLUGGABLE DATABASE ... RELOCATEのAVAILABILITY MAX句は、SQL*Netレイヤーに元のリスナーを再構成するように暗黙的に指示します。この状況は、データ・センター間でPDBを再配置する場合によくあります。この構成は、Oracle Internet Directory (OID)またはLDAPサーバーを更新する間、またはクライアント接続を変更する間に一時的に使用することが意図されています。

ローカル・リスナーがOracle RAC構成の単一クライアント・アクセス名(SCAN)リスナーにリダイレクトされる場合、このリスナーはクライアント接続リクエストを別のクラスタ・ノードにさらにリダイレクトすることが必要な場合があります。複数リダイレクトは、Oracle Netリスナーのデフォルトではサポートされません。SCANリスナーは任意のノードに接続リクエストをルーティングできるため、ALLOW_MULTIPLE_REDIRECTS_listener_nameパラメータをすべてのSCANリスナーのlistener_nameに設定して、それをクラスタ内のすべてのlistener.oraファイルで設定します。たとえば、SCANリスナーの名前がlistener_scan1、listener_scan2およびlistener_scan3の場合は、すべての宛先ホストのlistener.oraファイルに次の設定が必要になります。

```
ALLOW_MULTIPLE_REDIRECTS_LISTENER_SCAN1=YES  
ALLOW_MULTIPLE_REDIRECTS_LISTENER_SCAN2=YES  
ALLOW_MULTIPLE_REDIRECTS_LISTENER_SCAN3=YES
```

注意:



ALLOW_MULTIPLE_REDIRECTS_listener_name パラメータをノード・リスナーに設定すると、特定のネットワーク構成で無限のリダイレクト・ループが可能になるため、設定しないでください。

独立したリスナー・ネットワークでの再配置のステージ

1. ユーザーがCREATE PLUGGABLE DATABASE ... RELOCATE AVAILABILITY MAXを発行します。

このステップでは、元の場所からターゲットの場所にソースPDBのホット・クローンを実行します。ソースPDBは、SCN

マーカーの暗黙的な開始時点でのデータファイル、UNDOブロックおよびREDOブロックをターゲットPDBにコピーします。

2. ユーザーがALTER PLUGGABLE DATABASE OPENを発行します。

次の処理がバックグラウンドで実行されます。

- a. ターゲットPDBによって終了SCNのマーカーが暗黙的に設定され、暗黙的な終了SCNのマーカーを満たすためにメディア・リカバリの完了に必要なREDOまたはUNDOが適用されます。
 - b. ターゲットPDBでメディア・リカバリが発生した場合は、ソースPDBでの排出アクティブ・セッションが開始されます。
 - c. ソースPDBのLISTENER_NETWORKS初期化パラメータが転送先アドレスで暗黙的に更新され、ソースCDBのリスナーPDBサービスが転送先アドレスで更新されます。
 - d. メディア・リカバリが完了すると、ターゲットPDBが読取り専用モードでオープンします。
- このステージでは、ターゲットPDBの間合せのみが許可されます。間合せは、ソースPDBで実行された場合とまったく同様に動作します。ただし、DMLを試行する接続は完了しません。
- e. 読取り専用接続が新しいホスティング・リスナーに即時に転送され、新規の読取り/書込み接続が新しいホスティング・リスナーに転送され、そこでターゲットPDBが一貫性のある状態でオープンされるまでスピンします。
 - f. ソースPDBによって、永続的な接続を終了するSHUTDOWN IMMEDIATEが実行されます。
 - g. ターゲットPDBが読取り/書込みモードでオープンします。

このステップで、ターゲットCDBへのPDBの再配置が完了します。操作の終了時に、接続が、新しく再配置されたPDBを指すようになります。

PDBが読取り/書込みモードでオープンされると、そのステータスはNORMALになります。PDBを読取り専用モードでオープンしようとすると、エラーが返されます。

ノート:

更新が完了するまでツームストーン PDB と呼ばれるアーティファクトがソース CDB に残り、PDB のネームスペースを保護し、リスナー転送構成を維持します。ソース CDB のルートでは、ツームストーン PDB が RELOCATED ステータスで V\$CONTAINERS に表示されます。ターゲット PDB に直接接続を指定するためにアプリケーション接続文字列を変更した場合は、ソース CDB からツームストーン PDB を削除できます。

関連項目:

- [アプリケーションPDBの作成](#)
- リスナーのリダイレクトの詳細は、『[Oracle Database Net Services管理者ガイド](#)』を参照してください
- 計画メンテナンス前にセッションを排出および移行するためのアプリケーション・コンティニューイティの使用についてさらに学習するには、『[Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)』を参照してください

親トピック: [PDBの再配置のステージ](#)

PDBの再配置のユーザー・インタフェース

SQL、DBCAユーティリティまたはフリート・パッチ適用およびプロビジョニング・ユーティリティを使用すると、コマンドラインでPDBを再配置できます。

SQL文

SQL文の形式は、次のとおりです。

```
CREATE PLUGGABLE DATABASE ... FROM src_pdb_name@link2src ... RELOCATE AVAILABILITY [MAX | NORMAL]
```

FROM句で、ソースPDBの場所を特定します。src_pdb_nameには、ソースPDBの名前を指定します。link2srcには、ソースPDBの場所を示すデータベース・リンクを指定します。このデータベース・リンクは、ターゲットCDB (PDBの再配置先CDB)で作成しておく必要があります。リンクは、リモートCDBのルートまたはリモートPDBのどちらかに接続できます。

AVAILABILITY句では、データベースがクライアント接続を処理する方法を決定します。

DBCA

PDBは、サイレント・モードでDBCAを実行することで再配置できます。relocatePDBコマンドで、再配置を実行します。

表8-1 relocatePDBのパラメータ

パラメータ	説明
-remotePDBName remote_pdb_name	再配置する PDB の名前。
-remoteDBConnString remote_db_conn_string	リモート CDB へのネット・サービス接続。
-sysDBAUserName sysdbusername	ローカル CDB の SYS ユーザーの名前。
-sysDBAPassword sysdbapassowrd	ローカル CDB の SYS ユーザーのパスワード。
-remoteDBSYSDBAUserName sysdbusername	リモート CDB の SYS ユーザーの名前。
-remoteDBSYSDBAPassword sysdbapassowrd	リモート CDB の SYS ユーザーのパスワード。
-dbLinkUsername dblink_common_user_name	リモート CDB の共通ユーザーの名前。
-dbLinkUserPassword dblink_common_username_pwd	リモート CDB の共通ユーザーのパスワード。
-sourceDB dbname_pdb_toberelocated	ソース PDB の名前。
-pdbName pdbtoberecreated	再配置後の PDB の名前。

フリート・パッチ適用およびプロビジョニング・コントロール(RHPCTL)

Oracle Grid Infrastructureでは、フリート・パッチ適用およびプロビジョニングを使用して、あるCDBから別のCDBへのPDBの再配置を自動化できます。

関連項目:

- CREATE PLUGGABLE DATABASEの構文およびセマンティクスについては、『[Oracle Database SQL言語リファレンス](#)』を参照してください
- サイレント・モードのDBCAのコマンド・リファレンスは、『[Oracle Database管理者ガイド](#)』を参照してください
- フリート・パッチ適用およびプロビジョニングについてさらに学習するには、『[Oracle Clusterware管理およびデプロイメント・ガイド](#)』を参照してください

親トピック: [PDBの再配置](#)

CREATE PLUGGABLE DATABASEを使用したPDBの再配置

CREATE PLUGGABLE DATABASE ... RELOCATE文で、PDBの別のコンテナに移動します。

ターゲットCDB (宛先CDBとも呼ばれる)とは、PDBの再配置先となるCDBです。ターゲットPDBとは、再配置されるPDBです。CREATE PLUGGABLE DATABASE ... RELOCATE操作が完了すると、Oracle Databaseによって、PDBがソースCDBから宛先CDBに移動されます。

一般的な前提条件

[「表5-3」](#)のPDBの再配置に適用される質問に対処します。この表では、様々な要因に基づいて指定する必要があるCREATE PLUGGABLE DATABASEの句について説明しています。また、[PDB作成の一般的な前提条件](#)で説明している前提条件を満たします。

データベースのモードおよび状態の前提条件

次の前提条件を満たしている必要があります。

- ソースCDBはローカルUNDOモードである必要があります。
- ソースCDBでは、すべてのデータベース・インスタンスでPDBのサービスとオープン状態を保存する必要があります。管理者としてCDBルートにログインし、次の文を発行します。

```
ALTER PLUGGABLE DATABASE ALL SAVE STATE INSTANCES=ALL;
```

このステップにより、PDBの再配置操作の際、ターゲットCDBでPDBサービスが自動起動されるようになります。

- ターゲットCDBがARCHIVELOGモードでない場合は、操作中にターゲットPDBを読み取り専用でオープンする必要があります。この要件は、ターゲットCDBがARCHIVELOGモードの場合は適用されません。

ユーザー権限の前提条件

次の前提条件を満たしている必要があります。

- ターゲットCDBでは、現行ユーザーはCDBのルートのCREATE PLUGGABLE DATABASEシステム権限を持っている必要があります。
- 次の前提条件がデータベース・リンクに適用されます。
 - データベース・リンクで、宛先CDBからソースCDBへの接続が有効である必要があります。
 - ターゲットが標準PDBである場合、データベース・リンクはソースCDBのルートに接続する必要があります。ターゲットPDBがアプリケーションPDBである場合、データベース・リンクはそのアプリケーション・ルートに接続する必

必要があります。

- データベース・リンク・ユーザーがソースCDB内のCDBルートに接続する場合、このユーザーは共通ユーザーである必要があります。データベース・リンクがアプリケーション・ルートに接続されている場合、このユーザーには、CDB全体の共通ユーザー、またはアプリケーションの共通ユーザーのどちらかがなることができます。
- データベース・リンク・ユーザーには、CREATE PLUGGABLE DATABASEシステム権限またはSYSOPER管理権限が必要です。

プラットフォームおよび文字セットの前提条件

次の前提条件を満たしている必要があります。

- ソースCDBと宛先CDBのプラットフォームが次の要件を満たしている必要があります。
 - endiannessが同じである必要があります。
 - ソース・プラットフォームにインストールされているデータベース・オプションは、宛先プラットフォームにインストールされているデータベース・オプションと同じか、そのサブセットである必要があります。
- 宛先CDBの文字セットがAL32UTF8でない場合、ソースCDBと宛先CDBには、互換性のある文字セットと各国語文字セットが必要です。

宛先CDBの文字セットがAL32UTF8である場合、この要件は適用されません。

ノート:



Oracle Multitenant では、あるコンテナの LOB に対する異なる文字セットのコンテナによるデータ・リンク、拡張データ・リンクまたは CONTAINERS () 句を使用したアクセスはサポートされません。たとえば、CDB ルートと salespdb の文字セットが異なる場合、CDB ルートで実行される CONTAINERS () 問合せは salespdb の LOB にアクセスできません。

アプリケーション名とバージョンの前提条件

アプリケーションPDBを作成している場合は、ソースPDBとターゲット・アプリケーション・コンテナに同じアプリケーション名とバージョンを指定する必要があります。

PDBを再配置するには:

1. SQL*Plusで、CREATE PLUGGABLE DATABASEシステム権限を持つユーザーとしてターゲットCDBにログインします。
2. 現在のコンテナがターゲットCDBまたはターゲット・アプリケーション・コンテナのルートであることを確認します。
3. FROM句を指定してCREATE PLUGGABLE DATABASE ... RELOCATE文を実行します。

FROM句でソースPDBを指定し、RELOCATE句を含めます。PDBの古い場所から新しい場所に接続をリダイレクトするには、AVAILABILITY MAX句を指定します。他の句が必要な場合はそれらを指定します。

再配置したPDBはマウント・モードになり、そのステータスはRELOCATINGです。PDBのオープン・モードは、V\$PDBSビューのOPEN_MODE列を問い合わせることで確認できます。CDB_PDBSまたはDBA_PDBSビューのSTATUS列を問い合わせ、PDBのステータスを表示できます。

PDBの新しいデフォルト・サービスが作成されます。サービスの名前はPDBと同じになり、このサービスを使用してPDBにアクセスできます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があ

ります。

4. (オプション)ファイルのコピー操作のステータスを確認するには、V\$SESSION_LONGOPSを問い合わせます。

OPNAMES列は、データファイル・コピーにはkpdbfCopyTaskCbk、REDOファイルのコピーにはkcrfremnocが表示されま
す。

5. 新しいPDBを読み取り/書き込みモードでオープンします。

このステップは、CDBへの新しいPDBの統合を完了するために必要です。PDBが読み取り/書き込みモードでオープンされ
ると、そのステータスはNORMALになります。PDBを読み取り専用モードでオープンしようとすると、エラーが返されます。

6. PDBをバックアップします。

PDBは、バックアップしないとリカバリできません。

ノート:



PDBの再配置時にエラーが返された場合は、作成されたPDBがUNUSABLE状態である可能性があります。CDB_PDBS または DBA_PDBS ビューを問い合わせる PDB の状態を確認したり、アラート・ログをチェックして PDB の作成エラーについてさらに学習できます。使用不可の PDB は削除することのみができ、使用不可の PDB と同じ名前の PDB を作成できるようにするには、その前に削除する必要があります。

関連項目:

- [「CDBのUNDOモードについて」](#)
- [「PDBのオープン・モードの変更」](#)
- 文字セットおよび各国語文字セットの互換性の要件は、[『Oracle Databaseグローバル化・サポート・ガイド』](#)を参照してください
- PDBのバックアップの詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド』](#)を参照してください

親トピック: [PDBの再配置](#)

PDBの再配置: 例

この項の例では、SQLおよびDBCAを使用した再配置の方法を示します。

- [リモートCDBからのPDBの再配置](#)
この例では、リモートCDBから現在のCDBにpdb1という名前のPDBを再配置します。
- [DBCAを使用したPDBの再配置: 例](#)
この例では、DBCAを使用して、リモートCDBからローカルCDBにpdb1という名前のPDBを再配置し、再配置先でrelpdb1という名前に変更します。

親トピック: [PDBの再配置](#)

リモートCDBからのPDBの再配置

この例では、リモートCDBから現在のCDBにpdb1という名前のPDBを再配置します。

この例で、新しいPDBが属するルートはCREATE PLUGGABLE DATABASE文の実行時の現在のコンテナによって決まります。

- 現在のコンテナがCDBルートの場合、新しいPDBはCDBルートに作成されます。
- 現在のコンテナがアプリケーション・コンテナ内のアプリケーション・ルートの場合、新しいPDBはアプリケーション・ルート内にアプリケーションPDBとして作成されます。

この例では、様々な要因に基づき、pdb1という名前のPDBをリモートCDBから再配置します。この例では、次の要因を想定しています。

- 現行ユーザーには、ターゲットCDBのルートでのCREATE PLUGGABLE DATABASEシステム権限があります。
- ソースCDBへのデータベース・リンク名はlnk2srcです。このデータベース・リンクは、次のSQL文を使用して作成されました。

```
CREATE PUBLIC DATABASE LINK lnk2src CONNECT TO c##myadmin IDENTIFIED BY password USING 'MYCDB';
```

共通ユーザーc##myadminには、ソースCDB内のSYSOPER管理権限とCREATE PLUGGABLE DATABASEシステム権限があります。

- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。これらのファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所に移動されません。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。
- ソースPDBから再配置されたPDBに接続を自動的に再配置する必要があります。したがって、AVAILABILITY MAX句が含まれます。

次の文は、ソースCDBから現在のCDBにpdb1 PDBを再配置します。

```
CREATE PLUGGABLE DATABASE pdb1 FROM pdb1@lnk2src RELOCATE AVAILABILITY MAX;
```

親トピック: [PDBの再配置: 例](#)

DBCAを使用したPDBの再配置: 例

この例では、DBCAを使用して、リモートCDBからローカルCDBにpdb1という名前のPDBを再配置し、再配置先でrelpdb1という名前に変更します。

前提条件

この例では、次のことを前提としています。

- ローカル・データベースのユーザーには、ルート・コンテナでのCREATE PLUGGABLE DATABASE権限が付与されている。
- リモートCDBはローカルUNDOモードになっている。

- リモートCDBとローカルCDBは、ARCHIVELOGモードになっている。
- データベース・リンクの接続先リモートCDBの共通ユーザーには、CREATE PLUGGABLE DATABASE、SESSIONおよびSYSOPER権限が付与されている。
- ローカルCDBとリモートCDBは、同じオプションでインストールされている。

想定

この例では、次のことを前提としています。

- 再配置されたPDBを格納するCDBのホスト上でDBCAを実行しています。ローカルCDBの名前は、locpdb1です。
- リモート(ソース) CDBの名前はremcdb1で、ホストremcdb1hostに存在しています。リモートCDBのインスタンス名は、reminstです。
- 再配置されるPDBであるリモートPDBの名前は、rempdb1です。
- 共通ユーザーc##adminuser_remcdb1がremcdb1に存在しています。
- 管理ユーザーlocSYSには、locpdb1 (PDBの再配置先のCDB)に対するSYSDBA権限が付与されています。
- 管理ユーザーremSYSには、remcdb1 (再配置するPDBを格納しているCDB)に対するSYSDBA権限が付与されています。
- locpdb1への再配置後に、PDBの名前をrelpdb1に変更します。

次のsilentコマンドでは、rempdb1をlocpdb1に再配置します。

```
./dbca -silent
-relocatePDB
-sourceDB remcdb1
-remotePDBName rempdb1
-remoteDBConnString remcdb1host:1521/reminst
-remoteDBSYSDBAUserName remSYS
  -remoteDBSYSDBAUserPassword remsyspwd
-dbLinkUsername c##adminuser_remcdb1
  -dbLinkUserPassword pwd4dblinkusr
-sysDBAUserName locSYS
  -sysDBAPassword locsyspwd
-pdbName relpdb1
```

関連項目:

DBCAコマンドの構文およびセマンティクスは、[『Oracle Database管理者ガイド』](#)を参照

親トピック: [PDBの再配置: 例](#)

9 切断されたPDBの接続

切断されたPDBをCDBに接続することで、PDBを作成できます。

- [PDBのプラグイン操作について](#)
PDBを接続するには、CREATE PLUGGABLE DATABASEのUSING句を指定します。この句では、XMLメタデータ・ファイルまたは圧縮済アーカイブ・ファイル(.pdbファイル)を指定します。
- [切断されたPDBの接続](#)
CREATE PLUGGABLE DATABASE ... USING文を使用してPDBを接続します。
- [PDBとしての非CDBの適用](#)
DBMS_PDB.DESCRIBEプロシージャを使用して非CDBをPDBに適用(移動)できます。
- [切断されたPDBの接続後](#)
切断されたPDBの接続後に、ユーザーと表領域に関する特定のルールが適用されます。
- [切断されたPDBの接続: 例](#)
次の例では、様々な要因に基づき、/disk1/usr/salespdb.xmlファイルまたは/disk1/usr/sales.pdbファイルを使用して、salespdbという名前の切断されたPDBを接続します。

親トピック: [PDBおよびアプリケーション・コンテナの作成および削除](#)

PDBのプラグイン操作について

PDBを接続するには、CREATE PLUGGABLE DATABASEのUSING句を指定します。この句では、XMLメタデータ・ファイルまたは圧縮済アーカイブ・ファイル(.pdbファイル)を指定します。

- [XMLファイルとアーカイブ・ファイルについて](#)
XMLメタデータ・ファイルには、切断されたPDBおよびPDBに関連付けられているファイル(データファイルやウォレット・ファイルなど)について記述します。アーカイブ・ファイルには、XMLメタデータ・ファイルとPDBのファイルの両方が含まれます。
- [切断されたPDBをプラグインする場合のソース・ファイルの場所](#)
CREATE PLUGGABLE DATABASE ... USING文を使用して、切断されたPDBをCDBに接続します。
- [PDBとしての非CDBの適用について](#)
非CDBを記述するXMLファイルを生成するには、DBMS_PDB.DESCRIBEプロシージャを使用します。その後、切断されたPDBを接続するのと同じように、非CDBに接続します。

親トピック: [切断されたPDBの接続](#)

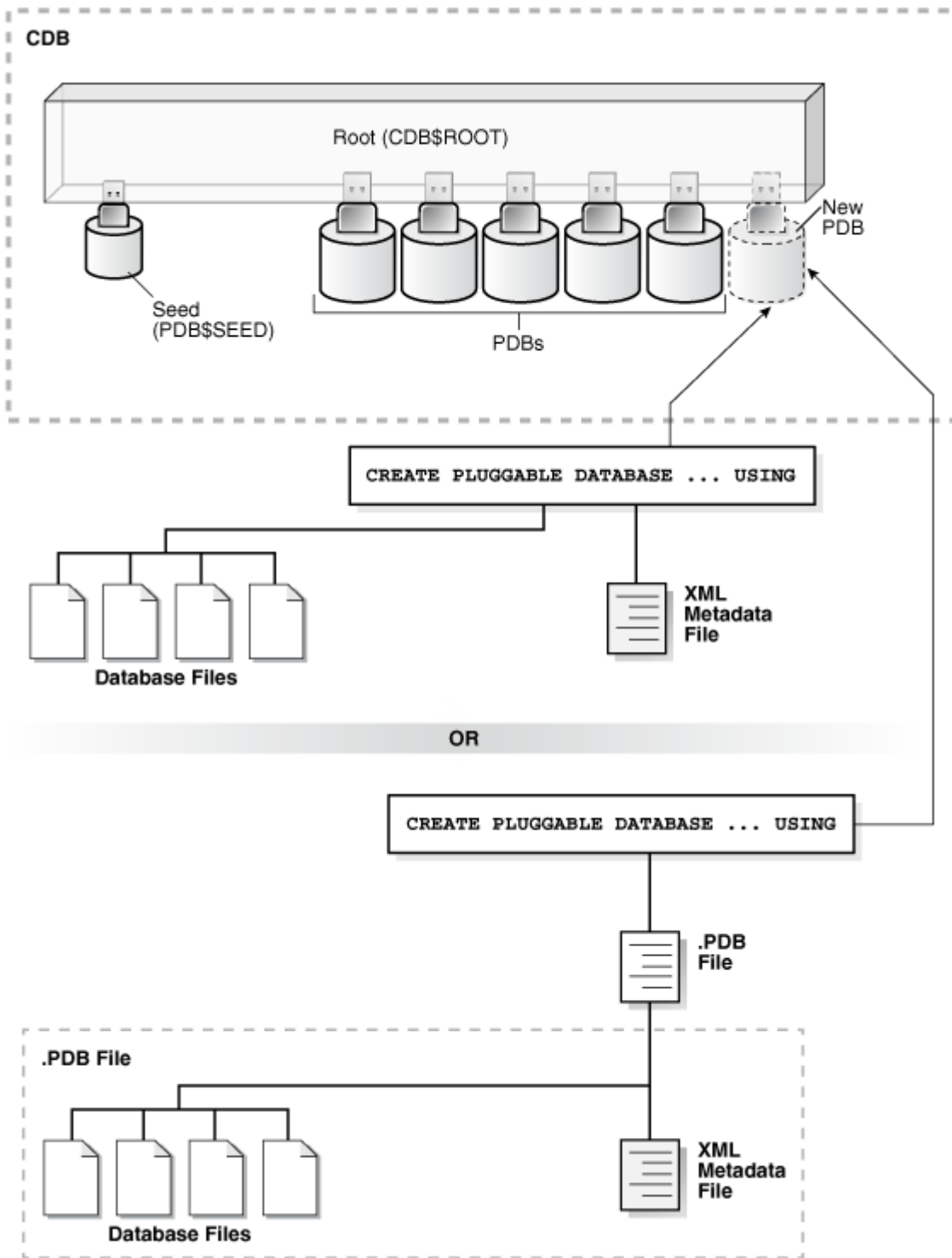
XMLファイルとアーカイブ・ファイルについて

XMLメタデータ・ファイルには、切断されたPDBおよびPDBに関連付けられているファイル(データファイルやウォレット・ファイルなど)について記述します。アーカイブ・ファイルには、XMLメタデータ・ファイルとPDBのファイルの両方が含まれます。

XMLメタデータ・ファイルが指定されている場合、XMLファイルにはPDBのファイルのフル・パスが含まれます。.pdbアーカイブ・ファイルが指定されている場合、XMLメタデータ・ファイルには、相対ファイル名のみが含まれます。

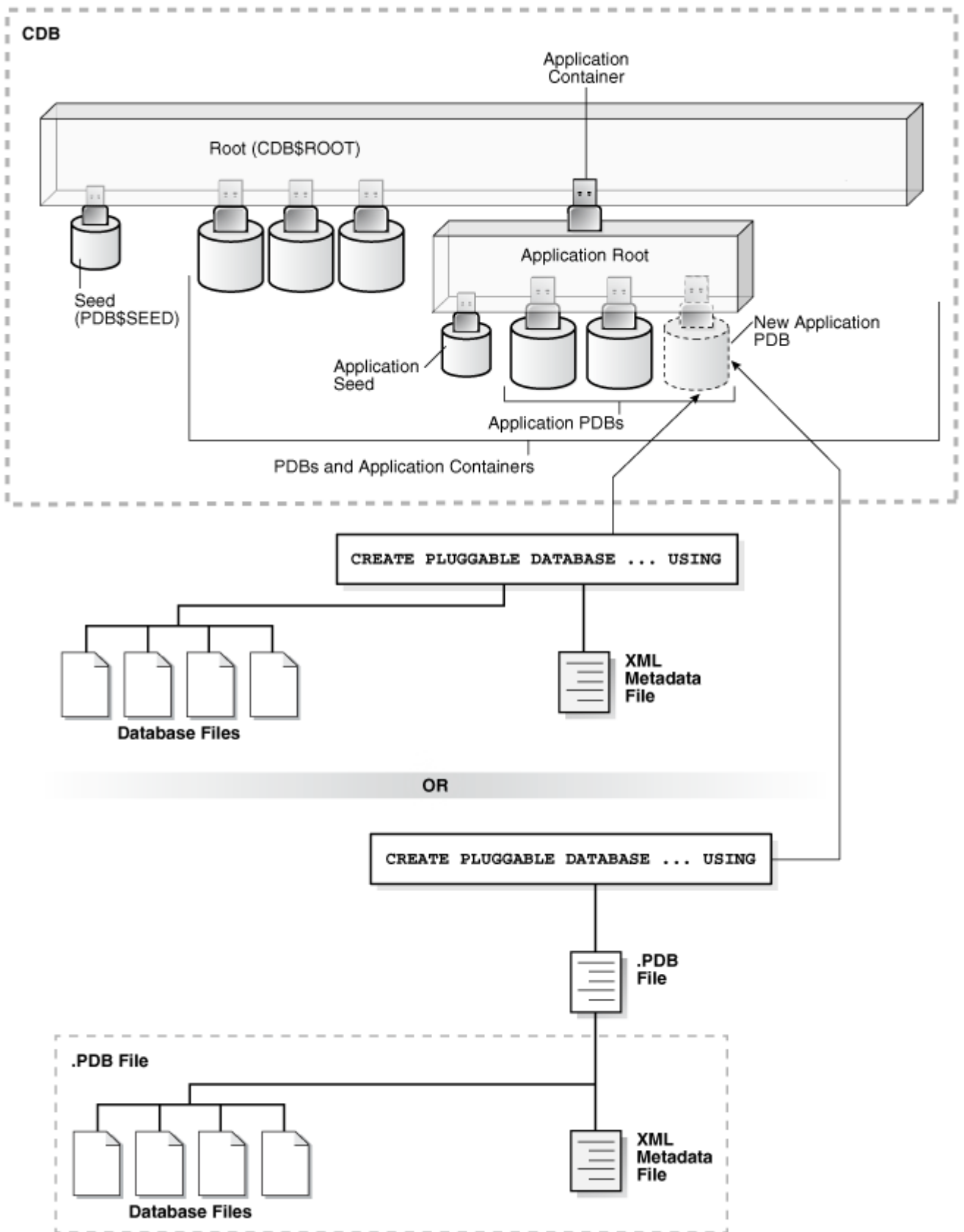
次の図は、切断されたPDBの接続方法を示しています。

図9-1 切断されたPDBのCDBルートへの接続



次に図は、この手法で新しいアプリケーションPDBをアプリケーション・コンテナに作成する方法を示しています。

図9-2 切断されたPDBのアプリケーション・ルートへの接続



ノート:



PDBの自動ダウングレードはサポートされていません。したがって、ソース CDB がターゲット CDB より上位の Oracle Database リリースである場合は、PDB を接続できません。

切断されたPDBに接続する場合は、[表5-3](#)の切断されたPDBの接続に適用される質問に対処する必要があります。この表では、様々な要因に基づいて指定する必要があるCREATE PLUGGABLE DATABASEの句について説明しています。

関連項目:

- [「PDB記憶域」](#)
- [アプリケーションPDBの作成](#)

親トピック: [PDBのプラグイン操作について](#)

切断されたPDBをプラグインする場合のソース・ファイルの場所

CREATE PLUGGABLE DATABASE ... USING文を使用して、切断されたPDBをCDBに接続します。

PDBの接続時に.pdbアーカイブ・ファイルを使用する場合、Oracle DatabaseはPDBの接続時にこのファイルを抽出し、PDBのファイルを.pdbアーカイブ・ファイルと同じディレクトリに配置します。したがって、ソース・ファイルの場所を指定する句は、.pdbアーカイブ・ファイルを使用するときには必要ありません。

PDBの接続時にXMLメタデータ・ファイルを指定する場合、このファイルは切断されたPDBソース・ファイルの名前と場所を記述します。切断されているファイルのあるストレージ・システムから異なるストレージ・システムにトランスポートした場合は、これらのファイルの場所がXMLファイルに正確に記述されていない可能性があります。これらのファイルは新しい場所にありますが、XMLファイル内のファイル・パスは依然として古い場所を示しています。

XMLメタデータ・ファイル(.pdbアーカイブ・ファイルではない)を使用して、切断されたPDBを接続する場合は、SOURCE_FILE_NAME_CONVERT句またはSOURCE_FILE_DIRECTORY句のいずれかを使用します。これらの句は相互に排他的です。

- [SOURCE_FILE_NAME_CONVERT句](#)
SOURCE_FILE_NAME_CONVERT句は、PDBファイルがXMLファイルの指定とは異なる場所に存在する場合にPDBファイルを検索する方法を指定します。
- [SOURCE_FILE_DIRECTORY句](#)
SOURCE_FILE_DIRECTORY句は、新しいPDBの作成に使用されるファイルのソース・ディレクトリを指定します。

親トピック: [PDBのプラグイン操作について](#)

SOURCE_FILE_NAME_CONVERT句

SOURCE_FILE_NAME_CONVERT句は、PDBファイルがXMLファイルの指定とは異なる場所に存在する場合にPDBファイルを検索する方法を指定します。

この句を使用して、次のいずれかのオプションを指定できます。

- 1つ以上のファイル名パターンおよび置換ファイル名パターンを次の形式で指定します。

```
'string1' , 'string2' , 'string3' , 'string4' , ...
```

string1ファイル名パターンがstring2ファイル名パターンによって置き換えられ、string3ファイル名パターンがstring4ファイル名パターンによって置き換えられます。ファイル名パターン文字列と置換ファイル名パターン文字列のペアを必要な数だけ使用できます。

この句を使用する場合は、PDBに使用するファイルが、置換ファイル名パターンに存在することを確認してください。必要な場合は、ファイルをこれらの場所にコピーまたは移動します。

- PDBのXMLファイルでファイル名が正確に記述されているためにファイル名を検索する必要がない場合はNONE。
SOURCE_FILE_NAME_CONVERT句の省略は、NONEの指定と同じです。

SOURCE_FILE_NAME_CONVERT句は、XMLメタデータ・ファイルを指定するUSING句とともにCREATE PLUGGABLE DATABASE文でのみ使用できます。したがって、この句は、切断されているPDBをXMLメタデータ・ファイルで接続する場合にのみ使用できます。PDBを.pdbアーカイブ・ファイルで接続する場合は、この句を使用できません。

例9-1 SOURCE_FILE_NAME_CONVERT句

次のSOURCE_FILE_NAME_CONVERT句では、/disk1/oracle/pdb7ディレクトリではなく、/disk2/oracle/pdb7ディレクトリのファイルを使用します。この場合、PDBを記述しているXMLファイルでは、/disk1/oracle/pdb7ディレクトリを指定していますが、PDBでは/disk2/oracle/pdb7ディレクトリのファイルを使用する必要があります。

```
SOURCE_FILE_NAME_CONVERT = ('/disk1/oracle/pdb7/', '/disk2/oracle/pdb7/')
```

関連項目:

- [切断されたPDBの接続](#)
- SOURCE_FILE_NAME_CONVERT句の構文は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。

親トピック: [切断されたPDBをプラグインする場合のソース・ファイルの場所](#)

SOURCE_FILE_DIRECTORY句

SOURCE_FILE_DIRECTORY句は、新しいPDBの作成に使用されるファイルのソース・ディレクトリを指定します。

句では、XMLファイルにリストされているすべてのファイルを含むディレクトリを指定します。この句は、多数のデータファイルがあり、各ファイルのSOURCE_FILE_NAME_CONVERTパターンの指定が現実的でない場合に使用すると便利です。

PDBを接続する際に、ソース・ファイルがすべて単一ディレクトリに存在する場合は、この句でディレクトリ名を指定できます。切断されたPDBのXMLファイルに基づいて適切なファイルを検索するためにディレクトリがスキャンされます。

この句を使用して、次のいずれかのオプションを指定できます。

- ソース・ファイル・ディレクトリの絶対パス。
- PDBの作成時にいずれのファイルもコピーまたは移動しない場合はNONEを指定します。SOURCE_FILE_DIRECTORY句の省略は、NONEの指定と同じです。

SOURCE_FILE_DIRECTORY句は、XMLメタデータ・ファイルを指定するUSING句とともにCREATE PLUGGABLE DATABASE文でのみ使用できます。したがって、この句は、切断されているPDBをXMLメタデータ・ファイルで接続する場合にのみ使用できます。PDBを.pdbアーカイブ・ファイルで接続する場合は、この句を使用できません。

この句は、Oracle Managed Filesを使用する構成とOracle Managed Filesを使用しない構成に対して指定できます。

例9-2 SOURCE_FILE_DIRECTORY句

このSOURCE_FILE_DIRECTORY句は、/oracle/pdb5/ディレクトリ内のソース・ファイルを使用して新しいPDBのファイル名を生成します。

```
SOURCE_FILE_DIRECTORY = '/oracle/pdb5/'
```

関連項目:

- [切断されたPDBの接続](#)

- SOURCE_FILE_DIRECTORY句の構文は、『Oracle Database SQL言語リファレンス』を参照してください

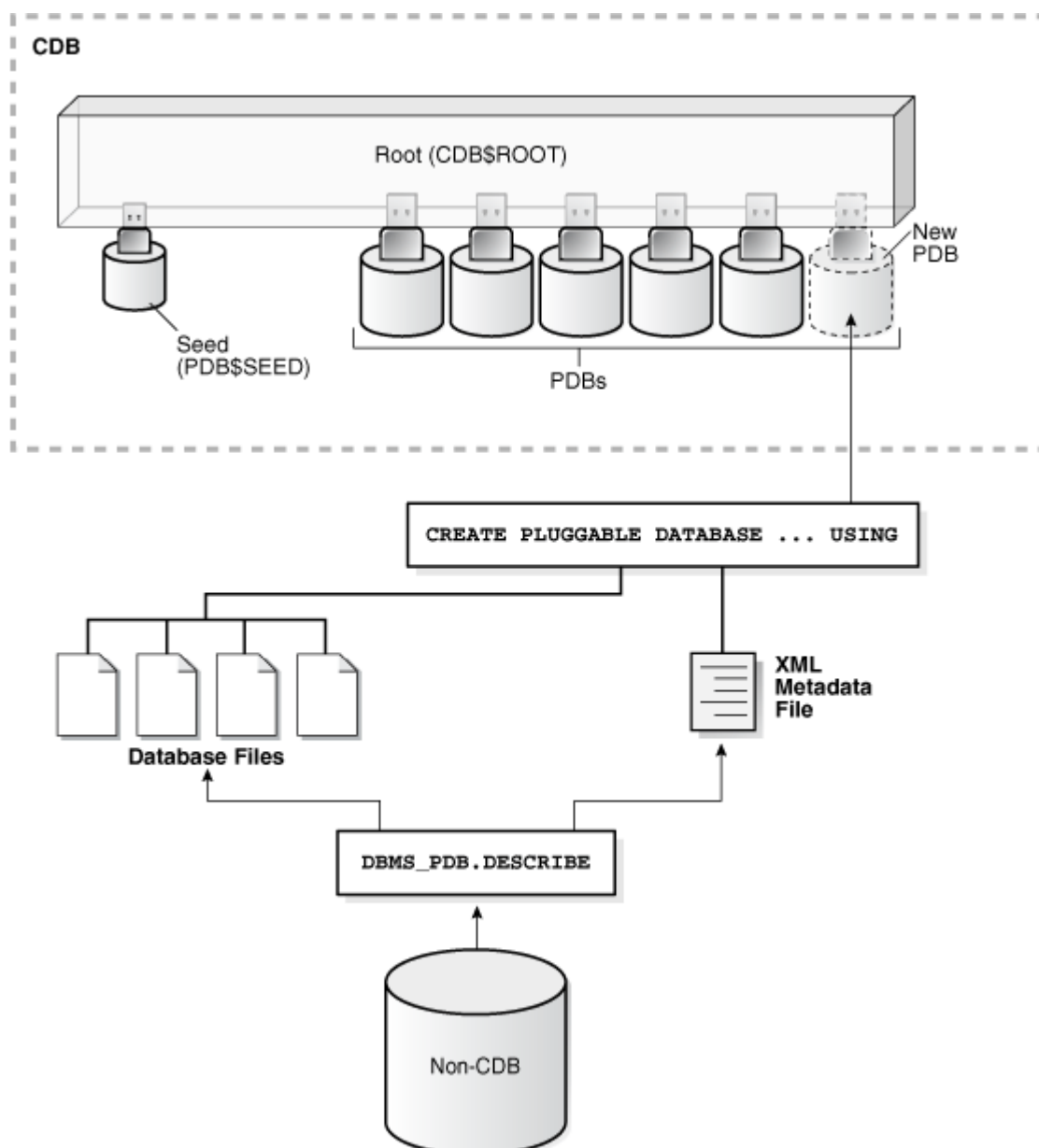
親トピック: [切断されたPDBをプラグインする場合のソース・ファイルの場所](#)

PDBとしての非CDBの適用について

非CDBを記述するXMLファイルを生成するには、DBMS_PDB.DESCRIBEプロシージャを使用します。その後、切断されたPDBを接続するのと同じように、非CDBに接続します。

CREATE PLUGGABLE DATABASE ... USING文を使用してPDBを作成します。非CDBがCDBに接続された場合、その非CDBはPDBになります。

図9-3 DBMS_PDB.DESCRIBEプロシージャを使用した非CDBの接続



アプリケーション・コンテナに新しいアプリケーションPDBを作成するのと同じ手法を使用できます。

ノート:

この手法を使用するには、非CDBがリリース Oracle Database 12c 以降である必要があります。

関連項目:

[「切断されたPDBの接続」](#)

親トピック: [PDBのプラグイン操作について](#)

切断されたPDBの接続

CREATE PLUGGABLE DATABASE ... USING文を使用してPDBを接続します。

一般的な前提条件

切断されたPDBを接続するには、次の要件が満たされている必要があります。

- [「PDB作成の一般的な前提条件」](#)で説明している前提条件を満たします。
- PDBを記述しているXMLファイルまたは.pdbアーカイブ・ファイルは、CDBにアクセスできる場所に存在する必要があります。

USING句でXMLファイルまたは.pdbアーカイブ・ファイルを指定する必要があります。PDBのXMLファイルが使用できないか見つからない場合は、DBMS_PDB.RECOVERプロシージャを使用して、PDBのデータファイルでXMLファイルを生成できます。

- USING句にXMLファイル(.pdbファイルではなく)を指定する場合は、PDBに関連付けられたファイル(データファイルやウォレット・ファイルなど)が、CDBにアクセスできる場所に存在している必要があります。
- 接続操作のターゲット・データベースがOracle Data Guard構成内のプライマリ・データベースの場合は、接続されるPDBのファイルの場所をスタンバイ・データベースで特定できることを確認します。

スタンバイ・データベースで、STANDBY_PDB_SOURCE_FILE_DIRECTORY初期化パラメータを、PDBをインスタンス化するためのソース・データファイルが含まれている場所に設定します。ファイルが見つからない場合、スタンバイ・データベースはOMFの場所でファイルを探します。OMFの場所で見つからない場合は、データファイルをスタンバイ・データベースのOMFの場所にコピーし、スタンバイ・データベースに対するREDO適用を再起動する必要があります。

- ソースCDBとターゲットCDBのプラットフォームは、次の要件を満たしている必要があります。
 - endiannessが同じである必要があります。
 - ソース・プラットフォームにインストールされているデータベース・オプションは、ターゲット・プラットフォームにインストールされているデータベース・オプションと同じか、サブセットである必要があります。
- アプリケーションPDBを作成している場合、切断されたPDBのアプリケーション名およびバージョンは、アプリケーションPDBを接続するアプリケーション・コンテナのアプリケーション名およびバージョンと一致する必要があります。

文字セットの前提条件

文字セットを一致させるために、次の前提条件を満たす必要があります。

- PDBが接続されるCDBの文字セットがAL32UTF8でない場合、切断されたPDBを含んでいたCDBとターゲットCDBには互換性のある文字セットおよび各国語文字セットが必要です。互換性を確保するには、文字セットと各国語文字セットが、[『Oracle Databaseグローバル化・サポート・ガイド』](#)で指定されている要件を満たしている必要があります。

PDBが接続されるCDBの文字セットがAL32UTF8である場合、この要件は適用されません。

ノート:



Oracle Multitenant では、あるコンテナの LOB に対する異なる文字セットのコンテナによるデータ・リンク、拡張データ・リンクまたは CONTAINERS () 句を使用したアクセスはサポートされません。たとえば、CDB ルートと salespdb の文字セットが異なる場合、CDB ルートで実行される CONTAINERS () 問合せは salespdb の LOB にアクセスできません。

- アプリケーションPDBを作成している場合、アプリケーションPDBはアプリケーション・コンテナと同じ文字セットおよび各国語文字セットである必要があります。

CDBのデータベース文字セットがAL32UTF8の場合、アプリケーション・コンテナの文字セットおよび各国語文字セットはCDBと異なっても構いません。ただし、アプリケーション・コンテナ内のすべてのアプリケーションPDBには、アプリケーション・コンテナと同じ文字セットおよび各国語文字セットが必要です。

前提条件が満たされているかどうかを判断するには、DBMS_PDB.CHECK_PLUG_COMPATIBILITY関数を使用します。次の手順のステップ2では、この関数の使用方法を示しています。

PDBをプラグインするには:

1. SQL*Plusで、現在のコンテナがCDBルートまたはターゲットCDBのアプリケーション・ルートであることを確認します。
現在のコンテナがCDBルートの場合、PDBはCDBに作成されます。現在のコンテナがアプリケーション・ルートの場合、アプリケーションPDBはアプリケーション・コンテナ内に作成されます。
2. (オプション)DBMS_PDB.CHECK_PLUG_COMPATIBILITY関数を実行して、切断されたPDBがCDBと互換性があるかどうかを確認します。
 - a. PDBがまだ切断されていない場合は、DBMS_PDB.DESCRIBEプロシージャを実行して、そのPDBを記述するXMLファイルを作成します。

PDBがすでに切断されている場合は、ステップ2bに進みます。

たとえば、salespdb.xmlという名前のXMLファイルを/disk1/oracleディレクトリに生成し、次のプロシージャを実行します。

```
BEGIN
  DBMS_PDB.DESCRIBE (
    pdb_descr_file => '/disk1/oracle/salespdb.xml',
    pdb_name       => 'SALESPDB');
END;
/
```

PDBがリモートCDBに存在している場合は、pdb_nameパラメータに@database_link_nameを含めることができます。database_link_nameは、リモートCDBまたはPDBへの有効なデータベース・リンクの名前です。たとえば、リモートCDBへのデータベース・リンクの名前がrcdbの場合、pdb_name値にSALESPDB@rcdbと設定します。

- b. DBMS_PDB.CHECK_PLUG_COMPATIBILITY関数を実行します。

この関数を実行する場合は、次のパラメータを設定します。

- pdb_descr_file - パラメータをXMLファイルへのフルパスに設定します。

- `pdb_name`: 新しいPDBの名前を指定します。このパラメータを省略した場合は、XMLファイル内のPDB名が使用されます。

たとえば、`/disk1/usr/salespdb.xml`ファイルで記述されているPDBが現在のCDBと互換性があるかどうかを確認するには、次のPL/SQLブロックを実行します。

```
SET SERVEROUTPUT ON
DECLARE
  compatible CONSTANT VARCHAR2(3) :=
    CASE DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
      pdb_descr_file => '/disk1/usr/salespdb.xml',
      pdb_name       => 'SALESPDB')
    WHEN TRUE THEN 'YES'
    ELSE 'NO'
END;
BEGIN
  DBMS_OUTPUT.PUT_LINE(compatible);
END;
/
```

出力がYESの場合はPDBに互換性があり、次のステップに進むことができます。出力がNOの場合はPDBに互換性がなく、`PDB_PLUG_IN_VIOLATIONS`ビューをチェックして、互換性がない理由を確認します。



ノート:

`.pdb` アーカイブ・ファイルは `pdb_descr_file` パラメータで指定できます。

3. PDBが切断されていない場合は、切断します。
4. `CREATE PLUGGABLE DATABASE ... USING`文を実行し、`USING`句でXMLファイルまたは`.pdb`アーカイブ・ファイルを指定します。他の句が必要な場合はそれらを指定します。

作成したPDBはマウント・モードになり、そのステータスはNEWです。PDBのオープン・モードは、`V$PDBS`ビューの`OPEN_MODE`列を問い合わせることで確認できます。CDB_PDBSまたはDBA_PDBSビューの`STATUS`列を問い合わせ、PDBのステータスを表示できます。

PDBの新しいデフォルト・サービスが作成されます。サービスの名前はPDBと同じになり、このサービスを使用してPDBにアクセスできます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。

5. 新しいPDBを読取り/書込みモードでオープンします。

Oracle Databaseで新しいPDBをCDBに統合するには、PDBを読取り/書込みモードでオープンする必要があります。PDBを読取り専用モードでオープンしようとすると、エラーが返されます。PDBが読取り/書込みモードでオープンされると、そのステータスはNORMALになります。

6. PDBをバックアップします。

PDBは、バックアップしないとリカバリできません。



ノート:

PDBの作成時にエラーが返された場合は、作成されたPDBがUNUSABLE状態である可能性があります。

CDB_PDBS または DBA_PDBS ビューを問い合わせる PDB の状態を確認したり、アラート・ログをチェックして PDB の作成エラーについてさらに学習できます。使用不可の PDB は削除することのみができ、使用不可の PDB と同じ名前の PDB を作成できるようにするには、その前に削除する必要があります。

関連項目:

- [「CDBからのPDBの切断」](#)
- 詳細は、[「PDBのオープン・モードの変更」](#)を参照してください。
- PDBのバックアップの詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』](#)を参照してください。
- Oracle Data Guard環境でのPDBの接続についてさらに学習するには、[『Oracle Data Guard概要および管理』](#)を参照してください
- PDB_FILE_NAME_CONVERT初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください
- このプロシージャの詳細は、[『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)を参照してください。

親トピック: [切断されたPDBの接続](#)

PDBとしての非CDBの適用

DBMS_PDB.DESCRIBEプロシージャを使用して非CDBをPDBに適用(移動)できます。

DBMS_PDBパッケージを使用して非CDBをPDBとして適用するには:

1. CDBが存在しない場合は作成します。
2. 非CDBがトランザクション的に一貫性のある状態であることを確認します。
3. 非CDBを読み取り専用モードにします。
4. 非CDBに接続し、DBMS_PDB.DESCRIBEプロシージャを実行して、非CDBを記述するXMLファイルを作成します。

現行ユーザーには、SYSDBA管理権限が必要です。ユーザーは、接続時にAS SYSDBAを使用して権限を行使する必要があります。

たとえば、ncdb.xmlという名前のXMLファイルを/disk1/oracleディレクトリに生成し、次のプロシージャを実行します。

```
BEGIN
  DBMS_PDB.DESCRIBE (
    pdb_descr_file => '/disk1/oracle/ncdb.xml');
END;
/
```

このプロシージャが正常に完了した後、XMLファイルおよび非CDBデータベース・ファイルを使用して、非CDBをCDBに接続できます。

5. DBMS_PDB.CHECK_PLUG_COMPATIBILITYファンクションを実行して、非CDBがCDBと互換性があるかどうかを確認します。

このファンクションを実行する場合は、次のパラメータを設定します。

- `pdb_descr_file` - パラメータをXMLファイルへのフルパスに設定します。

- `pdb_name`: 新しいPDBの名前を指定します。このパラメータを省略した場合は、XMLファイル内のPDB名が使用されます。

たとえば、`/disk1/oracle/ncdb.xml`ファイルで記述されている非CDBが現在のCDBと互換性があるかどうかを確認するには、次のPL/SQLブロックを実行します。

```
SET SERVEROUTPUT ON
DECLARE
  compatible CONSTANT VARCHAR2(3) :=
    CASE DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
      pdb_descr_file => '/disk1/oracle/ncdb.xml',
      pdb_name       => 'NCDB')
    WHEN TRUE THEN 'YES'
    ELSE 'NO'
END;
BEGIN
  DBMS_OUTPUT.PUT_LINE(compatible);
END;
/
```

出力がYESの場合は非CDBに互換性があり、次のステップに進むことができます。出力がNOの場合は非CDBに互換性がなく、`PDB_PLUG_IN_VIOLATIONS`ビューをチェックして、互換性がない理由を確認できます。続行するには、すべての違反を修正する必要があります。たとえば、アップグレード・ユーティリティまたはデータパッチ・ユーティリティを実行して、すべてのバージョン不一致またはパッチ不一致を解決する必要があります。違反を修正してから、`DBMS_PDB.CHECK_PLUG_COMPATIBILITY`を再度実行して、非CDBがCDBと互換性があることを確認します。

6. 非CDBを停止します。

7. 非CDBを接続します。

たとえば、次のSQL文は非CDBをプラグインし、そのファイルを新しい場所にコピーして、非CDBから`tbs3`ユーザー表領域のみを含めます。

```
CREATE PLUGGABLE DATABASE ncdb USING '/disk1/oracle/ncdb.xml'
COPY
FILE_NAME_CONVERT = ('/disk1/oracle/dbs/', '/disk2/oracle/ncdb/')
USER_TABLESPACES=('tbs3');
```

違反がない場合は、新しいPDBをオープンしないでください。次のステップでオープンします。

`USER_TABLESPACES`句を使用すると、非CDB内の複数のテナントに使用されたデータを異なるPDBに分けることができます。この句を指定した`CREATE PLUGGABLE DATABASE`文を複数使用して、非CDBに存在する他の表領域のデータを含む別のPDBを作成できます。

8. `ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql`スクリプトを実行します。このスクリプトは、PDBを初めてオープンする前に実行する必要があります。

PDBが非CDBでない場合、`noncdb_to_pdb.sql`スクリプトを実行する必要はありません。`noncdb_to_pdb.sql`スクリプトを実行するには、次のステップを実行します。

- PDBにアクセスします。

現在のユーザーにはSYSDBA管理権限が必要であり、その権限は共通で付与されているか、またはPDBでローカルに付与されている必要があります。ユーザーは、接続時にAS SYSDBAを使用して権限を行使する必要があります。

- `noncdb_to_pdb.sql`スクリプトを次のように実行します。

スクリプトによって、PDBのオープン、変更の実行、および変更の完了時にPDBのクローズが行われます。


9. 新しいPDBを読み取り/書き込みモードでオープンします。

Oracle Databaseで新しいPDBをCDBに統合するには、PDBを読み取り/書き込みモードでオープンする必要があります。PDBを読み取り専用モードでオープンしようとすると、エラーが返されます。PDBが読み取り/書き込みモードでオープンされると、そのステータスはNORMALになります。

10. PDBをバックアップします。

PDBは、バックアップしないとリカバリできません。

ノート:



PDBの作成時にエラーが返された場合は、作成されたPDBがUNUSABLE状態である可能性があります。CDB_PDBS または DBA_PDBS ビューを問い合せてPDBの状態を確認したり、アラート・ログをチェックしてPDBの作成エラーについてさらに学習できます。使用不可のPDBは、削除のみ可能であり、使用不可のPDBと同じ名前のPDBを作成するには削除する必要があります。

関連トピック

- [切断されたPDBの接続](#)
- [CDBの作成および構成](#)
- [切断されたPDBの接続後](#)
- [『Oracle Databaseバックアップおよびリカバリ・アドバンスド・ユーザーズ・ガイド』](#)

親トピック: [切断されたPDBの接続](#)

切断されたPDBの接続後

切断されたPDBの接続後に、ユーザーと表領域に関する特定のルールが適用されます。

切断されたPDBの接続後、次の内容が適用されます。

- ソースPDBのデフォルト一時表領域を使用していたPDBのユーザー・アカウントは、ターゲットPDBのデフォルト一時表領域を使用します。ソースPDB内のデフォルト以外の一時表領域を使用していたユーザー・アカウントは、ターゲットPDB内の同じローカル一時表領域を引き続き使用します。
- ソースCDBに存在するが、ターゲットには存在しない、手動で作成された共通ユーザー・アカウントには、権限は共通して付与されていません。ただし、ターゲットCDBに共通ユーザーが存在し、PDBの共通ユーザーと同じ名前である場合、後者は前者にリンクされ、ターゲットCDBの共通ユーザーに付与された権限を持ちます。

クローニングまたは接続されたPDBにターゲットCDBに存在しない共通ユーザー・アカウントが存在し、このユーザーがPDB内のオブジェクトを所有していない場合、Oracle Databaseでは同期ステップ中にユーザーが削除されます。それ以外の場合、ユーザー・アカウントはターゲットPDBでロックされます。ロックされたアカウントに関して、次のオプションがあります。

- PDBをクローニングし、ルートに接続し、同じ名前で作成された共通ユーザー・アカウントを作成します。PDBを読み取り/書き込みモードでオープンすると、ユーザー・アカウントに共通して付与されているロールや権限の相違点が解決さ

れ、アカウントをロック解除できます。ユーザー・アカウントに対してローカルに付与されている権限やロールは、このプロセス中も変更されません。

- PDBに新しいローカル・ユーザー・アカウントを作成し、データ・ポンプを使用して、ロックされたユーザーのデータを新しいローカル・ユーザーのスキーマにエクスポート/インポートします。
- ユーザー・アカウントはロックされたままにします。
- ユーザー・アカウントを削除します。

関連項目:

- [「PDBのサービスの管理」](#)
- [「CDB内表領域の管理について」](#)
- 共通ユーザーおよびローカル・ユーザーの詳細は、[『Oracle Database概要』](#)を参照してください
- CDBにおける共通ユーザーおよびローカル・ユーザーの作成方法の詳細は、[『Oracle Databaseセキュリティガイド』](#)を参照してください
- CDBでのOracle Data Pumpの使用の詳細は、[『Oracle Databaseセキュリティ』](#)を参照してください

親トピック: [切断されたPDBの接続](#)

切断されたPDBの接続: 例

次の例では、様々な要因に基づき、/disk1/usr/salespdb.xmlファイルまたは/disk1/usr/sales.pdbファイルを使用して、salespdbという名前の切断されたPDBを接続します。

各例で、新しいPDBが属するルートはCREATE PLUGGABLE DATABASE文の実行時の現在のコンテナによって決まります。

- 現在のコンテナがCDBルートの場合、新しいPDBはCDBに作成されます。
- 現在のコンテナがアプリケーション・ルートの場合、新しいアプリケーションPDBはアプリケーション・ルート内のアプリケーション・コンテナ内に作成されます。

例9-3 NOCOPY句を使用して切断されたPDBを接続する方法

この例では、次の要因を想定しています。

- 新しいPDBは、CDB内に既存のPDBを作成するために使用された同じ切断されたPDBには基づいていません。したがって、AS CLONE句は必要ありません。
- PATH_PREFIX句は必要ありません。
- XMLファイルにファイルの現在の場所が正確に記述されています。したがって、SOURCE_FILE_NAME_CONVERTまたはSOURCE_FILE_DIRECTORY句は不要です。
- ファイルは、正しい場所に存在します。したがって、NOCOPYが含まれています。
- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、XMLファイルで指定されている一時ファイルと同じ名前のファイルがあります。したがって、TEMPFILE REUSE句が必要です。

前述の要因に基づき、次の文によってPDBを接続します。


```
CREATE PLUGGABLE DATABASE salespdb USING '/disk1/usr/salespdb.xml'  
NOCOPY  
TEMPFILE REUSE;
```

例9-4 AS CLONE句およびNOCOPY句を使用して切断されたPDBを接続する方法

この例では、次の要因を想定しています。

- 新しいPDBは、CDB内に既存のPDBを作成するために使用された同じ切断されたPDBに基づいています。したがって、AS CLONE句が必要です。AS CLONE句は、新しいPDBに一意的識別子があることを保証します。
- PATH_PREFIX句は必要ありません。
- XMLファイルにファイルの現在の場所が正確に記述されています。したがって、SOURCE_FILE_NAME_CONVERTまたはSOURCE_FILE_DIRECTORY句は不要です。
- ファイルは、正しい場所に存在します。したがって、NOCOPYが含まれています。
- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、XMLファイルで指定されている一時ファイルと同じ名前のファイルがあります。したがって、TEMPFILE REUSE句が必要です。

前述の要因に基づき、次の文によってPDBを接続します。

```
CREATE PLUGGABLE DATABASE salespdb AS CLONE USING '/disk1/usr/salespdb.xml'  
NOCOPY  
TEMPFILE REUSE;
```

例9-5 SOURCE_FILE_NAME_CONVERT、NOCOPYおよびSTORAGEの各句を使用して切断されたPDBを接続する方法

この例では、次の要因を想定しています。

- 新しいPDBは、CDB内に既存のPDBを作成するために使用された同じ切断されたPDBには基づいていません。したがって、AS CLONE句は必要ありません。
- PATH_PREFIX句は必要ありません。
- XMLファイルでは、ファイルの現在の場所が正確に記述されていません。したがって、SOURCE_FILE_NAME_CONVERT句またはSOURCE_FILE_DIRECTORY句が必要です。この例では、ファイルが/disk1/oracle/salesに存在することがXMLファイルで示されていますが、ファイルは/disk2/oracle/salesにあり、SOURCE_FILE_NAME_CONVERT句が使用されています。
- ファイルは、正しい場所に存在します。したがって、NOCOPYが含まれています。
- PDBにストレージ制限を強制する必要があります。したがって、STORAGE句が必要です。特に、PDBに属する表領域はいずれも2GBを超えることはできません。
- ターゲットの場所に、XMLファイルで指定されている一時ファイルと同じ名前のファイルがあります。したがって、TEMPFILE REUSE句が必要です。

前述の要因に基づき、次の文によってPDBを接続します。

```
CREATE PLUGGABLE DATABASE salespdb USING '/disk1/usr/salespdb.xml'  
SOURCE_FILE_NAME_CONVERT = ('/disk1/oracle/sales/', '/disk2/oracle/sales/')  
NOCOPY  
STORAGE (MAXSIZE 2G)  
TEMPFILE REUSE;
```

例9-6 COPY、PATH_PREFIXおよびFILE_NAME_CONVERTの各句を使用して切断されたPDBを接続する方法

この例では、次の要因を想定しています。

- 新しいPDBは、CDB内に既存のPDBを作成するために使用された同じ切断されたPDBには基づいていません。したがって、AS CLONE句は必要ありません。
- パス接頭辞をPDBのディレクトリ・パスに追加する必要があります。したがって、PATH_PREFIX句が必要です。この例では、パス接頭辞/disk2/oracle/sales/をPDBのディレクトリ・オブジェクト・パスに追加しています。
- XMLファイルにファイルの現在の場所が正確に記述されています。したがって、SOURCE_FILE_NAME_CONVERTまたはSOURCE_FILE_DIRECTORY句は不要です。
- ファイルは、正しい場所に存在しません。したがって、COPYまたはMOVEが含まれている必要があります。この例では、ファイルはコピーされます。

CREATE_FILE_DEST句が使用されておらず、Oracle Managed Filesが有効になっておらず、PDB_FILE_NAME_CONVERT初期化パラメータが設定されていません。したがって、FILE_NAME_CONVERT句が必要です。この例では、ファイルは/disk1/oracle/salesから/disk2/oracle/salesにコピーされます。

- PDBにはストレージ制限は必要ありません。したがって、STORAGE句は必要ありません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

前述の要因に基づき、次の文によってPDBを接続します。

```
CREATE PLUGGABLE DATABASE salespdb USING '/disk1/usr/salespdb.xml'  
COPY  
PATH_PREFIX = '/disk2/oracle/sales/'  
FILE_NAME_CONVERT = ('/disk1/oracle/sales/', '/disk2/oracle/sales/');
```

例9-7 SOURCE_FILE_NAME_CONVERT、MOVE、FILE_NAME_CONVERTおよびSTORAGEの各句を使用して切断されたPDBを接続する方法

この例では、次の要因を想定しています。

- 新しいPDBは、CDB内に既存のPDBを作成するために使用された同じ切断されたPDBには基づいていません。したがって、AS CLONE句は必要ありません。
- PATH_PREFIX句は必要ありません。
- XMLファイルでは、ファイルの現在の場所が正確に記述されていません。したがって、SOURCE_FILE_NAME_CONVERT句またはSOURCE_FILE_DIRECTORY句が必要です。この例では、ファイルが/disk1/oracle/salesに存在することがXMLファイルで示されていますが、ファイルは/disk2/oracle/salesにあり、SOURCE_FILE_NAME_CONVERT句が使用されています。
- ファイルは、PDBの正しい最終的な場所に存在しません。したがって、COPYまたはMOVEが含まれている必要があります。この例では、MOVEを指定してファイルを移動します。

CREATE_FILE_DEST句が使用されておらず、Oracle Managed Filesが有効になっておらず、PDB_FILE_NAME_CONVERT初期化パラメータが設定されていません。したがって、FILE_NAME_CONVERT句が必要です。この例では、ファイルは/disk2/oracle/salesから/disk3/oracle/salesにコピーされます。

- PDBにストレージ制限を強制する必要があります。したがって、STORAGE句が必要です。特に、PDBに属する表領域はいずれも2GBを超えることはできません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE

句は必要ありません。

前述の要因に基づき、次の文によってPDBを接続します。

```
CREATE PLUGGABLE DATABASE salespdb USING '/disk1/usr/salespdb.xml'  
SOURCE_FILE_NAME_CONVERT = ('/disk1/oracle/sales/', '/disk2/oracle/sales/')  
MOVE  
FILE_NAME_CONVERT = ('/disk2/oracle/sales/', '/disk3/oracle/sales/')  
STORAGE (MAXSIZE 2G);
```

例9-8 SOURCE_FILE_DIRECTORY、MOVE、FILE_NAME_CONVERTおよびSTORAGEの各句を使用して切断されたPDBを接続する方法

この例では、次の要因を想定しています。

- 新しいPDBは、CDB内に既存のPDBを作成するために使用された同じ切断されたPDBには基づいていません。したがって、AS CLONE句は必要ありません。
- PATH_PREFIX句は必要ありません。
- XMLファイルでは、ファイルの現在の場所が正確に記述されていません。したがって、SOURCE_FILE_NAME_CONVERT句またはSOURCE_FILE_DIRECTORY句が必要です。この例では、ファイルが/disk1/oracle/salesに存在することがXMLファイルで示されていますが、ファイルは/disk2/oracle/salesにあり、SOURCE_FILE_DIRECTORY句が使用されています。
- ファイルは、PDBの正しい最終的な場所に存在しません。したがって、COPYまたはMOVEが含まれている必要があります。この例では、MOVEを指定してファイルを移動します。

CREATE_FILE_DEST句が使用されておらず、Oracle Managed Filesが有効になっておらず、PDB_FILE_NAME_CONVERT初期化パラメータが設定されていません。したがって、FILE_NAME_CONVERT句が必要です。この例では、ファイルは/disk2/oracle/salesから/disk3/oracle/salesにコピーされます。

- PDBにストレージ制限を強制する必要があります。したがって、STORAGE句が必要です。特に、PDBに属する表領域はいずれも2GBを超えることはできません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

前述の要因に基づき、次の文によってPDBを接続します。

```
CREATE PLUGGABLE DATABASE salespdb USING '/disk1/usr/salespdb.xml'  
SOURCE_FILE_DIRECTORY = '/disk2/oracle/sales/'  
MOVE  
FILE_NAME_CONVERT = ('/disk2/oracle/sales/', '/disk3/oracle/sales/')  
STORAGE (MAXSIZE 2G);
```

例9-9 アーカイブ・ファイルを使用して切断されたPDBを接続する方法

この例では、次の要因を想定しています。

- 切断されたPDBは、sales.pdbという名前の.pdbアーカイブ・ファイルにあります。アーカイブ・ファイルには、XMLメタデータ・ファイルおよびPDBのファイル(データファイルやウォレット・ファイルなど)が圧縮形式で含まれ、これらのファイルはCREATE PLUGGABLE DATABASE文の実行時に.pdbアーカイブ・ファイルの現在のディレクトリに抽出されます。
- 新しいPDBは、CDB内に既存のPDBを作成するために使用された同じ切断されたPDBには基づいていません。したがって、AS CLONE句は必要ありません。
- PATH_PREFIX句は必要ありません。

- PDBにストレージ制限を強制する必要があります。したがって、STORAGE句が必要です。特に、PDBに属する表領域はいずれも2GBを超えることはできません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

前述の要因に基づき、次の文によってアーカイブ・ファイルを使用してPDBを接続します。

```
CREATE PLUGGABLE DATABASE salespdb USING '/disk1/usr/sales.pdb'  
STORAGE (MAXSIZE 2G);
```

親トピック: [切断されたPDBの接続](#)

10 プロキシPDBとしてのPDBの作成

リモートCDBのPDBを参照することで、PDBをプロキシPDBとして作成できます。

- [プロキシPDBの作成について](#)
プロキシPDBは、リモートCDB内のPDBへのアクセスを提供します。シンボリック・リンクに似ています。
- [プロキシPDBの作成](#)
別のCDB内のPDBを参照することで、プロキシPDBを作成します。

親トピック: [PDBおよびアプリケーション・コンテナの作成および削除](#)

プロキシPDBの作成について

プロキシPDBは、リモートCDB内のPDBへのアクセスを提供します。シンボリック・リンクに似ています。

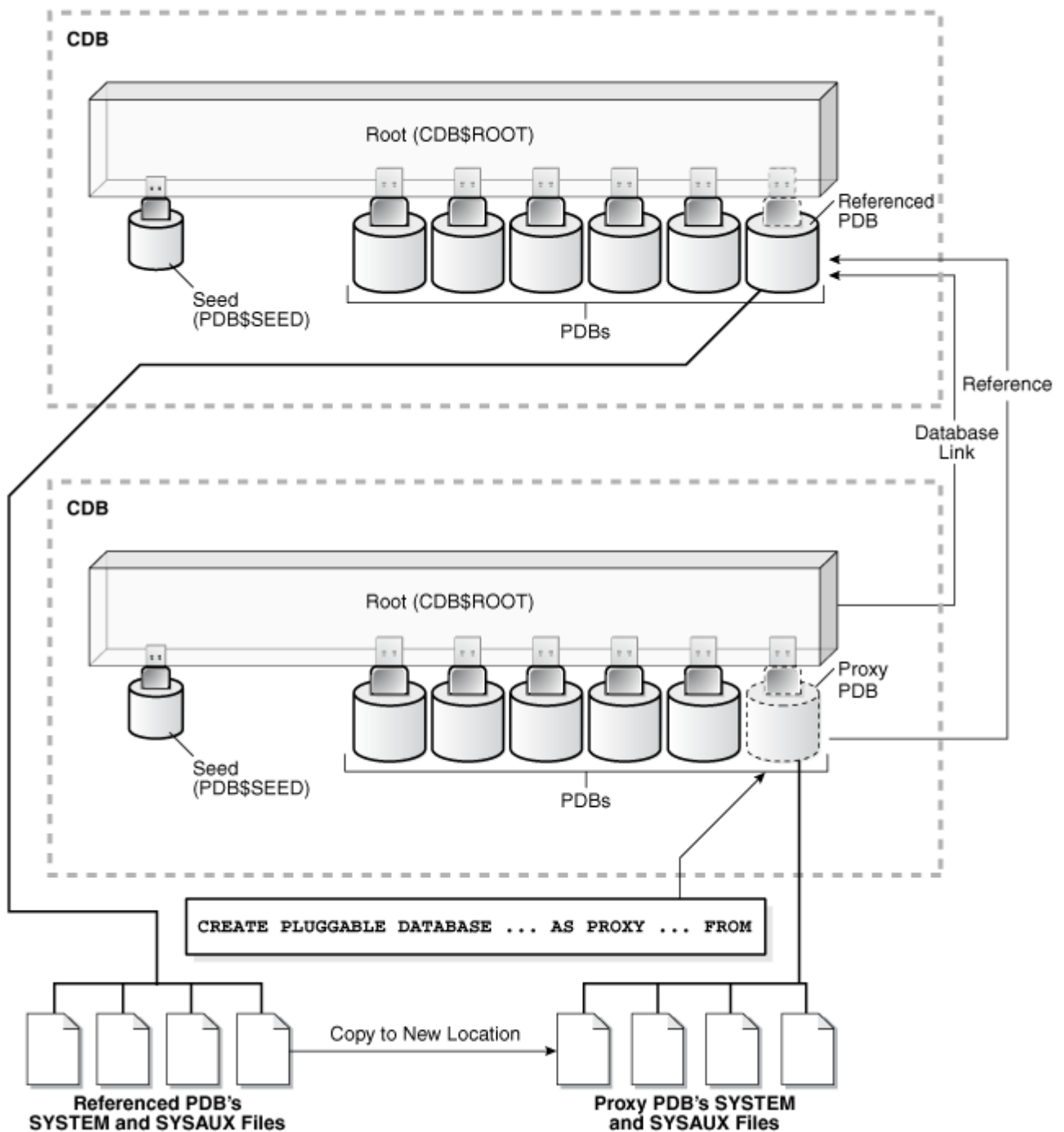
CREATE PLUGGABLE DATABASE文では、異なるCDB内のPDB (**参照先PDB**と呼ばれています)を参照することでプロキシPDBを作成します。リモートPDBのローカル・コンテンツが必要な場合は、プロキシPDBを使用できます。また、異なる複数のCDB内のアプリケーション・コンテナに同じアプリケーションがある場合、それらのアプリケーション・ルートはプロキシPDBを使用して同期を維持できます。

この手法を使用するには、プロキシPDBを含めるCDBでCREATE PLUGGABLE DATABASE文を実行します。次を含める必要があります。

- プロキシPDBを作成していることを指定するためのAS PROXY句。
- プロキシPDBが参照しているPDBを指定するFROM句。
- FROM句にある参照先PDBの現在の場所へのデータベース・リンク。データベース・リンクはプロキシPDBを含むCDBのルートに作成する必要があり、データベース・リンクはリモートCDBのルートまたはリモート参照先PDBに接続します。

次の図は、この手法でリモートCDB内のPDBを参照するプロキシPDBがどのように作成されるかを示しています。

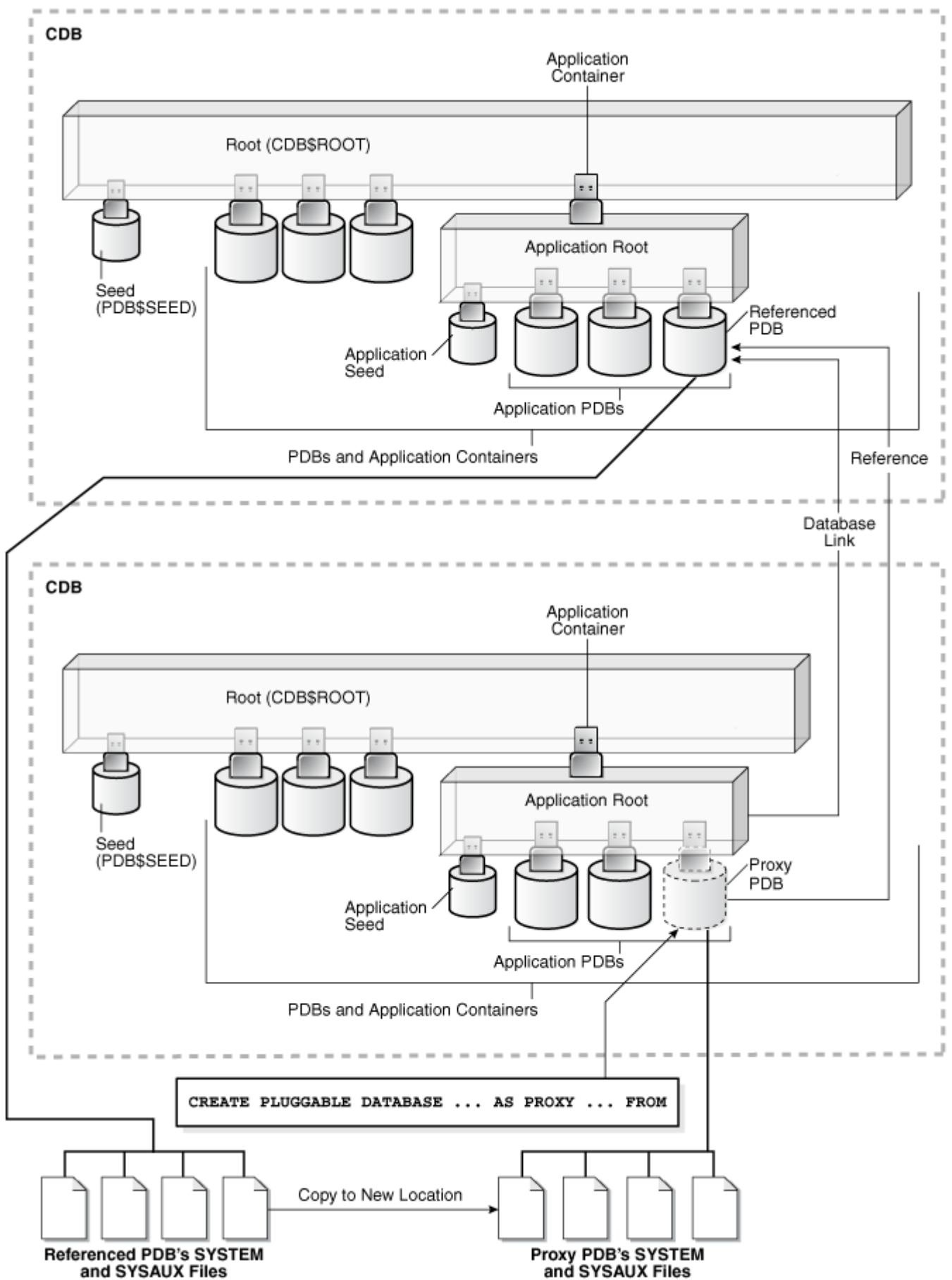
図10-1 リモート・プロキシPDBの作成



アプリケーション・コンテナにプロキシPDBを作成できます。そのためには、参照先PDBがアプリケーション・ルートまたは別のCDBのアプリケーション・コンテナ内のアプリケーションPDBである必要があります。データベース・リンクはプロキシPDBを含むアプリケーション・コンテナのルートに作成する必要があり、データベース・リンクはリモート・アプリケーション・コンテナのルートまたはリモート参照先アプリケーションPDBに接続します。

次の図は、この手法でアプリケーション・コンテナ内のリモート参照先PDBに基づいてプロキシPDBをアプリケーション・コンテナに作成する方法を示しています。

図10-2 アプリケーション・コンテナ内のリモート・プロキシPDBの作成



プロキシPDBを作成する前に、[表5-3](#)のプロキシPDBの作成に適用される質問に対処します。この表では、様々な要因に基づいて指定する必要があるCREATE PLUGGABLE DATABASEの句について説明しています。

- [プロキシPDBおよびSQL文](#)

原則として、プロキシPDBが現在のコンテナである場合は、プロキシPDBでの実行のために発行されたSQL文は参照先PDB内で実行されます。

- [プロキシPDBおよびデータベース・リンク](#)
データベース・リンクはプロキシPDBの作成時に必要です。
- [プロキシPDBおよび認証](#)
プロキシPDBのセッションでは、パスワード認証のみがサポートされます。
- [プロキシPDBおよびリスナー](#)
PDBのホスト名とポート番号の設定は、プロキシPDBがPDBを参照する場合にのみ重要です。

関連項目:

- [「PDB記憶域」](#)
- [アプリケーション・ルート・レプリカのプロキシPDBとの同期](#)

親トピック: [プロキシPDBとしてのPDBの作成](#)

プロキシPDBおよびSQL文

原則として、プロキシPDBが現在のコンテナである場合は、プロキシPDBでの実行のために発行されたSQL文は参照先PDB内で実行されます。

リモート実行の結果はプロキシPDBに返されます。たとえば、データ定義言語(DDL)文、データ操作言語(DML)文およびプロキシPDBで実行される問合せは、実行のために参照先PDBに送信され、結果はプロキシPDBに返されます。

ルールに対する例外が1つあります。プロキシPDBが現在のコンテナの場合は、ALTER PLUGGABLE DATABASE文およびALTER DATABASE文を実行すると、これらの文はプロキシPDBにのみ影響を与えます。これらは実行のために参照先PDBに送信されません。同様に、現在のコンテナが、プロキシPDBが属するルートである場合、ALTER PLUGGABLE DATABASE文はプロキシPDBにのみ影響します。たとえば、CDBルート、アプリケーション・ルートまたはプロキシPDBで実行されたALTER PLUGGABLE DATABASE文は、プロキシPDBをオープンまたはクローズできますが、この文は参照先PDBをオープンまたはクローズしません。

親トピック: [プロキシPDBの作成について](#)

プロキシPDBおよびデータベース・リンク

データベース・リンクはプロキシPDBの作成時に必要です。

プロキシPDBの作成後は、作成時に指定したデータベース・リンクがそのプロキシPDBで使用されなくなります。かわりに、プロキシPDBは参照先PDBと直接通信します。

この直接通信は、参照先PDBを含むCDBのリスナーのポート番号とホスト名を必要とします。プロキシPDBの作成時に、プロキシPDBはデフォルトで次の値を使用します。

- リスナー・ポート番号: 1521
参照先PDBのリスナーがデフォルト・ポート番号を使用しない場合は、PORT句を使用して、リスナーのポート番号を指定する必要があります。プロキシPDBの作成時にポート番号を指定するか、プロキシPDBを変更してポート番号を変更できます。
- リスナー・ホスト名: 参照先PDBを含むCDBのホスト名
参照先PDBのリスナーがデフォルト・ホスト名を使用しない場合は、HOST句を使用して、リスナーのホスト名を指定する必要があります。プロキシPDBの作成時にホスト名を指定するか、プロキシPDBを変更してホスト名を変更できます。

関連トピック

- [プロキシPDBおよびリスナー](#)
- [参照先PDBのリスナー設定の変更](#)

親トピック: [プロキシPDBの作成について](#)

プロキシPDBおよび認証

プロキシPDBのセッションでは、パスワード認証のみがサポートされます。

親トピック: [プロキシPDBの作成について](#)

プロキシPDBおよびリスナー

PDBのホスト名とポート番号の設定は、プロキシPDBがPDBを参照する場合にのみ重要です。

- [HOST句](#)
CREATE PLUGGABLE DATABASE文のHOST句では、作成されるPDBのリスナーのホスト名を指定します。
- [PORT句](#)
CREATE PLUGGABLE DATABASE文のPORT句では、作成されるPDBのリスナーのポート番号を指定します。

親トピック: [プロキシPDBの作成について](#)

HOST句

CREATE PLUGGABLE DATABASE文のHOST句では、作成されるPDBのリスナーのホスト名を指定します。

デフォルトでは、リスナーのホスト名は作成されるPDBのホスト名と同じです。次の両方の条件を満たす場合にHOST句を指定します。

- リスナーのホスト名は、作成されるPDBのホスト名と異なります。
- 作成されるPDBを参照するプロキシPDBの作成を計画します。

プロキシPDBは、データベース・リンクを使用して、その参照先PDBとの通信を確立します。通信が確立した後、プロキシPDBでは、データベース・リンクを使用しないで参照先PDBと直接通信します。プロキシPDBが正常に機能するためにはリスナーのホスト名が正しい必要があります。

例10-1 HOST句

```
HOST='myhost.example.com'
```

関連項目:

- [プロキシPDBの作成について](#)
- [参照PDBのリスナー・ホスト名の変更](#)
- HOST句についてさらに学習するには、[Oracle Database SQL言語リファレンス](#)を参照してください。

親トピック: [プロキシPDBおよびリスナー](#)

PORT句

CREATE PLUGGABLE DATABASE文のPORT句では、作成されるPDBのリスナーのポート番号を指定します。

デフォルトでは、作成されるPDBのポート番号は1521です。次の両方の条件を満たす場合にPORT句を指定します。

- リスナーのポート番号は1521ではありません。
- 作成されるPDBを参照するプロキシPDBの作成を計画します。

プロキシPDBは、データベース・リンクを使用して、その参照先PDBとの通信を確立します。通信が確立した後、プロキシPDBでは、データベース・リンクを使用しないで参照先PDBと直接通信します。プロキシPDBが正常に機能するためにはリスナーのポート番号が正しい必要があります。

例10-2 PORT句

PORT=1599

ノート:



- [プロキシ PDB の作成について](#)
- [参照 PDB のリスナー・ホスト名の変更](#)
- PORT 句についてさらに学習するには、[Oracle Database SQL 言語リファレンス](#)を参照してください。

親トピック: [プロキシPDBおよびリスナー](#)

プロキシPDBの作成

別のCDBのPDBを参照することで、プロキシPDBを作成します。

前提条件

次の前提条件を満たしている必要があります。

- [「PDB作成の一般的な前提条件」](#)で説明している前提条件を満たします。
- 現行ユーザーは、プロキシPDBが作成されているCDBのルートでCREATE PLUGGABLE DATABASEシステム権限を持っている必要があります。
- 参照先PDBを含むCDBはローカルUNDOモードである必要があります。
- 参照先PDBを含むCDBはARCHIVELOGモードである必要があります。
- プロキシPDBの作成時には、参照先PDBが読み取り/書き込みオープン・モードである必要があります。参照先PDBのオープン・モードは、プロキシPDBの作成後に変更できます。
- データベース・リンクは、プロキシPDBが作成されているCDBのルートから参照先PDBの場所への接続を有効にする必要があります。データベース・リンクは、リモートCDBのルート、またはリモートPDBのいずれかに接続できます。
- データベース・リンクが、参照先PDBを含むリモートCDBのルートに接続する場合、データベース・リンクが接続するユーザーは、共通ユーザーである必要があります。
- データベース・リンクが参照先PDBに接続する場合、データベース・リンクが参照先PDBで接続するユーザーは、CREATE PLUGGABLE DATABASEシステム権限を持っている必要があります。
- プロキシPDBをアプリケーション・コンテナに作成している場合は、次の前提条件が適用されます。
 - 参照先PDBがアプリケーション・ルートまたはアプリケーション・コンテナ内のアプリケーションPDBである必要があ

ります。

- プロキシPDBのアプリケーション・コンテナのアプリケーション名およびバージョンは、参照先PDBのアプリケーション名およびバージョンと一致する必要があります。
- プロキシPDBがアプリケーション・コンテナに作成されている場合、データベース・リンクは、プロキシPDBが作成されているアプリケーション・コンテナのルートから参照先PDBの場所への接続を有効にする必要があります。データベース・リンクは、リモート・アプリケーション・コンテナのルート、またはリモート・アプリケーションPDBのいずれかに接続できます。
- データベース・リンクが、参照先PDBを含むリモート・アプリケーション・コンテナのルートに接続する場合、データベース・リンクが接続するユーザーは、アプリケーション共通ユーザーである必要があります。
- データベース・リンクが参照先アプリケーションPDBに接続する場合、データベース・リンクが参照先アプリケーションPDBで接続するユーザーは、CREATE PLUGGABLE DATABASEシステム権限を持っている必要があります。



ノート:

アプリケーション・コンテナ内の参照先 PDB に基づいてプロキシ PDB を CDB ルートに作成できます。

プロキシPDBを作成するには:

1. SQL*Plusで、現在のコンテナがCDBルートまたはプロキシPDBが作成されているアプリケーション・ルートであることを確認します。

現在のコンテナがCDBルートの場合、プロキシPDBはCDBに作成されます。現在のコンテナがアプリケーション・ルートの場合、プロキシPDBはアプリケーション・コンテナ内に作成されます。

2. CREATE PLUGGABLE DATABASE文を実行します。AS PROXY句を指定し、FROM句でデータベース・リンク名を指定して参照先PDBを指定します。他の句が必要な場合はそれらを指定します。

作成したプロキシPDBはマウント・モードになり、そのステータスはNEWです。PDBのオープン・モードは、V\$PDBSビューのOPEN_MODE列を問い合わせることで確認できます。CDB_PDBSまたはDBA_PDBSビューのSTATUS列を問い合わせ、PDBのステータスを表示できます。

PDBの新しいデフォルト・サービスが作成されます。サービスの名前はPDBと同じになり、このサービスを使用してPDBにアクセスできます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。

3. 新しいPDBを読取り/書込みモードでオープンします。

Oracle Databaseで新しいPDBをCDBに統合するには、PDBを読取り/書込みモードでオープンする必要があります。PDBを読取り専用モードでオープンしようとすると、エラーが返されます。PDBが読取り/書込みモードでオープンされると、そのステータスはNORMALになります。

4. PDBをバックアップします。

PDBは、バックアップしないとリカバリできません。



ノート:

プロキシ PDB の作成時にエラーが返された場合は、作成された PDB が UNUSABLE 状態である可能性があります。CDB_PDBS または DBA_PDBS ビューを問い合わせる PDB の状態を確認したり、アラート・ログをチェックして PDB の作成エラーについてさらに学習できます。使用不可の PDB は削除することのみができ、使用不可の PDB と同じ名前の PDB を作成できるようにするには、その前に削除する必要があります。

例10-3 リモート・プロキシPDBの作成

この例で、新しいPDBが属するルートはCREATE PLUGGABLE DATABASE文の実行時の現在のコンテナによって決まります。

- 現在のコンテナがCDBルートの場合、新しいPDBはCDBルートに作成されます。
- 現在のコンテナがアプリケーション・コンテナ内のアプリケーション・ルートの場合、新しいPDBはアプリケーション・ルート内にアプリケーションPDBとして作成されます。

この例では、様々な要因に基づいてpdb1という名前のリモート・プロキシPDBを作成します。この例では、次の要因を想定しています。

- 参照先PDBのCDBへのデータベース・リンク名はpdb1_linkです。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

Oracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。SYSTEMファイルおよびSYSAUXファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

前述の要因に基づき、次の文によってpdb1プロキシPDBが作成されます。

```
CREATE PLUGGABLE DATABASE pdb1 AS PROXY FROM pdb1@pdb1_link;
```

関連項目:

- [「CDBのUNDOモードについて」](#)
- [「CDB内のコンテナへのアクセスについて」](#)
- [「PDBのオープン・モードの変更」](#)
- PDBのバックアップの詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド』](#)を参照してください。

親トピック: [プロキシPDBとしてのPDBの作成](#)

11 PDBの削除

接続されたPDBは、切断、削除または再配置することでCDBから削除できます。

- [CDBからのPDBの切断](#)
PDBをCDBに接続できるのと同様に、PDBをCDBから切断できます。
- [PDBの削除](#)
PDBを新しいCDBに移動する場合、またはPDBが必要でなくなった場合は、PDBを削除します。

関連項目:

[「PDBの再配置」](#)

親トピック: [PDBおよびアプリケーション・コンテナの作成および削除](#)

CDBからのPDBの切断

PDBをCDBに接続できるのと同様に、PDBをCDBから切断できます。

- [PDBの切断について](#)
PDBを切断すると、CDBへのPDBの関連付けが解除されます。PDBは、CDBに接続している場合にのみ使用できます。
- [PDBの切断](#)
ALTER PLUGGABLE DATABASE ... UNPLUG INTO文を使用してPDBを切断します。

親トピック: [PDBの削除](#)

PDBの切断について

PDBを切断すると、CDBへのPDBの関連付けが解除されます。PDBは、CDBに接続している場合にのみ使用できます。

次のいずれかを実行する場合は、PDBを切断します。

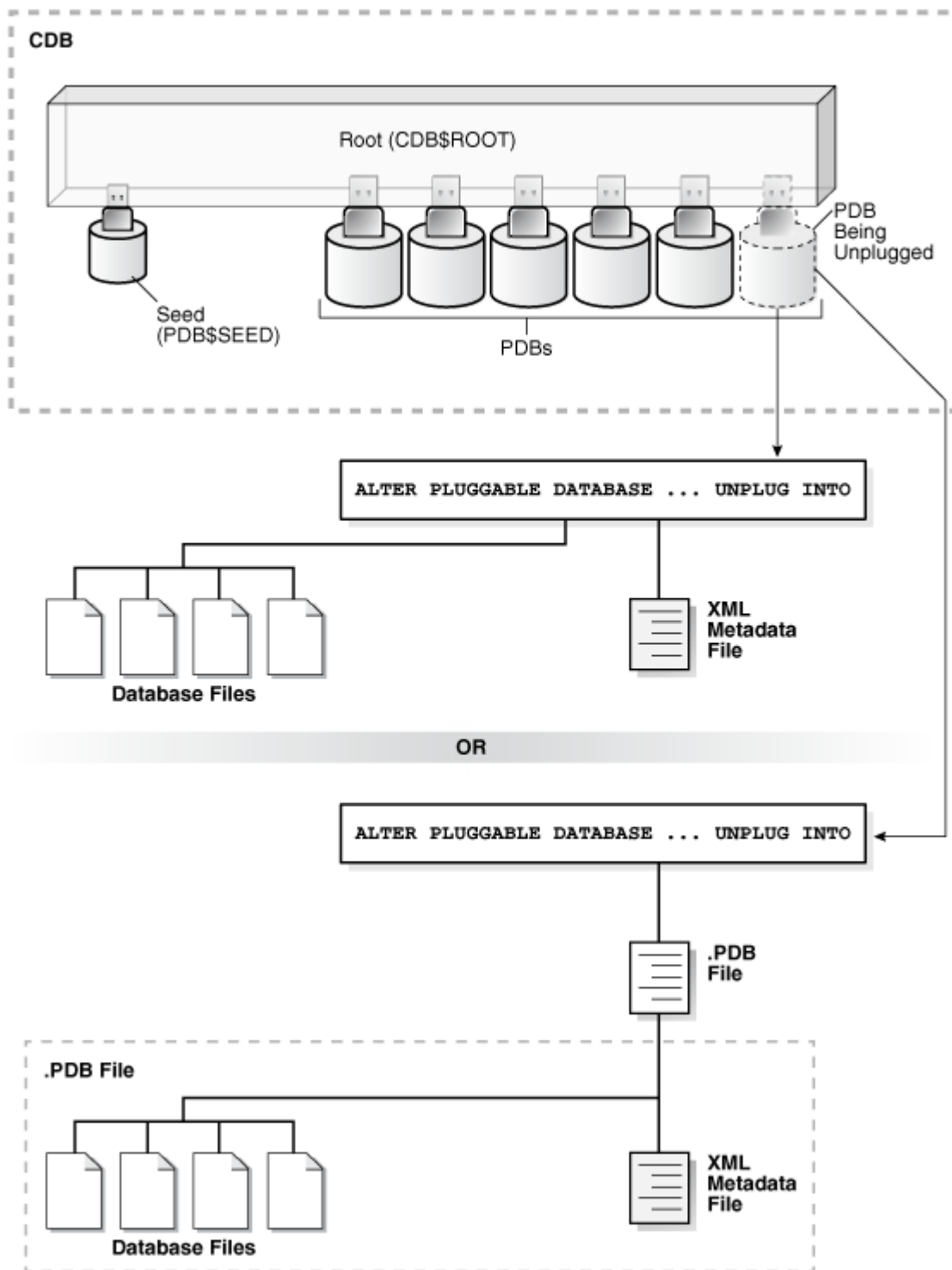
- PDBを異なるCDBに移動します
- 後で使用できるようにPDBをアーカイブします
- PDBを使用できないようにします

PDBを切断するには、CDBルートまたはアプリケーション・ルートに接続し、ALTER PLUGGABLE DATABASE文を使用して、次のいずれかを指定します。

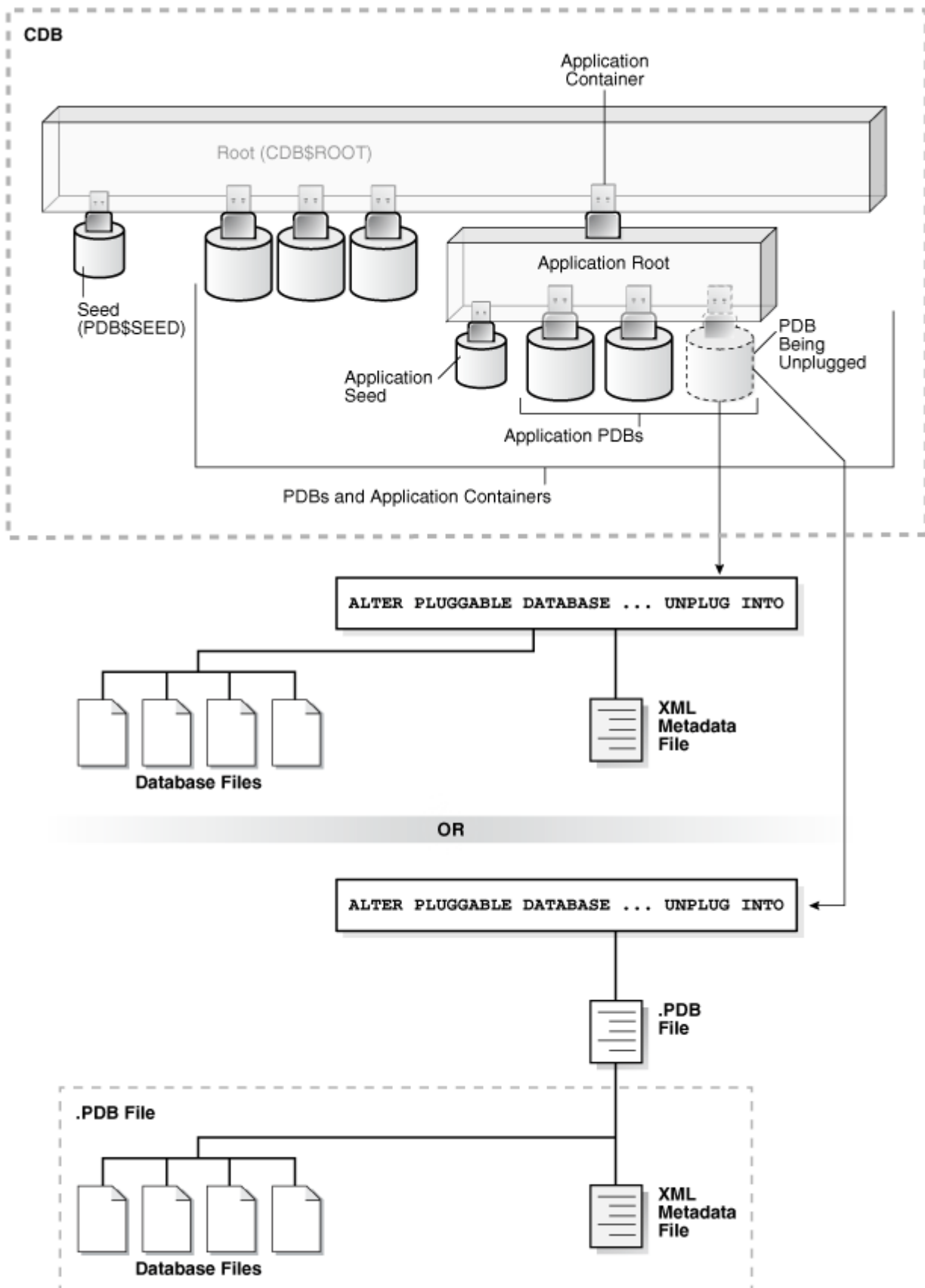
- XMLファイル
XMLファイル(.xml拡張子)には、切断された後にPDBに関するメタデータが格納されます。このメタデータは、PDBに接続するために、ターゲットCDBでCREATE PLUGGABLE DATABASE文を有効にするために必要な情報が含まれています。
- .pdbファイル
.pdbファイルには、PDBについて記述したXMLファイルおよびPDBで使用されるファイル(データファイルやウォレット・ファイルなど)の圧縮アーカイブが格納されます。.pdbファイルにより、単一の圧縮されたファイル(複数ファイルではない)を

新しい場所にコピーしてPDBをCDBに接続できます。

図11-1 PDBの切断



次の図では、この手法を使用してアプリケーション・コンテナからアプリケーションPDBを切断する方法を示しています。



PDBは切断する前にクローズする必要があります。PDBを切断すると、切断されたPDBはマウント・モードになります。この切断操作によって、PDBのデータファイルに対して、PDBが正常に切断されたなどの、記録対象となるなんらかの変更が加えられます。依然としてCDBの一部であるため、切断されたPDBはCDB全体のRMANバックアップに含まれています。このような方法によるバックアップは、切断されたPDBが将来必要になった場合に備えてアーカイブするのに便利です。

PDBをCDBから完全に除去するには、PDBを削除します。切断されたPDBに対してサポートされている唯一の操作は、PDBの削除です。PDBを再度同じCDBに接続できるようにするには、CDBから削除しておく必要があります。



ノート:

アプリケーション・コンテナは、それに属するアプリケーション PDB がない場合にのみ切断できます。

関連項目:

- [「PDBの削除」](#)
- PDBのクローズの詳細は、[「PDBのオープン・モードの変更」](#)を参照してください
- 初期化パラメータおよび切断されたPDBの詳細は、[「システム・レベルでのPDBの変更」](#)を参照してください。
- 共通ユーザーおよびローカル・ユーザーの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBからのPDBの切断](#)

PDBの切断

ALTER PLUGGABLE DATABASE ... UNPLUG INTO文を使用してPDBを切断します。

前提条件

次の前提条件を満たしている必要があります。

- 現在のユーザーにはSYSDBAまたはSYSOPER管理権限が必要であり、その権限は共通で付与されているか、またはPDBでローカルに付与されている必要があります。ユーザーは、接続時にAS SYSDBAまたはAS SYSOPERを使用して権限を行使する必要があります。
- PDBは、少なくとも1回はオープンしている必要があります。

ノート:



透過的データ暗号化で暗号化されているデータが含まれた PDB を切断する場合は、[『Oracle Database Advanced Security ガイド』](#)の指示に従います。

PDBを切断するには:

1. SQL*Plusで、現在のコンテナがPDBのルートであることを確認します。

PDBがCDBルートに接続される場合、現在のコンテナはCDBルートである必要があります。PDBがアプリケーション・ルートに接続される場合、現在のコンテナはアプリケーション・ルートである必要があります。

アプリケーション・コンテナを切断している場合、現在のコンテナはCDBルートである必要があり、アプリケーション・コンテナにはそれに接続されているアプリケーションPDBがないようにしてください。

2. PDBをクローズします。

Oracle Real Application Clusters (Oracle RAC)環境では、PDBがすべてのインスタンスでクローズされている必要があります。

3. UNPLUG INTO句を使用してALTER PLUGGABLE DATABASE文を実行し、切断するPDB、およびPDBのXMLメタデータ・ファイルまたは.pdbの名前と場所を指定します。

例11-1 XMLメタデータ・ファイルへのPDB salespdbの切断

次のALTER PLUGGABLE DATABASE文により、PDB salespdbが切断され、/oracle/data/ディレクトリにsalespdb.xmlメタデータ・ファイルが作成されます。

```
ALTER PLUGGABLE DATABASE salespdb UNPLUG INTO '/oracle/data/salespdb.xml';
```

例11-2 アーカイブ・ファイルへのPDB salespdbの切断

次のALTER PLUGGABLE DATABASE文により、PDB salespdbが切断され、/oracle/data/ディレクトリにsales.pdbアーカイブ・ファイルが作成されます。sales.pdbアーカイブ・ファイルは、XMLメタデータ・ファイルとPDBのファイル(データファイルやウォレット・ファイルなど)を含む圧縮ファイルです。

```
ALTER PLUGGABLE DATABASE salespdb UNPLUG INTO '/oracle/data/sales.pdb';
```

親トピック: [CDBからのPDBの切断](#)

PDBの削除

PDBを新しいCDBに移動する場合、またはPDBが必要でなくなった場合は、PDBを削除します。

PDBを削除すると、削除されたPDBへのすべての参照を削除するようにCDBの制御ファイルが変更されます。PDBに関連付けられているアーカイブREDOログ・ファイルおよびバックアップは削除されませんが、Oracle Recovery Manager (RMAN)を使用して削除できます。

PDBを削除する場合は、DROP PLUGGABLE DATABASE文の次のいずれかの句を使用して、PDBのデータファイルを保持または削除できます。

- KEEP DATAFILESはデフォルトであり、データファイルを保持します。
KEEP DATAFILESが指定されていても、PDBの一時ファイルは不要であるため削除されます。
KEEP DATAFILESが指定されている場合は、PDBを切断する必要があります。
- INCLUDING DATAFILESでは、ディスクからデータファイルを削除します。
PDBがSNAPSHOT COPY句を使用して作成された場合、PDBを削除する際にINCLUDING DATAFILESを指定する必要があります。

前提条件

次の前提条件を満たしている必要があります。

- PDBは、マウント・モードであるか、または切断されている必要があります。
[「PDBのオープン・モードの変更」](#)を参照してください。
[「CDBからのPDBの切断」](#)を参照してください。
- 現在のユーザーにはSYSDBAまたはSYSOPER管理権限が必要であり、その権限は共通で付与されているか、またはPDBでローカルに付与されている必要があります。ユーザーは、接続時にAS SYSDBAまたはAS SYSOPERを使用して権限を行使する必要があります。



ノート:

これは、破壊的な操作です。

PDBを削除するには:

1. SQL*Plusで、現在のコンテナがCDBルートであるか、アプリケーションPDBの場合はアプリケーションPDBを含むアプリケーション・ルートであることを確認します。

PDBがCDBルートに接続される場合、現在のコンテナはCDBルートである必要があります。PDBがアプリケーション・ルートに接続される場合、現在のコンテナはアプリケーション・ルートまたはCDBルートである必要があります。

アプリケーション・コンテナを削除している場合、現在のコンテナはCDBルートである必要があります、アプリケーション・コンテナにはそれに接続されているアプリケーションPDBがないようにしてください。

2. DROP PLUGGABLE DATABASE文を実行し、削除するPDBを指定します。

例11-3 データファイルは保持したままPDB salespdbを削除する方法

```
DROP PLUGGABLE DATABASE salespdb  
KEEP DATAFILES;
```

例11-4 PDB salespdbおよびそのデータファイルを削除する方法

```
DROP PLUGGABLE DATABASE salespdb  
INCLUDING DATAFILES;
```

関連項目:

- [「CDBからのPDBの切断」](#)
- [「スナップショットのコピーPDBの記憶域要件」](#)
- [Oracle Database SQL言語リファレンス](#)
- RMANの詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド』](#)を参照してください

親トピック: [PDBの削除](#)

12 アプリケーション・コンテナおよびシードの作成と削除

いくつかの異なる方法でアプリケーション・コンテナおよびアプリケーション・シードを作成できます。アプリケーション・コンテナをCDBから削除したり、アプリケーション・コンテナからアプリケーション・シードを削除することもできます。

- [アプリケーション・コンテナの作成と削除](#)
アプリケーション・コンテナは、PDBシードの使用、既存のPDBまたは非CDBのクローニング、切断されたPDBの接続など、複数の方法で作成できます。アプリケーション・コンテナをCDBから削除することもできます。
- [アプリケーション・シードの作成と削除](#)
アプリケーション・シードは、PDBシードの使用、既存のPDBまたは非CDBのクローニング、切断されたPDBの接続など、複数の方法で作成できます。アプリケーション・シードをアプリケーション・コンテナから削除することもできます。
- [アプリケーションPDBの作成](#)
アプリケーションPDBを作成するには、アプリケーション・ルートが現在のコンテナであるときに、CREATE PLUGGABLE DATABASE文を実行します。

関連項目:

- [「アプリケーション・コンテナについて」](#)
- [「アプリケーション・コンテナの管理」](#)

親トピック: [PDBおよびアプリケーション・コンテナの作成および削除](#)

アプリケーション・コンテナの作成および削除

アプリケーション・コンテナは、PDBシードの使用、既存のPDBまたは非CDBのクローニング、切断されたPDBの接続など、複数の方法で作成できます。アプリケーション・コンテナをCDBから削除することもできます。

- [アプリケーション・コンテナの作成](#)
CREATE PLUGGABLE DATABASE文を使用して、CDB内にアプリケーション・コンテナを作成できます。
- [CDBからのアプリケーション・コンテナの切断](#)
CDBからアプリケーション・コンテナを切断できます。
- [アプリケーション・コンテナの削除](#)
アプリケーション・コンテナは、あるCDBから別のCDBにアプリケーション・コンテナを移動する場合、またはアプリケーション・コンテナが不要になった場合に削除できます。

親トピック: [アプリケーション・コンテナおよびシードの作成と削除](#)

アプリケーション・コンテナの作成

CREATE PLUGGABLE DATABASE文を使用して、CDB内にアプリケーション・コンテナを作成できます。

- [アプリケーション・コンテナの作成について](#)
CREATE PLUGGABLE DATABASE ... AS APPLICATION CONTAINER文は、新しいアプリケーション・コンテナを作成します。
- [アプリケーション・コンテナの準備](#)
アプリケーション・コンテナを作成する前に前提条件を満たす必要があります。
- [アプリケーション・コンテナの作成](#)

AS APPLICATION CONTAINER句を指定したCREATE PLUGGABLE DATABASE文を使用して、アプリケーション・コンテナを作成できます。

親トピック: [アプリケーション・コンテナの作成および削除](#)

アプリケーション・コンテナの作成について

CREATE PLUGGABLE DATABASE ... AS APPLICATION CONTAINER文は、新しいアプリケーション・コンテナを作成します。

アプリケーション・コンテナは、アプリケーション・ルートおよび1つ以上のアプリケーションのデータを格納するアプリケーションPDBのコレクションから構成されます。アプリケーションPDBはアプリケーション・ルートに接続され、オプションで新規アプリケーションPDBを素早く簡単に作成するためにアプリケーション・シードを作成できます。アプリケーションPDBとアプリケーション・ルートはアプリケーション共通オブジェクトを共有できます。

アプリケーション共通オブジェクトには3つのタイプがあります。

- メタデータリンク・アプリケーション共通オブジェクトは、表などの特定のオブジェクトのメタデータを格納するため、アプリケーション共通オブジェクトを共有するコンテナは同じ構造を持ちますが、データは異なります。
- データリンク・アプリケーション共通オブジェクトはアプリケーション・ルートで一度定義され、ホストされるアプリケーションPDBのコンテキストで読み取り専用オブジェクトとして共有されます。
- 拡張データリンク・アプリケーション共通オブジェクトは、共有データをアプリケーション・ルートに格納しますが、アプリケーションPDBがそのオブジェクトに追加してデータを格納することも許可します。追加されるデータは、各アプリケーションPDBに固有のローカル・データです。

アプリケーション・コンテナは、CREATE PLUGGABLE DATABASE文にAS APPLICATION CONTAINER句を含めることで作成します。アプリケーション・コンテナを作成する次の手法を使用できます。

- PDBシードの使用
- 既存のPDBまたは非CDBのクローニング
- PDBの再配置
- 切断されたPDBの接続

アプリケーション・コンテナを作成するには、現在のコンテナがCDBルートであることが必要です。また、CREATE PLUGGABLE DATABASE文でAS APPLICATION CONTAINERを指定する必要があります。Oracle Managed Filesを使用してアプリケーション・コンテナを作成する必要があります。

ノート:



いずれかのアプリケーション PDB がアプリケーション・コンテナに属する場合、そのアプリケーション・コンテナはアップグレードまたは削除できません。

アプリケーション・コンテナへの既存のアプリケーションの移行

既存のPDBを使用してアプリケーション・ルートを作成して、アプリケーションをアプリケーション・ルートに移行できます。既存のアプリケーションをアプリケーション・コンテナに移行している場合、追加タスクを実行する必要があります。接続するPDBには、データを含むアプリケーション・オブジェクトが含まれている必要があり、DBMS_PDBパッケージのプロシージャを実行して、共有するオブジェクトを指定する必要があります。また、アプリケーションの共通ユーザー、ロールまたはプロファイルがアプリケーション・ルートに存在する場合は、DBMS_PDBパッケージのプロシージャを実行して、それらが共通であることを指定する必要があります。

アプリケーションをアプリケーション・コンテナに移行した後は、アプリケーション・ルートにアプリケーションPDBを作成したり、既存のPDBを使用して、アプリケーションPDBを作成できます。

関連項目:

[「アプリケーション・コンテナへの既存アプリケーションの移行」](#)

親トピック: [アプリケーション・コンテナの作成](#)

アプリケーション・コンテナの準備

アプリケーション・コンテナを作成する前に、前提条件を満たす必要があります。

- CDBが存在していることが必要です。
- CDBは、読取り/書き込みモードである必要があります。
- 現在のユーザーは、現在のコンテナがCDBルートである共通ユーザーである必要があります。
- 現行ユーザーにはCREATE PLUGGABLE DATABASEシステム権限が必要です。
- 各アプリケーション・コンテナに対して一意のアプリケーション・コンテナ名を決定する必要があります。各アプリケーション・コンテナ名は、単一CDB内のすべてのコンテナに関して一意であり、さらに、各アプリケーション・コンテナ名は、特定のリスナーを介してアクセスされたデータベース・インスタンスのすべてのCDBの範囲内で一意である必要があります。
アプリケーション・コンテナ名は、アプリケーション・コンテナをCDB内の他のコンテナから区別するために使用されます。アプリケーション・コンテナ名のルールは、大文字と小文字が区別されないなど、サービス名のルールと同じです。
- Oracle Managed Filesを使用してコンテナを作成する必要があります。
- 物理スタンバイ・データベースがあるOracle Data Guard構成でアプリケーション・コンテナを作成している場合は、アプリケーション・コンテナを作成する前に追加タスクを完了する必要があります。
- インストール・スクリプトを使用して、既存のアプリケーションをアプリケーション・コンテナに移行している場合、そのスクリプトは実行可能である必要があります。
- PDBを使用して既存のアプリケーションをアプリケーション・コンテナに移行している場合は、PDBをアプリケーション・ルートにクローニングするか、PDBをアプリケーション・ルートに接続することを可能にする必要があります。

関連項目:

- [「現在のコンテナについて」](#)
- [「アプリケーション・コンテナへの既存アプリケーションの移行」](#)
- [『Oracle Data Guard概要および管理』](#)

親トピック: [アプリケーション・コンテナの作成](#)


アプリケーション・コンテナの作成

AS APPLICATION CONTAINER句を指定したCREATE PLUGGABLE DATABASE文を使用して、アプリケーション・コンテナを作成できます。

アプリケーション・コンテナを作成する前に、[「アプリケーション・コンテナの準備」](#)で説明した前提条件を満たします。

1. SQL*Plusで、現在のコンテナがCDBルートであることを確認します。
2. CREATE PLUGGABLE DATABASE文を実行し、AS APPLICATION CONTAINER句を含めます。要求された場合には、他の句を指定します。
アプリケーション・コンテナを作成した後、それはマウント・モードになり、そのステータスはNEWになります。アプリケーション・コンテナのオープン・モードは、V\$PDBSビューのOPEN_MODE列を問い合わせることで表示できます。CDB_PDBSまたはDBA_PDBSビューのSTATUS列を問い合わせ、アプリケーション・コンテナのステータスを表示できます。
新規デフォルト・サービスがアプリケーション・コンテナに対して作成されます。サービスにはアプリケーション・コンテナと同じ名前があり、アプリケーション・コンテナへのアクセスに使用できます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。
3. 新規アプリケーション・コンテナを読取り/書き込みモードでオープンします。
Oracle Databaseの新規アプリケーション・コンテナを読取り/書き込みモードでオープンして、新規アプリケーション・コンテナのCDBへの統合を完了する必要があります。アプリケーション・コンテナを読取り専用モードでオープンしようとすると、エラーが返されます。アプリケーション・コンテナを読取り/書き込みモードでオープンした後、そのステータスはNORMALになります。
4. アプリケーション・コンテナをバックアップします。
アプリケーション・コンテナは、バックアップしていないとリカバリできません。

ノート:



アプリケーション・コンテナの作成中にエラーが返された場合、作成されているアプリケーション・コンテナはUNUSABLE 状態になります。CDB_PDBS または DBA_PDBS ビューを問い合わせ、アプリケーション・コンテナの状態を確認し、アラート・ログをチェックしてアプリケーション・コンテナの作成エラーについてさらに学習できます。使用不可のアプリケーション・コンテナは削除することのみができ、使用不可のアプリケーション・コンテナと同じ名前のアプリケーション・コンテナまたは PDB を作成できるようにするには、その前にそのアプリケーション・コンテナを削除する必要があります。

5. 既存のアプリケーションをアプリケーション・コンテナに移行している場合、[「アプリケーション・コンテナへの既存のアプリケーションの移行」](#)の手順に従います。

アプリケーション・コンテナはアプリケーション・ルートで作成されます。アプリケーションPDBをアプリケーション・コンテナに作成できます。

例12-1 PDBシードを使用したアプリケーション・コンテナの作成

この例では、次の要因を想定しています。

- ストレージ制限は、アプリケーション・コンテナには不要です。したがって、STORAGE句は必要ありません。
- アプリケーション・コンテナにはデフォルトの表領域は必要ありません。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

CDBに対してOracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。PDBシードに関連付けられているファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

- Oracleの事前定義されたロールをPDB_DBAロールに付与する必要はありません。

次の文では、PDBシードからアプリケーション・コンテナを作成します。

```
CREATE PLUGGABLE DATABASE salesact AS APPLICATION CONTAINER
ADMIN USER salesadm IDENTIFIED BY password;
```

例12-2 ローカルPDBのクローニングによるアプリケーション・コンテナの作成

この例では、次の要因を想定しています。

- PATH_PREFIX句は必要ありません。
- コピーしたファイルのターゲットの場所を指定するには、FILE_NAME_CONVERT句が必要です。この例では、ファイルが /disk1/oracle/pdb1/ から /disk2/oracle/hract/ にコピーされます。
CREATE_FILE_DEST句は使用されず、Oracle Managed FilesもPDB_FILE_NAME_CONVERT初期化パラメータも、コピーしたファイルのターゲットの場所の指定に使用されません。
PDBのデータファイルの場所を表示するには、[\[例19-7\]](#)の問合せを実行します。
- アプリケーション・ルートに対してストレージ制限を適用する必要があります。したがって、STORAGE句が必要です。特に、アプリケーション・ルートに属するすべての表領域が2GBを超えていない必要があります。このストレージ制限は、アプリケーション・ルートに接続されているアプリケーションPDBには適用されません。
- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

前述の要因に基づき、次の文によって、pdb1からhractがアプリケーション・コンテナとしてクローニングされます。

```
CREATE PLUGGABLE DATABASE hract AS APPLICATION CONTAINER FROM pdb1
FILE_NAME_CONVERT = ('/disk1/oracle/pdb1/', '/disk2/oracle/hract/')
STORAGE (MAXSIZE 2G);
```

ノート:



既存のアプリケーションを新しいアプリケーション・コンテナに移行している場合、[「アプリケーション・コンテナへの既存のアプリケーションの移行」](#)の手順に従います。

例12-3 切断されたPDBの接続によるアプリケーション・コンテナの作成

この例では、次の要因を想定しています。

- 新しいアプリケーション・コンテナは、CDB内に既存のPDBまたはアプリケーション・コンテナを作成するために使用された同じ切断されたPDBには基づいていません。したがって、AS CLONE句は必要ありません。
- PATH_PREFIX句は必要ありません。
- XMLファイルでは、ファイルの現在の場所が正確に記述されていません。したがって、SOURCE_FILE_NAME_CONVERT句またはSOURCE_FILE_DIRECTORY句が必要です。この例では、ファイルが /disk1/oracle/payroll/ に存在することがXMLファイルで示されていますが、ファイルは /disk2/oracle/payroll/ にあり、SOURCE_FILE_NAME_CONVERT句が使用されています。
- ファイルは、正しい場所に存在します。したがって、NOCOPYが含まれています。
- アプリケーション・コンテナに対してストレージ制限を適用する必要があります。したがって、STORAGE句が必要です。特に、

アプリケーション・コンテナに属するすべての表領域が2GBを超えていない必要があります。

- ターゲットの場所に、XMLファイルで指定されている一時ファイルと同じ名前のファイルがあります。したがって、TEMPFILE REUSE句が必要です。

次の文は、PDBを接続します。

```
CREATE PLUGGABLE DATABASE payrollact AS APPLICATION CONTAINER
USING '/disk1/usr/payrollpdb.xml'
SOURCE_FILE_NAME_CONVERT = ('/disk1/oracle/payroll/',
                             '/disk2/oracle/payroll/')
NOCOPY
STORAGE (MAXSIZE 2G)
TEMPFILE REUSE;
```

ノート:



既存のアプリケーションを新しいアプリケーション・コンテナに移行している場合、[「アプリケーション・コンテナへの既存のアプリケーションの移行」](#)の手順に従います。

関連トピック

- [現在のコンテナについて](#)
- [アプリケーション・コンテナの管理](#)
- [CDB内のコンテナへのアクセス](#)
- [PDBのオープン・モードの変更](#)

親トピック: [アプリケーション・コンテナの作成](#)

CDBからのアプリケーション・コンテナの切断

アプリケーション・コンテナをCDBから切断できます。

- [アプリケーション・コンテナの切断について](#)
アプリケーション・コンテナを切断すると、そのアプリケーション・コンテナはCDBから関連付け解除されます。
- [アプリケーション・コンテナの切断](#)
ALTER PLUGGABLE DATABASE ... UNPLUG INTO文を使用してアプリケーション・コンテナを切断します。

親トピック: [アプリケーション・コンテナの作成および削除](#)

アプリケーション・コンテナの切断について

アプリケーション・コンテナを切断すると、アプリケーション・コンテナがCDBから関連付け解除されます。

通常、アプリケーション・コンテナは、そのアプリケーション・コンテナを別のCDBに移動する場合に切断します。アプリケーション・コンテナは、不要になった場合も切断できます。

アプリケーション・コンテナの切断は、PDBの切断と似ています。アプリケーション・コンテナを切断するには、CDBルートに接続し、ALTER PLUGGABLE DATABASE文を使用して、XMLファイルまたは.pdbファイルを指定します。XMLファイル(拡張子.xml)を指定すると、そのファイルには、切断後にアプリケーション・コンテナに関するメタデータが含まれます。このSQL文によってXMLファイルが作成され、このXMLファイルには、ターゲットCDBでCREATE PLUGGABLE DATABASE文を使用してアプリケーション・コンテナを接続できるようにするのに必要な情報が含まれます。 .pdbファイルを指定すると、そのファイルにはアプリケーション・コンテナに

ついて記述したXMLファイルおよびアプリケーション・コンテナで使用されるファイル(データファイルやウォレット・ファイルなど)の圧縮アーカイブが含まれます。pdbファイルにより、単一の圧縮されたファイル(複数ファイルではない)を新しい場所にコピーしてアプリケーション・コンテナをCDBに接続できます。

切断する前に、アプリケーション・コンテナにアプリケーションPDBが接続されていない必要があり、クローズされている必要があります。アプリケーション・コンテナを切断する場合、切断されたアプリケーション・コンテナはマウント・モードになります。切断操作により、記録するアプリケーション・コンテナのデータファイルに、たとえばアプリケーション・コンテナが正常に切断されたことなどの変更が加えられることがあります。依然としてCDBの一部であるため、切断されたアプリケーション・コンテナはCDB全体のRMANバックアップに含まれています。このような方法によるバックアップは、切断されたアプリケーション・コンテナが将来必要になった場合に備えてアーカイブするのに便利です。

アプリケーション・コンテナをCDBから完全に削除するには、それを削除できます。切断されたアプリケーション・コンテナに対してサポートされる操作は、アプリケーション・コンテナの削除のみです。アプリケーション・コンテナは、同じCDBに再接続する前に、CDBから削除する必要があります。アプリケーション・コンテナは、CDBに接続されているときにのみ使用可能です。

関連項目:

- [「CDBからのPDBの切断」](#)
- [「アプリケーション・コンテナの削除」](#)
- PDBのクローズの詳細は、[「PDBのオープン・モードの変更」](#)を参照してください
- 初期化パラメータおよび切断されたPDBの詳細は、[「システム・レベルでのPDBの変更」](#)を参照してください。
- 共通ユーザーおよびローカル・ユーザーの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDBからのアプリケーション・コンテナの切断](#)

アプリケーション・コンテナの切断

ALTER PLUGGABLE DATABASE ... UNPLUG INTO文を使用してアプリケーション・コンテナを切断します。

前提条件

次の前提条件を満たしている必要があります。

- 現在のユーザーにはSYSDBAまたはSYSOPER管理権限が必要であり、その権限は共通で付与されているか、またはPDBでローカルに付与されている必要があります。ユーザーは、接続時にAS SYSDBAまたはAS SYSOPERを使用して権限を行使する必要があります。
- アプリケーション・コンテナは少なくとも1回オープンされている必要があります。
- アプリケーション・コンテナにアプリケーションPDBが接続されていない必要があります。
- アプリケーション・コンテナにアプリケーション・シードが接続されていない必要があります。

ノート:



透過的データ暗号化で暗号化されているデータが含まれたアプリケーション・コンテナを切断する場合は、[『Oracle Database Advanced Security ガイド』](#)の指示に従います。

アプリケーション・コンテナを切断するには:

1. SQL*Plusで、現在のコンテナがCDBのルートであることを確認します。
2. アプリケーション・コンテナをクローズします。

Oracle Real Application Clusters (Oracle RAC)環境では、アプリケーション・コンテナがすべてのインスタンスでクローズされている必要があります。

3. UNPLUG INTO句を指定してALTER PLUGGABLE DATABASE文を実行し、切断するアプリケーション・コンテナおよびアプリケーション・コンテナのXMLメタデータ・ファイルまたは.pdbファイルの名前と場所を指定します。

例12-4 アプリケーション・コンテナsalesactの切断

このALTER PLUGGABLE DATABASE文は、アプリケーション・コンテナsalesactを切断し、salesact.xmlメタデータ・ファイルを/oracle/data/ directoryに作成します。

```
ALTER PLUGGABLE DATABASE salesact UNPLUG INTO '/oracle/data/saleact.xml';
```

親トピック: [CDBからのアプリケーション・コンテナの切断](#)

アプリケーション・コンテナの削除

アプリケーション・コンテナは、アプリケーション・コンテナをあるCDBから別のCDBに移動する場合、またはアプリケーション・コンテナが不要になった場合に削除できます。

アプリケーション・コンテナの削除は、PDBの削除と非常に似ています。アプリケーション・コンテナを削除する場合、CDBの制御ファイルが変更され、削除されたアプリケーション・コンテナへの参照がすべて除外されます。アプリケーション・コンテナに関連付けられているアーカイブREDOログ・ファイルおよびバックアップは削除されませんが、Oracle Recovery Manager (RMAN)を使用して削除できます。

アプリケーション・コンテナを削除する場合、そのアプリケーション・コンテナのデータファイルは、DROP PLUGGABLE DATABASE文にある次のいずれかの句を使用して保持または削除できます。

- KEEP DATAFILESはデフォルトであり、データファイルを保持します。
KEEP DATAFILESが指定されていても、アプリケーション・コンテナの一時ファイルは不要になったため削除されます。
- INCLUDING DATAFILESでは、ディスクからデータファイルを削除します。
アプリケーション・コンテナがSNAPSHOT COPY句を指定して作成された場合は、アプリケーション・コンテナを削除する際にINCLUDING DATAFILESを指定する必要があります。

次の前提条件を満たしている必要があります。

- アプリケーション・コンテナはマウント・モードであるか、切断されている必要があります。
[「PDBのオープン・モードの変更」](#)を参照してください。
[「アプリケーション・コンテナの切断」](#)を参照してください。
- 現在のユーザーにはSYSDBAまたはSYSOPER管理権限が必要であり、その権限は共通で付与されているか、またはアプリケーション・コンテナでローカルに付与されている必要があります。ユーザーは、接続時にAS SYSDBAまたはAS SYSOPERを使用して権限を行使する必要があります。
- アプリケーション・コンテナにアプリケーションPDBが接続されていない必要があります。
- アプリケーション・コンテナにアプリケーション・シードが接続されていない必要があります。



ノート:

これは、破壊的な操作です。

アプリケーション・コンテナを削除するには:

1. SQL*Plusで、現在のコンテナがCDBルートであることを確認します。
[「現在のコンテナについて」](#)および[「SQL*Plusを使用したCDB内のコンテナへのアクセス」](#)を参照してください。
2. DROP PLUGGABLE DATABASE文を実行し、削除するアプリケーション・コンテナを指定します。

例12-5 データファイルを保持した状態でのアプリケーション・コンテナsalesactの削除

```
DROP PLUGGABLE DATABASE salesact  
KEEP DATAFILES;
```

例12-6 アプリケーション・コンテナsalesactとデータファイルの削除

```
DROP PLUGGABLE DATABASE salesact  
INCLUDING DATAFILES;
```

関連項目:

- [「アプリケーション・コンテナの切断」](#)
- [「PDBの削除」](#)
- [「スナップショットのコピーPDBの記憶域要件」](#)
- [Oracle Database SQL言語リファレンス](#)
- RMANの詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド』](#)を参照してください

親トピック: [アプリケーション・コンテナの作成および削除](#)

アプリケーション・シードの作成および削除

アプリケーション・シードは、PDBシードの使用、既存のPDBまたは非CDBのクローニング、切断されたPDBの接続など、複数の方法で作成できます。アプリケーション・シードをアプリケーション・コンテナから削除することもできます。

- [アプリケーション・シードの作成](#)
CREATE PLUGGABLE DATABASE文を使用して、アプリケーション・コンテナ内にアプリケーション・シードを作成できます。
- [アプリケーション・コンテナからのアプリケーション・シードの切断](#)
アプリケーション・コンテナからアプリケーション・シードを切断できます。
- [アプリケーション・シードの削除](#)
DROP PLUGGABLE DATABASE文を使用して、アプリケーション・シードを削除できます。アプリケーション・シードが必要でなくなった場合は削除できます。

親トピック: [アプリケーション・コンテナおよびシードの作成と削除](#)

アプリケーション・シードの作成

CREATE PLUGGABLE DATABASE文を使用して、アプリケーション・コンテナ内にアプリケーション・シードを作成できます。

- [アプリケーション・シードの作成について](#)
アプリケーション・コンテナに新しいアプリケーション・シードを作成するには、CREATE PLUGGABLE DATABASE文でAS SEED句を使用します。
- [アプリケーション・シードの準備](#)
アプリケーション・シードを作成する前に前提条件を満たす必要があります。
- [アプリケーション・シードの作成](#)
アプリケーション・シードは、AS SEED句をCREATE PLUGGABLE DATABASE文に含めることで作成します。

親トピック: [アプリケーション・シードの作成および削除](#)

アプリケーション・シードの作成について

アプリケーション・コンテナに新しいアプリケーション・シードを作成するには、CREATE PLUGGABLE DATABASE文でAS SEED句を使用します。

アプリケーション・シードを使用して、アプリケーションPDBにアプリケーション・ルートのアプリケーションがインストールされているアプリケーション・コンテナをプロビジョニングできます。通常は、アプリケーション・コンテナのアプリケーションは、アプリケーション・シードの作成前にアプリケーション・ルートにインストールされます。アプリケーション・シードの作成後、アプリケーションがアプリケーション・シードにインストールされるように、アプリケーション・シードがアプリケーション・ルートと同期されます。それが完了すると、アプリケーション・シードを使用して作成されたPDBにアプリケーションがインストールされます。アプリケーション・ルートのアプリケーションがアップグレードまたはパッチ適用される場合、アプリケーション・シードをアプリケーション・ルートと同期して、これらの変更を適用する必要があります。

アプリケーション・コンテナにはゼロまたは1つのアプリケーション・シードを含めることができます。CREATE PLUGGABLE DATABASEのAS SEED句を使用してアプリケーション・シードを作成する場合、名前は指定しません。アプリケーション・シード名は常にapplication_container_name\$SEEDで、application_container_nameはアプリケーション・シードのアプリケーション・コンテナの名前です。たとえば、salesactアプリケーション・コンテナのアプリケーション・シードの名前はsalesact\$SEEDである必要があります。

新規アプリケーション・シードの作成時に、CREATE PLUGGABLE DATABASE文でアプリケーション・コンテナの管理者を指定する必要があります。この文では、管理者をアプリケーション・コンテナのローカル・ユーザーとして作成し、PDB_DBAロールを管理者にローカルに付与します。

関連項目:

- [「PDBを最初から作成」](#)
- [アプリケーション・コンテナ内のアプリケーションの管理](#)
- [「アプリケーションPDB内のアプリケーションの同期」](#)
- AS SEED文の構文とセマンティクスの詳細は、[Oracle Database SQL言語リファレンス](#)を参照してください。

親トピック: [アプリケーション・シードの作成](#)

アプリケーション・シードの準備

アプリケーション・シードを作成する前に、前提条件を満たす必要があります。

アプリケーション・シードを作成する前に、次の前提条件が満たされていることを確認してください。

- CDBが存在していることが必要です。

[「CDBの作成および構成」](#)を参照してください。

- CDBは、読取り/書込みモードである必要があります。
- アプリケーション・シードが属するアプリケーション・コンテナは読取り/書込みモードである必要があります。
- 現在のユーザーは、アプリケーション・シードが属するアプリケーション・ルートを現在のコンテナとして持つ共通ユーザーである必要があります。
- 現行ユーザーにはCREATE PLUGGABLE DATABASEシステム権限が必要です。
- アプリケーション・シードにアプリケーション・コンテナのアプリケーションを含めるには、アプリケーションをアプリケーション・ルートにインストールする必要があります。

関連項目:

- [「現在のコンテナについて」](#)
- [アプリケーション・コンテナ内のアプリケーションの管理](#)

親トピック: [アプリケーション・シードの作成](#)

アプリケーション・シードの作成

アプリケーション・シードは、CREATE PLUGGABLE DATABASE文にAS SEED句を含めることで作成します。

アプリケーション・コンテナ内のアプリケーション・シードは、CDB内のシードと同様です。アプリケーション・シードにより、アプリケーション・コンテナの要件を素早く簡単に満たすアプリケーションPDBを作成できます。

アプリケーション・シードを作成する前に、[「アプリケーション・シードの準備」](#)で説明した前提条件を満たします。

1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
2. CREATE PLUGGABLE DATABASE文を実行し、アプリケーション・シードを作成するためのAS SEED句を含めます。要求された場合には、他の句を指定します。
アプリケーション・シードを作成した後、それはマウント・モードになり、そのステータスはNEWになります。アプリケーション・シードのオープン・モードは、V\$PDBSビューのOPEN_MODE列を問い合わせることで表示できます。CDB_PDBSまたはDBA_PDBSビューのSTATUS列を問い合わせ、アプリケーション・シードのステータスを表示できます。
新規デフォルト・サービスがアプリケーション・シードに対して作成されます。サービスにはアプリケーション・シードと同じ名前があり、アプリケーション・シードへのアクセスに使用できます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。
3. 新規アプリケーション・シードを読取り/書込みモードでオープンします。
4. Oracle Databaseの新規アプリケーション・シードを読取り/書込みモードでオープンして、新規アプリケーション・シードのアプリケーション・コンテナへの統合を完了する必要があります。アプリケーション・シードを読取り専用モードでオープンしようとする、エラーが返されます。アプリケーション・シードを読取り/書込みモードでオープンした後、そのステータスはNORMALになります。
5. 次のアクションを1つ以上実行します。
 - アプリケーション・シードがPDBシードから作成された場合は、コンテナをアプリケーション・シードに切り替え、SYNC句を指定したALTER PLUGGABLE DATABASE文を使用して、アプリケーション・シードを同期します。アプリ

ケーション・ルートを同期すると、アプリケーション・シードにアプリケーション・ルートのアプリケーションの1つ以上がインスタンス化されます。

- アプリケーション・シードがアプリケーション・ルートから作成された場合は、コンテナをアプリケーション・シードに切り替え、pdb_to_apppdb.sql スクリプトを実行してアプリケーション・ルートをアプリケーションPDBに変換します。

アプリケーション・シードがアプリケーションPDBをクローニングすることによって作成された場合、これらのアクションは必要ありません。

6. アプリケーション・シードをクローズし、オープン読み取り専用モードでオープンします。

7. アプリケーション・シードをバックアップします。

アプリケーション・シードは、バックアップしていないとリカバリできません。

ノート:

- アプリケーション・シードの作成中にエラーが返された場合、作成されているアプリケーション・シードは UNUSABLE 状態になります。CDB_PDBS または DBA_PDBS ビューを問い合わせるアプリケーション・シードの状態を確認したり、アラート・ログをチェックしてアプリケーション・シードの作成エラーについてさらに学習できます。使用できないアプリケーション・シードは削除するしかありません。
- アプリケーション・ルートのアプリケーションがアプリケーション・ルートでアップグレードまたはパッチ適用される場合、アプリケーション・シードをアプリケーション・ルートと同期して、変更を含める必要があります。

例12-7 PDBシードからのアプリケーション・シードの作成

この例では、次の要因を想定しています。

- アプリケーション・シードは、salesactというアプリケーション・コンテナに作成されています。
- ストレージ制限は、アプリケーション・シードには不要です。したがって、STORAGE句は必要ありません。
- アプリケーション・シードにはデフォルトの表領域は必要ありません。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

CDBに対してOracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。PDBシードに関連付けられているファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。
- Oracleの事前定義されたロールをPDB_DBAロールに付与する必要はありません。

次の文はPDBシードからアプリケーション・シードを作成し、アプリケーション・シードを開き、コンテナをアプリケーション・シードに切り替え、アプリケーション・シードをアプリケーション・ルート内のアプリケーションと同期し、アプリケーション・シードを閉じてから、読み取り専用オープン・モードでアプリケーション・シードを開きます。

```
CREATE PLUGGABLE DATABASE AS SEED
  ADMIN USER actseedadm IDENTIFIED BY password;
ALTER PLUGGABLE DATABASE salesact$SEED OPEN;
ALTER SESSION SET CONTAINER=salesact$SEED;
ALTER PLUGGABLE DATABASE APPLICATION ALL SYNC;
ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE OPEN READ ONLY;
```

アプリケーション・コンテナ名はsalesactであるため、アプリケーション・シード名はsalesact\$SEEDです。

指定されたローカル管理者の名前を持つローカル・ユーザーが作成され、このユーザーにアプリケーション・シードでローカルにPDB_DBA共通ロールが付与されます。このユーザーがアプリケーション・シードの作成時に管理者権限を付与されていない場合は、SYSおよびSYSTEM共通ユーザーを使用してアプリケーション・シードを管理します。

アプリケーション・シードは、作成時にアプリケーション・ルートと同期されます。このため、アプリケーション・シードにはアプリケーション・ルートにインストールされているアプリケーション、およびそれらのアプリケーションの一部であるアプリケーション共通オブジェクトが含まれています。アプリケーション・シードを使用して新しいアプリケーションPDBを作成した場合、アプリケーションPDBにはインストールされているアプリケーションおよびアプリケーション共通オブジェクトも含まれています。

例12-8 アプリケーションPDBからのアプリケーション・シードの作成

この例では、次の要因を想定しています。

- アプリケーション・シードは、salesactというアプリケーション・コンテナに作成されています。
- アプリケーション・シードは、salesapppdbというアプリケーション・コンテナのアプリケーションPDBに作成されています。
- ストレージ制限は、アプリケーション・シードには不要です。したがって、STORAGE句は必要ありません。
- アプリケーション・シードにはデフォルトの表領域は必要ありません。
- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

CDBに対してOracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。アプリケーション・ルートに関連付けられているファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

前述の要因により、次の文はアプリケーション・ルートからアプリケーション・シードを作成して、アプリケーション・シードを開き、アプリケーション・シードを閉じて、アプリケーション・シードを読み取り専用オープン・モードで開きます。

```
CREATE PLUGGABLE DATABASE AS SEED FROM salesapppdb;  
ALTER PLUGGABLE DATABASE salesact$SEED OPEN;  
ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;  
ALTER PLUGGABLE DATABASE OPEN READ ONLY;
```

アプリケーション・コンテナ名はsalesactであるため、アプリケーション・シード名はsalesact\$SEEDです。

アプリケーション・シードがアプリケーションPDBから作成されました。このため、アプリケーション・シードにはアプリケーション・ルートにインストールされているアプリケーション、およびそれらのアプリケーションの一部であるアプリケーション共通オブジェクトが含まれています。アプリケーション・シードを使用して新しいアプリケーションPDBを作成した場合、アプリケーションPDBにはインストールされているアプリケーションおよびアプリケーション共通オブジェクトも含まれています。

例12-9 アプリケーション・ルートからのアプリケーション・シードの作成

この例では、次の要因を想定しています。

- アプリケーション・シードは、salesactというアプリケーション・コンテナに作成されています。アプリケーション・シードは、アプリケーション・コンテナのルートからクローニングされます。
- ストレージ制限は、アプリケーション・シードには不要です。したがって、STORAGE句は必要ありません。
- アプリケーション・シードにはデフォルトの表領域は必要ありません。

- PATH_PREFIX句は必要ありません。
- FILE_NAME_CONVERT句およびCREATE_FILE_DEST句は必要ありません。

CDBに対してOracle Managed Filesが有効であるか、またはPDB_FILE_NAME_CONVERT初期化パラメータが設定されています。アプリケーション・ルートに関連付けられているファイルは、Oracle Managed Files構成または初期化パラメータ設定に基づいて新しい場所にコピーされます。

- ターゲットの場所に、作成される新しい一時ファイルと同じ名前のファイルはありません。したがって、TEMPFILE REUSE句は必要ありません。

前述の要因により、次の文はアプリケーション・ルートからアプリケーション・シードを作成し、アプリケーション・シードを開き、コンテナをアプリケーション・シードに切り替え、pdb_to_appdb.sqlスクリプトを実行してアプリケーション・ルートをアプリケーションPDBに変換し、アプリケーション・シードを閉じ、読み取り専用オープン・モードでアプリケーション・シードを開きます。

```
CREATE PLUGGABLE DATABASE AS SEED FROM salesact;
ALTER PLUGGABLE DATABASE salesact$SEED OPEN;
ALTER SESSION SET CONTAINER=salesact$SEED;
@$ORACLE_HOME/rdbms/admin/pdb_to_appdb.sql
ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE OPEN READ ONLY;
```

アプリケーション・コンテナ名はsalesactであるため、アプリケーション・シード名はsalesact\$SEEDです。

アプリケーション・シードはアプリケーション・ルートから作成されました。このため、アプリケーション・シードにはアプリケーション・ルートにインストールされているアプリケーション、およびそれらのアプリケーションの一部であるアプリケーション共通オブジェクトが含まれています。アプリケーション・シードを使用して新しいアプリケーションPDBを作成した場合、アプリケーションPDBにはインストールされているアプリケーションおよびアプリケーション共通オブジェクトも含まれています。

親トピック: [アプリケーション・シードの作成](#)

アプリケーション・コンテナからのアプリケーション・シードの切断

アプリケーション・コンテナからアプリケーション・シードを切断できます。

- [アプリケーション・シードの切断について](#)
アプリケーション・シードを切断すると、アプリケーション・シードがアプリケーション・コンテナから関連付け解除されます。アプリケーション・シードを使用しなくなった場合は、アプリケーション・シードを切断します。
- [アプリケーション・シードの切断](#)
アプリケーション・シードを切断するには、ALTER PLUGGABLE DATABASE ... UNPLUG INTO文を実行します。

親トピック: [アプリケーション・シードの作成および削除](#)

アプリケーション・シードの切断について

アプリケーション・シードを切断すると、アプリケーション・シードがアプリケーション・コンテナから関連付け解除されます。アプリケーション・シードを使用しなくなった場合は、アプリケーション・シードを切断します。

アプリケーション・シードの切断は、PDBの切断と似ています。アプリケーション・シードを切断するには、アプリケーション・ルートに接続し、ALTER PLUGGABLE DATABASE文を使用して、XMLファイルまたは.pdbファイルを指定します。XMLファイル(拡張子.xml)を指定すると、そのファイルには切断後にアプリケーション・シードに関するメタデータが含まれます。このSQL文によってXMLファイルが作成され、このXMLファイルには、ターゲットCDBでCREATE PLUGGABLE DATABASE文を使用し、それをPDBまたはアプリケーションPDBとして接続できるようにするために必要な情報が含まれます。.pdbファイルを指定すると、それにはアプリケーション・シードについて記述したXMLファイルおよびアプリケーション・シードで使用されるファイル(データファイルやウォレット・ファイルなど)の圧縮アーカイブが含まれます。.pdbファイルにより、単一の圧縮されたファイル(複数ファイルではない)を新しい

場所にコピーし、PDBまたはアプリケーションPDBとして接続できます。

切断する前に、アプリケーション・シードがクローズされている必要があります。アプリケーション・シードを切断する場合、切断されたアプリケーション・シードはマウント・モードになります。切断操作により、記録するアプリケーション・シードのデータファイルに、たとえばアプリケーション・シードが正常に切断されたことなどの変更が加えられることがあります。依然としてアプリケーション・コンテナの一部であるため、切断されたアプリケーション・シードはCDB全体のRMANバックアップに含まれています。このような方法によるバックアップは、切断されたアプリケーション・シードが将来必要になった場合に備えてアーカイブするのに便利です。

アプリケーション・シードをアプリケーション・コンテナから完全に削除するには、それを削除できます。切断されたアプリケーション・シードに対してサポートされる操作は、アプリケーション・シードの削除のみです。アプリケーション・シードは、同じアプリケーション・コンテナに再接続する前に、アプリケーション・コンテナから削除する必要があります。アプリケーション・シードは、アプリケーション・コンテナに接続されているときにのみ使用可能です。

関連項目:

- [「CDBからのPDBの切断」](#)
- [「アプリケーション・シードの削除」](#)
- PDBのクローズの詳細は、[「PDBのオープン・モードの変更」](#)を参照してください
- 初期化パラメータおよび切断されたPDBの詳細は、[「システム・レベルでのPDBの変更」](#)を参照してください。
- 共通ユーザーおよびローカル・ユーザーの詳細は、[『Oracle Databaseセキュリティガイド』](#)を参照してください。

親トピック: [アプリケーション・コンテナからのアプリケーション・シードの切断](#)

アプリケーション・シードの切断

アプリケーション・シードを切断するには、ALTER PLUGGABLE DATABASE ... UNPLUG INTO文を実行します。

前提条件

次の前提条件を満たしている必要があります。

- 現在のユーザーにはSYSDBAまたはSYSOPER管理権限が必要であり、その権限は共通で付与されているか、またはアプリケーション・コンテナでローカルに付与されている必要があります。ユーザーは、接続時にAS SYSDBAまたはAS SYSOPERを使用して権限を行使する必要があります。
- アプリケーション・シードは少なくとも1回オープンされている必要があります。

ノート:



透過的データ暗号化で暗号化されているデータが含まれたアプリケーション・シードを切断する場合は、[『Oracle Database Advanced Security ガイド』](#)の指示に従います。

アプリケーション・シードを切断するには:

1. SQL*Plusで、現在のコンテナが、アプリケーション・シードが属するアプリケーション・コンテナのアプリケーション・ルートであることを確認します。

2. アプリケーション・シードをクローズします。

Oracle Real Application Clusters (Oracle RAC)環境では、アプリケーション・シードがすべてのインスタンスでクローズされている必要があります。

3. UNPLUG INTO句を指定してALTER PLUGGABLE DATABASE文を実行し、切断するアプリケーション・シードおよびアプリケーション・シードのXMLメタデータ・ファイルまたは.pdbファイルの名前と場所を指定します。

例12-10 アプリケーション・シードsalesact\$SEEDの切断

このALTER PLUGGABLE DATABASE文は、アプリケーション・シードsalesact\$SEEDを切断し、salesact\$SEED.xmlメタデータ・ファイルを/oracle/data/ディレクトリに作成します。

```
ALTER PLUGGABLE DATABASE salesact$SEED
UNPLUG INTO '/oracle/data/saleact$SEED.xml';
```

親トピック: [アプリケーション・コンテナからのアプリケーション・シードの切断](#)

アプリケーション・シードの削除

DROP PLUGGABLE DATABASE文を使用して、アプリケーション・シードを削除できます。アプリケーション・シードが必要ではなくなった場合は削除できます。

アプリケーション・シードを削除する場合、CDBの制御ファイルが変更され、削除されたアプリケーション・シードへの参照がすべて除外されます。アプリケーション・シードに関連付けられているアーカイブREDOログ・ファイルおよびバックアップは削除されませんが、Oracle Recovery Manager (RMAN)を使用して削除できます。

アプリケーション・シードを削除する場合、次のいずれかの句を使用して、アプリケーション・シードのデータファイルを保持または削除できます。

- KEEP DATAFILESはデフォルトであり、データファイルを保持します。
KEEP DATAFILESが指定されていても、アプリケーション・シードの一時ファイルは不要になったため削除されます。
- INCLUDING DATAFILESでは、ディスクからデータファイルを削除します。
アプリケーション・シードがSNAPSHOT COPY句を指定して作成された場合は、アプリケーション・シードを削除する際にINCLUDING DATAFILESを指定する必要があります。

次の前提条件を満たしている必要があります。

- アプリケーション・シードはマウント・モードであるか、切断されている必要があります。
- 現在のユーザーにはSYSDBAまたはSYSOPER管理権限が必要であり、その権限は共通で付与されているか、またはアプリケーション・コンテナでローカルに付与されている必要があります。ユーザーは、接続時にAS SYSDBAまたはAS SYSOPERを使用して権限を行使する必要があります。

ノート:



これは、破壊的な操作です。

アプリケーション・シードを削除するには:

1. SQL*Plusで、現在のコンテナが、アプリケーション・シードが属するアプリケーション・コンテナのアプリケーション・ルートで

あることを確認します。

2. DROP PLUGGABLE DATABASE文を実行し、アプリケーション・シードを指定します。

例12-11 データファイルを保持した状態でのアプリケーション・シードsalesact\$SEEDの削除

```
DROP PLUGGABLE DATABASE salesact$SEED  
KEEP DATAFILES;
```

例12-12 アプリケーション・シードsalesact\$SEEDとデータファイルの削除

```
DROP PLUGGABLE DATABASE salesact$SEED  
INCLUDING DATAFILES;
```

関連項目:

- [「CDB内のコンテナへのアクセスについて」](#)
- [「PDBのオープン・モードの変更」](#)
- [「アプリケーション・シードの切断」](#)
- [「スナップショットのコピーPDBの記憶域要件」](#)
- [Oracle Database SQL言語リファレンス](#)
- RMANの詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド』](#)を参照してください

親トピック: [アプリケーション・シードの作成および削除](#)

アプリケーションPDBの作成

アプリケーションPDBを作成するには、アプリケーション・ルートが現在のコンテナであるときに、CREATE PLUGGABLE DATABASE文を実行します。


アプリケーションPDBは、CDBルートにPDBを作成するために使用するSQL文と同じSQL文を使用して作成できます。CREATE PLUGGABLE DATABASE文をアプリケーション・ルートで実行した場合、新しく作成したPDBはアプリケーションPDBです。この文はアプリケーション・ルートで実行する必要があり、そのアプリケーション・ルートに定義されているアプリケーション・データベースに対して明確な依存関係があります。

アプリケーションPDBを作成する前に、[「PDB作成の一般的な前提条件」](#)で説明している前提条件を満たします。また、作成する特定のタイプのPDBの前提条件を満たす必要があります。たとえば、PDBをクローニングする場合は、PDBのクローニングの前提条件を満たしている必要があります。

1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
2. CREATE PLUGGABLE DATABASE文を実行します。
作成したアプリケーションPDBはマウント・モードになり、そのステータスはNEWです。アプリケーションPDBのオープン・モードは、V\$PDBSビューのOPEN_MODE列を問い合わせることで表示できます。CDB_PDBSまたはDBA_PDBSビューのSTATUS列を問い合わせ、アプリケーションPDBのステータスを表示できます。
新規デフォルト・サービスがアプリケーションPDBに対して作成されます。サービスにはアプリケーションPDBと同じ名前があり、アプリケーションPDBへのアクセスに使用できます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。
3. 新規アプリケーションPDBを読取り/書き込みモードでオープンします。

4. Oracle Databaseの新規アプリケーションPDBを読み取り/書き込みモードでオープンして、新規アプリケーションPDBのアプリケーション・コンテナへの統合を完了する必要があります。アプリケーションPDBを読み取り専用モードでオープンしようとすると、エラーが返されます。アプリケーションPDBを読み取り/書き込みモードでオープンした後、そのステータスはNORMALになります。
5. コンテナをアプリケーションPDBに切り替えます。
6. SYNC句を指定したALTER PLUGGABLE DATABASE文を使用して、アプリケーションPDBを同期します。
アプリケーションPDBを同期すると、アプリケーションPDBにアプリケーション・ルートのアプリケーションの1つ以上がインスタンス化されます。
7. アプリケーションPDBをクローズし、オープン読み取り専用モードでオープンします。
8. アプリケーションPDBをバックアップします。
アプリケーションPDBは、バックアップしていないとリカバリできません。

ノート:

- 
- アプリケーション PDB の作成中にエラーが返された場合、作成されているアプリケーション PDB は UNUSABLE 状態になります。CDB_PDBS または DBA_PDBS ビューを問い合わせるアプリケーション PDB の状態を確認したり、アラート・ログをチェックしてアプリケーション PDB の作成エラーについてさらに学習できます。使用できないアプリケーション PDB は削除するしかありません。
 - アプリケーション・ルートのアプリケーションがアプリケーション・ルートでアップグレードまたはパッチ適用される場合、アプリケーション PDB をアプリケーション・ルートと同期して、変更を含める必要があります。

関連トピック

- [PDBおよびアプリケーション・コンテナの作成および削除](#)
- [アプリケーション・コンテナの管理](#)

親トピック: [アプリケーション・コンテナおよびシードの作成と削除](#)

第IV部 マルチテナント環境の管理

SQL*PlusまたはEnterprise Manager Cloud Control (Cloud Control)を使用して、マルチテナント環境を管理できます。

- [CDBの管理](#)

マルチテナント・コンテナ・データベース(CDB)の管理には、コンテナへのアクセス、CDBの変更、DDL文の実行、およびOracle社が提供するSQLスクリプトの実行などのタスクが含まれます。

- [CDBフリートの管理](#)

CDBフリートは、1つの論理CDBとして管理できるCDBおよびホストされるPDBの集合です。

- [PDBの管理](#)

PDBの管理には、PDBへの接続、PDBの変更、PDBに関連付けられたサービスの管理などのタスクが含まれます。

- [PDBスナップショット・カラーセルの管理](#)

指定したPDBのPDBスナップショット・カラーセルを構成し、スナップショットを手動または自動的に作成して、スナップショットの最大数を設定できます。

- [アプリケーション・コンテナの管理](#)

アプリケーション・ルートおよびアプリケーションPDBを含むアプリケーション・コンテナを管理できます。アプリケーション・コンテナにインストールされているアプリケーションも管理できます。

- [マルチテナント環境のセキュリティの管理](#)

SQL*PlusおよびOracle Enterprise Managerを使用して、マルチテナント環境の共通およびローカルのユーザーとロールを管理できます。

- [CDBおよびPDBの監視](#)

SQL*PlusまたはSQL Developerを使用してCDBおよびPDBに関する情報を表示できます。

13 CDBの管理

マルチテナント・コンテナ・データベース(CDB)の管理には、コンテナへのアクセス、CDBの変更、DDL文の実行、およびオラクル社が提供するSQLスクリプトの実行などのタスクが含まれます。



ノート:

この章のタスクは、SQL*Plus または Oracle SQL Developer を使用して実行できます。

- [CDB管理について](#)
CDBの管理は非CDBの管理と似ていますが、いくつかの違いがあります。
- [CDB内のコンテナへのアクセス](#)
SQL*Plus CONNECTコマンドを使用してコンテナに接続できます。または、ALTER SESSION SET CONTAINER SQL文を使用してコンテナに切り替えることができます。
- [システム・レベルでのCDBの変更](#)
CDBレベルで初期化パラメータを設定できます。場合によっては、これらのパラメータをPDBレベルで上書きできます。
- [CDBルートへの接続時のコンテナの変更](#)
ALTER DATABASE文を使用して、CDB全体またはルートを変更できます。
- [別のコンテナでのSQLの実行](#)
別のコンテナでSQLを実行するには、DMLのCONTAINERS句、またはDDLのCONTAINER句を使用します。
- [CDBインスタンスの停止](#)
CDBインスタンスは、非CDBインスタンスを停止する場合と同じ方法で停止できます。

関連項目:

[「マルチテナント環境のツール」](#)

親トピック: [マルチテナント環境の管理](#)

CDB管理について

CDBの管理は非CDBの管理と似ていますが、いくつかの違いがあります。

ほとんどの違いは、一部の管理タスクはCDB全体に適用されますが、それ以外のタスクは特定のコンテナに適用されるために発生します。

- [現在のコンテナについて](#)
CDB内の各コンテナのデータ・ディクショナリは分かれており、現在のコンテナのデータ・ディクショナリは、名前解決と権限の承認に使用されます。
- [CDBにおける管理タスクについて](#)
共通ユーザーがCDBの管理タスクを実行します。
- [CDBにおける管理性機能の使用について](#)
CDB内のOracle Databaseの各管理機能について、データの場所とデータの可視性を理解することが重要です。
- [CDB内表領域の管理について](#)

表領域とは、記憶域を使用するデータベース・オブジェクト(表や索引など)の論理記憶域コンテナです。

- [CDBにおけるデータベース・オブジェクトの管理について](#)

CDBでは、異なるコンテナに異なるデータベース・オブジェクトを含めることができます。

- [PDBのフラッシュ・バックについて](#)

FLASHBACK PLUGGABLE DATABASE文を使用して、PDBを過去の時点またはシステム変更番号(SCN)に戻すことができます。

- [セキュリティを強化するためのPDBユーザーの制限について](#)

セキュリティを強化するためにPDBユーザーを制限する方法はいくつかあります。

- [Oracle RACを含むOracle Multitenantの概要](#)

Oracle MultitenantはOracle Database 12cのオプションで、統合、プロビジョニング、アップグレードなどを簡略化します。

親トピック: [CDBの管理](#)

現在のコンテナについて

CDB内の各コンテナのデータ・ディクショナリは分かれており、現在のコンテナのデータ・ディクショナリは、名前解決と権限の承認に使用されます。

現在のコンテナは、CDBルート、アプリケーション・ルート、PDBまたはアプリケーションPDBにできます。各セッションには、任意の時点で現在のコンテナがそれぞれ1つのみ含まれます。ただし、セッションでは、あるコンテナから別のコンテナへ切替えができます。

各コンテナにはCDB内で一意のIDと名前があります。USERENVネームスペースのCON_IDおよびCON_NAMEパラメータを使用すると、SYS_CONTEXTファンクションで現在のコンテナのIDと名前を判断できます。たとえば、次の問合せは現在のコンテナ名を返します。

```
SELECT SYS_CONTEXT (' USERENV' , ' CON_NAME' ) FROM DUAL;
```

コンテナには様々な方法でアクセスできます。たとえば、SQL*PlusのCONNECTコマンドを使用し、ALTER SESSION SET CONTAINER文を使用すると、現在のセッションのコンテナを切り替えることができます。

CDBの現在のコンテナには、次のルールが適用されます。

- 共通ユーザーの場合にのみ、現在のコンテナはCDB\$ROOT (CDBルート)にできます。
- 共通ユーザーおよびローカル・ユーザーの場合に、現在のコンテナは特定のPDBにできます。
- アプリケーション・ルートで作成された共通ユーザーまたはアプリケーション共通ユーザーの場合、現在のコンテナにできるのはアプリケーション・ルートのみです。
- 共通ユーザー、アプリケーション共通ユーザーおよびローカル・ユーザーの場合に、現在のコンテナは特定のアプリケーションPDBにできます。
- SQL文にCONTAINER = ALLが含まれる場合、現在のコンテナはCDBルートまたはアプリケーション・ルートである必要があります。

CONTAINER句は、CREATE USER、ALTER USER、CREATE ROLE、GRANT、REVOKEおよびALTER SYSTEM文などのいくつかのSQL文に含めることができます。CONTAINER = ALLに関する次のルールに注意してください。

- SQL文にCONTAINER = ALLが含まれ、現在のコンテナがCDBルートの場合、SQL文は、すべてのPDB、アプリケーション・ルートおよびアプリケーションPDBを含む、CDB内のすべてのコンテナに影響します。
- SQL文にCONTAINER = ALLが含まれ、現在のコンテナがアプリケーション・ルートの場合、SQL文は、アプリケーション・ルートおよびアプリケーション・ルートに属するすべてのアプリケーションPDBを含む、アプリケーション・

コンテナ内のすべてのコンテナに影響します。SQL文は、CDBルート、PDB、または現在のアプリケーション・ルートに属さないアプリケーションPDBには影響しません。

- 共通にSET CONTAINER権限が付与されている共通ユーザーまたはアプリケーション共通ユーザーのみ、CONTAINER = ALLを含むSQL文を実行できます。

関連項目:

- [「CDB内のコンテナへのアクセスについて」](#)
- [「DBMS_SQLパッケージを使用したコンテナのコードの実行」](#)
- [「現在のコンテナのIDまたは名前の判別」](#)
- [「CDBのネームスペース」](#)
- [Oracle Database SQL言語リファレンス](#)
- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [CDB管理について](#)

CDBにおける管理タスクについて

共通ユーザーがCDBの管理タスクを実行します。

共通ユーザーは1つのIDを持ち、CDBルート、アプリケーション・ルート、PDBまたは権限を持つアプリケーションPDBにログインできます。CDBインスタンスの起動などの一部のタスクは、共通ユーザーのみが実行できます。

その他の管理タスクについては、CDBと非CDBとで違いはありません。次の表は、これらのタスクの一部についての説明と、関連ドキュメントへのポインタを示しています。

表13-1 CDBと非CDBに共通する管理タスク

タスク	説明	追加情報
CDB インスタンスの起動	CDB インスタンスを起動するには、現在のユーザーは、その現在のコンテナが CDB ルートである共通ユーザーである必要があります。	データベースの起動の詳細は、 『Oracle Database 管理者ガイド』 を参照してください 「PDB のオープン・モードの変更」 「ALTER PLUGGABLE DATABASE 文を使用した PDB の変更」 「現在のコンテナについて」
プロセスの管理	CDB には、CDB ルートとすべてのコンテナによって共有された 1 セットのバックグラウンド・プロセスが存在します。	プロセスの管理の詳細は、 『Oracle Database 管理者ガイド』 を参照してください

タスク	説明	追加情報
メモリの管理	CDB には、単一のシステム・グローバル領域(SGA)および単一の集計プログラム・グローバル領域(PGA)があります。CDB で必要なメモリは、CDB に含まれるすべてのコンテナのメモリ要件の合計となります。	メモリの管理の詳細は、 『Oracle Database 管理者ガイド』 を参照してください
セキュリティの管理	CDB では、共通ユーザー、アプリケーション共通ユーザーおよびローカル・ユーザーを作成および削除できます。また、これらのユーザーに対して権限の追加および取消しを行うこともできます。共通ユーザーおよびアプリケーション共通ユーザーの CONTAINER_DATA 属性を管理することもできます。	『Oracle Database セキュリティ・ガイド』
	さらに、次のロールを適切なユーザーに付与します。	
	<ul style="list-style-type: none"> ● CDB_DBA ロールを CDB 管理者に付与します。 ● アプリケーション・コンテナ管理者と PDB 管理者に PDB_DBA ロールを付与します。 	
エラーおよびアラートの監視	CDB には、CDB 全体について 1 つのアラート・ログが存在します。アプリケーション・コンテナ、PDB またはアプリケーション PDB の名前は、該当する場合にトレース・ファイルのレコードに含まれます。	エラーおよびアラートの監視の詳細は、 『Oracle Database 管理者ガイド』 を参照してください
診断データの管理	CDB では、Oracle Database の障害診断インフラストラクチャおよび自動診断リポジトリ(ADR)を使用できます。	診断データの管理の詳細は、 『Oracle Database 管理者ガイド』 を参照してください。
制御ファイルの管理	CDB には 1 つの制御ファイルが存在します。	制御ファイルの管理の詳細は、 『Oracle Database 管理者ガイド』 を参照してください
オンライン REDO ログ・ファイルおよびアーカイブ REDO ログ・ファイルの管	CDB には、1 つのオンライン REDO ログ・ファイルおよび 1 セットのアーカイブ REDO	REDO ログの管理の詳細は、 『Oracle Database 管理者ガイド』 を参照してくだ

タスク	説明	追加情報
理	ログ・ファイルが存在します。	さい アーカイブ REDO ログ・ファイルの管理の詳細は、 『Oracle Database 管理者ガイド』 を参照してください
表領域の管理	CDB ルートおよび個々のコンテナの表領域および一時表領域を作成、変更および削除できます。また、CDB ルートのデフォルトの表領域、デフォルトの表領域タイプおよびデフォルトの一時表領域を指定することもできます。CDB ルートには、オラクル社が提供する表領域(SYSTEM 表領域など)の独自のセットがあり、他のコンテナにはオラクル社が提供する表領域の独自のセットがあります。	表領域の管理の詳細は、 『Oracle Database 管理者ガイド』 を参照してください 「CDB ルートへの接続時のコンテナの変更について」
データファイルおよび一時ファイルの管理	CDB ルートには独自のデータファイルがあり、他のコンテナには独自のデータファイルがあります。CDB では、非 CDB の場合と基本的に同じ方法でデータファイルと一時ファイルを管理できます。ただし、CDB には次の例外が適用されます。 <ul style="list-style-type: none"> ● CREATE PLUGGABLE DATABASE または ALTER PLUGGABLE DATABASE 文で STORAGE 句を使用して、コンテナのデータファイルで使用される記憶域の量を制限できます。 ● CDB ルートおよび個々のコンテナについてデフォルト一時表領域が存在します。 	データファイルおよび一時ファイルの管理の詳細は、 『Oracle Database 管理者ガイド』 を参照してください 「CDB ルートへの接続時のコンテナの変更について」 「ストレージ制限」 「データベース・レベルでの PDB の変更」
UNDO の管理	CDB は、ローカル UNDO モードまたは共有 UNDO モードで実行できます。ローカル UNDO モードは、CDB 内のすべてのコンテナがローカル UNDO を使用することを意味します。共有 UNDO モードは、単一インスタンス CDB に 1 つのアクティブな UNDO 表領域があるか、Oracle RAC CDB の場合はインスタンスごとに 1 つの	「ALTER DATABASE を使用した CDB の UNDO モードの設定」 UNDO の管理の詳細は、 『Oracle Database 管理者ガイド』 を参照してください 「現在のコンテナについて」

タスク	説明	追加情報
	<p>クティブな UNDO 表領域があることを意味します。</p> <p>CDB では、UNDO_MANAGEMENT 初期化パラメータを AUTO に設定する必要があり、UNDO データの管理に UNDO 表領域が必要です。</p>	
コンテナ間のデータの移動	<p>非 CDB 間でデータを移動する場合と同じ方法を使用して、コンテナ間でデータを移動できます。たとえば、データをトランスポートしたり、データ・ポンプ・エクスポートおよびインポートを使用してデータを移動できます。</p>	<p>データのトランスポートの詳細は、『Oracle Database 管理者ガイド』を参照してください</p> <p>『Oracle Database ユーティリティ』</p>
Oracle Managed Files の使用	<p>Oracle Managed Files を使用すると、CDB と非 CDB 両方の管理を簡素化できます。</p>	<p>Oracle Managed Files の使用の詳細は、『Oracle Database 管理者ガイド』を参照してください</p>
透過的データ暗号化の使用	<p>透過的データ暗号化は、個々の表の列をデータファイルに格納する前に暗号化するか、表領域全体を暗号化する機能です。CDB には、コンテナごとに透過的データ暗号化の独自のマスター・キーが存在し、適用可能な場合、ADMINISTER KEY MANAGEMENT SQL 文によって、コンテナごとに CDB レベルでキー管理を行うことができます。</p>	<p>Oracle Database Advanced Security ガイド</p> <p>「現在のコンテナについて」</p>
スタンバイ・データベースの使用	<p>Oracle Data Guard を使用すると、CDB のフィジカル・スタンバイまたはロジカル・スタンバイを構成できます。Data Guard は、CDB 内の個々のコンテナではなく CDB 全体を操作します。</p>	<p>『Oracle Data Guard 概要および管理』</p>
Oracle Database Vault の使用	<p>Oracle Database Vault 共通レلمは、共通オブジェクトのアプリケーション・ルートにスコープ指定できます。Database Vault 共通コマンド・ルールは、CDB またはアプリケーション・ルートにスコープ指定できます。ローカル・レلمとコマンド・ルールは、個々の PDB またはアプリケーション</p>	<p>Oracle Database Vault の管理者ガイド</p>

タスク	説明	追加情報
	PDB にローカルにスコープ指定できます。 Oracle Database Vault セキュリティ・オブジェクトが CDB ルートまたはアプリケーション・ルートにある場合は、セキュリティ・オブジェクトの施行は Oracle Database Vault が有効になっているコンテナにのみ適用されます。	
データベースの削除	CDB を削除すると、CDB 内のすべてのコンテナがデータとともに削除されます。これらのコンテナには、CDB ルートおよび PDB シードと、すべてのアプリケーション・コンテナ、アプリケーション・シード、PDB およびアプリケーション PDB が含まれます。 DROP PLUGGABLE DATABASE 文を使用して、個々のアプリケーション・コンテナ、アプリケーション・シード、PDB およびアプリケーション PDB を削除することもできます。	データベースの削除の詳細は、 『Oracle Database 管理者ガイド』 を参照してください 「PDB の削除」

関連項目:

CDBのアーキテクチャの詳細は、[「マルチテナント・アーキテクチャの概要」](#)を参照してください

親トピック: [CDB管理について](#)

CDBにおける管理性機能の使用について

CDB内のOracle Databaseの各管理機能について、データの場所とデータの可視性を理解することが重要です。

CDBルートに機能データが存在している場合、そのデータはPDBの切断時に含まれなくなります。ただし、そのデータがPDBに存在する場合、データはPDBの切断時と接続時のどちらでも存続します。

通常、CDBでは、共有ユーザーの現在のコンテナがCDBルートである場合、共通ユーザーはCDBルートおよび複数のPDBのデータを表示できます。共通ユーザーは、コンテナ・データ・オブジェクトを問い合わせることでこのデータを表示できます。表示できる特定のデータは、管理性機能によって異なります。現在のコンテナがPDBであるユーザーは、そのPDBのデータのみを表示できます。

次の表で、CDBにおける管理性機能の動作について説明します。

表13-2 CDBにおける管理性機能

管理性機能	データの場所	データの可視性	追加情報
アクティブ・セッション履歴	ほとんどの ASH データは、メモリーに格納されます。ごく一部	現在のコンテナが CDB ルートである共通ユーザーは、CDB	Oracle Database 2 日でパフォーマンス・チューニング・ガ

管理性機能	データの場所	データの可視性	追加情報
<p>(ASH)</p> <p>ASH により、アクティブ・データベース・セッションに関する情報が収集されます。この情報を使用して、パフォーマンス上の問題を分析および識別できます。</p>	<p>の ASH データ・サンプルは、CDB ルートに格納されます。</p> <p>PDB が切断されていると、PDB に関連する ASH データは含まれません。</p>	<p>ルートと PDB の ASH データを表示できます。</p> <p>現在のコンテナが PDB であるユーザーは、その PDB の ASH データのみを表示できません。</p>	<p>ID</p> <p>Oracle Database パフォーマンス・チューニング・ガイド</p>
<p>アラート</p> <p>アラートとは、考えられる問題の通知です。</p>	<p>PDB に関連するしきい値設定は、PDB に格納されます。</p> <p>しきい値に違反したときに転送されるアラートは、CDB ルートのアラート・キューにエンキューされます。</p> <p>PDB に関連するしきい値設定は、PDB が切断されている場合も含まれます。PDB が切断されていると、PDB に関連するアラートは含まれません。</p>	<p>現在のコンテナが CDB ルートである共通ユーザーは、CDB ルートと PDB のアラートを表示できます。</p> <p>現在のコンテナが PDB であるユーザーは、その PDB のアラートしきい値およびアラートのみを表示できます。</p>	<p>エラーおよびアラートの監視の詳細は、『Oracle Database 管理者ガイド』を参照してください</p>
<p>自動データベース・メンテナンス・タスク</p> <p>自動化データベース・メンテナンス・タスクとは、データベースのメンテナンス操作を実行するために、一定の間隔をおいて自動的に開始されるタスクです。自動タスクには、自動オプティマイザ統計収集、自動セグメント・アドバイザ・タスクおよび自動 SQL チューニング・アドバイザ・タスクが含まれます。</p> <p>ENABLE_AUTOMATIC_MAINTENANCE_PDB 初期化パラメータは、CDB 内のすべての PDB または CDB 内の個別の PDB での自動メンテナンス・タスクの実行を有効または無効</p>	<p>ユーザーは、メンテナンス期間をスケジュールして、現在のコンテナのみのメンテナンス・タスクを有効または無効にできます。現在のコンテナが CDB ルートである場合、変更は CDB ルートにのみ適用されます。現在のコンテナが PDB の場合、変更は PDB にのみ適用されます。</p> <p>PDB に関連するデータは、自動オプティマイザ統計収集および自動セグメント・アドバイザの PDB に格納されます。このデータは、PDB が切断されている場合も含まれます。</p> <p>自動 SQL チューニング・アドバイザは、CDB ルートでのみ</p>	<p>管理性機能(自動オプティマイザ統計収集、オプティマイザ統計アドバイザ、自動セグメント・アドバイザおよび自動 SQL チューニング・アドバイザ)に関するデータの可視性の詳細は、この表の適切な行を参照してください。</p>	<p>データベースの自動メンテナンス・タスクの詳細は、『Oracle Database 管理者ガイド』を参照してください</p> <p>ENABLE_AUTOMATIC_MAINTENANCE_PDB 初期化パラメータの詳細は、『Oracle Database リファレンス』を参照してください</p> <p>AUTOTASK_MAX_ACTIVE_PDBS 初期化パラメータの詳細は、『Oracle Database リファレンス』を参照してください</p>

管理性機能	データの場所	データの可視性	追加情報
<p>にできます。</p> <p>AUTOTASK_MAX_ACTIVE_PDBS 初期化パラメータは、自動メンテナンス・タスクを同時(メンテナンス・ウィンドウ中)にスケジュールできる PDB の数を制限します。</p>	<p>実行されます。自動 SQL チューニング・アドバイザによって収集されたデータの詳細は、この表の SQL チューニング・アドバイザの行を参照してください。</p>		
<p>自動データベース診断モニター(ADDM)</p> <p>ADDM では、CDB または PDB のパフォーマンスを診断して、特定された問題の解決方法を判断できます。</p>	<p>ADDM は、PDB または CDB ルートで実行されます。ADDM は、次のいずれかのソースを使用してデータを分析します。</p> <ul style="list-style-type: none"> ● PDB 内で取得した AWR スナップショットを通じて PDB 内に格納されている AWR データ ● 非 CDB、CDB ルートまたは PDB から PDB の AWR 記憶域にインポートされている AWR データ ● ルートで取得した AWR スナップショットを通じてルート・コンテナに格納されている AWR データ <p>分析の開始前に、ADDM は AWR データのソース(PDB または CDB ルート)を判断して、それぞれのデータ型に適用可能なルールを適用します。</p> <p>ノート: PDB の自動 ADDM は、その PDB の自動スナップショットが有効になっている場合にのみ有効です。</p>	<p>現在のコンテナが CDB ルートである共通ユーザーは、CDB 全体の結果を確認できます。ADDM 結果には、複数の PDB に関する情報が含まれる場合があります。PDB が切断されていると、PDB に関連する ADDM 結果は含まれません。現在のコンテナが PDB である場合、ADDM 結果を表示することはできません。</p> <p>現在のコンテナが PDB であるユーザーは、現在の PDB のみの ADDM 結果データを表示できます。この結果には、CDB 全体に該当する診断結果(バッファ・キャッシュ・サイズに関連する I/O の問題など)は含まれません。</p>	<p>Oracle Database 2 日でデータベース管理者</p> <p>Oracle Database パフォーマンス・チューニング・ガイド</p>

管理性機能	データの場所	データの可視性	追加情報
<p>自動オプティマイザ統計収集</p> <p>自動オプティマイザ統計収集により、データベース内に統計がないか、古い統計のみがあるすべてのスキーマ・オブジェクトに関するオプティマイザ統計を収集します。このタスクで収集された統計は、SQLの実行パフォーマンスを改善するためにSQL問合せオプティマイザによって使用されます。</p>	<p>自動オプティマイザ統計収集タスクにより PDB のデータが収集されると、このデータは PDB に格納されます。このデータは、PDB が切断されていても含まれます。</p>	<p>現在のコンテナが CDB ルートである共通ユーザーは、PDB のオプティマイザ統計データを表示できます。</p> <p>現在のコンテナが PDB であるユーザーは、その PDB のオプティマイザ統計データのみを表示できます。</p>	<p>「PDB に対する Oracle Resource Manager の使用」</p> <p>Oracle Database SQL チューニング・ガイド</p>
<p>自動セグメント・アドバイザー</p> <p>自動セグメント・アドバイザーで、再生可能な領域が存在しているセグメントを識別し、それらのセグメントの断片化を解消する方法について推奨事項を生成します。</p>	<p>自動セグメント・アドバイザーによって PDB のデータが収集されると、このデータは PDB に格納されます。このデータは、PDB が切断されていても含まれます。</p>	<p>現在のコンテナが CDB ルートである共通ユーザーは、PDB の自動セグメント・アドバイザー・データを表示できます。</p> <p>現在のコンテナが PDB であるユーザーは、その PDB の自動セグメント・アドバイザー・データのみを表示できます。</p>	<p>未使用領域の解放の詳細は、『Oracle Database 管理者ガイド』を参照してください</p> <p>「PDB に対する Oracle Resource Manager の使用」</p>
<p>自動ワークロード・リポジトリ (AWR)</p> <p>AWR により、問題の検出および自己チューニングを目的として、パフォーマンス統計を収集、処理およびメンテナンスします。このデータはデータベースに格納されます。収集されたデータは、レポートとビューに表示できます。</p>	<p>AWR レポートは、CDB ルートまたは PDB で生成できます。CDB ルートで生成された AWR レポートは CDB 全体に関係し、PDB が現在のコンテナであるときに生成された AWR レポートはその PDB に関係します。</p> <p>CDB ルートで生成された AWR データは、その CDB ルートに格納されます。PDB で生成された AWR データは、PDB に格納されます。</p> <p>PDB が切断されている場合、CDB ルートに格納されている AWR データは含まれません。</p>	<p>現在のコンテナが CDB ルートである共通ユーザーは、CDB ルートと PDB の AWR データを表示できます。</p> <p>現在のコンテナが PDB であるユーザーは、その PDB の AWR データのみを表示できます。</p>	<p>Oracle Database パフォーマンス・チューニング・ガイド</p>

管理性機能	データの場所	データの可視性	追加情報
	PDB に格納された AWR データは、PDB が切断されていても含められます。		
データベース・リプレイ データベース・リプレイは、Oracle Real Application Testing の機能です。データベース・リプレイは、CDB または PDB のワークロードを取得して、そのワークロードをテスト・データベースで正確にリプレイします。	取得ファイルは常にオペレーティング・システム・ファイルに格納されます。取得とリプレイが CDB レベルであるか PDB レベルであるかは関係ありません。	CDB レベルのワークロードの場合、現在のコンテナが CDB ルートである共通ユーザーは、データベースの取得およびリプレイの情報を表示できます。PDB レベルのワークロードの場合、SELECT_CATALOG_ROLE 権限を持つローカルまたは共通 PDB 管理者は、この情報を DBA_WORKLOAD_CAPTURES および DBA_WORKLOAD_REPLAYS で確認できます。	『Oracle Database テスト・ガイド』
オプティマイザ統計アドバイザ オプティマイザ統計アドバイザは、統計の収集方法を分析し、統計収集を微調整するために行うことのできる変更を提案します。	PDB に関連するデータは、オプティマイザ統計アドバイザの PDB に格納されます。このデータは、PDB が切断されていても含まれます。	現在のコンテナが CDB ルートである共通ユーザーは、PDB のオプティマイザ統計アドバイザ・データを表示できます。 現在のコンテナが PDB であるユーザーは、その PDB のオプティマイザ統計アドバイザ・データのみを表示できます。	Oracle Database SQL チューニング・ガイド
SQL 管理ベース(SMB) SMB により、文のログ、計画履歴、SQL 計画ベースラインおよび SQL プロファイルが、データ・ディクショナリに格納されます。	PDB に関連する SMB データは、PDB に格納されます。PDB に関連する SMB データは、PDB が切断されていても含められます。	現在のコンテナが CDB ルートである共通ユーザーは、PDB の SMB データを表示できます。 現在のコンテナが PDB であるユーザーは、その PDB の SMB データのみを表示できます。	Oracle Database SQL チューニング・ガイド
SQL パフォーマンス・アナライザ(SPA) SPA では、SQL チューニング	現在のコンテナが CDB ルートである共通ユーザーは、任意の PDB の SPA を実行できます。この場合、SPA の結果	現在のコンテナが CDB ルートである共通ユーザーは、PDB の SPA 結果データを表示で	『Oracle Database テスト・ガイド』

管理性機能	データの場所	データの可視性	追加情報
<p>およびその他のシステム変更が SQL パフォーマンスに及ぼす影響を分析できます。通常、SPA はデータベース・リプレイで使用されます。</p>	<p>データは CDB ルートに格納され、PDB が切断されている場合は含まれません。</p> <p>現在のコンテナが PDB であるユーザーは、その PDB について SPA を実行できます。この場合、SPA 結果データは PDB に格納され、PDB が切断されていても含まれます。</p>	<p>きます。</p> <p>現在のコンテナが PDB であるユーザーは、その PDB の SPA 結果データのみを表示できます。</p>	
<p>SQL チューニング・セット (STS)</p> <p>STS は、1 つ以上の SQL 文とその実行統計および実行コンテキストを含むデータベース・オブジェクトで、ユーザーの優先順位ランキングを含む場合があります。</p> <p>STS を使用すると、SQL 文のグループをチューニングしたり、SPA を使用してパフォーマンスをテストできます。</p>	<p>STS は CDB ルートまたは任意の PDB に格納できます。CDB ルートに格納されている場合は、任意の PDB から SQL 文をロードできます。</p> <p>PDB が切断されていると、STS に PDB からの SQL 文が含まれている場合でも、CDB ルートに格納された STS は含まれません。</p> <p>PDB に格納された STS は、PDB が切断されていても含まれます。</p>	<p>現在のコンテナが CDB ルートである共通ユーザーは、CDB ルートに格納された STS データのみを表示できます。</p> <p>現在のコンテナが PDB であるユーザーは、その PDB の STS データのみを表示できます。</p>	<p>Oracle Database SQL チューニング・ガイド</p>
<p>SQL チューニング・アドバイザー</p> <p>SQL チューニング・アドバイザーでは、高負荷 SQL 文として識別された SQL 文が最適化されます。</p>	<p>自動 SQL チューニング・アドバイザー・データは、CDB ルートに格納されます。このデータには、PDB で実行され、アドバイザーによって分析された SQL 文に関する結果が含まれていることがありますが、PDB が切断されている場合、これらの結果は含まれません。</p> <p>現在のコンテナが CDB ルートである共通ユーザーは、任意の PDB からの SQL 文について SQL チューニング・アドバイザーを手動で実行できます。文をチューニングすると、その文を</p>	<p>SQL チューニング・アドバイザーが自動的に実行される場合、その結果は、現在のコンテナが CDB ルートである共通ユーザーにのみ表示されます。現在のコンテナが PDB である場合、これらの結果を表示することはできません。</p> <p>SQL チューニング・アドバイザーが、現在のコンテナが PDB であるユーザーによって手動で実行された場合、結果は現在のコンテナがその PDB であるユーザーにのみ表示されます。</p>	<p>Oracle Database 2 日でパフォーマンス・チューニング・ガイド</p> <p>Oracle Database SQL チューニング・ガイド</p>

管理性機能	データの場所	データの可視性	追加情報
	<p>実行する任意のコンテナでチューニングされます。</p> <p>現在のコンテナが PDB であるユーザーも、PDB からの SQL 文について SQL チューニング・アドバイザを手動で実行できます。SQL チューニング・アドバイザが PDB から手動で実行される場合、その結果は実行元の PDB に格納されます。この場合、文は現在の PDB 用にのみチューニングされ、PDB が切断されていると PDB に関連する結果が含まれません。</p>		

SPAまたはSQLチューニング・アドバイザをPDBからのSQL文について実行するには、共通ユーザーに次の権限が必要です。

- PDBにおける共通権限SET CONTAINERまたはローカル権限SET CONTAINER
- PDBでSQL文を実行するために必要な権限

関連項目:

- [「現在のコンテナについて」](#)
- コンテナ・データ・オブジェクトの概要については、[「ビュー内のCDBおよびコンテナの情報について」](#)を参照してください
- コンテナ・データ・オブジェクトの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください

親トピック: [CDB管理について](#)

CDB内表領域の管理について

表領域とは、記憶域を使用するデータベース・オブジェクト(表や索引など)の論理記憶域コンテナです。

物理レベルでは、表領域のデータは、1つ以上のデータファイルまたは一時ファイルに格納されます。ALTER DATABASE文を使用すると、CDB内の表領域を管理できます。

CDBの表領域については、次のことを考慮する必要があります。

- 永続表領域を関連付けることができるコンテナは厳密に1つです。
- コンテナに表領域を作成する場合、表領域はそのコンテナに関連付けられます。
- ローカルUNDOがCDBに対して無効になっている場合、CDBには、アクティブなUNDO表領域が1つのみ存在するか、またはOracle RAC CDBのインスタンスごとにアクティブなUNDO表領域が1つ存在します。ローカルUNDOがCDBに対して有効になっている場合、CDB内の各コンテナには独自のUNDO表領域があります。

- PDBがオープンされるOracle Real Application Clusters (Oracle RAC)クラスタの各ノードに、ローカルUNDO表領域が必要となります。
- CDBルート、各PDB、各アプリケーション・ルート、各アプリケーションPDBなど、CDB内のコンテナごとに1つのデフォルト一時表領域があります。
- [CDB内永続表領域の管理について](#)
永続表領域を関連付けることができるコンテナは1つのみです。したがって、永続表領域はルートまたは1つのPDBと関連付けることができます。
- [CDB内の一時表領域の管理について](#)
CDB内の各コンテナには、独自のデフォルト一時表領域(または表領域グループ)があります。

親トピック: [CDB管理について](#)

CDB内永続表領域の管理について

永続表領域を関連付けることができるコンテナは1つのみです。したがって、永続表領域はルートまたは1つのPDBと関連付けることができます。

CDB内のコンテナごとに独自のデフォルト永続表領域が存在する必要があり、デフォルト永続表領域をコンテナ間で共有することはできません。表領域を明示的に割り当てられていないユーザーがコンテナに接続された場合、コンテナのデフォルト永続表領域を使用します。

親トピック: [CDB内表領域の管理について](#)

CDB内の一時表領域の管理について

CDB内の各コンテナには、独自のデフォルト一時表領域(または表領域グループ)があります。

追加の一時表領域を個々のコンテナに作成して、コンテナの特定のユーザーをこれらの一時表領域に割り当てることができます。PDBを切断すると、その一時表領域も切断されます。

ユーザーにコンテナ内の一時表領域が明示的に割り当てられていない場合、ユーザーの一時表領域はコンテナのデフォルト一時表領域です。

関連項目:

- 表領域の管理の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- [「CDBからのPDBの切断」](#)
- [「ALTER DATABASEを使用したCDB全体の変更」](#)
- [「ALTER DATABASEを使用したCDBルートの変更」](#)

親トピック: [CDB内表領域の管理について](#)

CDBにおけるデータベース・オブジェクトの管理について

CDBでは、異なるコンテナに異なるデータベース・オブジェクトを含めることができます。

Oracle Databaseには、表、索引、ディレクトリなどのデータベース・オブジェクトが格納されます。スキーマに所有されているデータベース・オブジェクトはスキーマ・オブジェクトと呼ばれ、スキーマによって所有されていないデータベース・オブジェクトは非スキーマ・オブジェクトと呼ばれます。ルートおよびPDBにはスキーマが含まれ、スキーマにはスキーマ・オブジェクトが含まれています。

ルートおよびPDBに、ユーザー、ロール、表領域、ディレクトリ、エディションなどの非スキーマ・オブジェクトを含めることもできます。CDBでは、ルートにはオラクル社が提供するスキーマおよびデータベース・オブジェクトが含まれています。SYSやSYSTEMなどのオラクル社が提供する共通ユーザーは、これらのスキーマおよび共通データベース・オブジェクトを所有します。また、これらのユーザーは、ルートとPDBの両方にローカル・オブジェクトを所有することもできます。

ルートに共通ユーザーを作成して、CDBのコンテナを管理できます。ユーザーが作成した共通ユーザーは、ルートにデータベース・オブジェクトを作成できます。ルートでは、ユーザーが作成した共通ユーザーによって所有されるスキーマには、データベース・トリガーおよびそれらの定義で使用されるオブジェクトのみを含めることをお勧めします。ユーザーが作成した共通ユーザーは、PDBの任意のタイプのローカル・オブジェクトを所有することもできます。

PDBにローカル・ユーザーを作成できます。PDBのローカル・ユーザーは、PDBにスキーマ・オブジェクトおよび非スキーマ・オブジェクトを作成できます。ルートにローカル・ユーザーを作成することはできません。

CDBでの名前解決は、非CDBでの名前解決と似ていますが、ユーザーの現在のコンテナの、ディクショナリコンテキストで名前が解決されることが異なります。

関連項目:

- [「現在のコンテナについて」](#)
- [「CDBの共通ユーザーおよびローカル・ユーザーの概要」](#)
- スキーマ・オブジェクトの管理の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。
- スキーマ・オブジェクトおよび非スキーマ・オブジェクトの詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。
- 共通ユーザーおよびローカル・ユーザーの作成の詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [CDB管理について](#)

PDBのフラッシュ・バックについて

FLASHBACK PLUGGABLE DATABASE文を使用して、PDBを過去の時点またはシステム変更番号(SCN)に戻すことができます。PDBのリストア・ポイントを作成して、CDBまたは他のPDBに影響を与えずに、PDBをリストア・ポイントにフラッシュバックできます。



ノート:

[『Oracle Database バックアップおよびリカバリ・アドバンスド・ユーザーズ・ガイド』](#)

親トピック: [CDB管理について](#)

セキュリティを強化するためのPDBユーザーの制限について

セキュリティを強化するためにPDBユーザーを制限する方法はいくつかあります。

PDBロックダウン・プロファイルは、PDB内のユーザーが使用可能な機能とオプションを制限します。PDB_OS_CREDENTIAL初期化パラメータでは、PDBに一意的オペレーティング・システム・ユーザーを指定して、オペレーティング・システム・アクセスを制限でき

ます。また、PATH_PREFIXおよびCREATE_FILE_DEST句がPDBの作成時に指定された場合は、ファイル・システム・アクセスを制限します。

- [PDBロックダウン・プロファイル](#)

アイデンティティがPDB間で共有される場合は、過剰な権限が存在することがあります。ロックダウン・プロファイルを使用すると、この権限の昇格を防ぐことができます。

- [PDB_OS_CREDENTIAL初期化パラメータ](#)

データベースがextprocエージェントで外部プロセスにアクセスする際に、PDB_OS_CREDENTIAL初期化パラメータはPDBからオペレーティング・システムと対話するときに採用されるオペレーティング・システム・ユーザーのIDを決定します。

- [PATH_PREFIXおよびCREATE_FILE_DEST PDB作成句](#)

CREATE PLUGGABLE DATABASEのPATH_PREFIX句は、PDBと関連付けられているすべてのディレクトリ・オブジェクト・パスが、指定したディレクトリまたはそのサブディレクトリに制限されるようにします。

親トピック: [CDB管理について](#)

PDBロックダウン・プロファイル

アイデンティティがPDB間で共有される場合は、過剰な権限が存在することがあります。ロックダウン・プロファイルを使用すると、この権限の昇格を防ぐことができます。

IDは、次の状況で共有できます。

- オペレーティング・システム・レベルでは、データベースがファイルやプロセスなどのオペレーティング・システム・リソースと対話するとき
- ネットワーク・レベルでは、データベースが他のシステムと通信するとき
- データベースの内部では、PDBが共通オブジェクトにアクセスまたは作成するか、データベース・リンクなどの機能を使用してコンテナ境界を越えて通信するとき

セキュリティを高めるために、CDB管理者はPDBロックダウン・プロファイルを使用して特定のPDBのユーザーを制限できます。PDBロックダウン・プロファイルでは、ALTER SYSTEM文などの指定したSQL文をユーザーが実行できないようにしたり、UTL_SMTPなど、ネットワークにアクセスできるパッケージへのアクセスを無効にしたりできます。PDBロックダウン・プロファイルでは、共通ユーザー、共通オブジェクトおよび管理ツール(Oracle XML DB、管理機能(例: カーソル共有)、データベース・オプション(例: Oracle Databaseアドバンスド・キューイング)など)へのアクセスも制限できます。PDBロックダウン・プロファイルでは、XDB_PROTocols機能でPDBによるXDBプロトコル(FTP、HTTP、HTTPS)の使用を禁止できます。

CDBルートまたはアプリケーション・ルートにログインするときに、次のオプション句をサポートしているCREATE LOCKDOWN PROFILE文を発行して、ロックダウン・プロファイルを作成します。

- FROM static_base_profileは、既存のプロファイルの値を使用して、新しいロックダウン・プロファイルを作成します。既存のプロファイルへの以降の変更は、新しいプロファイルには影響しません。
- INCLUDING dynamic_base_profileは、既存のプロファイルの値を使用して新しいロックダウン・プロファイルを作成しますが、この新しいロックダウン・プロファイルは、基本プロファイルを構成するDISABLE STATEMENTルール、および基本プロファイルへの以降の変更を継承するという点が異なります。

文を発行するユーザーは、現在のコンテナでCREATE LOCKDOWN PROFILEシステム権限を持っている必要があります。制約を追加および削除するには、ALTER LOCKDOWN PROFILE文を使用します。ユーザーはCDBルートまたはアプリケーション・ルートでALTER文を発行する必要があり、現在のコンテナでALTER LOCKDOWN PROFILEシステム権限を持っている必要があります。

ロックダウン・プロファイルは、PDB_LOCKDOWN初期化パラメータを使用して指定します。このパラメータによって、PDBロックダウン・プロファイルが指定のPDBに適用されるかどうかが決まります。このパラメータは、次のレベルで設定できます。

- PDB

プロファイルは、設定されるPDBにのみ適用されます。

- アプリケーション・コンテナ

プロファイルは、アプリケーション・コンテナ内のすべてのアプリケーションPDBに適用されます。値を変更できるのは、アプリケーション共通のSYSDBA権限または共通のALTER SYSTEM権限を持つアプリケーション共通ユーザー、あるいは共通のSYSDBA権限または共通のALTER SYSTEM権限を持つCDB共通ユーザーのみです。

- CDB

プロファイルはすべてのPDBに適用されます。共通のSYSDBA権限または共通のALTER SYSTEM権限を持つ共通ユーザーは、特定のPDBに対するCDB全体の設定をオーバーライドできます。

PDBのPDB_LOCKDOWNパラメータが、このPDBのコンテナ(CDBまたはアプリケーション・コンテナ)とは異なるロックダウン・プロファイルの名前に設定されている場合は、一連のルールによって制限の間の相互作用が制御されます。

関連項目:

- ロックダウン・プロファイルの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください
- CREATE LOCKDOWN PROFILE文の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。
- PDB_LOCKDOWN初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください

親トピック: [セキュリティを強化するためのPDBユーザーの制限について](#)

PDB_OS_CREDENTIAL初期化パラメータ

データベースがextprocエージェントで外部プロシージャにアクセスする際に、PDB_OS_CREDENTIAL初期化パラメータはPDBからオペレーティング・システムと対話するときに採用されるオペレーティング・システム・ユーザーのIDを決定します。

名前がPDB_OS_CREDENTIAL初期化パラメータの値として指定された資格証明で記述されるOSユーザーを使用すると、オペレーティング・システムの対話が強力な権限を持たないユーザーとして実行されます。このようにして、あるPDBに属するデータを別のPDBに接続しているユーザーからアクセスできないように保護する機能が提供されます。資格証明は、DBMS_CREDENTIALパッケージ内のCREATE_CREDENTIALプロシージャを使用して作成されるオブジェクトです。

通常、Oracle OSユーザーは高い権限を持つユーザーです。オペレーティング・システムの対話はこのアカウントを使用することはお薦めしません。また、異なるPDBからのオペレーティング・システムの対話に同じOSユーザーを使用すると、特定のPDBに属するデータが危険にさらされる可能性があります。

関連項目:

- 外部プロシージャ用のプロセスの管理方法の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- PDB_OS_CREDENTIAL初期化パラメータの詳細は、[Oracle Databaseリファレンス](#)を参照してください
- [Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)

親トピック: [セキュリティを強化するためのPDBユーザーの制限について](#)

PATH_PREFIXおよびCREATE_FILE_DEST PDB作成句

CREATE PLUGGABLE DATABASEのPATH_PREFIX句は、PDBと関連付けられているすべてのディレクトリ・オブジェクト・パスが、指定したディレクトリまたはそのサブディレクトリに制限されるようにします。

また、PATH_PREFIXは、PDBと関連付けられている次のファイルが、指定したディレクトリに制限されるようにします。

- PDBのOracle XMLリポジトリ
- CREATE PFILE文で作成されたファイル
- Oracleウォレットのエクスポート・ディレクトリ
- CREATE LIBRARY文で作成されたライブラリ・オブジェクト

ノート:



ライブラリでは、ディレクトリ・オブジェクトを使用する必要があります。PDBで事前定義済みのPATH_PREFIXが使用されている場合、ディレクトリ・オブジェクトを使用しないライブラリ・オブジェクトを使用しようとすると、ORA-65394エラーが発生します。ライブラリ・オブジェクトは無効になりませんが、ライブラリ・オブジェクトを使用可能にするには、ディレクトリ・オブジェクトを使用して再作成する必要があります。

CREATE PLUGGABLE DATABASE文のCREATE_FILE_DEST句は、PDBに属するすべてのデータベース・ファイルが、指定したディレクトリまたはそのサブディレクトリに制限されるようにします。この句を使用すると、PDBでOracle Managed Filesが有効になり、PDBファイルのデフォルトのファイル・システム・ディレクトリまたはOracle ASMディスク・グループを指定できます。

関連項目:

- [PDBファイルの場所に対する制限](#)
- [CREATE_FILE_DEST句](#)

親トピック: [セキュリティを強化するためのPDBユーザーの制限について](#)

Oracle RACを含むOracle Multitenantの概要

Oracle MultitenantはOracle Database 12cのオプションで、統合、プロビジョニング、アップグレードなどを簡略化します。これは、マルチテナント・コンテナ・データベース(CDB)に複数のプラグブル・データベース(PDB)を保持することができるアーキテクチャに基づいています。アプリケーション層を変更することなく、既存のデータベースをPDBとして採用できます。このアーキテクチャでは、1つのシステム上で様々なビジネスで重要なアプリケーションを統合するときに必要なローカルの高可用性を、Oracle RACが提供します。

Oracle RACでPDBを使用する場合、マルチテナントCDBはOracle RACに基づきます。各PDBをOracle RAC CDBの各インスタンスまたはインスタンスのサブセットで使用可能にすることができます。いずれの場合も、PDBへのアクセスおよび管理は、動的データベース・サービスを使用して調整されます。これは接続にOracle Net Servicesを使用する1つのインスタンス Oracleデータベース内なので、アプリケーションにより使用されて各PDBに接続することもあります。

同じOracle RACデータベースまたはデータベース・インスタンスを共有している別のPDBを妨害する可能性のある、特定のPDBでの特定の操作が実行されないようにするために、PDBを分離できます。PDBの分離により、Oracle Multitenantを使

用した高度な統合が可能になります。

Oracle RACデータベースをCDBとして作成し、そのCDBに1つ以上のPDBを接続する場合、Oracle RACのCDBのどのインスタンスでもPDBはデフォルトで自動起動されません。PDBに(データベース名と同じ名前のデフォルトのデータベース・サービス以外の)最初の動的なデータベース・サービスが割り当てられると、PDBはサービスが実行されるインスタンスで有効になります。

Oracle RACの1つ以上のインスタンスでPDBが有効かどうかにかかわらず、CDBは通常、PDBで実行されるサービスで管理されます。インスタンス上でPDBを手動で起動することによって、Oracle RAC CDBの各インスタンス上でPDBアクセスを手動で有効化できます。

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)

親トピック: [CDB管理について](#)

CDB内のコンテナへのアクセス

SQL*Plus CONNECTコマンドを使用してコンテナに接続できます。または、ALTER SESSION SET CONTAINER SQL文を使用してコンテナに切り替えることができます。

- [CDB内のコンテナへのアクセスについて](#)
SQL*Plusを使用して、CDB内のルートまたはPDBにアクセスできます。
- [CDB内のコンテナへのアクセス](#)
SQL*PlusのCONNECTまたはALTER SESSIONコマンドを発行して、CDB内のコンテナにアクセスします。

親トピック: [CDBの管理](#)

CDB内のコンテナへのアクセスについて

SQL*Plusを使用してCDB内のルートまたはPDBにアクセスできます。

ノート:



この項では、SQL*Plus で非 CDB に接続する方法を理解していることを想定しています。

- [CDBのサービス](#)
クライアントからは、データベース・サービスを使用してルートまたはPDBにアクセスします。
- [CDBでのセッション制限](#)
SESSIONS初期化パラメータの設定によって、PDBに接続されたセッションなどのCDBで使用できるセッションの合計数が制限されます。
- [マルチテナント環境のユーザー名](#)
各PDB内のユーザー名は、そのPDB内の他のユーザー名およびロールと比較して一意である必要があります。
- [管理ユーザーのパスワード・ファイルに対するマルチテナント・オプションの影響](#)
マルチテナント環境では、ローカル管理ユーザーと共通管理ユーザーのパスワード情報はそれぞれ異なる場所に保存されます。

関連項目:

- データベースに対するコマンドおよびSQLの発行の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください
- Oracle Net Servicesの構成の詳細は、[Oracle Database Net Services管理者ガイド](#)を参照してください。

親トピック: [CDB内のコンテナへのアクセス](#)

CDBのサービス

クライアントからは、データベース・サービスを使用してルートまたはPDBにアクセスします。

データベース・サービスには、オプションのPDBプロパティがあります。PDBが作成されると、PDBの新しいデフォルト・サービスが自動的に作成されます。サービスには、PDBと同じ名前が付いています。サービス名を使用すると、簡易接続構文またはtnsnames.oraファイル内のネット・サービス名を使用してPDBにアクセスできます。このサービスにアクセスするクライアントには、Oracle Net Serviceが正しく構成されている必要があります。

ユーザーがnull以外のPDBプロパティを含むサービスを使用して接続する場合、ユーザー名は指定したPDBのコンテキストで解決されます。ユーザーがサービスを指定しないか、またはnullのPDBプロパティを含むサービス名を使用して接続する場合、ユーザー名はルートのコンテキストで解決されます。サービスのPDBプロパティを表示するには、CDB_SERVICESデータ・ディクショナリビューを問い合わせるか、またはconfig serviceコマンドをSRVCTLユーティリティで実行します。

ノート:



同じコンピュータ・システム上の 2 つ以上の CDB で同じリスナーが使用され、これらの CDB 内で 2 つ以上の PDB が同じサービス名を持つ場合、このサービス名を指定する接続は、そのサービス名の PDB の 1 つにランダムに接続します。不正に接続されることを回避するには、コンピュータ・システム上の PDB のすべてのサービス名が一意であることを確認するか、コンピュータ・システム上の各 CDB に別のリスナーを構成します。

関連項目:

- [「CDBのサービスの概要」](#)
- [「PDBのサービスの管理」](#)
- [「例19-9」](#)

親トピック: [CDB内のコンテナへのアクセスについて](#)

CDBでのセッション制限

SESSIONS初期化パラメータの設定によって、PDBに接続されたセッションなどのCDBで使用できるセッションの合計数が制限されます。

CDBの制限に達すると、ユーザーはPDBに接続できなくなります。1つのPDBで過剰なセッションが使用されないようにするには、PDBでSESSIONS初期化パラメータを設定して、PDBで使用できるセッションの数を制限できます。

関連項目:

親トピック: [CDB内のコンテナへのアクセスについて](#)

マルチテナント環境のユーザー名

各PDB内のユーザー名は、そのPDB内の他のユーザー名およびロールと比較して一意である必要があります。

以下の制限に注意してください。

- 共通ユーザー名の場合、ユーザー作成の共通ユーザーの名前は共通のユーザー接頭辞で始める必要があります。デフォルトでは、CDB共通ユーザーの場合、この接頭辞はC##です。アプリケーション共通ユーザーの場合、この接頭辞は空の文字列です。これは、CDB共通ユーザー用に予約された接頭辞を使用しないかぎり、アプリケーション共通ユーザーに割り当てる名前には制限がないことを意味します。たとえば、CDB共通ユーザーにc##hr_admin、アプリケーション共通ユーザーにchr_adminという名前を付けることができます。

CDB\$ROOTのCOMMON_USER_PREFIXパラメータは、共通のユーザー接頭辞を定義します。この設定は変更できますが、十分に注意して行うようにしてください。

- ローカル・ユーザー名の場合、名前をC## (またはc##) で始めることはできません。
- ユーザーとロールに同じ名前を付けることはできません。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [CDB内のコンテナへのアクセスについて](#)

管理ユーザーのパスワード・ファイルに対するマルチテナント・オプションの影響

マルチテナント環境では、ローカル管理ユーザーと共通管理ユーザーのパスワード情報はそれぞれ異なる場所に保存されます。

- CDB管理ユーザーの場合: CDBルートで管理権限が付与されたCDB共通管理ユーザーのパスワード情報(パスワードのハッシュ)は、パスワード・ファイルに格納されます。
- CDBルート外で管理権限が付与された、CDB内のすべてのユーザーの場合: これらのユーザーのパスワード・ハッシュ情報に関する情報を確認するには、\$PWFILERS動的ビューを問い合わせます。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [CDB内のコンテナへのアクセスについて](#)

CDB内のコンテナへのアクセス

SQL*PlusのCONNECTまたはALTER SESSIONコマンドを発行して、CDB内のコンテナにアクセスします。

- [SQL*Plus CONNECTコマンドを使用したコンテナへの接続](#)
SQL*Plus CONNECTコマンドを使用してルートまたはPDBに接続できます。
- [ALTER SESSION文を使用したコンテナへの切替え](#)
共通ユーザーとしてコンテナに接続している場合、ALTER SESSION文を使用して、別のコンテナおよびアプリケーション・サービスに切り替えることができます。

親トピック: [CDB内のコンテナへのアクセス](#)

SQL*Plus CONNECTコマンドを使用したコンテナへの接続

SQL*Plus CONNECTコマンドを使用してルートまたはPDBに接続できます。

- [SQL*Plus CONNECTコマンドを使用したルートへの接続](#)
非CDBに接続する場合と同じ方法でルートに接続できます。
- [SQL*Plus CONNECTコマンドを使用したPDBへの接続](#)
SQL*Plus CONNECTコマンドを使用してPDBに接続するには、簡易接続またはネット・サービス名を使用できます。

親トピック: [CDB内のコンテナへのアクセス](#)

SQL*Plus CONNECTコマンドを使用したルートへの接続

非CDBに接続する場合と同じ方法でルートに接続できます。

具体的には、SQL*Plus CONNECTコマンドでルートに接続するために、次の手法を使用できます。

- ローカル接続
- オペレーティング・システム認証を使用したローカル接続
- 簡易接続を使用したデータベース接続
- サービス名を使用したデータベース接続
- 外部認証を使用したリモート・データベース接続

ユーザーがルートに接続するには、次の前提条件を満たしている必要があります。

- ユーザーは共通ユーザーである必要があります。
- ユーザーは、ルートでのCREATE SESSION権限を付与されている必要があります。

SQL*Plus CONNECTコマンドを使用してルートに接続するには:

1. SQL*Plusを起動できるよう、環境を構成します。
2. /nolog引数を使用してSQL*Plusを起動します。

```
sqlplus /nolog
```

3. 次の例のように、SQL*PlusのCONNECTコマンドを発行し、ルートとして接続します。

例13-1 ローカル接続を使用したルートへの接続

この例では、ユーザーSYSTEMとしてローカルCDBのルートに接続します。SQL*Plusによって、ユーザーSYSTEMのパスワードの入力が求められます。

```
connect system
```

例13-2 オペレーティング・システム認証を使用したrootへの接続

この例では、オペレーティング・システム認証を使用してSYSDBA管理権限でルートにローカルに接続します。

```
connect / as sysdba
```

例13-3 ネット・サービス名を使用したrootへの接続

CDBのルートのネット・サービス名が含まれるようにクライアントが構成されていると想定します。たとえば、ネット・サービス名をtnsnames.oraファイルのエントリに含めることができます。

この例では、共通ユーザーc##dbaとして、ネット・サービス名mycdbによって指定されたデータベース・サービスに接続します。

SQL*Plusによって、ユーザーc##dbaのパスワードの入力が求められます。

```
connect c##dba@mycdb
```

関連項目:

データベースに対するコマンドおよびSQLの発行の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください

親トピック: [SQL*Plus CONNECTコマンドを使用したコンテナへの接続](#)

SQL*Plus CONNECTコマンドを使用したPDBへの接続

SQL*Plus CONNECTコマンドを使用してPDBに接続するには、簡易接続またはネット・サービス名を使用できます。

PDBに接続するには、ユーザーは次のいずれかである必要があります。

- CREATE SESSION権限が共通に付与されているか、PDBでローカルに付与されている共通ユーザー
- CREATE SESSION権限を持つPDBで定義されたローカル・ユーザー

SYSDBA、SYSOPER、SYSBACKUPまたはSYSDG権限を持つユーザーのみが、マウント・モードのPDBに接続できます。PDBのオープン・モードを変更するには、『[PDBのオープン・モードの変更](#)』を参照してください。

SQL*Plus CONNECTコマンドを使用してPDBに接続するには:

1. SQL*Plusを起動できるよう、環境を構成します。
2. /NOLOG引数を使用してSQL*Plusを起動します。

```
sqlplus /nolog
```

3. 簡易接続またはネット・サービス名を使用してSQL*Plus CONNECTコマンドを発行し、PDBに接続します。

例13-4 PDBへの接続

PDBごとに各PDB名と一致するネット・サービス名を持つように、クライアントが構成されていると想定します。たとえば、ネット・サービス名をtnsnames.oraファイルのエントリに含めることができます。

次のコマンドにより、salespdb PDB内のshローカル・ユーザーに接続します。

```
CONNECT sh@salespdb
```

次のコマンドにより、salespdb PDB内のSYSTEM共通ユーザーに接続します。

```
CONNECT system@salespdb
```

関連項目:

SQL*Plus CONNECTコマンドの発行の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください

親トピック: [SQL*Plus CONNECTコマンドを使用したコンテナへの接続](#)

ALTER SESSION文を使用したコンテナへの切替え

共通ユーザーとしてコンテナに接続している場合、ALTER SESSION文を使用して、別のコンテナおよびアプリケーション・サービスに切り替えることができます。

次の文を使用して、別のコンテナおよびアプリケーション・サービスに切り替えることができます。

```
ALTER SESSION SET CONTAINER = container_name [SERVICE = service_name]
```

container_nameについては、次のいずれかを指定します。

- CDBルートに切り替えるにはCDB\$ROOT
- PDBシードに切り替えるにはPDB\$SEED
- PDBに切り替えるにはPDB名

現在のコンテナがルートである場合、DBA_PDBSビューを問い合わせると、CDB内のPDBの名前を表示できます。

service_nameには、PDBで実行されているサービスを指定します。CDBルートを除くCDBのコンテナで実行されているサービスをリストするには、CDBルートが現在のコンテナであるときに次の問合せを発行します。

```
COL NAME FORMAT A30
COL CON_NAME FORMAT A20

SELECT NAME, CON_NAME, CON_ID
FROM V$ACTIVE_SERVICES
WHERE UPPER(NAME) != CON_NAME
AND CON_ID !=1
ORDER BY CON_ID;
```

デフォルトでは、コンテナに切り替えると、セッションはコンテナのデフォルト・サービスを使用します。ただし、PDBのデフォルト・サービスでは、すべてのサービス属性および機能(サービス・メトリック、高速アプリケーション通知(FAN)、ロード・バランシング、リソース・マネージャ、トランザクション・ガード、アプリケーション・コンティニューイティなど)がサポートされているわけではありません。ベスト・プラクティスは、SERVICE = service_name (service_nameはサービスの名前)を指定することによって、コンテナにデフォルト以外のサービスを使用することです。

この新しい機能によって、接続プールでサービスを切り替えたり、プールの接続を使用した接続でPDBが必要な場合に切り替えることができます。Oracle Database 12cリリース2 (12.2.0.1)以降では、接続プールはUCP (Universal Connection Pool)を使用して複数のデータベース・サービスをサポートします。スタンドアロンで使用することもできます。

サービスを切り替える場合は、データベース・サービスの識別、優先順位、測定および高可用性が維持されたままで、アプリケーションをCDBに統合できます。デフォルト以外のサービスに切り替えると、次のような利点があります。

- サービス属性および機能が維持されます。
- これらのテナントに使用される接続が多すぎる過剰な接続プールがなくなります。
- アプリケーションが、接続プールを過剰に消費することなく、ワークロード制御のためにデータベース・サービスをさらに使用できます。顧客は、データベース接続のサイズが大きくなりすぎることなく、サービスを使用したワークロードを識別して優先できます。

ALTER SESSION SET CONTAINER文を使用する場合は、次のことを考慮する必要があります。

- 文が正常に完了した後、セッションの現行スキーマは、指定されたコンテナの共通ユーザーが所有するスキーマに設定されます。
- 文が正常に完了した後、セキュリティ・コンテキストは、指定されたコンテナの共通ユーザーが所有するスキーマのセキュリティ・コンテキストにリセットされます。
- 文が正常に完了した後、指定されたコンテナのログイン・トリガーは起動されません。

トリガーが必要な場合は、ALTER SESSION SET CONTAINER文が実行される前または後に起動するように、BEFOREまたはAFTER SET CONTAINERトリガーをPDBで定義できます。

- この文が正常に完了し、SERVICE句によってPDBにデフォルト以外のサービスが指定されている場合、セッションは設定されている属性(新しいサービスのメトリック、FAN、TAF、アプリケーション・コンティニューイティ、トランザクション・ガード、drain_timeoutおよびstop_optionを含む)を使用して新しいサービスを使用します。
- パッケージの状態は複数のコンテナにわたって共有されません。
- PDBをクローズするとき、PDBに切り替えられたセッションおよびPDBに直接接続されたセッションは同様に処理されません。
- トランザクションを複数のコンテナにわたって実行することはできません。トランザクションを開始し、ALTER SESSION SET CONTAINERを使用して別のコンテナに切り替えると、そのトランザクションを開始したコンテナに再び切り替えるまで、DML、DDL、COMMITまたはROLLBACK文を発行できなくなります。
- カーソルをオープンし、ALTER SESSION SET CONTAINERを使用して別のコンテナに切り替えると、カーソルをオープンしたコンテナに再び切り替えるまで、このカーソルからデータをフェッチできなくなります。
- 接続プーリングおよび拡張CDB管理に、SERVICE句を指定したALTER SESSION SET CONTAINER文を使用できます。
たとえば、マルチテナント・アプリケーションのPDBで、この文を接続プーリングに使用できます。マルチテナント・アプリケーションでは、サーバー上のソフトウェアの単一インスタンスを使用して、複数の顧客(テナント)に対処します。非CDBでは、通常、アプリケーションで使用されるすべての表にテナントを識別する余分な列を追加することでマルチテナントがサポートされていて、テナントは接続プールから接続をチェック・アウトします。PDBが含まれるCDBでは、テナントごとに独自のPDBを設定でき、接続プーリング構成でALTER SESSION SET CONTAINER文を使用できます。
- 複数のアプリケーションを処理する接続プールを使用している場合、アプリケーションは別のサービスのデータ・ソースを使用できます。SERVICE句を指定してALTER SESSION SET CONTAINER文を使用すると、接続プールで多数のアプリケーションに同じ接続を使用して、サービスを共有できます。

ALTER SESSION SET CONTAINER文を使用するには、次の前提条件を満たしている必要があります。

- 現在のユーザーは共通ユーザーである必要があります。初期接続は、SQL*Plus CONNECTコマンドを使用して確立する必要があります。
- Oracle Databaseで提供されていない共通ユーザーとして、セッションを変更してPDBに切り替える場合、現在のユーザーには、SET CONTAINER権限が共通に付与されているか、またはこの権限がPDBでローカルに付与されている必要があります。

ノート:



ALTER SESSION SET CONTAINER 文を使用して現在のコンテナを切り替える場合、このような前提条件は強制されず、満たされていない場合、エラー・メッセージは返されません。

SERVICE句を指定してALTER SESSION SET CONTAINER文を発行する前に、次の前提条件を満たしている必要があります。

- 切り替える先のサービスがアクティブである必要があります。実行されていないサービスに切り替えることはできません。
- サービス間で切り替える場合は、切り替える元のサービスと切り替える先のサービスのサービス属性が一致している必要があります。たとえば、切り替える元と切り替える先のサービスの両方が、TAFを持っているか、アプリケーション・コンティニューイティを使用しているか、drain_timeout設定がある必要があります。

ALTER SESSION文を使用してコンテナに切り替えるには:

1. SQL*Plusで、必要な権限を持つ共通ユーザーとしてコンテナに接続します。

2. 切替え先のコンテナの現在のオープン・モードを確認します。

ルートまたはPDBの現在のオープン・モードを確認するには、現在のコンテナがルート有的时候に、V\$CONTAINERSビューでOPEN_MODE列を問い合わせます。

ルートのオープン・モードを変更する必要がある場合は、『[Oracle Database管理者ガイド](#)』のデータベースの可用性の変更に関する項の手順に従って、オープン・モードを変更します。

PDBのオープン・モードを変更する必要がある場合は、『[PDBのオープン・モードの変更](#)』の手順に従って、オープン・モードを変更します。

ルートのオープン・モードにより、PDBのオープン・モードが制限されます。たとえば、PDBをオープンするには、ルートがオープンしている必要があります。したがって、PDBのオープン・モードを変更する前に、ルートのオープン・モードを変更することが必要な場合があります。

3. 特定のサービスに切り替える場合は、そのサービスが実行されていることを確認します。

サービスのアクティブ・ステータスを確認するには、現在のコンテナがCDBルートであるときにV\$ACTIVE_SERVICESビューを問い合わせます。

サービスが実行されていない場合は、SRVCTLユーティリティまたはDBMS_SERVICEパッケージを使用して、サービスを開始します。

4. ALTER SESSION SET CONTAINER文を実行して、切替え先のコンテナを指定します。

SERVICE句を含めて、特定のアプリケーション・サービスに切り替えます。

次の例では、ALTER SESSIONを使用して様々なコンテナに切り替えます。

例13-5 PDB salespdbへの切替えとsalesrepサービスの使用

```
ALTER SESSION SET CONTAINER = salespdb SERVICE = salesrep;
```

例13-6 PDB salespdbへの切替えとデフォルト・サービスの使用

```
ALTER SESSION SET CONTAINER = salespdb;
```

例13-7 CDBルートへの切替え

```
ALTER SESSION SET CONTAINER = CDB$ROOT;
```

例13-8 PDBシードへの切替え

```
ALTER SESSION SET CONTAINER = PDB$SEED;
```

例13-9 CDBルートのダミー・サービスを使用したサービスの切替え

コンテナおよびサービスを切り替える接続プールを設計するには、CDBルートにダミー・サービスを作成し、必要なすべての属性(たとえば、drain_timeout、TAFまたはアプリケーション・コンティニューイティ)をこのダミー・サービスに設定することが1つの方法です。サービス属性は、CDBルートおよびPDB間で一致している必要があります。この方法を使用するには、次のステップを実行します。

1. 最初に接続プールを作成するとき、および新しい接続を作成するとき、ダミー・サービスに接続します。
2. サービスが各PDBに追加されると、これらの実際のサービスに同じ属性が設定されます。
3. アプリケーションが接続を必要とする場合は、次のいずれかのアクションを実行します。

- ダミー・サービスへの新しい接続を作成し、そのPDBおよびサービスに切替えます。
- プールの空き接続を使用して、PDBおよびサービスに切り替えます。

PDB間で切り替える場合、CDBに戻る必要はありません。

PDB間で切り替える場合、CDBに戻る必要はありません。

関連項目:

データベース常駐接続プーリングの詳細は、[Oracle Database管理者ガイド](#)を参照してください

親トピック: [CDB内のコンテナへのアクセス](#)

システム・レベルでのCDBの変更

初期化パラメータはCDBレベルで設定できます。場合によっては、これらのパラメータをPDBレベルで上書きできます。

- [CDBのシステム・レベルの変更について](#)
ALTER SYSTEM SET文は、1つ以上のコンテナの初期化パラメータを動的に設定します。
- [ALTER SYSTEMを使用したCDBの変更](#)
システム・レベルでCDBを変更するには、ALTER SYSTEM文を使用します。

親トピック: [CDBの管理](#)

CDBのシステム・レベルの変更について

ALTER SYSTEM SET文は、1つ以上のコンテナの初期化パラメータを動的に設定します。

CDBでは初期化パラメータの継承モデルが使用されます。このモデルでは、PDBがルートから初期化パラメータの値を継承します。この場合、継承は、ルートの特定のパラメータ値が特定のPDBに適用されることを意味します。

PDBは、一部のパラメータのルート設定を上書きできます。このような場合、PDBには初期化パラメータごとに継承プロパティ (trueまたはfalse)があります。あるパラメータでPDBがルートの値を継承する場合、そのパラメータの継承プロパティはtrueです。それ以外の場合、プロパティはfalseです。

一部のパラメータについては、継承プロパティがtrueである必要があります。その他のパラメータでは、現在のコンテナがPDBである場合、ALTER SYSTEM SET文を実行して継承プロパティを変更できます。初期化パラメータのV\$SYSTEM_PARAMETER.ISPDB_MODIFIABLEがTRUEの場合は、そのパラメータの継承プロパティをfalseに設定できます。

現在のコンテナがルートである場合、ALTER SYSTEM SET文のCONTAINER句で設定されるパラメータ値を継承するPDBが制御されます。CONTAINER句の構文は次のとおりです。

```
CONTAINER = { CURRENT | ALL }
```

次の設定を使用できます。

- CURRENT
パラメータ設定は現在のコンテナのみに適用されます。これは、CONTAINERのデフォルト設定です。現在のコンテナがルートである場合は、ルートと、そのパラメータについてtrueの継承プロパティを持つすべてのPDBにそのパラメータ設定が適用されます。
- ALL

パラメータ設定は、ルートおよびすべてのPDBを含むCDB内のすべてのコンテナに適用されます。ALLを指定すると、すべてのPDB内のパラメータの継承プロパティはtrueに設定されます。

関連項目:

CONTAINER句およびそれに適用するルールの詳細は、[「現在のコンテナについて」](#)を参照してください

親トピック: [システム・レベルでのCDBの変更](#)

ALTER SYSTEMを使用したCDBの変更

システム・レベルでCDBを変更するには、ALTER SYSTEM文を使用します。

前提条件

現在のユーザーには、ALTER SYSTEM権限が共通して付与されている必要があります。

CDBのルートでALTER SYSTEM SETを使用するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. ALTER SYSTEM SET文を実行します。

ノート:



PDBのパラメータの継承プロパティを false から true に変更するには、現在のコンテナが PDB であるときに ALTER SYSTEM RESET 文を実行して、パラメータをリセットします。次のサンプル文は、OPEN_CURSORS パラメータをリセットします。

```
ALTER SYSTEM RESET OPEN_CURSORS SCOPE = SPFILE;
```

例13-10 すべてのコンテナへの初期化パラメータの設定

このALTER SYSTEM SET文では、すべてのコンテナのOPEN_CURSORS初期化パラメータを200に設定し、各PDBの継承プロパティをTRUEに設定します。

```
ALTER SYSTEM SET OPEN_CURSORS = 200 CONTAINER = ALL;
```

例13-11 ルートの初期化パラメータの設定

このALTER SYSTEM SET文では、ルートおよびPDBのOPEN_CURSORS初期化パラメータを200に設定します(このとき、パラメータの継承プロパティはtrueです)。

```
ALTER SYSTEM SET OPEN_CURSORS = 200 CONTAINER = CURRENT;
```

関連項目:

- [「システム・レベルでのPDBの変更」](#)
- ALTER SYSTEM SET文の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください

CDBルートへの接続時のコンテナの変更

ALTER DATABASE文を使用してCDB全体またはルートを変更できます。

- [CDBルートへの接続時のコンテナの変更について](#)
ALTER DATABASE文はCDBを変更します。CDBルートに接続するとき、ALTER PLUGGABLE DATABASE文では、1つ以上のPDBのオープン・モードを変更できます。
- [ALTER DATABASEを使用したCDB全体の変更](#)
ALTER DATABASE文を使用して、ルートおよびPDBすべてを含むCDB全体を変更できます。ほとんどのALTER DATABASE文は、CDB全体を変更します。
- [ALTER DATABASEを使用したCDBのUNDOモードの設定](#)
ローカルUNDOが有効になっている場合、コンテナがオープンされる各インスタンスで、各コンテナに独自のUNDO表領域があります。ローカルUNDOが無効になっている場合は、CDB全体に対して1つのUNDO表領域があります。
- [ALTER DATABASEを使用したCDBルートの変更](#)
CDBのルートのみを変更するには、ALTER DATABASE文を使用します。

親トピック: [CDBの管理](#)

CDBルートへの接続時のコンテナの変更について

ALTER DATABASE文はCDBを変更します。CDBルートに接続するとき、ALTER PLUGGABLE DATABASE文では、1つ以上のPDBのオープン・モードを変更できます。

ALTER DATABASEおよびALTER PLUGGABLE DATABASEの動作は、文を使用するときに接続されているコンテナによって異なります。

- 共通ユーザーとしてCDBルートに接続されている場合
この場合、ALTER DATABASE文は非CDBの場合と同様に機能します。RENAME GLOBAL_NAME句を含むALTER DATABASE文を使用すると、CDBのドメインが変更され、CDBのドメインにデフォルト設定された各PDBのドメインに影響を及ぼします。pdb_change_state句を含むALTER PLUGGABLE DATABASE文は、1つ以上のPDBのオープン・モードを変更します。
- PDBに接続されている場合
この場合、ALTER DATABASEおよびALTER PLUGGABLE DATABASE文は現在のPDBのみを変更します。

次の表に、ALTER DATABASEおよびALTER PLUGGABLE DATABASE文の句によって変更されるコンテナを示します。

表13-3 CDBのコンテナを変更する文

CDB全体の変更	ルートのみの変更	1つ以上のPDBの変更
現在のコンテナがルートである共通ユーザーとして接続した場合、次の句を含むALTER DATABASE文を使用すると、CDB全体が変更されます。	現在のコンテナがルートである共通ユーザーとして接続した場合、次の句を含むALTER DATABASE文を使用すると、ルートのみが変更されます。	現在のコンテナがルートである共通ユーザーとして接続した場合、次の句を含むALTER PLUGGABLE DATABASE文を使用すると、1つ以上のPDBのオープン・モードを変更できます。
● startup_clauses	● database_file_clauses	● pdb_change_state

CDB全体の変更	ルートのみの変更	1つ以上のPDBの変更
<ul style="list-style-type: none"> ● recovery_clauses ● logfile_clauses ● controlfile_clauses ● standby_database_clause ● instance_clauses ● security_clause 	<ul style="list-style-type: none"> ● DEFAULT EDITION 句 ● DEFAULT TABLESPACE 句 ● DEFAULT TEMPORARY TABLESPACE 句 <p>次の句を含む ALTER DATABASE 文を使用すると、ルートが変更され、PDB にデフォルト値が設定されます。</p>	<p>現在のコンテナが PDB である場合、この句を含む ALTER PLUGGABLE DATABASE 文を使用すると、現在の PDB のオープン・モードを変更できます。</p> <p>現在のコンテナがルートである共通ユーザーとして接続した場合、次の句を含む ALTER PLUGGABLE DATABASE 文を使用すると、CDB の再起動時に PDB のオープン・モードを保持することも破棄することもできます。</p>
<ul style="list-style-type: none"> ● RENAME GLOBAL_NAME 句 ● ENABLE BLOCK CHANGE TRACKING 句 ● DISABLE BLOCK CHANGE TRACKING 句 	<ul style="list-style-type: none"> ● flashback_mode_clause ● SET DEFAULT {BIGFILE SMALLFILE} TABLESPACE 句 ● set_time_zone_clause <p>これらの句を使用すると、特定の PDB にデフォルト以外の値を設定できます。</p>	<ul style="list-style-type: none"> ● pdb_save_or_discard_statement

関連項目:

- [「現在のコンテナについて」](#)
- [「データベース・レベルでのPDBの変更」](#)
- [Oracle Database SQL言語リファレンス](#)

親トピック: [CDBルートへの接続時のコンテナの変更](#)

ALTER DATABASEを使用したCDB全体の変更

ALTER DATABASE文を使用して、ルートおよびPDBすべてを含むCDB全体を変更できます。ほとんどのALTER DATABASE文は、CDB全体を変更します。

ルートまたは個々のPDBではなくCDB全体を変更する文のリストは、[「CDBルートへの接続時のコンテナの変更について」](#)の「CDB全体の変更」列を参照してください。

前提条件

CDB全体を変更するには、次の前提条件を満たしている必要があります。

- 現在のユーザーは、ALTER DATABASE権限を持つ共通ユーザーである必要があります。
- recovery_clauseを含むALTER DATABASE文を実行するには、現在のユーザーにSYSDBA管理権限が共通に付与

されている必要があります。この場合、接続時にAS SYSDBAを使用して、この権限を行使する必要があります。

CDB全体を変更するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. CDB全体を変更する句を含むALTER DATABASE文を使用します。

例13-12 CDBの制御ファイルのバックアップ

次のALTER DATABASE文では、recovery_clauseを使用して制御ファイルをバックアップします。

```
ALTER DATABASE BACKUP CONTROLFILE TO '+DATA/dbs/backup/control.bkp';
```

例13-13 CDBへのREDOログ・ファイルの追加

次のALTER DATABASE文では、logfile_clauseを使用してREDOログ・ファイルを追加します。

```
ALTER DATABASE cdb ADD LOGFILE  
GROUP 4 ('/u01/logs/orcl/redo04a.log', '/u02/logs/orcl/redo04b.log')  
SIZE 100M BLOCKSIZE 512 REUSE;
```

関連項目:

[Oracle Database SQL言語リファレンス](#)

親トピック: [CDBルートへの接続時のコンテナの変更](#)

ALTER DATABASEを使用したCDBのUNDOモードの設定

ローカルUNDOが有効になっている場合、コンテナがオープンされる各インスタンスで、各コンテナに独自のUNDO表領域があります。ローカルUNDOが無効になっている場合は、CDB全体に対して1つのUNDO表領域があります。

- [CDB UNDOモードについて](#)
すべてのコンテナでローカルUNDOを使用するか、CDB全体で共有UNDO (デフォルト)を使用するようにCDBを構成できます。
- [ローカルUNDOモードを使用するようにCDBを構成する方法](#)
ALTER DATABASE LOCAL UNDO ON文を発行し、データベースを再起動することによって、CDBをローカルUNDOモードに変更できます。
- [共有UNDOモードを使用するようにCDBを構成する方法](#)
共有UNDOモードを使用するようにCDBを変更するには、ALTER DATABASE LOCAL UNDO OFF文を使用します。

親トピック: [CDBルートへの接続時のコンテナの変更](#)

CDBのUNDOモードについて

各コンテナでローカルUNDOを使用するように、またはCDB全体で共有UNDO(デフォルト)を使用するようにCDBを構成できます。

CDBは、ローカルUNDOモードまたは共有UNDOモードのどちらかで動作します。UNDOモードはCDB全体に適用されます。したがって、すべてのコンテナは共有UNDOまたはローカルUNDOを使用します。

CREATE DATABASE文のENABLE PLUGGABLE DATABASE句で、CDB作成時にCDBのUNDOモードを指定できます。UNDO句を指定しない場合、共有UNDOモードがデフォルトとなります。CDBの作成後にALTER DATABASE文を発行し、CDBを再起

動することで、CDBのUNDOモードを変更できます。

現在のCDB UNDOモードを判断するには、CDBルートで次の問合せを実行します。

```
SELECT PROPERTY_NAME, PROPERTY_VALUE
FROM   DATABASE_PROPERTIES
WHERE  PROPERTY_NAME = 'LOCAL_UNDO_ENABLED';
```

問合せでPROPERTY_VALUEに対してTRUEが返された場合、CDBはローカルUNDOモードです。そうでない場合、CDBは共有UNDOモードです。

- [ローカルUNDOモードについて](#)

ローカルUNDOモードは、コンテナがオープンされる各インスタンスで、各コンテナに独自のUNDO表領域があることを意味します。

- [共有UNDOモードについて](#)

共有UNDOモードは、シングル・インスタンスCDBに対して1つのアクティブなUNDO表領域のみが存在することを意味します。Oracle RAC CDBの場合、アクティブなUNDO表領域はインスタンスごとに1つ存在します。

親トピック: [ALTER DATABASEを使用したCDBのUNDOモードの設定](#)

ローカルUNDOモードについて

ローカルUNDOモードは、コンテナがオープンされる各インスタンスで、各コンテナに独自のUNDO表領域があることを意味します。

このモードでは、Oracle DatabaseはCDB内のすべてのコンテナ用のUNDO表領域を自動的に作成します。Oracle RAC CDBの場合、アクティブなUNDO表領域はローカルUNDOモードの各PDBのインスタンスごとに1つ存在します。

ローカルUNDOモードでは、各コンテナの分離が向上し、コンテナの切断やコンテナに対するポイント・イン・タイム・リカバリの実行など、一部の操作の効率が向上します。またローカルUNDOモードは、PDBの再配置やオープン読取り/書込みモードでのPDBのクローニングなど、いくつかの操作をサポートするために必要です。

CDBがローカルUNDOモードの場合は、次のことが適用されます。

- 現在のコンテナに対する適切な権限を持つユーザーは、コンテナのUNDO表領域を作成できます。
- UNDO表領域は、CDB内のすべてのコンテナの静的データ・ディクショナリ・ビューおよび動的パフォーマンス(V\$)ビューに表示されます。

関連項目:

必要な権限の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください

親トピック: [CDBのUNDOモードについて](#)

共有UNDOモードについて

共有UNDOモードは、シングル・インスタンスCDBに対して1つのアクティブなUNDO表領域のみが存在することを意味します。Oracle RAC CDBの場合、アクティブなUNDO表領域はインスタンスごとに1つ存在します。

CDBが共有UNDOモードの場合は、次のことが適用されます。

- 適切な権限を持ち、現在のコンテナがCDBルートである共通ユーザーのみが、UNDO表領域を作成できます。
- 現在のコンテナがCDBルートでない場合、UNDO表領域を作成しようとすると失敗し、エラーが返されます。
- 現在のコンテナがCDBルートである場合、UNDO表領域は静的データ・ディクショナリ・ビューおよび動的パフォーマンス

(V\$)・ビューに表示されます。現在のコンテナがPDB、アプリケーション・ルートまたはアプリケーションPDBである場合、UNDO表領域は動的パフォーマンス・ビューにのみ表示されます。

ノート:



- CDB の UNDO モードを変更した場合、新しい UNDO モードは変更後にコンテナが初めてオープンされたときに個別のコンテナに適用されます。
- CDB の UNDO モードを変更した場合、CDB 内のコンテナは変更前の時間または SCN にフラッシュ・バックできません。

親トピック: [CDBのUNDOモードについて](#)

ローカルUNDOモードを使用するようにCDBを構成する方法

ALTER DATABASE LOCAL UNDO ON文を発行し、データベースを再起動することによって、CDBをローカルUNDOモードに変更できます。

CDBがローカルUNDOモードである場合、コンテナがオープンされる各インスタンスで、各コンテナに独自のUNDO表領域があります。Oracle Databaseは、UNDO表領域を持たないCDB内のコンテナにUNDO表領域を自動的に作成します。UNDO表領域のないPDBがローカルUNDOモードを使用するように構成されたCDBにクローニング、再配置または接続される場合、Oracle Databaseは初めてオープンされる時にPDBのUNDO表領域を自動的に作成します。

CDBを共有UNDOモードからローカルUNDOモードに切り替えると、Oracle Databaseによって必要なUNDO表領域が自動的に作成されます。

1. CDBインスタンスがオープンしている場合は、停止します。
2. CDBインスタンスをOPEN UPGRADEモードで起動します。次に例を示します。
STARTUP UPGRADE
3. SQL*Plusで、現在のコンテナがCDBルートであることを確認します。たとえば、次のように入力します。

```
SHOW CON_NAME
```

```
CON_NAME
```

```
-----  
CDB$ROOT
```

4. CDBの現在のUNDOモードを問い合わせます。

```
SELECT PROPERTY_NAME, PROPERTY_VALUE  
FROM   DATABASE_PROPERTIES  
WHERE  PROPERTY_NAME = 'LOCAL_UNDO_ENABLED';
```

5. ローカルUNDOを有効にするには、次のSQL文を発行します。
ALTER DATABASE LOCAL UNDO ON;
6. CDBインスタンスを停止し、再起動します。
7. **オプション:** PDBシードにUNDO表領域を手動で作成します。

Oracle DatabaseはローカルUNDOモードでPDBシードにUNDO表領域を自動的に作成しますが、UNDO表領域を手動で作成することによって、UNDO表領域のサイズおよび構成を制御する場合があります。PDBシードから作成されるPDBが、自動作成されたUNDO表領域ではなく手動で作成したUNDO表領域を使用するには、UNDO_TABLESPACE初期化パラメータに手動で作成したUNDO表領域を設定するか、自動作成されたUNDO表領

域を削除する必要があります。

- a. SQL*Plusで、現在のコンテナがルートであることを確認します。
- b. PDBシードを読み取り/書き込みオープン・モードにします。
ALTER PLUGGABLE DATABASE PDB\$SEED OPEN READ WRITE FORCE;
- c. コンテナをPDBシードに切り替えます。
ALTER SESSION SET CONTAINER=PDB\$SEED;
- d. PDBシードにUNDO表領域を作成します。次に例を示します。

```
CREATE UNDO TABLESPACE seedundots1
  DATAFILE 'seedundots_1a.dbf'
  SIZE 10M AUTOEXTEND ON
  RETENTION GUARANTEE;
```

- e. コンテナをルートに切り替えます。
ALTER SESSION SET CONTAINER=CDB\$ROOT;
- f. PDBシードを読み取り専用オープン・モードにします。
ALTER PLUGGABLE DATABASE PDB\$SEED OPEN READ ONLY FORCE;

親トピック: [ALTER DATABASEを使用したCDBのUNDOモードの設定](#)

共有UNDOモードを使用するようにCDBを構成する方法

共有UNDOモードを使用するようにCDBを変更するには、ALTER DATABASE LOCAL UNDO OFF文を使用します。

1. CDBインスタンスがオープンしている場合は、停止します。
2. CDBインスタンスをOPEN UPGRADEモードで起動します。次に例を示します。
STARTUP UPGRADE
3. SQL*Plusで、現在のコンテナがCDBルートであることを確認します。たとえば、次のように入力します。

```
SHOW CON_NAME
CON_NAME
-----
CDB$ROOT
```

4. 必要に応じて、CDBの現在のUNDOモードを問い合わせます。

```
SELECT PROPERTY_NAME, PROPERTY_VALUE
FROM   DATABASE_PROPERTIES
WHERE  PROPERTY_NAME = 'LOCAL_UNDO_ENABLED';
```

5. ローカルUNDOをオフにするには、次のSQL文を発行します。
ALTER DATABASE LOCAL UNDO OFF;
6. CDBインスタンスを停止し、再起動します。

共有UNDOモードの場合、CDBはローカルUNDOモードのときに作成されたローカルUNDO表領域を無視します。使用しないローカルUNDO表領域を削除することをお勧めします。

親トピック: [ALTER DATABASEを使用したCDBのUNDOモードの設定](#)

ALTER DATABASEを使用したCDBルートの変更

CDBのルートのみを変更するには、ALTER DATABASE文を使用します。

現在のコンテナがルートである場合、一部のALTER DATABASE文を使用すると、PDBのいずれかを直接変更することなくルートが変更されます。これらの文のリストは、[表13-3](#)の「ルートのみの変更」列を参照してください。

一部の文は、CDB内のPDBにデフォルトを設定します。PDBのこれらのデフォルトを上書きするには、ALTER PLUGGABLE DATABASE文を使用します。

前提条件

ルートを変更するには、現在のユーザーにルートでのALTER DATABASE権限が必要です。

ルートを変更するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. ルートを変更する句を含むALTER DATABASE文を実行します。

次の例では、ルートが変更されています。

現在のコンテナがルートであり、表領域を明示的に割り当てられていないユーザーは、ルートのデフォルト永続表領域を使用します。ALTER DATABASE文で指定された表領域は、ルートに存在している必要があります。

この文を実行すると、これ以降にルートで作成される表領域のデフォルトタイプはbigfileになります。この設定はPDBのデフォルトでもあります。

ALTER DATABASE文で指定された表領域または表領域グループは、ルートに存在している必要があります。

例13-14 ルートのデフォルト永続表領域の変更

このALTER DATABASE文では、DEFAULT TABLESPACE句を使用して、ルートのデフォルト永続表領域をroot_tbsに設定します。

```
ALTER DATABASE DEFAULT TABLESPACE root_tbs;
```

例13-15 ルートのデータファイルのオンライン化

このALTER DATABASE文では、database_file_clauseを使用して、/u02/oracle/cdb_01.dbfデータファイルをオンライン化します。

```
ALTER DATABASE DATAFILE '/u02/oracle/cdb_01.dbf' ONLINE;
```

例13-16 ルートのデフォルト表領域タイプの変更

このALTER DATABASE文では、SET DEFAULT TABLESPACE句を使用して、ルートのデフォルト表領域タイプをbigfileに変更します。

```
ALTER DATABASE SET DEFAULT BIGFILE TABLESPACE;
```

例13-17 ルートのデフォルト一時表領域の変更

このALTER DATABASE文では、DEFAULT TEMPORARY TABLESPACE句を使用して、ルートのデフォルト一時表領域をroot_tempに設定します。

```
ALTER DATABASE DEFAULT TEMPORARY TABLESPACE root_temp;
```

関連項目:

- [「データベース・レベルでのPDBの変更」](#)
- [Oracle Database SQL言語リファレンス](#)

親トピック: [CDBルートへの接続時のコンテナの変更](#)

別のコンテナでのSQLの実行

別のコンテナでSQLを実行するには、DMLのCONTAINERS句、またはDDLのCONTAINER句を使用します。

- [CDB内のコンテナに対するDML文の発行](#)
CDBまたはアプリケーション・ルートで発行されるDML（データ操作言語）文では、CDB内の別のコンテナを変更できます。また、DML文に対してデフォルト・コンテナ・ターゲットを指定できます。
- [CDBでのDDL文の実行](#)
CDBでは、現在のコンテナまたはすべてのコンテナでデータ定義言語（DDL）文を実行できます。
- [CDBでのオラクル社が提供するSQLスクリプトの実行](#)
catcon.plスクリプトを使用して、Oracleが提供するSQLまたはSQLスクリプトをCDB内で実行できます。スクリプトは、指定した任意のコンテナに対して実行できます。
- [DBMS_SQLパッケージを使用したコンテナのコードの実行](#)
CDBのコンテナ内でPL/SQLコードを実行する場合に別のコンテナで1つ以上のSQL文を実行する場合、DBMS_SQLパッケージを使用してコンテナを切り替えます。

親トピック: [CDBの管理](#)

CDB内のコンテナに対するDML文の発行

CDBまたはアプリケーション・ルートで発行されるDML（データ操作言語）文では、CDB内の別のコンテナを変更できます。また、DML文に対してデフォルト・コンテナ・ターゲットを指定できます。

- [CDB内のコンテナに対するDML文の発行について](#)
DML文は、CDB内の指定したコンテナ内のデータベース・オブジェクトに影響することがあります。
- [CDB内のDML文のデフォルト・コンテナの指定](#)
CDB内のDML文のデフォルト・コンテナを指定するには、CONTAINERS DEFAULT TARGET句を指定したALTER DATABASE文を発行します。

親トピック: [別のコンテナでのSQLの実行](#)

CDB内のコンテナに対するDML文の発行について

DML文は、CDB内の指定したコンテナ内のデータベース・オブジェクトに影響することがあります。

コンテナはコンテナIDによって指定されます。コンテナIDは複数の場所に表示されることがあるため、データベースでは次の優先順位が使用されます。

1. DML文のWHERE句に指定されたCON_ID
2. CONTAINERS_DEFAULT_TARGETデータベース・プロパティ
3. 現在のコンテナ(CDBルートまたはアプリケーション・ルートのいずれか)

CDBルートまたはアプリケーション・ルートで、CONTAINERS句を含むDML文は、CDBまたはアプリケーション・コンテナ内の単一コンテナ内の表またはビューを変更できます。CONTAINERS句を使用するには、変更される表またはビューをCONTAINERS句に指定し、影響を受けるコンテナIDをWHERE句に指定します。

INSERT VALUES文でターゲット・コンテナを指定するには、VALUES句にCON_ID値を指定します。また、UPDATE文またはDELETE文でターゲット・コンテナを指定するには、WHERE句にCON_ID述語を指定します。たとえば、次のDML文はコンテナ内のsales.customers表を7のCON_IDで更新します。

```
UPDATE CONTAINERS(sales.customers) ctab
SET ctab.city_name=' MIAMI'
```

```
WHERE ctab.CON_ID=7
AND CUSTOMER_ID=3425;
```

CONTAINERS句には、次の制限事項が適用されます。

- 指定したスキーマは、CON_IDで指定されたコンテナ、および文が実行されるCDBまたはアプリケーション・ルートの両方に存在する必要があります。
- WHERE句のCON_IDに指定する値は、CDB内のPDB、アプリケーション・ルートまたはアプリケーションPDBを参照する必要があります。
- INSERTのターゲットがCONTAINERS () 内にあるSELECT文を使用したINSERT文はサポートされません。
- INSERTのターゲットがCONTAINERS () 内にある、複数表に対するINSERT文はサポートされません。
- CONTAINERS句を使用するDML文では、データベース・リスナーがTCP (IPCではなく)を使用して構成され、PORT値およびHOST値がPORT句およびHOST句を使用して各ターゲットPDBに指定されている必要があります。

親トピック: [CDB内のコンテナに対するDML文の発行](#)

CDBにおけるDML文のデフォルト・コンテナの指定

CDB内のDML文のデフォルト・コンテナを指定するには、CONTAINERS DEFAULT TARGET句を指定したALTER DATABASE文を発行します。

WHERE句でコンテナを指定せずにDML文がCDBルートで発行されている場合、DML文はCDBのデフォルト・コンテナに影響します。デフォルト・コンテナには、CDBルート、PDB、アプリケーション・ルートまたはアプリケーションPDBを含む、CDB内の任意のコンテナを指定できます。1つのデフォルトのコンテナのみを使用できます。

CONTAINERS_DEFAULT_TARGETデータベース・プロパティは、デフォルト・コンテナを設定します。デフォルトでは、このプロパティは設定されていません。次の問合せを実行することで、CDBのデフォルト・ターゲット・コンテナを決定できます。

```
SELECT PROPERTY_VALUE
FROM DATABASE_PROPERTIES
WHERE PROPERTY_NAME='CONTAINERS_DEFAULT_TARGET';
```

1. SQL*Plusで、現在のコンテナがCDBルートまたはアプリケーション・ルートであることを確認します。
現在のユーザーには、ALTER DATABASE権限が共通して付与されている必要があります。
2. CONTAINERS DEFAULT TARGET句を指定してALTER DATABASE文を実行します。

例13-18 CDB内のDML文のデフォルト・コンテナの指定

この例では、PDB1がCDB内のDML文のデフォルト・コンテナであることを指定します。

```
ALTER DATABASE CONTAINERS DEFAULT TARGET = (PDB1);
```

例13-19 デフォルト・コンテナのクリア

この例は、デフォルト・コンテナ設定をクリアしています。設定されていない場合、デフォルト・コンテナはCDBルートです。

```
ALTER DATABASE CONTAINERS DEFAULT TARGET = NONE;
```

親トピック: [CDB内のコンテナに対するDML文の発行](#)

CDBでのDDL文の実行

CDBでは、現在のコンテナまたはすべてのコンテナでデータ定義言語(DDL)文を実行できます。

- [CDBにおけるDDL文の実行について](#)

CDBでは、DDL文をすべてのコンテナに適用できる場合も、現在のコンテナにのみ適用できる場合もあります。

- [現在のコンテナにおけるDDL文の実行](#)

DDL文のCONTAINER句にCURRENTを指定して、現在のコンテナで文を実行します。

- [CDB内のすべてのコンテナにおけるDDL文の実行](#)

DDL文のCONTAINER句にALLを指定して、CDB内のすべてのコンテナで文を実行します。

親トピック: [別のコンテナでのSQLの実行](#)

CDBにおけるDDL文の実行について

CDBでは、DDL文をすべてのコンテナに適用できる場合も、現在のコンテナにのみ適用できる場合もあります。

影響を受けるコンテナを指定するには、CONTAINER句を使用します。

```
CONTAINER = { CURRENT | ALL }
```

次の設定を使用できます。

- CURRENTは、文は現在のコンテナのみに適用されることを意味します。
- ALLは、ルートおよびすべてのPDBを含むCDB内のすべてのコンテナに文が適用されることを意味します。

DDL文のCONTAINER句には、次の制限があります。

- [「現在のコンテナについて」](#)で説明している制限があります。
- CONTAINER句は、[表13-4](#)にリストされているDDL文でのみ使用できます。

表13-4 CDBでのDDL文およびCONTAINER句

DDL文	CONTAINER = CURRENT	CONTAINER = ALL
CREATE USER	現在の PDB にローカル・ユーザーを作成します。	共通ユーザーを作成します。
ALTER USER	現在の PDB のローカル・ユーザーを変更します。	共通ユーザーを変更します。
CREATE ROLE	現在の PDB にローカル・ロールを作成します。	共通ロールを作成します。
GRANT	ローカル・コンテナでの権限をローカル・ユーザー、共通ユーザーまたはローカル・ロールに付与します。 SET CONTAINER 権限を、現在の PDB でユーザーが作成した共通ユーザーに付与できます。	共通オブジェクトに対するシステム権限またはオブジェクト権限を、共通ユーザーまたは共通ロールに付与します。指定された権限は、CDB 全体にわたってユーザーまたはロールに付与されます。
REVOKE	ローカル・コンテナでの権限をローカル・ユーザー、共通ユーザーまたはローカル・	共通オブジェクトに対するシステム権限またはオブジェクト権限を、共通ユーザーまたは共通ロールから取り消します。指定さ

DDL文	CONTAINER = CURRENT	CONTAINER = ALL
	<p>ロールから取り消します。</p> <p>この文では、ローカル・コンテナの指定されたユーザーまたはロールから、CONTAINER 句に CURRENT を指定して付与された権限のみを取り消すことができます。この文は、CONTAINER 句に ALL を指定して付与された権限には影響しません。</p> <p>SET CONTAINER 権限を、現在の PDB でユーザーが作成した共通ユーザーから取り消すことができます。</p>	<p>れた権限は、CDB 全体にわたってユーザーまたはロールから取り消されます。</p> <p>この文では、指定された共通ユーザーまたは共通ロールから、CONTAINER 句に ALL を指定して付与された権限のみを取り消すことができます。この文は、CONTAINER 句に CURRENT を指定して付与された権限には影響しません。ただし、取消し対象の共通付与権限に依存するローカル付与権限も取り消されます。</p>

他のすべてのDDL文は、現在のコンテナにのみ適用されます。

ユーザー、ロールおよびプロファイルの名前の通常のルールに加えて、CDBでユーザー、ロールまたはプロファイルを作成する場合には、次のルールおよびベスト・プラクティスが適用されます。

- 共通のユーザー、ロールおよびプロファイルの名前のベスト・プラクティスは、共通のユーザー、ロールおよびプロファイルとローカルのユーザー、ロールおよびプロファイルの間に名前の競合が発生しないように、接頭辞で開始することです。この接頭辞は、CDBルートでCOMMON_USER_PREFIX初期化パラメータを使用して指定します。デフォルトでは、CDBルートでの接頭辞はC##またはc##です。
- アプリケーション・コンテナでの共通ユーザー、ロールおよびプロファイルの名前のベスト・プラクティスは、アプリケーションの共通ユーザー、ロールおよびプロファイルとローカルのユーザー、ロールおよびプロファイルとの間に名前の競合が発生しないように接頭辞で開始することです。この接頭辞は、アプリケーション・ルートでCOMMON_USER_PREFIX初期化パラメータを使用して指定します。デフォルトでは、アプリケーション・ルートでの接頭辞はNULLです。
- COMMON_USER_PREFIX初期化パラメータがアプリケーション・ルートで設定されている場合、設定はアプリケーション・コンテナ内のアプリケーション共通ユーザー、ロールおよびプロファイルの名前に適用されます。接頭辞はCDBルートとアプリケーション・ルートで異なる場合があり、各アプリケーション・コンテナでも異なる場合があります。
- 共通のユーザー、ロールおよびプロファイルの名前は、ASCII文字のみで構成する必要があります。この制限は、アプリケーションの共通のユーザー、ロールおよびプロファイルの名前には適用されません。
- ローカルのユーザー、ロールおよびプロファイルの名前は、COMMON_USER_PREFIX初期化パラメータで共通ユーザーに指定した接頭辞で始めることができません。
- ローカルのユーザー、ロールおよびプロファイルの名前は、C##またはc##で始めないでください。
- CDBルートのCOMMON_USER_PREFIXの値に関係なく、アプリケーションの共通ユーザー、ロールおよびプロファイルの名前をC##またはc##で開始することはできません。
- アプリケーションの共通ユーザー、ロールおよびプロファイルの名前は、COMMON_USER_PREFIX初期化パラメータで共通ユーザーに指定した接頭辞で始めることができません。

関連項目:

- CDBでのALTER SYSTEM文の使用の詳細は、[「ALTER SYSTEMを使用したCDBの変更」](#)を参照してください

- [Oracle Database SQL言語リファレンス](#)
- [『Oracle Database概要』](#)
- CDBでのユーザー管理の詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください
- COMMON_USER_PREFIX初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください

親トピック: [CDBでのDDL文の実行](#)

現在のコンテナにおけるDDL文の実行

DDL文のCONTAINER句にCURRENTを指定して、現在のコンテナで文を実行します。

サポートされるDDL文を[表13-4](#)に示します。

現在のユーザーは、現在のコンテナでDDL文を実行するために必要な権限を付与されている必要があります。たとえば、ユーザーを作成するには、現在のユーザーに現在のコンテナでのCREATE USERシステム権限が付与されている必要があります。

現在のコンテナでDDL文を実行するには:

1. SQL*Plusでコンテナにアクセスします。
[「SQL*Plusを使用したCDB内のコンテナへのアクセス」](#)を参照してください。
2. CONTAINERがCURRENTに設定されたDDL文を実行します。

ローカル・ユーザーのユーザー名は、COMMON_USER_PREFIX初期化パラメータで指定した接頭辞で始めることができません。デフォルトでは、CDBルートでは接頭辞はC##またはc##です。アプリケーション・ルートでは、アプリケーション・コンテナに独自の接頭辞を指定できます。さらに、共通ユーザーの名前はASCII文字のみで構成する必要があります。指定する表領域は、PDBに存在している必要があります。

例13-20 PDBでのローカル・ユーザーの作成

この例では、現在のPDBにローカル・ユーザーtestpdbを作成します。

```
CREATE USER testpdb IDENTIFIED BY password
  DEFAULT TABLESPACE pdb1_tbs
  QUOTA UNLIMITED ON pdb1_tbs
  CONTAINER = CURRENT;
```

親トピック: [CDBでのDDL文の実行](#)

CDB内のすべてのコンテナにおけるDDL文の実行

DDL文のCONTAINER句にALLを指定して、CDB内のすべてのコンテナで文を実行します。

サポートされるDDL文を[表13-4](#)に示します。

次の前提条件を満たしている必要があります。

- 現在のユーザーは共通ユーザーである必要があります。
- 現在のユーザーは、DDL文を実行するために必要な権限を共通に付与されている必要があります。たとえば、ユーザーを作成するには、現在のユーザーには共通でCREATE USERシステム権限を付与する必要があります。

CDB内のすべてのコンテナでDDL文を実行するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。

[「CDB内のテナントへのアクセスについて」](#)を参照してください。

2. CONTAINERがALLに設定されたDDL文を実行します。

共通ユーザーのユーザー名は、COMMON_USER_PREFIX初期化パラメータで指定した接頭辞で始める必要があります。デフォルトでは、CDBルートでは接頭辞はC##またはc##です。アプリケーション・ルートでは、アプリケーション・テナントに独自の接頭辞を指定できます。さらに、共通ユーザーの名前はASCII文字のみで構成する必要があります。指定する表領域は、ルートおよびすべてのPDBに存在している必要があります。

例13-21 CDBでの共通ユーザーの作成

この例では、共通ユーザーc##testcdbを作成します。

```
CREATE USER c##testcdb IDENTIFIED BY password
  DEFAULT TABLESPACE cdb_tbs
  QUOTA UNLIMITED ON cdb_tbs
  CONTAINER = ALL;
```

親トピック: [CDBでのDDL文の実行](#)

CDBでのOracle社が提供するSQLスクリプトの実行

catcon.plスクリプトを使用して、Oracleが提供するSQLまたはSQLスクリプトをCDB内で実行できます。スクリプトは、指定した任意のテナントに対して実行できます。

- [CDBでのOracle社が提供するSQLスクリプトの実行について](#)
CDBでは、SQLスクリプトおよびSQL文を実行する場合、catcon.plスクリプトが最良の方法となります。
- [catcon.plの構文およびパラメータ](#)
catcon.plスクリプトはPerlスクリプトであり、オペレーティング・システムのプロンプトで実行される必要があります。
- [catcon.plスクリプトの実行](#)
catcon.plスクリプトの実行を例で示します。

親トピック: [別のテナントでのSQLの実行](#)

CDBでのOracle社が提供するSQLスクリプトの実行について

CDBでは、SQLスクリプトおよびSQL文を実行する場合、catcon.plスクリプトが最良の方法となります。

Oracle Databaseインストールには、複数のSQLスクリプトが含まれています。これらのスクリプトにより、データ・ディクショナリ・ビューの作成やオプションのインストールなどの操作が実行されます。

catcon.plスクリプトは、ルートおよび指定したPDBでスクリプトを正しい順序で実行でき、SQLスクリプトまたはSQL文で予期しないエラーが生成されなかったことを確認するために表示できるログ・ファイルを生成します。また、これにより、複数のプロセスが開始され、それらのプロセスが、事前に割り当てられたスクリプトの実行を完了すると、新しいスクリプトがプロセスに割り当てられます。

ノート:



- catcon.pl を実行するときに PDB シードを除外しないかぎり、SQL スクリプトまたは SQL 文は PDB シード上で実行されます。
- catcon.pl スクリプトを使用して、CDB と非 CDB の両方でスクリプトを実行できます。

catcon.plの構文およびパラメータ

catcon.pl スクリプトはPerlスクリプトであり、オペレーティング・システムのプロンプトで実行される必要があります。

catcon.pl スクリプトには、次の構文およびパラメータが含まれています。

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl
[--usr username[/password]]
[--int_usr username[/password]]
[--script_dir directory]
[--log_dir directory]
[/{--incl_con|--excl_con} container]
[--echo]
[--spool]
[--error_logging { ON | errorlogging-table-other-than-SPERRORLOG } ]
[--app_con application_root]
[--no_set_errlog_ident]
[--diag]
[-ignore_unavailable_pdb]
[--verbose]
[--force_pdb_mode pdb_mode]
[--num_procs number]
[--user_scripts]
[--recover]
--log_file_base log_file_name_base
-- { SQL_script [arguments] | --x' SQL_statement' }
```

--x SQL_statementは、1文字のパラメータが続く場合、前に--を付けます。--x SQL_statementの前にスクリプト名または別の--x SQL_statementを指定する場合、前の--は付けません。また、SQL文は一重引用符で囲む必要があります。

SQLスクリプトにコマンドライン・パラメータは、--pを使用して導入します。SQLスクリプトに対話型(シークレット)パラメータは、--Pを使用して導入します。

catcon.pl スクリプトのヘルプを表示するには、ディレクトリを\$ORACLE_HOME/perl/bin/に変更してから、次のコマンドを実行します。

```
perl $ORACLE_HOME/rdbms/admin/catcon.pl --help
```

次の表は、catcon.plパラメータの説明です。必須と指定されていないかぎり、パラメータはオプションです。

次の表に示す短いパラメータ名は、下位互換性があります。短縮名がないパラメータもあります。

表13-5 catcon.plのパラメータ

パラメータ	短縮名	説明
--usr	-u	ルートおよび指定された PDB に接続するユーザー名およびパスワードを指定します。SQL スクリプトまたは SQL 文を実行するために必要な権限を持つ共通ユーザーを指定します。デフォルトは / AS SYSDBA です。パスワードを入力しないと、catcon.pl によってパスワードが求められます。
--int_usr	-U	ルートおよび指定された PDB に接続するユーザー名およびパスワードを指定します。CDB メタデータを問い合わせなどの内部タスクを実行する場合は、必要な権限を持つ共通ユーザーを指定します。デフォルトは、 / AS

パラメータ	短縮名	説明
		SYSDBA です。パスワードを入力しないと、catcon.pl によってパスワードが求められます。
--script_dir	-d	SQL スクリプトを含むディレクトリ。デフォルトは、現行のディレクトリです。
--log_dir	-l	catcon.pl によってログ・ファイルが書き込まれるディレクトリ。デフォルトは、現行のディレクトリです。
{--incl_con --excl_con}	{-c -C}	<p>SQL スクリプトを実行するコンテナまたは実行しないコンテナ。</p> <p>--incl_con パラメータには、SQL スクリプトを実行するコンテナをリストします。</p> <p>--excl_con パラメータには、SQL スクリプトを実行しないコンテナをリストします。</p> <p>一重引用符で囲まれた PDB 名のスペース区切りリストで、コンテナを指定します。</p> <p>--incl_con パラメータと--excl_con パラメータは、両方同時には指定できません。</p> <p>このパラメータを使用する場合、--app_con パラメータは使用できません。</p>
--echo	-e	スクリプト実行時のエコーを ON に設定します。デフォルトでのエコーは OFF です。
--spool	-s	<p>すべてのスクリプトの出力を次の名前ファイルにスプーリングします。</p> <p>log-file-name-base_script-name-without-extension_[container-name-if-any]. default-extension</p>
--error_logging	-E	<p>ON に設定されている場合、デフォルトのエラー・ロギング表が使用されます。デフォルト設定は ON です。ON に設定されている場合、エラーは、SQL スクリプトを実行する各コンテナの現行スキーマ内の表 SPERRORLOG に書き込まれます。この表がコンテナに存在しない場合は、自動的に作成されます。</p> <p>SPERRORLOG 以外の表が指定されている場合、エラーは指定された表に書き込まれます。この表は SQL スクリプトを実行する各コンテナに存在している必要があり、現在のユーザーには、これらのコンテナごとの表に DML 操作を実行するために必要な権限がある必要があります。</p> <p>エラー・ロギング表の詳細は、SQL*Plus ユーザーズ・ガイドおよびリファレンスを参照してください。</p>

パラメータ	短縮名	説明
<code>--app_con</code>	<code>-F</code>	<p>アプリケーション・ルートを指定します。スクリプトは、アプリケーション・ルートと、アプリケーション・ルートにプラグインされるアプリケーション PDB で実行されます。</p> <p>このパラメータを使用する場合、<code>--incl_con</code> パラメータと <code>--excl_con</code> パラメータは使用できません。</p>
<code>--no_set_errlog_ident</code>	<code>-I</code>	<p>SET ERRORLOGGING 識別子を発行しないでください。このオプションは、SET ERRORLOGGING 識別子がすでに設定され、上書きしない場合を対象としています。</p>
<code>--diag</code>	<code>-g</code>	<p>デバッグ情報の生成をオンにします。</p>
<code>--verbose</code>	<code>-v</code>	<p>冗長出力をオンにします</p>
<code>--ignore_unavailable_pdb</code>	<code>-f</code>	<p>クローズされている PDB あるいは <code>--incl_con</code> または <code>--excl_con</code> オプションを使用している場合は存在しない PDB を無視し、明示的または暗黙的に指定したオープン PDB のみを処理します。</p> <p>このオプションを指定せず、指定された PDB の一部が存在しなかったり、オープンされていない場合、エラーが返されてコンテナは処理されません。</p>
<code>--force_pdb_mode</code>	該当なし	<p>スクリプトを実行する対象の PDB すべてに必要なオープン・モード。次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> ● UNCHANGED ● READ WRITE ● READ ONLY ● UPGRADE ● DOWNGRADE <p>UNCHANGED 以外の値を指定した場合、スクリプトを実行する対象の PDB はすべて、指定したオープン・モードに変更されます。PDB が異なるモードでオープンされている場合、PDB はクローズされ、指定したモードで再オープンされます。スクリプトがすべて実行されると、PDB は元のオープン・モードにリストアされます。</p> <p>デフォルトの UNCHANGED を指定した場合、PDB のオープン・モードは変更されません。</p>

パラメータ	短縮名	説明
--num_procs	-n	コール元から提供された文またはスクリプト(またはその両方)を実行するために catcon.pl によって生成される、SQL*Plus プロセスの数を指定します。これは、CDB 内の PDB の数および CPU_COUNT 初期化パラメータの値に基づいて catcon.pl によって生成される数より優先されます。
--user_scripts	-S	コール元から提供されるすべてのスクリプトまたは文(またはその両方)を CDB\$ROOT、PDB\$SEED またはアプリケーション・ルート・クローンで実行しないことを指定します。スクリプトまたは文(またはその両方)によって作成された表やビューなどのすべてのオブジェクトは、Oracle の保持対象としてマークされません。
--recover	-R	catcon.pl によって起動された SQL*Plus プロセスが予期せず終了した場合に、catcon.pl で SQL*Plus プロセスのリカバリを試みます。このパラメータを指定しない場合、catcon.pl はプロセスのリカバリを試行せずにクローズします。
--log_file_base	-b	(必須)ログ・ファイル名のベース名。

親トピック: [CDBでのオラクル社が提供するSQLスクリプトの実行](#)

catcon.plスクリプトの実行

catcon.plスクリプトの実行を例で示します。

catcon.plで実行されるSQLスクリプトまたはSQL文によって、データ操作言語(DML)またはデータ定義言語(DDL)の各操作を実行する場合、変更対象のコンテナは、読取り/書込みモードである必要があります。

catcon.plスクリプトを実行するには:

1. コマンドライン・プロンプトをオープンします。
2. catcon.plスクリプトを実行し、1つ以上のSQLスクリプトまたはSQL文を指定します。

```
cd $ORACLE_HOME/perl/bin/
perl $ORACLE_HOME/rdbms/admin/catcon.pl parameters SQL_script
perl $ORACLE_HOME/rdbms/admin/catcon.pl parameters -- --xSQL_statement
```

例13-22 CDB内のすべてのコンテナにおけるcatblock.sqlスクリプトの実行

次の例では、CDBのすべてのコンテナでcatblock.sqlスクリプトを実行します(バックスラッシュは行の継続を示します)。

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl ¥
--usr SYS --script_dir $ORACLE_HOME/rdbms/admin ¥
--log_file_base catblock_output catblock.sql
```

次のパラメータが指定されます。

- --usrパラメータは、SYSユーザーが各コンテナでスクリプトを実行することを指定します。
- --script_dirパラメータは、SQLスクリプトが\$ORACLE_HOME/rdbms/adminディレクトリに存在することを指定します。
- --log_file_baseパラメータでは、ログ・ファイル名のベース名がcatblock_outputであることを指定します。

デフォルトのパラメータ値は、その他すべてのパラメータに使用されます。--incl_conと--excl_conのいずれのパラメータも指定されていません。したがって、catcon.plにより、デフォルトですべてのコンテナにおいてスクリプトが実行されます。

例13-23 特定のPDBにおけるcatblock.sqlスクリプトの実行

次の例では、CDB内のPDB hrpdbおよびsalespdbで、catblock.sqlスクリプトが実行されます。

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl ¥  
--usr SYS --int_usr SYS --script_dir $ORACLE_HOME/rdbms/admin ¥  
--log_dir '/disk1/script_output' --incl_con 'HRPDB SALES PDB' ¥  
--log_file_base catblock_output catblock.sql
```

次のパラメータが指定されます。

- --usrパラメータは、SYSユーザーが各コンテナでスクリプトを実行することを指定します。
- --int_usrパラメータは、SYSユーザーが内部タスクを実行することを指定します。
- --script_dirパラメータは、SQLスクリプトが\$ORACLE_HOME/rdbms/adminディレクトリに存在することを指定します。
- --log_dirパラメータは、出力ファイルが/disk1/script_outputディレクトリに配置されることを指定します。
- --incl_conパラメータは、SQLスクリプトがPDB hrpdbおよびsalespdbで実行されることを指定します。スクリプトは、CDB内の他のコンテナでは実行されません。
- --log_file_baseパラメータでは、ログ・ファイル名のベース名がcatblock_outputであることを指定します。

例13-24 特定のPDBを除くすべてのコンテナにおけるcatblock.sqlスクリプトの実行

次の例では、PDB hrpdbおよびsalespdbを除くCDB内のすべてのコンテナで、catblock.sqlスクリプトを実行します。

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl ¥  
--usr SYS --script_dir $ORACLE_HOME/rdbms/admin ¥  
--log_dir '/disk1/script_output' --excl_con 'HRPDB SALES PDB' ¥  
--log_file_base catblock_output catblock.sql
```

次のパラメータが指定されます。

- --usrパラメータは、SYSユーザーが各コンテナでスクリプトを実行することを指定します。
- --script_dirパラメータは、SQLスクリプトが\$ORACLE_HOME/rdbms/adminディレクトリに存在することを指定します。
- --log_dirパラメータは、出力ファイルが/disk1/script_outputディレクトリに配置されることを指定します。
- --excl_conパラメータは、PDB hrpdbおよびsalespdbを除くCDB内のすべてのコンテナで、SQLスクリプトを実行することを指定します。
- --log_file_baseパラメータでは、ログ・ファイル名のベース名がcatblock_outputであることを指定します。

例13-25 コマンドライン・パラメータを使用したSQLスクリプトの実行

次の例では、CDB内のすべてのコンテナに対し、custom_script.sqlスクリプトを実行します。

```
cd $ORACLE_HOME/perl/bin/  
perl $ORACLE_HOME/rdbms/admin/catcon.pl --usr SYS --script_dir /u01/scripts ¥  
--log_file_base custom_script_output custom_script.sql '--phr' ¥  
'--PEnter password for user hr:'
```

次のパラメータが指定されます。

- --usrパラメータは、SYSユーザーが各コンテナでスクリプトを実行することを指定します。
- --script_dirパラメータは、SQLスクリプトが/u01/scriptsディレクトリ内にあることを指定します。

- --log_file_baseパラメータは、ログ・ファイル名のベース名がcustom_script_outputであることを指定します。
- --pパラメータは、コマンドライン・パラメータに対してhrを指定します。
- --Pパラメータは、ユーザーhrのパスワードを求める対話型パラメータを指定します。

デフォルトのパラメータ値は、その他すべてのパラメータに使用されます。-incl_conと-excl_conのいずれのパラメータも指定されていません。したがって、catcon.plにより、デフォルトですべてのコンテナにおいてスクリプトが実行されます。

例13-26 CDB内のすべてのコンテナにおけるSQL文の実行

次の例では、CDBのすべてのコンテナでSQL文を実行します。

```
cd $ORACLE_HOME/perl/bin/
perl $ORACLE_HOME/rdbms/admin/catcon.pl --usr SYS --echo ¥
--log_file_base select_output -- --x"SELECT * FROM DUAL"
```

次のパラメータが指定されます。

- --usrパラメータは、SYSユーザーが各コンテナでスクリプトを実行することを指定します。
- --echoパラメータは、SQL文に対する出力を表示します。
- --log_file_baseパラメータは、ログ・ファイル名のベース名がselect_outputであることを指定します。
- SQL文SELECT * FROM DUALは引用符で囲まれ、前に--xが付きます。--xの前にはパラメータ(--log_file_base)が指定されているため、--を前に付ける必要があります。

デフォルトのパラメータ値は、その他すべてのパラメータに使用されます。-incl_conと-excl_conのいずれのパラメータも指定されていません。したがって、catcon.plにより、デフォルトですべてのコンテナにおいてSQL文が実行されます。

関連項目:

- [「PDBのオープン・モードの変更」](#)
- catblock.sqlスクリプトの詳細は、[Oracle Database管理者ガイド](#)を参照してください
- SQLスクリプトの詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください

親トピック: [CDBでのオラクル社が提供するSQLスクリプトの実行](#)

DBMS_SQLパッケージを使用したコンテナのコードの実行

CDBのコンテナ内でPL/SQLコードを実行する場合に別のコンテナで1つ以上のSQL文を実行する場合、DBMS_SQLパッケージを使用してコンテナを切り替えます。

たとえば、複数のコンテナで同じアクションを実行する必要がある場合は、DBMS_SQLパッケージを使用してコンテナを切り替えることができます。

コンテナの切替えにDBMS_SQLを使用する場合、次を考慮します。

- トランザクションを複数のコンテナにわたって実行することはできません。
ターゲット・コンテナ内で実行する必要があるアクションのセットがトランザクションである必要がある場合、最終アクションとしてコミットまたはロールバックを行う、自律型のトランザクションの使用を検討します。
- SET ROLE文は許可されていません。

例13-27 複数のテナントでの同一のアクションの実行

次の例のPL/SQLブロックは、2つのPDB (pdb1およびpdb2)のhrスキーマにidentact表を作成します。この例では、両方のPDBのidentact表に行も挿入します。

```
DECLARE
  c1 INTEGER;
  rowcount INTEGER;
  taskList VARCHAR2(32767) :=
    'DECLARE
      PRAGMA AUTONOMOUS TRANSACTION;
      BEGIN
        -- Create the hr.identact table.
        EXECUTE IMMEDIATE
          '''CREATE TABLE hr.identact
            (actionno NUMBER(4) NOT NULL,
             action VARCHAR2 (10))''';
        EXECUTE IMMEDIATE
          '''INSERT INTO identact VALUES(1, 'ACTION1')''';
        -- A commit is required if the tasks include DML.
        COMMIT;
      EXCEPTION
        WHEN OTHERS THEN
          -- If there are errors, then drop the table.
          BEGIN
            EXECUTE IMMEDIATE '''DROP TABLE identact''';
          EXCEPTION
            WHEN OTHERS THEN
              NULL;
          END;
      END;';
  TYPE containerListType IS TABLE OF VARCHAR2(128) INDEX BY PLS_INTEGER;
  containerList containerListType;
BEGIN
  containerList(1) := 'PDB1';
  containerList(2) := 'PDB2';
  c1 := DBMS_SQL.OPEN_CURSOR;
  FOR conIndex IN containerList.first..containerList.last LOOP
    DBMS_OUTPUT.PUT_LINE('Creating in container: ' || containerList(conIndex));
    DBMS_SQL.PARSE(
      c => c1 ,
      statement => taskList,
      language_flag => DBMS_SQL.NATIVE,
      edition => NULL,
      apply_crossedition_trigger => NULL,
      fire_apply_trigger => NULL,
      schema => 'HR',
      container => containerList(conIndex));
    rowcount := DBMS_SQL.EXECUTE(c=>c1);
  END LOOP;
  DBMS_SQL.CLOSE_CURSOR(c=>c1);
END;
```

関連項目:

- DBMS_SQLパッケージの詳細は、[『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)を参照してください。

- 自律型トランザクションの詳細は、[『Oracle Database PL/SQL言語リファレンス』](#)を参照してください。

親トピック: [別のコンテナでのSQLの実行](#)

CDBインスタンスの停止

CDBインスタンスは、非CDBインスタンスを停止する場合と同じ方法で停止できます。

前提条件

次の前提条件を満たしている必要があります。

- CDBインスタンスは、マウントされているか、またはオープンしている必要があります。
- 現在のユーザーは、SYSDBA、SYSOPER、SYSBACKUPまたはSYSDBGの管理権限を持つ共通ユーザーである必要があります。CDBを停止するには、ユーザーは接続時にAS SYSDBA、AS SYSOPER、AS SYSBACKUPまたはAS SYSDBGをそれぞれ使用して、権限を行使する必要があります。

CDBを停止するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
[『SQL*Plus CONNECTコマンドを使用したコンテナへの接続』](#)を参照してください。
2. CDBインスタンスを停止します。

関連項目:

- [『PDBのオープン・モードの変更』](#)
- [『現在のコンテナについて』](#)
- インスタンスの停止方法を学習するには、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBの管理](#)

14 CDBフリートの管理

CDBフリートは、1つの論理CDBとして管理できるCDBおよびホストされるPDBの集合です。

- [CDBフリートについて](#)
リードCDBは、フリート内のCDBを監視および管理する中心的な場所です。
- [CDBフリートの目的](#)
CDBフリートによって、多数のCDBのスケラビリティおよび集中管理のためのデータベース・インフラストラクチャが提供されます。
- [CDBフリートへのリードCDBの設定](#)
LEAD_CDBデータベース・プロパティをtrueに設定することによって、CDBフリートにリードCDBを設定します。
- [CDBフリート・メンバーの指定](#)
LEAD_CDB_URIデータベース・プロパティを、リードCDBを指し示すデータベース・リンクに設定することによって、フリート・メンバーを指定します。

親トピック: [マルチテナント環境の管理](#)

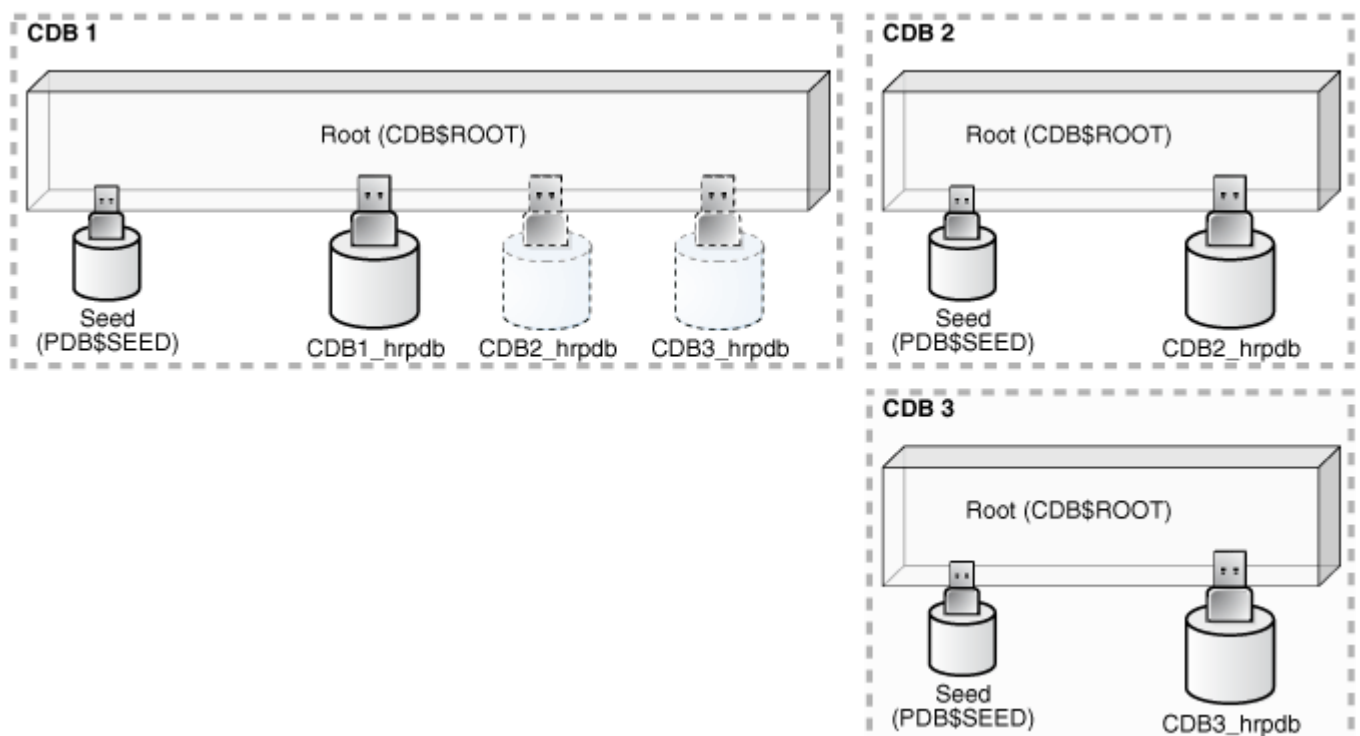
CDBフリートについて

リードCDBは、フリート内のCDBを監視および管理する中心的な場所です。

LEAD_CDBデータベース・プロパティをTRUEに設定することによって、フリート内の1つのCDBをリードCDBに指定します。LEAD_CDB_URIデータベース・プロパティを設定することによって、フリート内の他のCDBがリードCDBを指し示します。CDBフリートを構成した後、様々なCDBからのPDB情報はリードCDBと同期されます。CDB内のすべてのPDBがリードCDBに表示されるため、フリート内のPDBにはリードCDBから単一の論理CDBとしてアクセスできます。

次の図は、CDB1、CDB2およびCDB3で構成されたCDBフリートを示しています。リードCDBはCDB1です。CDB2に存在するCDB2_hrpdがCDB1に表示されます。CDB3に存在するCDB3_hrpdもCDB1に表示されます。

図14-1 CDBフリート



Oracle Real Application Cluster (Oracle RAC)、RMAN、Point-in-Timeリカバリ、フラッシュバック機能など、

Oracle Databaseのすべての機能がフリート内のCDBでサポートされています。

次のコンテナ間機能を使用して、CDBフリート内のCDBおよびPDBにアクセスできます。

- CDBビュー
- GV\$ビュー
- CONTAINERS句
- コンテナ・マップ

アプリケーション・コンテナを使用して共通アプリケーション・スキーマが構成されている場合、これらのコンテナ間機能によって、フリートで管理されている異なるCDB内のPDB間で問合せおよびデータ集計が可能になります。

ノート:



- 各 PDB 名は、CDB フリート内のすべての CDB で一意である必要があります。
- フリート内の任意の CDB に PDB を作成できますが、PDB が作成された CDB でのみ PDB を開くことができます。

関連項目:

- [「CDBおよびPDBの監視」](#)
- [「コンテナ・マップを使用したPDBでのパーティション化」](#)

親トピック: [CDBフリートの管理](#)

CDBフリートの目的

CDBフリートによって、多数のCDBのスケーラビリティおよび集中管理のためのデータベース・インフラストラクチャが提供されます。

CDBフリートは次の状況で役立ちます。

- プロビジョニングが必要なPDBの数がMAX_PDBS初期化パラメータ設定を超えているため、複数のCDBを作成する必要がある場合。
- 最適に機能するには、1つの構成内の異なるPDBで異なるタイプのサーバーが必要な場合。
たとえば、一部のPDBは大規模なトランザクションの負荷を処理する可能性があり、それ以外のPDBは主に監視用として使用される場合、これらのPDBに対して適切なサーバー・リソース(CPU、メモリー、I/Oレート、ストレージ・システムなど)が必要です。
- 同じアプリケーションを使用する異なるPDBは、異なる場所に存在する必要があります。

CDB間の収集の監視と診断

リードCDBは、フリート内のCDB間で実行する監視アプリケーションおよびレポート・アプリケーションを実行できます。1つのコンテナに監視アプリケーションをインストールし、CDBビューおよびGV\$ビューを使用して、CDBフリート全体の診断データを監視および処理できます。リードCDBで発行されたコンテナ間の問合せは、CDBフリート全体のすべてのPDBで自動的に実行できます。

Software as a Service (SaaS)アプリケーション

CDBフリート全体で異なるアプリケーション・コンテナの共通スキーマおよび共通アプリケーション・オブジェクトを使用すると、CONTAINERS句またはコンテナ・マップを使用して、CDBフリート内のすべてのPDB間で問合せを実行できます。CDB間で共通アプリケーション・スキーマを使用するために、アプリケーションをアプリケーション・ルートにインストールできます。

一般的なユースケースでは、マスター・アプリケーション・ルートをリードCDBにインストールします。アプリケーション・ルート・クローンはフリート内の他のすべてのCDBに存在します。アプリケーション・ルート・クローンのプロキシPDBはマスター・アプリケーション・ルートに存在します。

Database as a Service (DBaaS)アプリケーション

リードCDBは、CDBフリートにプロビジョニングされた全部または一部のPDBの使用状況メトリックとステータスを収集および表示できる中心的な場所として機能します。

マイクロサービス

マイクロサービスは、柔軟で独立してデプロイ可能なソフトウェア・システムを構築するために使用する、サービス指向アーキテクチャを特殊化したものです。マイクロサービスを使用すると、各チームは、カスタマイズされたスケーリングおよび可用性SLAを使用してCDBフリートをデプロイおよび管理できます。CDBでは、異なるストレージ・システムおよび構成設定を使用して、様々なタイプのワークロードに対応できます。リードCDBは、個々のマイクロサービスに関連付けられているCDBの集合を中央のDBAが管理するのに役立ちます。

関連項目:

- [「CDBおよびPDBの監視」](#)
- [「コンテナ・マップを使用したPDBでのパーティション化」](#)
- MAX_PDBSについてさらに学習するには、[Oracle Databaseリファレンス](#)を参照してください

親トピック: [CDBフリートの管理](#)

CDBフリートへのリードCDBの設定

LEAD_CDBデータベース・プロパティをtrueに設定することによって、CDBフリートにリードCDBを設定します。

CDBフリートにリードCDBを設定するには:

1. SQL*Plusで、現在のコンテナが、リードCDBになるCDBのルートであることを確認します。
2. 必要に応じて、次の問合せを実行して、現在のLEAD_CDBデータベース・プロパティを確認します。

```
SELECT PROPERTY_VALUE
FROM   DATABASE_PROPERTIES
WHERE  PROPERTY_NAME=' LEAD_CDB' ;
```

3. LEAD_CDBデータベース・プロパティをTRUEに設定します。

例14-1 リードCDBのデータベース・プロパティをtrueに設定

1. CDBルートにアクセスします。

```
ALTER SESSION SET CONTAINER = CDB$ROOT;
```

2. 次のSQL文を実行します。

```
ALTER DATABASE SET LEAD_CDB = TRUE;
```

関連項目:

[「CDB内のコンテナへのアクセスについて」](#)

親トピック: [CDBフリートの管理](#)

CDBフリート・メンバーの指定

LEAD_CDB_URIデータベース・プロパティを、リードCDBを指し示すデータベース・リンクに設定することによって、フリート・メンバーを指定します。

前提条件

固定ユーザーのセマンティクスを含むデータベース・リンク(つまり、ユーザー名とパスワードがリンク定義に含まれる)を使用する必要があります。リンクでは、接続ユーザーのセマンティクス(ユーザー名とパスワードがリンク定義に含まれない場合)を使用できません。

CDBフリート・メンバーを指定するには:

1. SQL*Plusで、現在のコンテナが、フリートメンバーとして指定するCDBのルートであることを確認します。
2. 必要に応じて、次の問合せを実行して、現在のLEAD_CDB_URIデータベース・プロパティを確認します。

```
SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES WHERE PROPERTY_NAME=' LEAD_CDB_URI' ;
```

3. データベース・リンクが存在しない場合は、フリート内のリードCDBのルートへのリンクを作成します。

データベース・リンクは、固定共通ユーザー・データベース・リンクである必要があります。

4. LEAD_CDB_URIデータベース・プロパティを、リードCDBへのデータベース・リンクの名前に設定します。

例14-2 CDBフリート・メンバーの指定

この例では、リードCDBがcdb1で、リードCDBへのデータベース・リンクが存在しないと想定しています。また、現在のCDBがlead_podサービス名を使用してcdb1に接続できるネットワークが構成されていると想定しています。

1. フリート・メンバーとして指定するCDBのルートにアクセスします。

```
ALTER SESSION SET CONTAINER = CDB$ROOT;
```

2. cdb1へのデータベース・リンクを作成します。

```
CREATE PUBLIC DATABASE LINK lead_link  
CONNECT TO C##CF1 IDENTIFIED BY password  
USING ' lead_pod' ;
```

3. LEAD_CDB_URIプロパティをデータベース・リンクの名前に設定します。

```
ALTER DATABASE SET LEAD_CDB_URI = ' dblink:LEAD_LINK' ;
```

関連項目:

- [「CDB内のコンテナへのアクセスについて」](#)

- 固定ユーザー・データベース・リンクの詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBフリートの管理](#)

15 PDBの管理

PDBの管理には、PDBへの接続、PDBの変更、PDBに関連付けられたサービスの管理などのタスクが含まれます。

- [PDB管理について](#)
プラグブル・データベース(PDB)の管理には、非CDBの管理に必要なタスクのサブセットが含まれます。
- [PDBへの接続の管理](#)
PDBの接続は非CDBと同じ方法で管理しますが、特別な考慮事項がいくつかあります。
- [システム・レベルでのPDBの変更](#)
ALTER SYSTEM文を使用してPDBを変更できます。
- [データベース・レベルでのPDBの変更](#)
ALTER PLUGGABLE DATABASE文を使用してPDBを変更できます。
- [STARTUPおよびSHUTDOWNを使用したPDBのオープン・モードの変更](#)
現在のコンテナがPDBの場合、SQL*PlusのSTARTUPコマンドを使用してPDBをオープンし、SQL*PlusのSHUTDOWNコマンドを使用してPDBをクローズできます。
- [Oracle RACでのPDBの起動および停止](#)
プラグブル・データベース(PDB)の管理には、非CDBを管理するために必要なタスクのごく一部が必要です。

関連トピック

- [マルチテナント環境のツール](#)

親トピック: [マルチテナント環境の管理](#)

PDB管理について

プラグブル・データベース(PDB)の管理には、非CDBの管理に必要なタスクのサブセットが含まれます。

タスクのこのサブセットは、ほとんどがPDBと非CDBで同じですが、違いもあります。たとえば、PDBのオープン・モードを変更する場合に違いがあります。また、PDB管理者は、単一のPDBの管理に限定され、マルチテナント・コンテナ・データベース(CDB)内の他のPDBは管理できません。

- [PDBと非CDBに共通するタスク](#)
ほとんどの管理タスクがPDBと非CDBで同じです。
- [CDBに固有のタスク](#)
現在のコンテナがPDBである場合に、いくつかの管理タスクを実行できません。

関連項目:

現在のPDBのオープン・モードを変更するには、[「データベース・レベルでのPDBの変更」](#)を参照してください

親トピック: [PDBの管理](#)

PDBと非CDBに共通するタスク

ほとんどの管理タスクがPDBと非CDBで同じです。

PDBを管理する場合、ALTER DATABASE、ALTER PLUGGABLE DATABASEまたはALTER SYSTEM文を使用してPDBを変更できます。また、PDBでDDL文を実行することもできます。次の表は、PDBと非CDBに共通する一部のタスクについて説明してい

ます。

表15-1 PDBと非CDBに共通する管理タスク

タスク	説明	追加情報
表領域の管理	PDB の表領域を作成、変更および削除できます。各 PDB のデフォルト表領域およびデフォルト表領域タイプを指定できます。また、各 PDB のデフォルト一時表領域があります。必要に応じて、個々の PDB で使用する追加の一時表領域を作成できます。	「データベース・レベルでの PDB の変更」 表領域の管理の詳細は、『 Oracle Database 管理者ガイド 』を参照してください
データファイルおよび一時ファイルの管理	各 PDB には独自のデータファイルがあります。非 CDB で管理しているのと同じ方法で、データファイルおよび一時ファイルを管理できます。CREATE PLUGGABLE DATABASE または ALTER PLUGGABLE DATABASE 文で STORAGE 句を使用して、PDB のデータファイルが使用する記憶域の量を制限することもできます。	「データベース・レベルでの PDB の変更」 データファイルおよび一時ファイルの管理の詳細は、『 Oracle Database 管理者ガイド 』を参照してください
スキーマ・オブジェクトの管理	非 CDB で行うのと同じ方法で、PDB 内のスキーマ・オブジェクトを作成、変更および削除できます。特定の PDB で起動するトリガーを作成することもできます。 CDB のデータベース・リンクを管理する場合、ルートには一意のグローバル・データベース名があり、各 PDB も同様です。ルートのグローバル名は、DB_NAME および DB_DOMAIN 初期化パラメータで定義されます。PDB のグローバル・データベース名は、PDB 名および DB_DOMAIN 初期化パラメータによって定義されます。各 PDB のグローバル・データベース名はドメイン内で一意である必要があります。	スキーマ・オブジェクトの詳細は、『 Oracle Database 管理者ガイド 』を参照してください Oracle Database 管理者ガイド CDB でのトリガーの作成の詳細は、『 Oracle Database PL/SQL 言語リファレンス 』を参照してください

親トピック: [PDB管理について](#)

CDBに固有のタスク

現在のコンテナがPDBである場合に、いくつかの管理タスクを実行できません。

次のタスクは、現在のコンテナがルートである場合にCDB全体またはCDBルートで共通ユーザーが実行します。

- CDBインスタンスの起動および停止
- ALTER DATABASE文を使用したCDBまたはルートの変更
- ALTER SYSTEM文を使用したCDBまたはルートの変更
- CDBまたはルートでのデータ定義言語(DDL)文の実行
- 次のコンポーネントの管理:
 - プロセス
 - メモリー
 - エラーおよびアラート
 - 診断データ
 - 制御ファイル
 - オンラインREDOログ・ファイルおよびアーカイブREDOログ・ファイル
 - UNDO
- PDBの作成、接続、切断および削除

現在のコンテナがルートである共通ユーザーは、1つ以上のPDBのオープン・モードを変更することもできます。同様に、現在のコンテナがPDBである共通ユーザーまたはローカル・ユーザーは、現在のPDBのオープン・モードを変更できます。

関連項目:

- [「現在のコンテナについて」](#)
- このタスクおよびCDBまたはルートの管理に関連するその他のタスクの詳細は、[「CDBの管理」](#)を参照してください

親トピック: [PDB管理について](#)

PDBへの接続の管理

PDBの接続は非CDBと同じ方法で管理しますが、特別な考慮事項がいくつかあります。

- [PDBへの接続](#)
いくつかの手法を使用して、SQL*Plus CONNECTコマンドでPDBに接続できます。
- [PDBのサービスの管理](#)
PDBのサービスを作成、変更または削除できます。
- [参照先PDBのリスナー設定の変更](#)
プロキシPDBによって参照されているPDBは参照先PDBと呼ばれます。

親トピック: [PDBの管理](#)

PDBへの接続

いくつかの手法を使用して、SQL*Plus CONNECTコマンドでPDBに接続できます。

この項では、SQL*Plusで非CDBに接続する方法を理解していることを想定しています。

次の手法を使用して、SQL*Plus CONNECTコマンドでPDBに接続できます。

- オペレーティング・システム認証を使用したローカル接続
- 簡易接続を使用したデータベース接続
- サービス名を使用したデータベース接続

前提条件

次の前提条件を満たしている必要があります。

- PDBに接続するユーザーは、そのPDBでCREATE SESSION権限を付与されている必要があります。
- SYSDBA、SYSOPER、SYSBACKUPまたはSYSDGの管理権限を持たないユーザーとしてPDBに接続するには、PDBをオープンにしている必要があります。

ノート:



この項では、PDBに接続するユーザーはローカル・ユーザー・アカウントを使用することを想定しています。共通ユーザーとしてPDBに接続することも可能で、共通ユーザーとしてルートに接続してからPDBに切り替えることができます。

SQL*Plus CONNECTコマンドを使用してPDBに接続するには:

1. SQL*Plusを起動できるよう、環境を構成します。
2. /NOLOG引数を使用してSQL*Plusを起動します。

```
sqlplus /nolog
```

3. 簡易接続またはネット・サービス名を使用してCONNECTコマンドを発行し、PDBに接続します。

PDBに接続するには、PDBプロパティを指定してサービスに接続します。

例15-1 PDBのネット・サービス名を使用したSQL*PlusでのPDBへの接続

次のコマンドでは、hrappサービスを使用してhrユーザーに接続します。hrappサービスには、hrpdb PDBのPDBプロパティが含まれています。この例では、hrappサービスのネット・サービス名を持つようにクライアントが設定されていると想定しています。

```
CONNECT hr@hrapp
```

関連項目:

- PDBのオープン・モードの変更の詳細は、[「PDBのオープン・モードの変更」](#)および[「データベース・レベルでのPDBの変更」](#)を参照してください。
- 共通ユーザーとしてPDBに接続する方法の詳細は、[「CDB内のコンテナへのアクセスについて」](#)を参照してください
- [「PDBのサービスの管理」](#)
- SQL*Plusを使用してデータベースに接続する方法の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [PDBへの接続の管理](#)

PDBのサービスの管理

PDBのサービスを作成、変更または削除できます。

- [PDBのサービスについて](#)
各PDBにはデフォルトのサービスがありますが、SRVCTLまたはDBMS_SERVICEを使用して独自のサービスを作成できません。
- [SRVCTLおよびDBMS_SERVICEを使用したPDBのサービスの管理](#)
PDBプロパティを指定して、サービスを作成、変更または削除できます。

関連項目:

[Oracle Database管理者ガイド](#)

親トピック: [PDBへの接続の管理](#)

PDBのサービスについて

各PDBにはデフォルトのサービスがありますが、SRVCTLまたはDBMS_SERVICEを使用して独自のサービスを作成できます。

- [PDBプロパティ](#)
PDBプロパティによって、PDBにサービスが関連付けられます。PDBプロパティを使用してクライアントがサービスに接続する場合、接続の現在のコンテナはPDBになります。
- [デフォルトおよびユーザー定義のサービス](#)
PDBを作成すると、PDBの新しいデフォルト・サービスが自動的に作成されます。
- [サービスを管理するためのツール](#)
サービスの作成および変更にはSRVCTLユーティリティを使用することをお勧めします。または、DBMS_SERVICEパッケージも使用できます。

親トピック: [PDBのサービスの管理](#)

PDBプロパティ

PDBプロパティによって、PDBにサービスが関連付けられます。PDBプロパティを使用してクライアントがサービスに接続する場合、接続の現在のコンテナはPDBになります。

PDBプロパティは、次のいずれかを行う場合にのみ必要です。

- サービスの作成
- サービスのPDBプロパティの変更

サービスを開始、停止または削除するときには、PDBプロパティを指定しません。また、サービスのPDBプロパティを変更せずにサービスを変更する場合、PDBプロパティを指定する必要はありません。

ALL_SERVICESデータ・ディクショナリ・ビューを問い合わせると、サービスのPDBプロパティを表示できます。また、SRVCTLユーティリティを使用する場合は、`srvctl config service`コマンドを使用できます。

関連項目:

[「現在のコンテナについて」](#)

親トピック: [PDBのサービスについて](#)

デフォルトおよびユーザー定義のサービス

PDBを作成すると、PDBの新しいデフォルト・サービスが自動的に作成されます。

各データベース・サービス名は、1つのCDB内で一意である必要があり、特定のリスナーからアクセスされるインスタンスを持つすべてのCDBの範囲内で一意である必要があります。デフォルト・サービスの名前はPDBと同じになります。このサービスは、ユーザーが管理できず、管理タスクにのみ使用する必要があります。

アプリケーションには常にユーザー定義のサービスを使用します。これは、ユーザー定義のサービスはアプリケーションの要件に適合するようにカスタマイズできるためです。デフォルトのPDBサービスをアプリケーションに使用しないことをお勧めします。

ノート:



サービスをプロキシ PDB と関連付けしないでください。

Oracle Clusterware環境では、PDBに作成される各サービスにOracle Clusterwareリソースを作成する必要があります。データベースがOracle RestartまたはOracle Clusterwareで管理され、SRVCTLユーティリティを使用して、クローズするPDBのPDBプロパティを指定してサービスを開始する場合、PDBは、サービスが開始されたノードにおいて読取り/書込みモードでオープンされます。ただし、PDBサービスを停止しても、PDBのオープン・モードは変更されません。

PDBを切断または削除した場合、切断または削除されたPDBのサービスは自動的に削除されません。これらのサービスは、手動で削除する必要があります。

関連項目:

- PDBのオープン・モードの変更の詳細は、[「ALTER PLUGGABLE DATABASE文を使用したPDBの変更」](#)を参照してください
- [「アプリケーション・ルート・レプリカを参照するプロキシPDBの作成」](#)

親トピック: [PDBのサービスについて](#)

サービスを管理するためのツール

サービスの作成および変更にはSRVCTLユーティリティを使用することをお勧めします。または、DBMS_SERVICEパッケージも使用できます。

SRVCTL

単一インスタンス・データベースがOracle Restartで管理されている場合、またはOracle RACデータベースがOracle Clusterwareで管理されている場合は、サーバー制御(SRVCTL)ユーティリティを使用して、サービスを作成、変更または削除します。

SRVCTLユーティリティを使用してPDBのサービスを作成する場合は、add serviceコマンドを使用し、-pdbパラメータでPDBを指定します。-pdbを指定しない場合、そのサービスはルートに関連付けられます。

SRVCTLユーティリティを使用してサービスのPDBプロパティを変更する場合は、modify serviceコマンドを使用し、-pdbパラ

メータでPDBを指定します。SRVCTLユーティリティを使用してPDBのサービスを削除するには、`remove service`コマンドを使用します。

-pdbパラメータが含まれていない場合でも、`start service`、`stop service`、`relocate service`コマンドなどの他のSRVCTLコマンドを使用してサービスを管理できます。

SRVCTLユーティリティを使用してサービスを作成または変更する場合、PDB名は検証されません。ただし、無効なPDB名を使用してサービスを開始しようとすると、エラーが発生します。

DBMS_SERVICE

データベースがOracle RestartまたはOracle Clusterwareで管理されていない場合は、DBMS_SERVICEパッケージを使用して、データベース・サービスを作成または削除します。

DBMS_SERVICEは、ルート・レベルで各PDBに存在します。各レベルのSYSによって所有され、実行されます。PDB管理者は、別のPDBによって所有されているサービスの接続を停止、再配置またはテストできません。

DBMS_SERVICEパッケージを使用してサービスを作成する場合、サービスのPDBプロパティは現在のコンテナに設定されます。したがって、特定のPDBに設定されたPDBプロパティを持つサービスをDBMS_SERVICEパッケージを使用して作成するには、PDBが現在のコンテナであるときにCREATE_SERVICEプロシージャを実行します。現在のコンテナがルートの場合にCREATE_SERVICEプロシージャを使用してサービスを作成すると、そのサービスはルートと関連付けられます。

DBMS_SERVICEパッケージを使用してサービスのPDBプロパティは変更できません。ただし、1つのPDBからサービスを削除し、同様なサービスを別のPDBに作成できます。この場合、新しいサービスのPDBプロパティは、それが作成されたPDBのものになります。

START_SERVICEプロシージャやSTOP_SERVICEプロシージャなど、他のDBMS_SERVICEサブプログラムを使用してサービスを管理することもできます。DBMS_SERVICE.*_CONNECTION_TESTプロシージャを使用して、計画メンテナンス中にデータベース接続の状態をチェックできます。DELETE_SERVICEプロシージャを使用してサービスを削除します。

関連項目:

- [「例19-9」](#)
- Oracleデータベースの自動再起動の構成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- DBMS_SERVICEパッケージの詳細は、[『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)を参照してください
- Oracle Real Application Clusters (Oracle RAC)環境でサービスを作成する方法の詳細は、[『Oracle Real Application Clusters管理およびデプロイメント・ガイド』](#)を参照

親トピック: [PDBのサービスについて](#)

SRVCTLおよびDBMS_SERVICEを使用したPDBのサービスの管理

PDBプロパティを使用してサービスを作成、変更または削除できます。

SRVCTLユーティリティを使用し、PDBプロパティを使用してサービスを管理するには:

1. 正しいユーザー・アカウントを使用して、ホスト・コンピュータにログインします。
2. 正しいOracleホームからSRVCTLを実行していることを確認します。
3. 次のどちらかの操作を実行します。

- サービスを作成または変更するには、`add service`コマンドを実行し、-pdbパラメータでPDBを指定します。

- サービスのPDBプロパティを変更する場合は、`modify service`コマンドを実行し、`-pdb`パラメータでPDBを指定します。
- サービスを削除するには、`remove service`コマンドを実行します。

DBMS_SERVICEパッケージを使用してPDBのサービスを作成または削除するには:

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。
[「PDBへの接続」](#)を参照してください。
2. DBMS_SERVICEパッケージ内の適切なサブプログラムを実行します。

ノート:



データベースが Oracle Restart または Oracle Clusterware で管理されている場合は、SRVCTL ユーティリティを使用してサービスを管理します。DBMS_SERVICE パッケージは使用しないでください。

例15-2 SRVCTLユーティリティを使用したPDBのサービスの作成

この例では、DB_UNIQUE_NAME mycdbで指定されたCDB内のPDB salespdbにsalesrepサービスを追加します。

```
srvctl add service -db mycdb -service salesrep -pdb salespdb
```

例15-3 SRVCTLユーティリティを使用したサービスのPDBプロパティの変更

この例では、DB_UNIQUE_NAME mycdbで指定されたCDBでsalesrepサービスを変更して、このサービスをhrpdb PDBに関連付けます。

```
srvctl modify service
  -db mycdb
  -service salesrep
  -pdb hrpdb
```

例15-4 SRVCTLユーティリティを使用したOracle RACでのサービスの再配置

`relocate service`コマンドを使用して、サービスが現在実行中のOracle RACインスタンスから、実行可能な別のインスタンスにサービスを再配置できます。この手法は、管理者管理データベースのサービス、およびポリシー管理データベースのシングルトン・サービスの両方に適用されます。

次のコマンドは、現在実行中のOracle RACインスタンス`cdb_inst1`から、現在実行されていないインスタンス`cdb_inst2`に、サービス`svc1`を再配置します。

```
srvctlrelocateservice
  dbcdb
  servicesvc1
  oldinstcdb_inst1
  newinstcdb_inst2
  -drain_timeout NNN
  -stopoption immediate
```

次のコマンドは、ポリシー管理データベースの場合と同じ操作を実行します。

```
srvctlrelocateservice
  dbcdb
  servicesvc1
  currentnodecdb_inst1
  targetnode cdb_inst2
```

```
-drain_timeout NNN
-stopoption immediate
```

例15-5 SRVCTLユーティリティを使用したサービスの削除

この例では、DB_UNIQUE_NAME mycdbで指定されたCDBでsalesrepサービスを削除します。

```
srvctl remove service
  -db mycdb
  -service salesrep
```

例15-6 DBMS_SERVICEパッケージを使用したPDBのサービスの作成

この例では、現在のPDBにsalesrepサービスを作成します。

```
BEGIN
  DBMS_SERVICE.CREATE_SERVICE (
    service_name => 'salesrep',
    network_name => 'salesrep.example.com');
END;
/
```

サービスのPDBプロパティが現在のコンテナに設定されます。たとえば、現在のコンテナがsalespdb PDBの場合、サービスのPDBプロパティはsalespdbになります。

例15-7 DBMS_SERVICEパッケージを使用したサービスの削除

この例では、現在のPDBのsalesrepサービスを削除します。

```
BEGIN
  DBMS_SERVICE.DELETE_SERVICE (
    service_name => 'salesrep');
END;
/
```

関連項目:

- [「例19-9」](#)
- DBMS_SERVICEパッケージの詳細は、[『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)を参照してください
- Oracle Real Application Clusters (Oracle RAC)環境でサービスを管理する方法の詳細は、[『Oracle Real Application Clusters管理およびデプロイメント・ガイド』](#)を参照

親トピック: [PDBのサービスの管理](#)

参照先PDBのリスナー設定の変更

プロキシPDBによって参照されているPDBは参照先PDBと呼ばれます。

参照先PDBのリスナーのポートまたはホスト名が変更された場合は、プロキシPDBが引き続き正しく機能するように参照先PDBのリスナー設定をリセットする必要があります。

- [参照先PDBのリスナー・ホスト名の変更](#)
参照先PDBのリスナーのホスト名が変更された場合は、ALTER PLUGGABLE DATABASE CONTAINERS HOST文を実

行して、プロキシPDBが引き続き正しく機能するように参照先PDBのホスト名をリセットする必要があります。

- [参照先PDBのリスナー・ポート番号の変更](#)

参照先PDBのリスナーのポート番号が変更された場合は、ALTER PLUGGABLE DATABASE CONTAINERS PORT文を実行して、プロキシPDBが引き続き正しく機能するように参照先PDBのポート番号をリセットする必要があります。

関連トピック

- [プロキシPDBとしてのPDBの作成](#)

親トピック: [PDBへの接続の管理](#)

参照PDBのリスナー・ホスト名の変更

参照先PDBのリスナーのホスト名が変更された場合は、ALTER PLUGGABLE DATABASE CONTAINERS HOST文を実行して、プロキシPDBが引き続き正しく機能するように参照先PDBのホスト名をリセットする必要があります。

プロキシPDBは、PDBの作成中に、データベース・リンクを使用してその参照先PDBと通信を確立します。通信が確立された後、プロキシPDBはPDBの作成時に使用されたデータベース・リンクを使用せずに参照先PDBと直接通信し、データベース・リンクを削除できます。参照先PDBのリスナー・ホスト名が変更された場合、各プロキシPDBは参照先PDBとの通信を再確立する必要があります。

Oracle Database 19cバージョン19.10以降では、ALTER PLUGGABLE DATABASE CONTAINERS HOSTコマンドを、CDBルート、アプリケーション・ルートまたはPDBで、PDB名を含めることにより実行できます。

現行ユーザーは、ALTER DATABASEシステム権限を持っている必要があり、その権限は共通に付与されているか、またはPDBでローカルに付与されている必要があります。

1. SQL*Plusで、現在のコンテナが参照先PDBであることを確認します。
[「PDBへの接続」](#)を参照してください。
2. ALTER PLUGGABLE DATABASE CONTAINERS HOST文を実行して新しいホスト名を指定するか、RESETキーワードを指定してホスト名を参照先PDBのホスト名であるデフォルト設定に戻します。
3. 参照先PDBを参照するプロキシPDBを削除してから再作成し、各プロキシPDBと参照先PDBとの通信を再確立します。

例15-8 参照先PDBのリスナー・ホスト名の変更

この例では、参照先PDBのホスト名をmyhost.example.comに変更しています。

```
ALTER PLUGGABLE DATABASE CONTAINERS HOST='myhost.example.com';
```

例15-9 リスナー・ホスト名のデフォルト値へのリセット

この例は、参照先PDBのホスト名をデフォルト値にリセットしています。デフォルト値は参照先PDBのホスト名です。

```
ALTER PLUGGABLE DATABASE CONTAINERS HOST RESET;
```

関連項目:

- [「プロキシPDBとしてのPDBの作成」](#)
- [「HOST句」](#)

例15-10 リスナー・ホスト名の変更時におけるPDB名の使用

この例では、PDB01という名前のPDBのホスト名をmyhost.example.comに変更します。

```
ALTER PLUGGABLE DATABASE PDB01 CONTAINERS HOST='myhost.example.com' ;
```

親トピック: [参照先PDBのリスナー設定の変更](#)

参照先PDBのリスナー・ポート番号の変更

参照先PDBのリスナーのポート番号が変更された場合は、ALTER PLUGGABLE DATABASE CONTAINERS PORT文を実行して、プロキシPDBが引き続き正しく機能するように参照先PDBのポート番号をリセットする必要があります。

プロキシPDBは、PDBの作成中に、データベース・リンクを使用してその参照先PDBと通信を確立します。通信が確立された後、プロキシPDBはPDBの作成時に使用されたデータベース・リンクを使用せずに参照先PDBと直接通信し、データベース・リンクを削除できます。参照先PDBのリスナーのポート番号が変更された場合、各プロキシPDBは参照先PDBと通信を再確立する必要があります。

Oracle Database 19cバージョン19.10以降では、ALTER PLUGGABLE DATABASE CONTAINERS PORTコマンドを、CDBルート、アプリケーション・ルートまたはPDBで、PDB名を含めることにより実行できます。

現行ユーザーは、ALTER DATABASEシステム権限を持っている必要があります、その権限は共通に付与されているか、またはPDBでローカルに付与されている必要があります。

1. SQL*Plusで、現在のコンテナが参照先PDBであることを確認します。
2. ALTER PLUGGABLE DATABASE CONTAINERS PORT文を実行して新しいポート番号を指定するか、RESETキーワードを指定してポート番号をデフォルト設定(1521)に戻します。
3. 参照先PDBを参照するプロキシPDBを削除して再作成し、各プロキシPDBと参照先PDBの通信を再確立します。

例15-11 参照先PDBのリスナー・ポート番号の変更

この例では、参照先PDBのポート番号を1543に変更しています。

```
ALTER PLUGGABLE DATABASE CONTAINERS PORT=1543;
```

例15-12 リスナー・ポート番号のデフォルト値へのリセット

この例は、参照先PDBのポート番号をデフォルト値にリセットしています。ポート番号のデフォルト値は1521です。

```
ALTER PLUGGABLE DATABASE CONTAINERS PORT RESET;
```

例15-13 ポート番号の変更時におけるPDB名の使用

この例では、PDB01という名前のPDBのポート番号を1543に変更します。

```
ALTER PLUGGABLE DATABASE PDB01 CONTAINERS PORT=1543;
```

関連トピック

- [PDBへの接続](#)
- [プロキシPDBとしてのPDBの作成](#)
- [PORT句](#)

親トピック: [参照先PDBのリスナー設定の変更](#)

システム・レベルでのPDBの変更

ALTER SYSTEM文を使用してPDBを変更できます。

- [PDBのシステム・レベルの変更について](#)
ALTER SYSTEM文によって、PDBを動的に変更できます。PDBが動作する方法を変更する場合は、ALTER SYSTEM

文を発行できます。

- [ALTER SYSTEMを使用したPDBの変更](#)

システム・レベルでPDBを変更するには、ALTER SYSTEM文を使用します(非CDBと同様)。

親トピック: [PDBの管理](#)

PDBのシステム・レベルの変更について

ALTER SYSTEM文によって、PDBを動的に変更できます。PDBが動作する方法を変更する場合は、ALTER SYSTEM文を発行できます。

現在のコンテナがPDBである場合、次のALTER SYSTEM文を実行できます。

- ALTER SYSTEM FLUSH { SHARED_POOL | BUFFER_CACHE | FLASH_CACHE }
- ALTER SYSTEM { ENABLE | DISABLE } RESTRICTED SESSION
- ALTER SYSTEM SET USE_STORED_OUTLINES
- ALTER SYSTEM { SUSPEND | RESUME }
- ALTER SYSTEM CHECKPOINT
- ALTER SYSTEM CHECK DATAFILES
- ALTER SYSTEM REGISTER
- ALTER SYSTEM { KILL | DISCONNECT } SESSION
- ALTER SYSTEM SET initialization_parameter (初期化パラメータのサブセットの場合)

その他のすべてのALTER SYSTEM文は、CDB全体に影響し、ルートの共通ユーザーによって実行される必要があります。

ALTER SYSTEM SET initialization_parameter文では、PDBの一部の初期化パラメータのみを変更できます。すべての初期化パラメータは、ルートに対して設定できます。初期化パラメータがPDB用に明示的に設定されていない場合、PDBではルートからパラメータ値を継承します。

V\$SYSTEM_PARAMETERビューのパラメータのISPDB_MODIFIABLE列がTRUEの場合は、PDBの初期化パラメータを変更できます。次の問合せにより、PDBで変更可能なすべての初期化パラメータがリストされます。

```
SELECT NAME
FROM V$SYSTEM_PARAMETER
WHERE ISPDB_MODIFIABLE=' TRUE'
ORDER BY NAME;
```

現在のコンテナがPDBである場合、ALTER SYSTEM SET initialization_parameter文を実行して、PDBを変更します。この文は、ルートまたは他のPDBには影響しません。次の表は、サーバー・パラメータ・ファイル(SPFIL)を使用し、ALTER SYSTEM SET文をPDBで実行した場合のSCOPE句の動作について説明しています。

SCOPE設定	動作
MEMORY	初期化パラメータ設定は、メモリーで変更され、PDB で即時に有効になります。新しい設定は、PDB へのみ影響を及ぼします。 この設定によって、次のいずれかの場合、ルートで設定された値に戻ります。 <ul style="list-style-type: none">● ALTER SYSTEM SET 文によって、SCOPE が BOTH または MEMORY に等しい場合はルートのパラメータの値が設定され、PDB がクローズして再オープンされます。SCOPE が SPFILE と等しい場

SCOPE設定	動作
	<p>合、PDB をクローズして再オープンしても PDB のパラメータ値は変更されません。</p> <ul style="list-style-type: none"> ● PDB が停止し、再オープンされます。 ● CDB が停止し、再オープンされます。
SPFILE	<p>初期化パラメータ設定は、PDB に対して変更されて永続的に格納されます。新しい設定は、次の場合に有効になります。</p> <ul style="list-style-type: none"> ● PDB が停止し、再オープンされます。 ● CDB が停止し、再オープンされます。 <p>これらの場合、新しい設定は PDB にのみ影響を及ぼします。</p>
BOTH	<p>初期化パラメータ設定は、メモリーで変更され、PDB に対して変更されて永続的に格納されます。新しい設定は PDB で即時に有効になり、PDB が停止して再オープンされた後、または CDB が停止して再オープンされた後も保持されます。新しい設定は、PDB にのみ影響を及ぼします。</p>

PDBがCDBから切断されると、SCOPE=BOTHまたはSCOPE=SPFILEを使用してPDBに指定された初期化パラメータの値は、PDBのXMLメタデータ・ファイルに追加されます。これらの値は、PDBがCDBに接続したときにPDBでリストアされます。

ノート:



テキスト初期化パラメータ・ファイル(PFILE)には、PDB 固有のパラメータ値を含めることはできません。

関連項目:

- [「CDBからのPDBの切断」](#)
- [「現在のコンテナについて」](#)
- [「ALTER SYSTEMを使用したCDBの変更」](#)
- [Oracle Database SQL言語リファレンス](#)

親トピック: [システム・レベルでのPDBの変更](#)

ALTER SYSTEMを使用したPDBの変更

システム・レベルでPDBを変更するには、ALTER SYSTEM文を使用します(非CDBと同様)。

前提条件

現行ユーザーは次の権限を付与されている必要があり、権限は共通に付与されているか、またはPDBでローカルに付与されて

いる必要があります。

- CREATE SESSION
- ALTER SYSTEM

ALTER SYSTEMを使用してPDBを変更するには:

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。

[「PDBへの接続」](#)を参照してください。

2. ALTER SYSTEM文を実行します。

例15-14 PDBでのセッション制限の有効化

PDBでセッションを制限するには、次の文を発行します。

```
ALTER SYSTEM ENABLE RESTRICTED SESSION;
```

例15-15 PDBの統計収集レベルの変更

次のALTER SYSTEM文によって、現在のPDBのSTATISTICS_LEVEL初期化パラメータがALLに設定されます。

```
ALTER SYSTEM SET STATISTICS_LEVEL = ALL SCOPE = MEMORY;
```

関連項目:

- [「ALTER SYSTEMを使用したCDBの変更」](#)
- [Oracle Database SQL言語リファレンス](#)

親トピック: [システム・レベルでのPDBの変更](#)

データベース・レベルでのPDBの変更

ALTER PLUGGABLE DATABASE文を使用してPDBを変更できます。

- [PDBのデータベース・レベルの変更について](#)
PDBのALTER PLUGGABLE DATABASEは、非CDBのALTER DATABASEに似ています。
- [ALTER PLUGGABLE DATABASE文を使用したPDBの変更](#)
単一のPDBの属性を変更するには、ALTER PLUGGABLE DATABASE文を使用します。
- [PDBのグローバル・データベース名の変更](#)
ALTER PLUGGABLE DATABASE RENAME GLOBAL_NAME TO文を使用して、PDBのグローバル・データベース名を変更できます。
- [リフレッシュ可能なクローンPDBの管理](#)
リフレッシュ可能なクローンPDBは、ソースPDBと定期的に同期できる読取り専用クローンです。
- [PDBのオープン・モードの変更](#)
PDBのオープン・モードは、ALTER PLUGGABLE DATABASEのSQL文またはSQL*PlusのSTARTUPコマンドを使用して変更できます。

親トピック: [PDBの管理](#)

PDBのデータベース・レベルの変更について

PDBのALTER PLUGGABLE DATABASEは、非CDBのALTER DATABASEに似ています。

ノート:



ALTER PLUGGABLE DATABASE 文でサポートされている句が含まれ、現在のコンテナが PDB である場合に発行された ALTER DATABASE 文は、対応する ALTER PLUGGABLE DATABASE 文と同じ効果があります。ただし、これらの文には、pdb_storage_clause、pdb_change_state_clause、logging_clause、pdb_recovery_clause など、PDB に固有の句を含めることはできません。

- [記憶域の句](#)
ALTER PLUGGABLE DATABASEを使用して、PDBレベルで記憶域を構成します。
- [ロギングおよびリカバリの句](#)
ALTER PLUGGABLE DATABASEを使用して、PDBレベルでロギング、リカバリおよびリカバリ・モードを設定します。
- [その他の句](#)
ALTER PLUGGABLE DATABASEを使用して、オープン・モード、グローバル名、タイム・ゾーンおよびデフォルト・エディションを変更できます。

親トピック: [データベース・レベルでのPDBの変更](#)

記憶域の句

ALTER PLUGGABLE DATABASEを使用して、PDBレベルで記憶域を構成します。

ALTER PLUGGABLE DATABASEの次の句は、PDB記憶域を変更します。

- database_file_clauses
この句は、ALTER DATABASE文での場合と同様に機能しますが、文は現在のPDBに適用されます。
- DEFAULT TABLESPACE句
現在のコンテナがPDBである間に作成されたユーザーの場合、この句では、デフォルト表領域がCREATE USER文で指定されていない場合に、ユーザーのデフォルト表領域を指定します。
- DEFAULT TEMPORARY TABLESPACE句
現在のコンテナがPDBである間に作成されたユーザーの場合、この句では、デフォルト一時表領域がCREATE USER文で指定されていない場合に、ユーザーのデフォルト一時表領域を指定します。
- SET DEFAULT { BIGFILE | SMALLFILE } TABLESPACE句
この句によって、PDBで以降に作成される表領域のデフォルト・タイプがbigfileまたはsmallfileに変更されます。この句は、ALTER DATABASE文での場合と同様に機能しますが、現在のPDBに適用されます。
- pdb_storage_clause
この句によって、PDBに属するすべての表領域が使用する記憶域の量に対する制限が設定されます。この制限は、PDBに属している表領域を構成するすべてのデータファイルおよび一時ファイルの合計サイズに適用されます。
この句では、PDB内の統合監査OSスピルオーバー(.bin形式)ファイルに使用できるストレージ量の制限を設定することもできます。この制限に達した場合は、これらのファイルにストレージを追加できません。

この句では、PDBに接続しているセッションが使用できる共有一時表領域の記憶域の量に対する制限も設定できます。制限に達すると、PDBに接続しているセッションは、共有一時表領域で追加の記憶域を使用できなくなります。

親トピック: [PDBのデータベース・レベルの変更について](#)

ロギングおよびリカバリの句

ALTER PLUGGABLE DATABASEを使用して、PDBレベルでロギング、リカバリおよびリカバリ・モードを設定します。

logging_clause

ノート:



この句は、Oracle Database 12c リリース 1 (12.1.0.2)以上で使用できます。

この句では、PDBのロギング属性を指定します。ロギング属性は、特定のDML操作をREDOログ・ファイルに記録する (LOGGING)か記録しない(NOLOGGING)かを制御します。

この句を使用して、次の属性のいずれかを指定できます。

- LOGGINGは、PDB内で今後作成される表領域がいずれも、デフォルトでLOGGING属性を使用して作成されることを示します。このデフォルトのロギング属性は、CREATE TABLE文などのスキーマ・オブジェクト・レベルでNOLOGGINGを指定することで上書きできます。
- NOLOGGINGは、PDB内で今後作成される表領域がいずれも、デフォルトでNOLOGGING属性を使用して作成されることを示します。このデフォルトのロギング属性は、CREATE TABLE文などのスキーマ・オブジェクト・レベルでLOGGINGを指定することで上書きできます。

指定の属性は、CREATE TABLESPACE文にlogging_clauseが指定されていない場合に、PDB内で作成される表領域のロギング属性の設定に使用されます。

DBA_PDBSビューには、PDBの現在のロギング属性が表示されます。

ノート:



この句を使用するには、PDB を制限モードでオープンする必要があります。

pdb_force_logging_clause

ノート:



この句は、Oracle Database 12c リリース 1 (12.1.0.2)以上で使用できます。

この句は、PDBを強制ロギング・モードまたは強制ロギングなしモードにするか、PDBの強制ロギングモードまたは強制ロギングなしモードを解除します。

この句を使用して、次の属性のいずれかを指定できます。

- ENABLE FORCE LOGGINGはPDBを強制ロギング・モードにし、これにより、一時表領域および一時セグメントでの変更を除いたPDBでの変更はすべて記録されます。強制ロギング・モードは、スキーマ・オブジェクト・レベルで上書きできません。

PDBレベルの強制ロギング・モードは、PDBの各表領域で指定するNOLOGGINGまたはFORCE LOGGING設定、およびPDBの各データベース・オブジェクトで指定するNOLOGGING設定より優先され、これらの設定には影響されません。

ENABLE FORCE LOGGINGは、PDBが強制ロギングなしモードである場合、指定できません。DISABLE FORCE NOLOGGINGは、最初に指定する必要があります。

- DISABLE FORCE LOGGINGは、現在強制ロギング・モードであるPDBのそのモードを解除します。PDBが現在強制ロギング・モードでない場合、DISABLE FORCE LOGGINGを指定するとエラーになります。
- ENABLE FORCE NOLOGGINGはPDBを強制ロギングなしモードにし、これにより、PDBでの変更は記録されません。強制ロギングなしモードは、スキーマ・オブジェクト・レベルで上書きできません。

CDB全体の強制ロギング・モードは、PDBレベルの強制ロギングなしモードより優先されます。PDBレベルの強制ロギングなしモードは、PDBの各表領域で指定するLOGGINGまたはFORCE LOGGING設定、およびPDBの各データベース・オブジェクトで指定するLOGGING設定より優先され、これらの設定には影響されません。

ENABLE FORCE NOLOGGINGは、PDBが強制ロギング・モードである場合、指定できません。DISABLE FORCE LOGGINGは、最初に指定する必要があります。

- DISABLE FORCE NOLOGGINGは、現在強制ロギングなしモードであるPDBのそのモードを解除します。PDBが現在強制ロギングなしモードでない場合、DISABLE FORCE NOLOGGINGを指定するとエラーになります。

DBA_PDBSビューには、PDBが強制ロギング・モードであるか、強制ロギングなしモードであるかが表示されます。

ノート:



この句を使用するには、PDBを制限モードでオープンする必要があります。

pdb_recovery_clause

ノート:



この句は、Oracle Database 12c リリース 1 (12.1.0.2)以上で使用できます。

ALTER PLUGGABLE DATABASE DISABLE RECOVERYは、PDBに属するデータファイルをオフラインにし、PDBのリカバリを無効にします。PDBデータファイルは、リカバリが再度有効になるまで、リカバリ・セッションに含まれません。リカバリ中に作成された新しいデータファイルはいずれも無効になり、PDBの無名ファイルとして作成されます。

ALTER PLUGGABLE DATABASE ENABLE RECOVERYは、PDBに属するデータファイルをオンラインにし、PDBをアクティブ・リカバリとマークします。このようなファイルはリカバリ・セッションに含まれます。

PDBのリカバリ・ステータスを確認するには、V\$PDBSビューのRECOVERY_STATUS列を問い合わせます。

関連項目:

- `pdb_recovery_clause`の詳細は、[『Oracle Data Guard概要および管理』](#)を参照してください。
- REDOレコードの書込みの管理の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- ロギング属性の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください

親トピック: [PDBのデータベース・レベルの変更について](#)

その他の句

ALTER PLUGGABLE DATABASEを使用して、オープン・モード、グローバル名、タイム・ゾーンおよびデフォルト・エディションを変更できます。

現在のコンテナがPDBである場合、次のいずれかの句が含まれたALTER PLUGGABLE DATABASE文によりPDBが変更されます。

- `pdb_change_state_clause`

この句によって、現在のPDBのオープン・モードが変更されます。

オプションのRESTRICTEDキーワードを指定すると、PDBへのアクセスはPDB内でRESTRICTED SESSION権限を持つユーザーに限られます。

この句でFORCEを指定すると、ALTER PLUGGABLE DATABASE文のセマンティクスが変わるため、現在閉じているPDBをオープンする以外に、すでに開いているPDBのオープン・モードを変更するために使用できます。

- `RENAME GLOBAL_NAME`句

この句によって、PDBの一意のグローバル・データベース名が変更されます。新しいグローバル・データベース名は、CDB内のいずれのコンテナ名とも異なる必要があります。PDBのグローバル・データベース名を変更すると、そのPDB名は、グローバル・データベース名の、最初のピリオドの前の部分の名前に変更されます。

グローバル・データベース名を変更する場合は、PDBへの接続に使用するデータベース・サービスのPDBプロパティを変更する必要があります。

- `set_time_zone_clause`

この句は、ALTER DATABASE文での場合と同様に機能しますが、現在のPDBに適用されます。

- `DEFAULT EDITION`句

この句は、ALTER DATABASE文での場合と同様に機能しますが、現在のPDBに適用されます。各PDBはエディションに基づく再定義を使用でき、あるPDBのエディションは他のPDBのエディションに影響しません。各PDBに独自のアプリケーションがあるマルチテナント環境では、個別のアプリケーションごとに、エディションに基づく再定義を独立して使用できます。

関連項目:

- [「PDBのサービスの管理」](#)
- [「ALTER PLUGGABLE DATABASEを使用したPDBのオープン・モードの変更」](#)

親トピック: [PDBのデータベース・レベルの変更について](#)

ALTER PLUGGABLE DATABASE文を使用したPDBの変更

単一のPDBの属性を変更するには、ALTER PLUGGABLE DATABASE文を使用します。

現在のコンテナがPDBである場合は、ALTER PLUGGABLE DATABASE文でPDBを変更します。この変更により、PDBのルートデフォルト設定が上書きされます。この変更は、CDBルートまたは他のPDBには影響しません。

前提条件

次の前提条件を満たしている必要があります。

- PDBのオープン・モードをマウントからオープンに、またはオープンからマウントに変更するには、現在のユーザーにSYSDBA、SYSOPER、SYSBACKUPまたはSYSDG管理権限が必要です。権限は共通に付与されているか、またはPDBでローカルに付与されている必要があります。ユーザーは、接続時にAS sys_privilege_nameを使用して権限を行使する必要があります。
- ALTER PLUGGABLE DATABASE文を使用して実行される他のすべての操作については、現在のユーザーにALTER DATABASEシステム権限が必要であり、その権限は共通に付与されているか、またはPDBでローカルに付与されている必要があります。
- PDBをクローズするには、そのPDBがオープンされている必要があります。

ノート:



この項では、ALTER PLUGGABLE DATABASE文を使用したPDBのグローバル・データベース名の変更については説明していません。

PDBを変更するには:

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。
2. ALTER PLUGGABLE DATABASE文を実行します。

例15-16 PDBのオープン・モードの変更

ALTER PLUGGABLE DATABASE文によって、現在のPDBのオープン・モードがマウントに変更されます。

```
ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;
```

次の文では、現在のPDBのオープン・モードを読取り専用に変更します。

```
ALTER PLUGGABLE DATABASE OPEN READ ONLY;
```

FORCEキーワードを指定する場合を除き、PDBのオープン・モードを読取り専用または読取り/書込みに変更するには、マウント・モードである必要があります。

次の文では、現在のPDBのオープン・モードをマウントまたは読取り専用オープンから読取り/書込みオープンに変更します。

```
ALTER PLUGGABLE DATABASE OPEN FORCE;
```

次の文では、現在のPDBのオープン・モードをマウントから移行に変更します。

```
ALTER PLUGGABLE DATABASE OPEN UPGRADE;
```

例15-17 PDBのデータファイルのオンライン化

このALTER PLUGGABLE DATABASE文では、database_file_clauseを使用して、/u03/oracle/pdb1_01.dbfデータファイルをオンラインにします。

```
ALTER PLUGGABLE DATABASE DATAFILE '/u03/oracle/pdb1_01.dbf' ONLINE;
```

例15-18 PDBのデフォルト表領域の変更

次のALTER PLUGGABLE DATABASE文では、DEFAULT TABLESPACE句を使用して、PDBのデフォルト永続表領域をpdb1_tbsに設定します。

```
ALTER PLUGGABLE DATABASE DEFAULT TABLESPACE pdb1_tbs;
```

次のALTER PLUGGABLE DATABASE文では、DEFAULT TEMPORARY TABLESPACE句を使用して、PDBのデフォルト一時表領域をpdb1_tempに設定します。

```
ALTER PLUGGABLE DATABASE DEFAULT TEMPORARY TABLESPACE pdb1_temp;
```

ALTER PLUGGABLE DATABASE文で指定した表領域または表領域グループは、PDBに存在する必要があります。現在のコンテナがPDBであり、デフォルトの表領域または一時表領域が明示的に割り当てられていないユーザーは、PDBのデフォルトの表領域または一時表領域を使用します。

例15-19 PDBのデフォルト表領域タイプの変更

次のALTER DATABASE文では、SET DEFAULT TABLESPACE句を使用して、PDBのデフォルトの表領域タイプをbigfileに変更します。

```
ALTER PLUGGABLE DATABASE SET DEFAULT BIGFILE TABLESPACE;
```

例15-20 PDBのストレージ制限の設定

次の文では、PDBに属しているすべての表領域のストレージ制限を2GBに設定します。

```
ALTER PLUGGABLE DATABASE STORAGE (MAXSIZE 2G);
```

次の文では、PDBに属している表領域にストレージ制限がないことを指定します。

```
ALTER PLUGGABLE DATABASE STORAGE (MAXSIZE UNLIMITED);
```

次の文では、PDBに属している表領域にストレージ制限がないこと、およびPDBに接続しているセッションが使用できる共有一時表領域にストレージ制限がないことを指定します。

```
ALTER PLUGGABLE DATABASE STORAGE UNLIMITED;
```

例15-21 PDBのロギング属性の設定

PDBが制限モードでオープンしている場合、次の文では、PDBにNOLOGGING属性を指定します。

```
ALTER PLUGGABLE DATABASE NOLOGGING;
```

例15-22 PDBの強制ロギング・モードの設定

次の文では、PDBに対して強制ロギング・モードを有効にします。

```
ALTER PLUGGABLE DATABASE ENABLE FORCE LOGGING;
```

例15-23 PDBのデフォルト・エディションの設定

この例では、現在のPDBのデフォルト・エディションをPDB1E3に設定します。

```
ALTER PLUGGABLE DATABASE DEFAULT EDITION = PDB1E3;
```


関連項目:

- 単一のPDBの属性を変更する句の詳細は、[「PDBのデータベース・レベルの変更について」](#)を参照してください
- [「PDBのグローバル・データベース名の変更」](#)
- ALTER PLUGGABLE DATABASE文の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください
- エディションに基づく再定義の詳細な説明は、[『Oracle Database開発ガイド』](#)を参照してください

親トピック: [データベース・レベルでのPDBの変更](#)

PDBのグローバル・データベース名の変更

ALTER PLUGGABLE DATABASE RENAME GLOBAL_NAME TO文を使用して、PDBのグローバル・データベース名を変更できます。

PDBのグローバル・データベース名を変更した場合、新しいグローバル・データベース名は、CDB内のいずれのコンテナ名とも異なる必要があります。

前提条件

次の前提条件を満たしている必要があります。

- 現行ユーザーは、ALTER DATABASEシステム権限を持っている必要があります。その権限は共通に付与されているか、またはPDBでローカルに付与されている必要があります。
- Oracle Real Application Clusters (Oracle RAC)データベースでは、現在のインスタンスのみでPDBをオープンする必要があります。その他すべてのインスタンスでは、PDBをクローズする必要があります。
- 変更するPDBは、PDBでのRESTRICTED SESSION権限を持つユーザーのみがアクセスできるように、RESTRICTEDを指定して読取り/書込みモードで現在のインスタンス上でオープンされている必要があります。

PDBのグローバル・データベース名を変更するには:

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。
2. ALTER PLUGGABLE DATABASE RENAME GLOBAL_NAME TO文を実行します。
次の例は、PDBのグローバル・データベース名をsalespdb.example.comに変更します。

```
ALTER PLUGGABLE DATABASE RENAME GLOBAL_NAME TO salespdb.example.com;
```

3. PDBをクローズします。
4. 読取り/書込みモードでPDBをオープンします。

PDBのグローバル・データベース名を変更すると、そのPDB名は、新しいグローバル名の最初の部分(最初のピリオドの前の部分)に変更されます。また、Oracle DatabaseはPDBのデフォルトのデータベース・サービス名を自動的に変更します。Oracle Databaseは、PDB内のすべてのデータベース・サービスのPDBプロパティを、PDBの新しいグローバル名に変更します。PDBをクローズして読取り/書込みモードでオープンし、Oracle Databaseで新しいPDBのサービス名がCDBに統合されるようにする必要があります。

データベース・サービスにアクセスするクライアントで、Oracle Net Serviceが正しく構成されている必要があります。PDBの名前を変更したため、Oracle Net Servicesの構成の変更も必要になる場合があります。

関連項目:

- [「PDBへの接続」](#)
- PDBおよびデータベース・サービスの詳細は、[「PDBのサービスの管理」](#)を参照してください

親トピック: [データベース・レベルでのPDBの変更](#)

リフレッシュ可能なクローンPDBの管理

リフレッシュ可能なクローンPDBは、ソースPDBと定期的に同期できる読取り専用クローンです。

- [PDBのリフレッシュ](#)
リフレッシュ可能なクローンとして作成されたPDBをリフレッシュできます。
- [リフレッシュ可能なクローンPDBのスイッチオーバー](#)
ソースPDBとそのリフレッシュ可能なクローンPDBのロールを切り替えることができます。

親トピック: [データベース・レベルでのPDBの変更](#)

PDBのリフレッシュ

リフレッシュ可能なクローンとして作成されたPDBをリフレッシュできます。

PDBを手動でリフレッシュする場合、最後のリフレッシュ以降にソースPDBに対して行った変更は、リフレッシュされているPDBに伝播されます。自動リフレッシュが構成されているPDBを手動でリフレッシュできます。

前提条件

PDBをリフレッシュするには、REFRESH MODE MANUALまたはREFRESH MODE EVERY minutes句を含めてPDBがクローンとして作成されている必要があります。

1. SQL*Plusで、現在のコンテナがリフレッシュ対象のPDBであることを確認します。
2. PDBがクローズされていない場合は、PDBをクローズします。たとえば、次のSQL文を発行します。
ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;
3. 次のSQL文を発行します。
ALTER PLUGGABLE DATABASE REFRESH;

関連トピック

- [リフレッシュ可能なクローンPDBについて](#)

親トピック: [リフレッシュ可能なクローンPDBの管理](#)

リフレッシュ可能なクローンPDBのスイッチオーバー

ソースPDBとそのリフレッシュ可能なクローンPDBのロールを切り替えることができます。

次の文は、スイッチオーバーを実行します。

```
ALTER PLUGGABLE DATABASE refresh_mode FROM clonepdb@dblink SWITCHOVER;
```

refresh_modeにREFRESH MODE NONEを指定しないでください。FROM句で指定されたデータベース・リンクは、クローンPDBが存在するCDBのルートを指し示す必要があります。

スイッチオーバーが完了した後、ソースPDBはリフレッシュ可能なクローンPDBになり、READ ONLYモードでのみオープンできます。

前提条件

次の前提条件を満たしている必要があります。

- ALTER PLUGGABLE DATABASE ... SWITCHOVERを発行するときに、ソースPDBに接続している必要があります。
- ソースPDBとクローンPDBが別々のCDB内にある場合、データベース・リンクで指定されたユーザーは、ソースPDBとクローンPDBで同じ名前とパスワードを使用する必要があります。

ソースPDBおよびクローンPDBのロールを切り替えるには:

1. SQL*PlusまたはSQL Developerで、ソースPDBにログインします。
2. ALTER PLUGGABLE DATABASE refresh_mode FROM clonepdb@dblink SWITCHOVER文を実行します。
文が完了すると、現在接続されているPDBがリフレッシュ可能なクローンPDBになります。
3. 必要に応じて、クローンPDBをリフレッシュします。

```
ALTER PLUGGABLE DATABASE REFRESH;
```

例15-24 リフレッシュ可能なクローンPDBのスイッチオーバー

この例では、使用しているデータ・センターにcdb1およびcdb2という名前のCDBが含まれていることを想定しています。cdb1_pdb1という名前のPDBはcdb1内に存在します。このPDBのリフレッシュ可能なクローンをcdb2に作成し、cdb1_pdb1_refという名前を付けます。目的は、ソースPDBおよびcdb1_pdb1がクローンPDBになるように、cdb1_pdb1_refをスイッチオーバーすることです。

1. SQL*Plusで、管理者権限を持つユーザーとしてcdb1に接続し、cdb1_pdb1が読み取り/書き込みモードでオープンしていることを確認します(出力例も示します)。

```
CONNECT SYS@cdb1 AS SYSDBA
Enter password: *****
```

```
ALTER PLUGGABLE DATABASE ALL CLOSE;
ALTER PLUGGABLE DATABASE cdb1_pdb1 OPEN READ WRITE;
SHOW PDBS;
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	CDB1_PDB1	READ WRITE	NO

2. c##u1という名前の共通ユーザーを作成します(pwdはユーザー指定のパスワードに置き換えます)。

```
DROP USER c##u1 CASCADE;
CREATE USER c##u1 IDENTIFIED BY pwd;
GRANT CREATE SESSION, RESOURCE, CREATE ANY TABLE, UNLIMITED TABLESPACE TO c##u1 CONTAINER=ALL;
GRANT CREATE PLUGGABLE DATABASE TO c##u1 CONTAINER=ALL;
GRANT SYSOPER TO c##u1 CONTAINER=ALL;
```

3. コンテナをcdb1_pdb1に設定し、表t1を作成して、テスト用に使用します(出力例も示します)。

```
ALTER SESSION SET CONTAINER = cdb1_pdb1;
CREATE TABLE t1(n1 NUMBER);
INSERT INTO t1 VALUES(1);
COMMIT;
SELECT * FROM t1;
```

```
      N1
-----
      1
```

4. 管理者権限を持つユーザーとしてcdb2に接続し、c##u1という名前の共通ユーザーを作成します(pwdはユーザー指定のパスワードに置き換えます)。

```
CONNECT SYS@cdb2 AS SYSDBA
Enter password: *****

DROP USER c##u1 CASCADE;
CREATE USER c##u1 IDENTIFIED BY pwd;
GRANT CREATE SESSION, RESOURCE, CREATE ANY TABLE, UNLIMITED TABLESPACE TO c##u1 CONTAINER=ALL;
GRANT CREATE PLUGGABLE DATABASE TO c##u1 CONTAINER=ALL;
GRANT SYSOPER TO c##u1 CONTAINER=ALL;
```

これで、cdb1とcdb2の両方に、同じ名前(c##u1)とパスワードを持つ共通ユーザーが含まれます。

5. cdb1へのデータベース・リンクを作成します。

次のコマンドは、ユーザーc##u1、パスワードpwd、およびサービス名cdb1を指定します。

```
CREATE DATABASE LINK cdb1_dataLink CONNECT TO c##u1 IDENTIFIED BY pwd USING 'cdb1';
```

6. cdb1_pdb1_refという名前の手動でリフレッシュ可能なPDBを作成します。

次の文は、データベース・リンクcdb1_dataLink、およびファイル宛先/dsk1/dfを指定します。

```
CREATE PLUGGABLE DATABASE cdb1_pdb1_ref FROM cdb1_pdb1@cdb1_dataLink
  CREATE_FILE_DEST='/dsk1/df'
  REFRESH MODE MANUAL;
```

7. cdb1_pdb1_refをリフレッシュします。

```
ALTER SESSION SET CONTAINER = cdb1_pdb1_ref;
ALTER PLUGGABLE DATABASE REFRESH;
```

8. t1を問い合せて、リフレッシュ可能なクローンPDBに正しい内容が含まれていることを確認します(出力例も示します)。

```
ALTER PLUGGABLE DATABASE OPEN READ ONLY;
SELECT * FROM t1;
```

```
      N1
-----
      1
```

9. 管理者権限を持つユーザーとしてcdb1に接続し、cdb2へのデータベース・リンクを作成します。

```
CONNECT SYS@cdb1 AS SYSDBA
Enter password: *****

CREATE DATABASE LINK cdb2_dataLink CONNECT TO c##u1 IDENTIFIED BY pwd USING 'cdb2';
```

前述の文は、ユーザーc##u1、パスワードpwd、およびサービス名cdb2を指定しています。

10. コンテナをcdb1_pdb1に設定し、cdb1_pdb1_refがプライマリPDBになり、現在のPDBがクローンになるようにスイッチオーバーします。

```
ALTER SESSION SET CONTAINER = cdb1_pdb1;
ALTER PLUGGABLE DATABASE
  REFRESH MODE MANUAL
  FROM cdb1_pdb1_ref@cdb2_dataLink
  SWITCHOVER;
```

11. t1を問い合せて、リフレッシュ可能なクローンPDBになった現在のPDBに正しい内容が含まれていることを確

認めます(出力例も示します)。

```
ALTER PLUGGABLE DATABASE OPEN READ ONLY;  
SELECT * FROM t1;
```

N1
1

12. 管理者権限を持つユーザーとしてcdb2に接続し、コンテナを新しいソースPDB cdb1_pdb1_refに設定して、表t1に新規の行を挿入します(出力例も示します)。

```
CONNECT SYS@cdb2 AS SYSDBA  
Enter password: *****
```

```
ALTER SESSION SET CONTAINER = cdb1_pdb1_ref;  
SELECT * FROM t1;
```

N1
1

```
INSERT INTO t1 VALUES (2);  
COMMIT;  
SELECT * FROM t1;
```

N1
1
2

13. 管理者権限を持つユーザーとしてcdb1に接続し、コンテナをcdb1_pdb1(新しいクローン)に設定し、これをリフレッシュしてからt1を問い合わせます。

```
CONNECT SYS@cdb1 AS SYSDBA  
Enter password: *****
```

```
ALTER SESSION SET CONTAINER = cdb1_pdb1;  
ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;  
ALTER PLUGGABLE DATABASE REFRESH;  
ALTER PLUGGABLE DATABASE OPEN READ ONLY;  
SELECT * FROM t1;
```

N1
1
2

前述の出力は、クローンcdb1_pdb1はソースcdb1_pdb1_refからリフレッシュされたことを示しています。

親トピック: [リフレッシュ可能なクローンPDBの管理](#)

PDBのオープン・モードの変更

PDBのオープン・モードは、ALTER PLUGGABLE DATABASEのSQL文またはSQL*PlusのSTARTUPコマンドを使用して変更できます。

- [PDBのオープン・モードについて](#)

PDBがマウントされると、読取り/書込み、読取り専用またはMIGRATEモードでオープンできます。PDBをオープンせずに

マウントすることもできます。

- [PDBのオープン状態を変更する句](#)

現在のコンテナがCDBルートである場合にPDBのオープン・モードを変更するには、ALTER PLUGGABLE DATABASEのpdb_change_state句を指定します。

- [ALTER PLUGGABLE DATABASEを使用したPDBのオープン・モードの変更](#)

pdb_change_state句を指定したALTER PLUGGABLE DATABASE文を使用して、PDBのオープン・モードを変更できます。

- [CDB再起動時のPDBのオープン・モードの保持または破棄](#)

pdb_save_or_discard_state句を指定してALTER PLUGGABLE DATABASE SQL文を使用すると、CDBの再起動時に1つ以上のPDBのオープン・モードを保持できます。

親トピック: [データベース・レベルでのPDBの変更](#)

PDBのオープン・モードについて

PDBをマウントするときは、それを読み取り/書き込み、読み取り専用またはMIGRATEモードでオープンできます。PDBをオープンせずにマウントすることもできます。

PDBのオープン・モード

V\$PDBSビューのOPEN_MODE列を問い合わせ、PDBのオープン・モードを表示できます。

次の表では、使用可能なオープン・モードを説明します。

表15-2 PDBのマウントおよびオープン・モード

モード	説明	ノート
OPEN READ WRITE	<p>読み取り/書き込みオープン・モードの PDB では、問合せおよびユーザー・トランザクションを実行でき、ユーザーは REDO ログを生成できます。</p> <p>これは、PDB がフィジカル・スタンバイ・データベースに属している場合を除き、デフォルトのオープン・モードです。</p>	<p>オプションの RESTRICTED キーワードを指定すると、PDB へのアクセスは PDB 内で RESTRICTED SESSION 権限を持つユーザーに限られます。FORCE も指定した場合は、PDB 内の RESTRICTED SESSION 権限がない PDB に接続されているすべてのセッションが終了し、それらのトランザクションがロールバックされます。</p>
OPEN READ ONLY	<p>読み取り専用オープン・モードの PDB では、問合せは実行できますが、ユーザー変更を実行することはできません。</p> <p>これは、PDB がフィジカル・スタンバイ・データベースに属している場合の、デフォルトのオープン・モードです。</p>	<p>データベース管理者は、CDB 内の共通ユーザーおよびロールを作成、変更または削除できます。これらの変更は、PDB のオープン・モードが読み取り/書き込みモードに変更されると、CDB によって PDB に適用されます。変更が適用される前には、PDB における共通ユーザーおよびロールの記述が、CDB の残りの部分における説明と異なる場合があります。</p> <p>オプションの RESTRICTED キーワードを指定すると、PDB へのアクセスは PDB 内で RESTRICTED SESSION 権限を持つユーザーに限られます。FORCE も指定した場合、PDB の RESTRICTED SESSION 権限を持たない PDB</p>

モード	説明	ノート
		に接続されたすべてのセッションは停止され、それらのトランザクションはロールバックされます。
OPEN MIGRATE	<p>PDB が移行オープン・モードの場合は、PDB でデータベース・アップグレード・スクリプトを実行できます。</p> <p>ALTER DATABASE OPEN UPGRADE 文を実行すると、PDB がこのモードになります。</p>	<p>オプションの RESTRICTED キーワードを指定すると、PDB へのアクセスは PDB 内で RESTRICTED SESSION 権限を持つユーザーに限られます。</p>
MOUNTED	<p>PDB がマウントされている場合、オブジェクトに対する変更は許可されず、データベース管理者のみがその PDB にアクセスできます。データファイルからの読取りも、データファイルへの書込みもできません。PDB に関する情報は、メモリ・キャッシュから削除されます。PDB の一貫性バックアップがサポートされています。</p>	<p>データベース管理者は、CDB で共通ユーザーおよびロールを作成、変更または削除できます。PDB のオープン・モードが読取り/書込みモードに変更されると、CDB によってこれらの変更が PDB に適用されます。変更が適用される前には、PDB における共通ユーザーおよびロールの記述が、CDB の残りの部分における説明と異なる場合があります。</p>

自動互換性チェック

PDBをオープンすると、Oracle DatabaseによってPDBとCDBとの互換性がチェックされます。互換性違反は次のいずれかとなります。

- 警告

データベースによってアラート・ログに警告が記録され、PDBが通常どおりにオープンされます。警告メッセージは表示されません。

- エラー

PDBがオープンされるときに、データベースによって、PDBが変更されエラーがあることを示すメッセージが表示され、アラート・ログにそのエラーが記録されます。エラーの原因となった状況を修正する必要があります。エラーがある場合、PDBはオープンできますが、PDBへのアクセスは互換性違反に対処できるよう、RESTRICTED SESSION権限を持つユーザーに制限されます。PDB_PLUG_IN_VIOLATIONSビューを問い合わせることで、違反の説明を表示できます。

関連項目:

- 現在のコンテナがルートの場合に1つ以上のPDBのオープン・モードを変更するには、[「PDBのオープン・モードの変更」](#)を参照してください
- 現在のコンテナがPDBの場合にPDBのオープン・モードを変更するには、[「ALTER PLUGGABLE DATABASE文を使用したPDBの変更」](#)を参照してください
- [「CDBインスタンスの停止」](#)
- PDBのその他の属性を変更するには、[「データベース・レベルでのPDBの変更」](#)を参照してください

親トピック: [PDBのオープン・モードの変更](#)

PDBのオープン状態を変更する句

現在のコンテナがCDBルートである場合にPDBのオープン・モードを変更するには、ALTER PLUGGABLE DATABASEのpdb_change_state句を指定します。

- [OPEN句とCLOSE句](#)
オープンされているPDBが、フィジカル・スタンバイ・データベースとして使用されているCDBに属する場合(この場合は、READ ONLYがデフォルト)を除き、READ WRITEがALTER PLUGGABLE DATABASE OPENのデフォルトとなります。
- [SERVICES句](#)
services句を使用して、単一PDBのオープン時に開始されるサービスを指定できます。
- [INSTANCES句](#)
Oracle RAC CDBでは、instances句を使用して、PDBを変更するインスタンスを指定できます。
- [RELOCATE句](#)
Oracle Real Application Clusters環境では、RELOCATEを使用して、異なるOracle RACインスタンス上でPDBを再オープンするようにデータベースに指示できます。

親トピック: [PDBのオープン・モードの変更](#)

OPEN句とCLOSE句

オープンされているPDBが、フィジカル・スタンバイ・データベースとして使用されているCDBに属する場合(この場合は、READ ONLYがデフォルト)を除き、READ WRITEがALTER PLUGGABLE DATABASE OPENのデフォルトとなります。

オープンまたはクローズするPDBを指定する場合には、次の操作を実行できます。

- 1つ以上のPDBをリストします。
- すべてのPDBを変更するには、ALLを指定します。
- リストされたPDBを除くすべてのPDBを変更するには、ALL EXCEPTを指定します。

次の表は、PDBのモードを変更するALTER PLUGGABLE DATABASE文の句について説明しています。

表15-3 PDBのモードを変更するALTER PLUGGABLE DATABASEの句

句	説明
OPEN READ WRITE [RESTRICTED] [FORCE]	読取り/書込みモードで PDB をオープンします。 RESTRICTED が指定されている場合、PDB にアクセスできるのは、PDB での RESTRICTED SESSION 権限を持つユーザーのみです。PDB での RESTRICTED SESSION 権限を持たない PDB に接続されたすべてのセッションは停止され、それらのトランザクションはロールバックされます。 FORCE を指定した場合、文で現在クローズされている PDB がオープンされ、オープンの読取り専用モードである PDB のオープン・モードが変更されます。
OPEN READ ONLY [RESTRICTED] [FORCE]	読取り専用モードで PDB をオープンします。 RESTRICTED が指定されている場合、PDB にアクセスできるのは、PDB での RESTRICTED SESSION 権限を持つユーザーのみです。PDB での RESTRICTED

句	説明
	<p>SESSION 権限を持たない PDB に接続されたすべてのセッションは、停止されます。</p> <p>FORCE を指定した場合、文で現在クローズされている PDB がオープンされ、オープンの読取り/書込みモードである PDB のオープン・モードが変更されます。</p>
OPEN UPGRADE [RESTRICTED]	<p>移行モードで PDB をオープンします。</p> <p>RESTRICTED が指定されている場合、PDB にアクセスできるのは、PDB での RESTRICTED SESSION 権限を持つユーザーのみです。</p>
CLOSE [IMMEDIATE ABORT]	<p>PDB をマウント・モードにします。</p> <p>CLOSE 文は、PDB での SQL*Plus SHUTDOWN コマンドに相当します。IMMEDIATE または ABORT を指定しない場合、PDB は通常モードで停止されます。</p> <p>IMMEDIATE が指定されている場合、この文は、PDB では、SQL*Plus の SHUTDOWN IMMEDIATE コマンドに相当します。</p> <p>CDB が ARCHIVELOG モードであり、ABORT が指定されている場合、PDB は強制的にクローズされます。このプロセスの間は、PDB のデータ・ファイルにチェックポイント処理またはアクセスは行われません。他のインスタンスによって PDB がオープンされている場合は、使用可能なインスタンスによって自動的にインスタンス・リカバリが実行されます。この期間に、他のインスタンス上の PDB にアクセスすると、ブラウンアウトになっていると検出される場合があります。どのインスタンスでも PDB がオープンされていない場合は、次に PDB がオープンされると、自動メディア・リカバリが行われる可能性があります。自動メディア・リカバリが失敗した場合(たとえば、ファイルにアクセスできないため)、PDB を、オープンする前に手動でリカバリする必要があります。</p> <p>PDB キーストアがオープン状態であった場合は、ALTER PLUGGABLE DATABASE CLOSE を実行してもそれはクローズされません。キーストアをクローズするには、ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE IDENTIFIED BY "pdb_ks_pwd"コマンドを実行します。</p>

親トピック: [PDBのオープン状態を変更する句](#)

SERVICES句

services句を使用して、単一PDBのオープン時に開始されるサービスを指定できます。

この句には次のようなバリエーションがあります。

- 次の形式で、services句に1つ以上のサービスをリストします。

```
SERVICES = ( 'service_name' [, 'service_name'] ... )
```

- 次の例のように、services句にALLを指定してすべてのPDBのサービスを開始します。

```
SERVICES = ALL
```

- 次の形式でservices句にALL EXCEPTを指定して、リストされているサービスを除くすべてのPDBのサービスを開始します。

```
SERVICES = ALL EXCEPT('service_name' [, 'service_name'] ... )
```

- 次の例のように、services句にNONEを指定して、PDBのデフォルト・サービスのみ開始し、他のPDBのサービスは開始しません。

```
SERVICES = NONE
```

NONEは、services句のデフォルト設定です。PDBのデフォルト・サービスは、services句の設定にかかわらず、常に開始されます。

親トピック: [PDBのオープン状態を変更する句](#)

INSTANCES句

Oracle RAC CDBでは、instances句を使用して、PDBを変更するインスタンスを指定できます。

PDBは、いくつかのインスタンスでクローズし、別のインスタンスでオープンしておくことができます。instances句には次のようなバリエーションがあります。

- 次の形式で、instances句に1つ以上のインスタンスをリストします。

```
INSTANCES = ('instance_name' [, 'instance_name'] ... )
```

- すべての実行中のインスタンスのPDBを変更するには、次の例のように、instances句にALLを指定します。

```
INSTANCES = ALL
```

- リストされたインスタンスを除くすべてのインスタンスのPDBを変更するには、次の形式で、instances句にALL EXCEPTを指定します。

```
INSTANCES = ALL EXCEPT('instance_name' [, 'instance_name'] ... )
```

親トピック: [PDBのオープン状態を変更する句](#)

RELOCATE句

Oracle Real Application Clusters環境では、RELOCATEを使用して、異なるOracle RACインスタンス上でPDBを再オープンするようにデータベースに指示できます。

次のオプションを使用できます。

- 現行インスタンスでPDBをクローズするには、デフォルトであるNORELOCATEを指定します。
- 指定したインスタンスでPDBを再オープンするには、RELOCATE TOを指定してインスタンス名を指定します。
- Oracle Databaseで選択された別のインスタンス上のPDBを再オープンするには、RELOCATEを指定します。

ノート:



services 句と instances 句の両方が同じ ALTER PLUGGABLE DATABASE 文に指定されている場合、指定されたサービスは指定したインスタンスで開始されます。

親トピック: [PDBのオープン状態を変更する句](#)

ALTER PLUGGABLE DATABASEを使用したPDBのオープン・モードの変更

pdb_change_state句を指定したALTER PLUGGABLE DATABASE文を使用して、PDBのオープン・モードを変更できます。

前提条件

ALTER PLUGGABLE DATABASE文でPDBのオープン・モードを変更するには、次の前提条件を満たす必要があります。

- 現在のユーザーに、次のいずれかの管理権限が必要です。これは、共通で付与されているか、またはPDB内でローカルに付与されている必要があります。
 - SYSDBA。接続時にAS SYSDBAを使用することで権限が行使されている。
 - SYSOPER。接続時にAS SYSOPERを使用することで権限が行使されている。
 - SYSBACKUP。接続時にSYSBACKUPを使用することで権限が行使されている。
 - SYSDG。接続時にAS SYSDGを使用することで権限が行使されている。



ノート:

現在のコンテナが PDB である場合は、PDB のオープン・モードを変更できます。

- RESTRICTED SESSIONが有効になっており、PDBがオープンされている場合は、RESTRICTEDを指定する必要があります。
- Oracle RAC CDBでは、1つ以上のOracle RACインスタンス上でPDBがオープンされている場合に、それをさらに他のインスタンスでもオープンできます。ただし、そのPDBは、それがすでにオープンされているインスタンス上と同じモードでオープンする必要があります。一部のインスタンスでPDBをクローズし、他のインスタンスでオープンできます。

ALTER PLUGGABLE DATABASE文を使用してPDBを特定のターゲット・モードにするには、次の表で説明されている要件を満たす必要があります。

表15-4 ALTER PLUGGABLE DATABASEを使用したPDBのオープン・モードの変更

PDBのターゲット・モード	ALLキーワードが含まれている	FORCEキーワードが含まれている	ルートに必要なモード	変更対象のPDBごとに必要なモード
読取り/書込み	はい	はい	読取り/書込み	マウント、読取り専用または読取り/書込み
読取り/書込み	はい	いいえ	読取り/書込み	マウントまたは読取り/書込み
読取り/書込み	いいえ	はい	読取り/書込み	マウント、読取り専用または読取り/書込み
読取り/書込み	いいえ	いいえ	読取り/書込み	マウント済

PDBのターゲット・モード	ALLキーワードが含まれている	FORCEキーワードが含まれている	ルートに必要なモード	変更対象のPDBごとに必要なモード
読取り専用	はい	はい	読取り専用または読取り/書込み	マウント、読取り専用または読取り/書込み
読取り専用	はい	いいえ	読取り専用または読取り/書込み	マウント済または読取り専用
読取り専用	いいえ	はい	読取り専用または読取り/書込み	マウント、読取り専用または読取り/書込み
読取り専用	いいえ	いいえ	読取り専用または読取り/書込み	マウント
移行	はい	該当なし	読取り専用または読取り/書込み	マウント
移行	いいえ	該当なし	読取り専用または読取り/書込み	マウント
マウント	はい	該当なし	読取り専用または読取り/書込み	マウント、読取り専用、移行または読取り/書込み
マウント	いいえ	該当なし	読取り専用または読取り/書込み	読取り専用、移行または読取り/書込み

オープン・モードを変更するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。

[「CDB内のコンテナへのアクセスについて」](#)を参照してください。

2. `pdb_change_state`句を含むALTER PLUGGABLE DATABASE文を実行します。

例15-25 リストされたPDBのオープン・モードの変更

この文は、PDB salespdbおよびhrpdbのオープン・モードを読取り/書込みモードでオープンするよう変更します。

```
ALTER PLUGGABLE DATABASE salespdb, hrpdb
OPEN READ WRITE;
```

この文は、PDB salespdbのオープン・モードを読取り専用モードでオープンするよう変更します。RESTRICTEDにより、PDBでのRESTRICTED SESSION権限を持つユーザーのみがPDBにアクセスできることを指定します。

```
ALTER PLUGGABLE DATABASE salespdb
OPEN READ ONLY RESTRICTED;
```

この文は、PDB salespdbのオープン・モードを移行モードでオープンするよう変更します。

```
ALTER PLUGGABLE DATABASE salespdb
OPEN UPGRADE;
```

例15-26 すべてのPDBのオープン・モードの変更

次の問合せを実行して、CDBに関連付けられるPDBごとのオープン・モードを表示します。

```
SELECT NAME, OPEN_MODE FROM V$PDBS WHERE CON_ID > 2;
```

NAME	OPEN_MODE
HRPDB	READ WRITE
SALESPDB	MOUNTED
DWPDB	MOUNTED

hrpdbはすでに読取り/書込みモードであることに注意してください。salespdbおよびdwpdbのオープン・モードを読取り/書込みモードでオープンするよう変更するには、次の文を使用します。

```
ALTER PLUGGABLE DATABASE ALL  
OPEN READ WRITE;
```

hrpdb PDBはすでに読取り/書込みオープン・モードであるため、変更されません。2つのPDBはマウント・モードであり、1つのPDB (hrpdb)は指定されたモード(読取り/書込み)であるため、この文はエラーを返しません。同様に、すべてのPDBがマウント・モードである場合、この文はエラーを返しません。

ただし、いずれかのPDBが読取り専用モードである場合、この文はエラーを返します。エラーを回避してCDB内ですべてのPDBを読取り/書込みモードで開くには、FORCEキーワードを指定します。

```
ALTER PLUGGABLE DATABASE ALL  
OPEN READ WRITE FORCE;
```

FORCEのキーワードが含まれていると、読取り専用モードのPDBを含むすべてのPDBは読取り/書込みモードでオープンされます。

例15-27 リストされたPDBを除くすべてのPDBのオープン・モードの変更

この文は、salespdbおよびhrpdbを除くすべてのPDBのモードをマウント・モードに変更します。

```
ALTER PLUGGABLE DATABASE ALL EXCEPT salespdb, hrpdb  
CLOSE IMMEDIATE;
```

ノート:



PDBのオープン・モードを変更するALTER PLUGGABLE DATABASE文は、インスタンス固有です。したがって、Oracle RAC インスタンスに接続しているときにこの文を発行する場合、この文はそのインスタンスのPDBのオープン・モードにのみ影響を及ぼします。

関連項目:

- [PDBのオープン状態を変更するための句](#)
- PDBのその他の属性を変更するには、[「データベース・レベルでのPDBの変更」](#)を参照してください
- データベース・モードおよびその使用方法の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください
- [Oracle Database SQL言語リファレンス](#)
- 停止モードの詳細は、[『Oracle Database概要』](#)を参照してください

CDB再起動時のPDBのオープン・モードの保持または破棄

pdb_save_or_discard_state句を指定してALTER PLUGGABLE DATABASE SQL文を使用すると、CDBの再起動時に1つ以上のPDBのオープン・モードを保持できます。

次に示す方法で実行できます。

- CDBの再起動時にPDBのモードを保持するには、SAVE STATEを指定します。
たとえば、CDBの再起動前にPDBがオープン/読み取り/書き込みモードである場合、CDBの再起動後のPDBはオープン/読み取り/書き込みモードです。CDBの再起動前にPDBがマウント・モードの場合、CDBの再起動後のPDBはマウント・モードです。
- CDBの再起動時にPDBのオープン・モードを無視するには、DISCARD STATEを指定します。
PDBにDISCARD STATEを指定すると、CDBの再起動後のPDBは常にマウントされます。

変更するPDBは、次の方法で指定できます。

- 1つ以上のPDBをリストします。
- すべてのPDBを変更するには、ALLを指定します。
- リストされたPDBを除くすべてのPDBを変更するには、ALL EXCEPTを指定します。

Oracle RAC CDBの場合、pdb_save_or_discard_state句でinstances句を使用すると、PDBのオープン・モードを保持するインスタンスを次の方法で指定できます。

- 次の形式で、instances句に1つ以上のインスタンスをリストします。

```
INSTANCES = ( 'instance_name' [, 'instance_name'] ... )
```

- すべての実行中のインスタンスのPDBを変更するには、次の例のように、instances句にALLを指定します。

```
INSTANCES = ALL
```

- リストされたインスタンスを除くすべてのインスタンスのPDBを変更するには、次の形式で、instances句にALL EXCEPTを指定します。

```
INSTANCES = ALL EXCEPT ( 'instance_name' [, 'instance_name'] ... )
```

Oracle RAC CDBのPDBの場合、SAVE STATEおよびDISCARD STATEは現行インスタンスのモードにのみ作用します。instances句に複数のインスタンスを指定している場合でも、他のインスタンスのモードには作用しません。

pdb_save_or_discard_state句を指定してALTER PLUGGABLE DATABASE SQL文を発行するには、現行ユーザーにルートでのALTER DATABASE権限が必要です。

CDBのPDBについて保存された状態を確認するには、DBA_PDB_SAVED_STATESビューを問い合わせます。

CDBの再起動時にPDBのオープン・モードを保持または破棄するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
[「CDB内のコンテナへのアクセスについて」](#)を参照してください。
2. pdb_save_or_discard_state句を指定してALTER PLUGGABLE DATABASE文を実行します。

次の例では、CDBの再起動時に、1つ以上のPDBのオープン・モードを保持または破棄しています。

例15-28 CDB再起動時のPDBのオープン・モードの保持

この文は、CDBの再起動時にsalespdbのオープン・モードを保持します。

```
ALTER PLUGGABLE DATABASE salespdb SAVE STATE;
```

例15-29 CDB再起動時のPDBのオープン・モードの破棄

この文は、CDBの再起動時にsalespdbのオープン・モードを破棄します。

```
ALTER PLUGGABLE DATABASE salespdb DISCARD STATE;
```

例15-30 CDB再起動時の全PDBのオープン・モードの保持

この文は、CDBの再起動時にすべてのPDBのオープン・モードを保持します。

```
ALTER PLUGGABLE DATABASE ALL SAVE STATE;
```

例15-31 CDB再起動時のリストされたPDBのオープン・モードの保持

この文は、CDBの再起動時にsalespdbおよびhrpdbのオープン・モードを保持します。

```
ALTER PLUGGABLE DATABASE salespdb, hrpdb SAVE STATE;
```

例15-32 CDB再起動時のリストされたPDB以外全部のオープン・モードの保持

この文は、salespdbおよびhrpdbを除くすべてのPDBのオープン・モードを保持します。

```
ALTER PLUGGABLE DATABASE ALL EXCEPT salespdb, hrpdb SAVE STATE;
```

親トピック: [PDBのオープン・モードの変更](#)

STARTUPおよびSHUTDOWNを使用したPDBのオープン・モードの変更

現在のコンテナがPDBの場合、SQL*PlusのSTARTUPコマンドを使用してPDBをオープンし、SQL*PlusのSHUTDOWNコマンドを使用してPDBをクローズできます。

- [SQL*PlusのSTARTUPコマンドを使用したPDBのオープン・モードの変更について](#)
現在のコンテナがルートである場合、STARTUP PLUGGABLE DATABASEコマンドを使用すると、単一のPDBをオープンできます。
- [STARTUPコマンドを使用したPDBの起動](#)
現在のコンテナがPDBの場合は、SQL*PlusのSTARTUPコマンドでPDBをオープンします。
- [SQL*PlusのSTARTUPコマンドを使用したPDBのオープン・モードの変更](#)
STARTUP PLUGGABLE DATABASEコマンドを使用して、単一のPDBをオープンできます。
- [SHUTDOWNコマンドを使用したPDBの停止](#)
現在のコンテナがPDBの場合は、SQL*PlusのSHUTDOWNコマンドでPDBをクローズします。

親トピック: [PDBの管理](#)

SQL*PlusのSTARTUPコマンドを使用したPDBのオープン・モードの変更について

現在のコンテナがルートである場合、STARTUP PLUGGABLE DATABASEコマンドを使用すると、単一のPDBをオープンできます。

STARTUP PLUGGABLE DATABASEコマンドの次のオプションを使用して、PDBをオープンします。

- FORCE
読み取り/書き込みモードで再オープンする前に、オープン状態のPDBをクローズします。このオプションが指定されている場

合、他のオプションは使用できません。

- RESTRICT

PDBでRESTRICTED SESSIONシステム権限を持つユーザーのみがそのPDBにアクセスできるようにします。

OPEN READ WRITEもOPEN READ ONLYも指定されていない場合、PDBは、属しているCDBがフィジカル・スタンバイ・データベースのときは読取り専用モードでオープンされます。そうでないときは、PDBは読取り/書込みモードでオープンされます。

- OPEN open_pdb_options

PDBを読取り/書込みモードまたは読取り専用モードでオープンします。OPEN READ WRITEまたはOPEN READ ONLYを指定できます。他のオプションを使用しないでOPENを指定した場合は、READ WRITEがデフォルトになります。

次の前提条件を満たしている必要があります。

- 現行ユーザーには、SYSDBA、SYSOPER、SYSBACKUPまたはSYSDBG管理権限があり、その権限は共通に付与されているか、またはPDBでローカルで付与されている必要があります。ユーザーは、接続時にAS SYSDBA、AS SYSOPER、AS SYSBACKUPまたはAS SYSDBGをそれぞれ使用して、権限を行使する必要があります。
- RESTRICTED SESSIONを有効にした場合、RESTRICTはPDBがオープンされているときに指定する必要があります。

また、STARTUP PLUGGABLE DATABASEコマンドを使用してPDBを特定のターゲット・モードにするには、次の表で説明されている要件を満たす必要があります。

表15-5 STARTUP PLUGGABLE DATABASEを使用したPDBのオープン・モードの変更

PDBのターゲット・モード	FORCEオプションが含まれている	ルートに必要なモード	変更対象のPDBの必要なモード
読取り/書込み	はい	読取り/書込み	マウント、読取り専用または読取り/書込み
読取り/書込み	いいえ	読取り/書込み	マウント
読取り専用	いいえ	読取り専用または読取り/書込み	マウント

ノート:



現在のコンテナが PDB である場合に PDB のオープン・モードを変更するには、STARTUP コマンドを使用することもできます。

関連項目:

- [「STARTUPコマンドを使用したPDBの起動」](#)
- [「SQL*PlusのSTARTUPコマンドを使用したPDBのオープン・モードの変更」](#)

STARTUPコマンドを使用したPDBの起動

現在のコンテナがPDBである場合、SQL*PlusのSTARTUPコマンドでPDBをオープンできます。

STARTUPコマンドの次のオプションを使用して、PDBをオープンします。

- FORCE

読取り/書き込みモードで再オープンする前に、オープン状態のPDBをクローズします。このオプションが指定されている場合、他のオプションは使用できません。

- RESTRICT

PDBでRESTRICTED SESSIONシステム権限を持つユーザーのみがそのPDBにアクセスできるようにします。

OPEN READ WRITEもOPEN READ ONLYも指定されず、RESTRICTが指定されている場合、PDBは、属しているCDBがフィジカル・スタンバイ・データベースのときに読取り専用でオープンされます。そうでないときは、PDBは読取り/書き込みモードでオープンされます。

- OPEN open_pdb_options

PDBを読取り/書き込みモードまたは読取り専用モードでオープンします。OPEN READ WRITEまたはOPEN READ ONLYを指定します。RESTRICTが指定されていない場合、READ WRITEが常にデフォルトです。

現在のコンテナがPDBの場合にSTARTUPコマンドを発行するには、次の前提条件を満たしている必要があります。

- 現行ユーザーには、SYSDBA、SYSOPER、SYSBACKUPまたはSYSDG管理権限があり、その権限は共通に付与されているか、またはPDBでローカルで付与されている必要があります。ユーザーは、接続時にAS SYSDBA、AS SYSOPER、AS SYSBACKUPまたはAS SYSDGをそれぞれ使用して、権限を行使する必要があります。
- FORCEオプションを使用する場合を除き、PDBをオープンするには、PDBがマウント・モードである必要があります。
- PDBをマウント・モードにするには、そのPDBが読取り専用オープン・モードまたは読取り/書き込みオープン・モードである必要があります。

STARTUPコマンドを使用してPDBを変更するには:

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。
2. STARTUPコマンドを実行します。

例15-33 STARTUPコマンドを使用してPDBを読取り/書き込みモードでオープンする方法

```
STARTUP OPEN
```

例15-34 STARTUPコマンドを使用してPDBを読取り専用モードでオープンする方法

```
STARTUP OPEN READ ONLY
```

例15-35 STARTUPコマンドを使用してPDBを読取り専用制限モードでオープンする方法

```
STARTUP RESTRICT OPEN READ ONLY
```

例15-36 STARTUPコマンドおよびFORCEオプションを使用してPDBを読取り/書き込みモードでオープンする方法

この例では、PDBが現在オープンしていると想定しています。FORCEオプションによって、PDBがクローズされ、その後、読取り/書き込みモードでオープンされます。

```
STARTUP FORCE
```

関連項目:

- [「現在のコンテナについて」](#)
- [「PDBへの接続」](#)
- データベースの起動の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください
- [SQL*Plusユーザズ・ガイドおよびリファレンス](#)

親トピック: [STARTUPおよびSHUTDOWNを使用したPDBのオープン・モードの変更](#)

SQL*PlusのSTARTUPコマンドを使用したPDBのオープン・モードの変更

STARTUP PLUGGABLE DATABASEコマンドを使用して、単一のPDBをオープンできます。

STARTUP PLUGGABLE DATABASEコマンドを使用してPDBを変更するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。

[「CDB内のコンテナへのアクセスについて」](#)を参照してください。

2. STARTUP PLUGGABLE DATABASEコマンドを実行します。

ノート:



現在のコンテナがルートである場合、CDB インスタンスは SQL*Plus の SHUTDOWN コマンドで常に停止されます。これを使用して個々の PDB をクローズすることはできません。

例15-37 STARTUPコマンドを使用してPDBを読取り/書き込みモードでオープンする方法

```
STARTUP PLUGGABLE DATABASE hrpdb OPEN
```

例15-38 STARTUPコマンドを使用した読取り/書き込み制限モードでのPDBのオープン

```
STARTUP PLUGGABLE DATABASE hrpdb RESTRICT
```

例15-39 STARTUPコマンドを使用してPDBを読取り専用制限モードでオープンする方法

```
STARTUP PLUGGABLE DATABASE hrpdb OPEN READ ONLY RESTRICT
```

例15-40 STARTUPコマンドを使用してPDBを読取り専用モードでオープンする方法

```
STARTUP PLUGGABLE DATABASE hrpdb OPEN READ ONLY
```

例15-41 STARTUPコマンドおよびFORCEオプションを使用してPDBを読取り/書き込みモードでオープンする方法

この例では、hrpdb PDBが現在オープンしていることを想定しています。FORCEオプションによって、PDBがクローズされ、その後、読取り/書き込みモードでオープンされます。

```
STARTUP PLUGGABLE DATABASE hrpdb FORCE
```

関連項目:

- [「SQL*PlusのSTARTUPコマンドを使用したPDBのオープン・モードの変更について」](#)
- 現在のコンテナがPDBである場合のSTARTUPコマンドまたはSHUTDOWNコマンドの使用については、[STARTUPおよびSHUTDOWNの使用によるPDBのオープン・モードの変更](#)を参照してください。
- [Oracle Database管理者ガイド](#)
- [SQL*Plusユーザズ・ガイドおよびリファレンス](#)

親トピック: [STARTUPおよびSHUTDOWNを使用したPDBのオープン・モードの変更](#)

SHUTDOWNコマンドを使用したPDBの停止

現在のコンテナがPDBである場合、SQL*PlusのSHUTDOWNコマンドでPDBをクローズできます。

SHUTDOWNコマンドがPDBで正常に発行されると、PDBはマウント・モードになります。

SHUTDOWNモードは次のいずれかになります。

- SHUTDOWNのみ指定した場合、PDBはNORMALモードで停止されます。
- SHUTDOWN IMMEDIATEを指定した場合、PDBはIMMEDIATEモードで停止されます。
- SHUTDOWN ABORTを指定した場合、PDBは強制的にクローズされます。

単一インスタンスCDBでは、SHUTDOWN ABORTを指定する場合にPDBメディア・リカバリが必要です。Oracle Real Application Clusters (Oracle RAC) CDBでは、SHUTDOWN ABORTコマンドが最後のオープン・インスタンスをクローズする場合にPDBメディア・リカバリが必要です。

PDBキーストアがオープン状態であった場合にはPDBレベルでSHUTDOWNを発行してもそれはクローズされないということに注意してください。キーストアを閉じるには、ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE IDENTIFIED BY "pdb_ks_pwd"コマンドを実行します。

前提条件

現在のコンテナがPDBである場合にSHUTDOWNコマンドを発行するには、次の前提条件を満たしている必要があります。

- 現行ユーザーには、SYSDBA、SYSOPER、SYSBACKUPまたはSYSDBG管理権限があり、その権限は共通に付与されているか、またはPDBでローカルで付与されている必要があります。ユーザーは、接続時にAS SYSDBA、AS SYSOPER、AS SYSBACKUPまたはAS SYSDBGをそれぞれ使用して、権限を行使する必要があります。
- PDBをクローズするには、そのPDBがオープンされている必要があります。

SHUTDOWNコマンドを使用してPDBを変更するには:

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。
2. SHUTDOWNコマンドを実行します。

ノート:



- 現在のコンテナがPDBの場合、SHUTDOWNコマンドは、CDBインスタンスではなくPDBのみをクローズします。
- CDBで使用できるSHUTDOWN TRANSACTIONALと同等のSHUTDOWNコマンドは、PDBにはありません。

例15-42 SHUTDOWN IMMEDIATEコマンドを使用してPDBをクローズする方法

SHUTDOWN IMMEDIATE

関連項目:

- [「現在のコンテナについて」](#)
- [「PDBへの接続」](#)
- [「ALTER PLUGGABLE DATABASEを使用したPDBのオープン・モードの変更」](#)
- 停止モードの詳細は、『[Oracle Database管理者ガイド](#)』を参照してください
- [SQL*Plusユーザズ・ガイドおよびリファレンス](#)

親トピック: [STARTUPおよびSHUTDOWNを使用したPDBのオープン・モードの変更](#)

Oracle RACでのPDBの起動および停止

プラグブル・データベース(PDB)の管理には、非CDBを管理するために必要なタスクのごく一部が必要です。

Oracle RACベースのマルチテナント・コンテナ・データベース(CDB)の管理は、非CDBの管理に似ています。違いは、ある管理タスクはCDB全体に適用され、ある管理タスクはCDBルートにのみ適用され、ある管理タスクは特定のPDBに適用されることです。この一部のタスクでは、ほとんどがPDBおよび非CDBに対して同じです。ただし、PDBのオープン・モードを変更する場合など、いくつかの違いがあります。また、PDB管理者は、単一PDBの管理のみを行い、CDB内の他のPDBによる影響は受けません。

Oracle RAC CDB内のPDBは、サービスを管理することによって管理します。これは、PDBがポリシー管理か、または管理者管理かに関係なく同じです。1つの動的データベース・サービスを各PDBに割り当てて、クラスタ化コンテナ・データベース内のインスタンスにわたってPDBの起動、停止および配置を調整します。

たとえば、prodというサーバー・プールにsparkというポリシー管理PDBを備えたraccontというCDBを所有している場合、次のコマンドを使用してplugというサービスをこのデータベースに割り当てます。

```
srvctl add service -db raccont -pdb spark -service plug -serverpool prod
```

サービスplugは、サーバー・プール内のすべてのノードにわたって均一に管理されます。同じサーバー・プールでこのサービスをシングルトン・サービスとして実行する場合は、前述のコマンドとともに-cardinality singletonパラメータを使用します。

PDB sparkを開くには、次のように、サービスplugを起動する必要があります。

```
srvctl start service -db raccont -service plug
```

サービスplugを停止するには:

```
srvctl stop service -db raccont -service plug
```

PDB sparkは、SQLコマンドALTER PLUGGABLE DATABASE PDB_NAME CLOSE IMMEDIATEを使用してPDBを閉じるまで開いたままです。srvctl status serviceコマンドを使用すると、データベースのステータスを確認できます。

PDBは動的データベース・サービスを使用して管理されるため、通常のOracle RACベースの管理プラクティスが適用されます。このため、サービスplugがオンライン状態で、このサービスをホストしているサーバー上でOracle Clusterwareが停止している

場合、このサーバー上のOracle Clusterwareの再起動後に、サービスは元の状態にリストアされます。したがって、PDBの起動は、他のOracle RACデータベースと同様に自動化されます。

ノート:



SQL*Plusとは異なり、SRVCTL は、クラスタ・データベース全体を操作します。したがって、サービスが同時に複数のサーバー上で実行されるように定義され、クラスタの現行のステータスがこの配置を可能にしている場合、サービスを使用した PDB の起動は、クラスタ化された CDB の複数のインスタンスに同時に適用されます。

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)

親トピック: [PDBの管理](#)

16 PDBスナップショット・カルーセルの管理

指定したPDBのPDBスナップショット・カルーセルを構成し、スナップショットを手動または自動的に作成して、スナップショットの最大数を設定できます。

- [PDBスナップショット・カルーセルについて](#)
PDBスナップショット・カルーセルとは、PDBスナップショットのライブラリのことです。
- [PDBスナップショット・カルーセル内のスナップショットの最大数の設定](#)
PDBのPDBスナップショットの最大数を設定できます。
- [自動PDBスナップショットの構成](#)
PDBを作成または変更する際にSNAPSHOT MODE EVERY句を使用して、自動スナップショットのPDBを構成します。
- [PDBスナップショットの手動作成](#)
PDBスナップショットを手動で作成するには、ALTER PLUGGABLE DATABASEまたはCREATE PLUGGABLE DATABASEでSNAPSHOT snapshot_name句を指定します。
- [PDBスナップショットの削除](#)
DROP SNAPSHOT句を指定したALTER PLUGGABLE DATABASE文を実行して、PDBスナップショットを削除できます。
- [PDBスナップショットのメタデータの表示](#)
データ・ディクショナリ・ビューのDBA_PDB_SNAPSHOTSとDBA_PDB_SNAPSHOTFILEにPDBスナップショットのメタデータが表示されます。

親トピック: [マルチテナント環境の管理](#)

PDBスナップショット・カルーセルについて

PDBスナップショット・カルーセルとは、PDBスナップショットのライブラリのことです。

PDBスナップショットはPDBのPoint-in-Timeコピーです。スナップショットの作成中に、ソースPDBを読み取り専用または読み取り/書き込みでオープンできます。スナップショットは、CREATE PLUGGABLE DATABASE (またはALTER PLUGGABLE DATABASE)のSNAPSHOT句を使用して手動で作成するか、またはEVERY間隔句を使用して自動的に作成できます。ストレージ・システムでスパース・クローンがサポートされている場合は、前述のコマンドではスパース・コピーが作成されます。それ以外の場合は、完全コピーが作成されます。

- [PDBスナップショット・カルーセルの目的](#)
PDBスナップショット・カルーセルは、Point-in-Timeリカバリ用およびクローニング用に最新PDBコピーのライブラリを維持するために役立ちます。
- [PDBスナップショット・カルーセルの仕組み](#)
特定のPDBのカルーセルは、このPDBのコピーの循環ライブラリです。
- [PDBスナップショット・カルーセルのユーザー・インタフェース](#)
SNAPSHOT MODE句を使用してスナップショットの作成を制御し、作成が手動、自動または無効かを決定します。

関連項目:

各種エディションおよびサービスでサポートされる機能の詳細は、『[Oracle Databaseライセンス情報ユーザー・マニュアル](#)』を参照

親トピック: [PDBスナップショット・カルーセルの管理](#)

PDBスナップショット・カールセルの目的

PDBスナップショット・カールセルは、Point-in-Timeリカバリ用およびクローニング用に最新PDBコピーのライブラリを維持するために役立ちます。

開発およびテスト用にPDBをクローニング

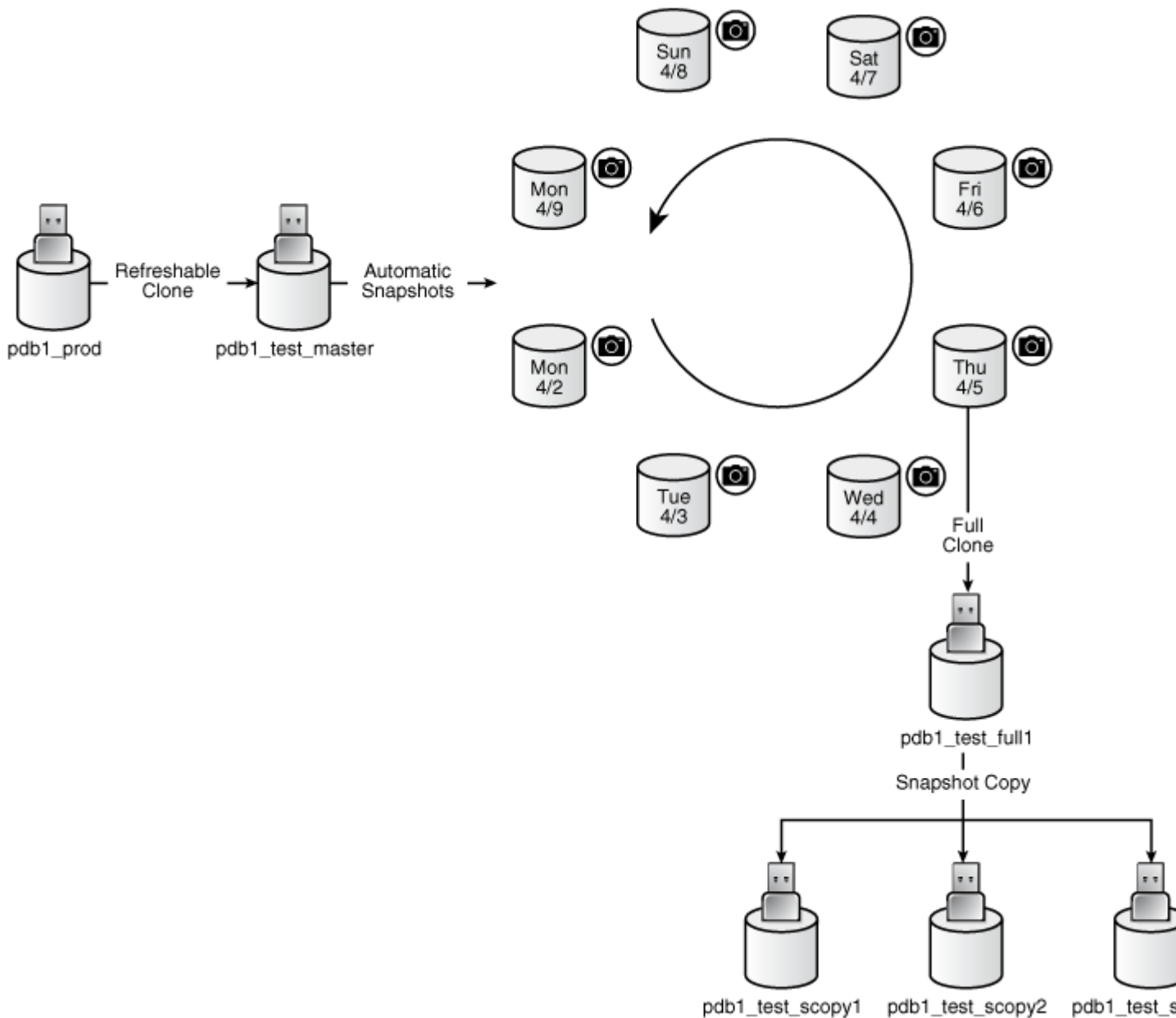
一般的な開発ユースケースでは、`CREATE PLUGGABLE DATABASE newpdb FROM srcpdb`という形式のコマンドを使用して、テスト用に本番PDBをクローニングします。CDBがARCHIVELOGモードおよびローカルUNDOモードの場合、ソースの本番PDBは読取り/書き込みモードでオープンでき、そのクローニング時に十分に機能します(**ホット・クローニング**と呼ばれる手法)。ホット・クローンPDBは、`ALTER PLUGGABLE DATABASE ... OPEN`文が完了したSCNの時点でのソースPDBとトランザクシヨンの一貫した状態です。

次のステップは、一般的な開発シナリオを示しています。

1. `pdb1_prod`という本番PDBがオープンされ使用されている間に、`pdb1_test_master`というリフレッシュ可能クローンPDBを作成します。
リフレッシュ可能クローンPDBは、読取り専用モードでのみオープンできます。そのクローンPDBを`pdb1_prod`からリフレッシュするには、それをクローズする必要があります。
2. `ALTER PLUGGABLE DATABASE pdb1_test_master SNAPSHOT MODE EVERY 24 HOURS`を実行します。これにより、`pdb1_test_master`の自動スナップショットを毎日生成するようにPDBが構成されます。
3. テスト用に新しいPDBが必要な場合は、`CREATE PLUGGABLE DATABASE ... USING SNAPSHOT`コマンドを使用してフル・クローンPDBを作成します。
4. `CREATE PLUGGABLE DATABASE ... SNAPSHOT COPY`を使用して、フル・クローンPDBのスペース・スナップショット・コピーPDBを作成します。

次の図は、4月5日に取得したPDBスナップショットからクローンの`pdb1_test_full1`を作成する手順を示しています。図には、`pdb1_test_full1`から作成された3つのスナップショット・コピーPDBが示されています。

図16-1 リフレッシュ可能なクローンPDBの自動スナップショット

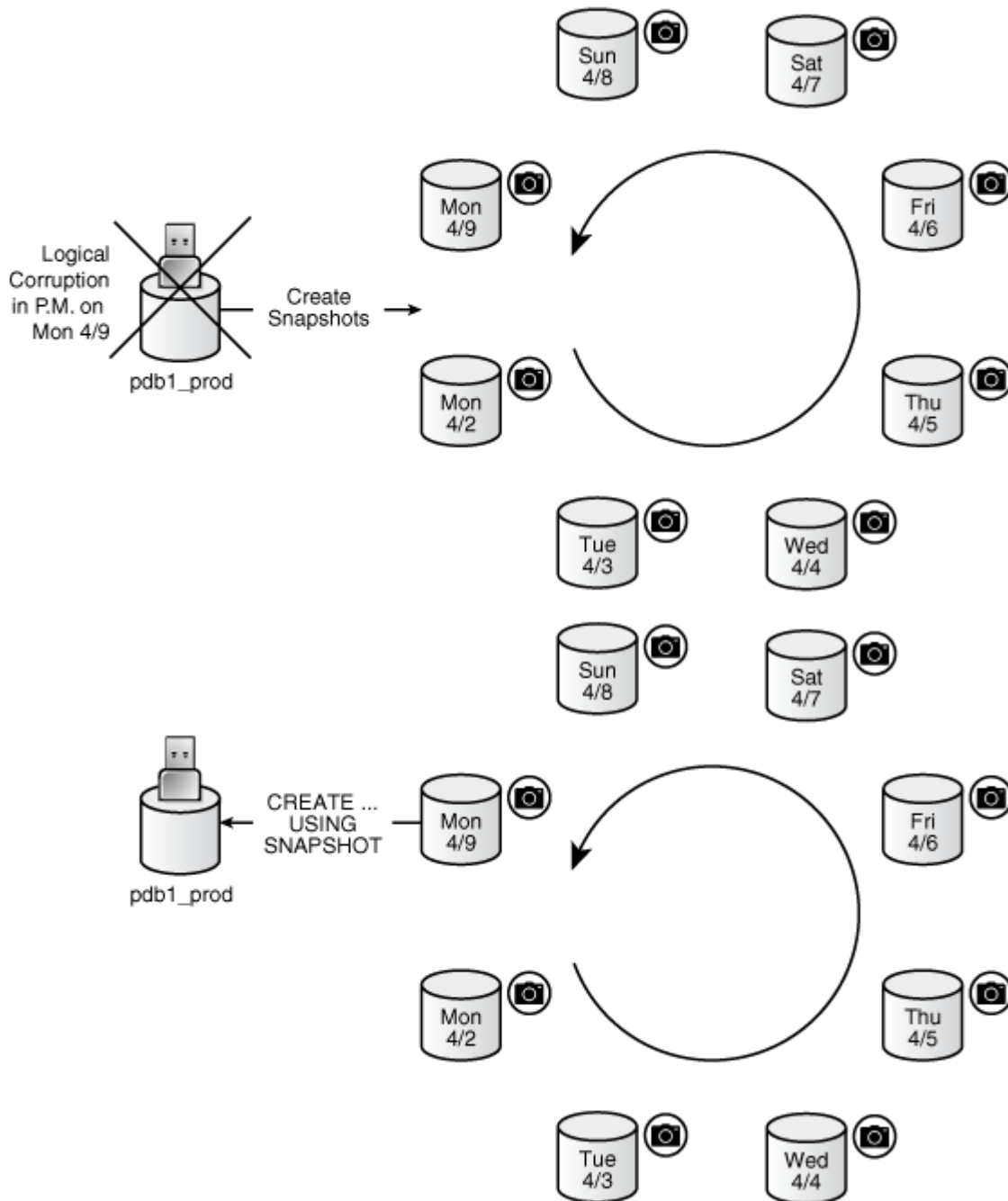


PDBスナップショット・カルーセルを使用したPoint-in-Timeリストア

PDBのスナップショットを毎日同じ時刻に取得するという方法があります。データ・ロード前に、PDBスナップショットを手動で取得する方法もあります。いずれの場合も、PDBスナップショット・カルーセルを使用すると、利用可能なPDBスナップショットを使用してPDBをリストアできます。

たとえば、`pdb1_prod`という名前の販売履歴PDBは、毎日午前12時1分に自動スナップショットを生成します。4月9日月曜日の午後に行われた日次データ・ロード時に、誤ったデータをロードしてPDBが破損しました。4月9日月曜日のスナップショットに基づいて新しい本番PDBを作成し、破損したPDBを削除して、データ・ロードを再試行できます。

図16-2 スナップショットを使用した本番PDBのリストア



関連項目:

- [「PDBまたは非CDBのクローニングについて」](#)
- CREATE PLUGGABLE DATABASEの構文およびセマンティクスについては、『[Oracle Database SQL言語リファレンス](#)』を参照してください
- 各種エディションおよびサービスでサポートされる機能の詳細は、『[Oracle Databaseライセンス情報ユーザー・マニュアル](#)』を参照

親トピック: [PDBスナップショット・カールールについて](#)

PDBスナップショット・カールールの仕組み

特定のPDBのカールールは、このPDBのコピーの循環ライブラリです。

データベースでは、必要に応じてまたは自動的に、カルーセル内に連続してコピーを作成します。スナップショット制限に達すると、データベースでは最も古いスナップショットが上書きされます。

- [PDBスナップショットの内容](#)

PDBスナップショットの内容は、基礎となるファイル・システムでスパース・ファイルがサポートされているかどうかで異なります。

- [PDBスナップショット・カルーセルの内容](#)

PDBスナップショット・カルーセルは、PDBのすべての既存のスナップショットのセットです。

親トピック: [PDBスナップショット・カルーセルについて](#)

PDBスナップショットの内容

PDBスナップショットの内容は、基礎となるファイル・システムでスパース・ファイルがサポートされているかどうかで異なります。

スナップショット名

データベース管理PDBスナップショットの名前は、ユーザー指定またはシステム生成のいずれかになります。システム生成のスナップショット名の場合は、スナップショットSCNを含む一意の識別子にSNAP_という接頭辞が付きます。たとえば、次の問合せにより、システム生成の名前と取得時のSCNが付いた3つのスナップショットが示されます。

```
SET LINESIZE 200
SET PAGESIZE 50000

COL CON_ID FORMAT 999999
COL CON_NAME FORMAT a15
COL SNAPSHOT_NAME FORMAT a27

SELECT CON_ID, CON_NAME, SNAPSHOT_NAME, SNAPSHOT_SCN FROM DBA_PDB_SNAPSHOTS;
```

CON_ID	CON_NAME	SNAPSHOT_NAME	SNAPSHOT_SCN
5	HRPDB	SNAP_1389467754_993556301	2925293
5	HRPDB	SNAP_1389467754_993556306	2925679
5	HRPDB	SNAP_1389467754_993556309	2925698

ノート:



各種エディションおよびサービスでサポートされる機能の詳細は、[『Oracle Database ライセンス情報ユーザー・マニュアル』](#)を参照してください。

フル・スナップショットとスパース・スナップショット

ALTER PLUGGABLE DATABASE ... SNAPSHOTによって生成されるスナップショットの内容は、基礎となるファイル・システムに応じて異なります。基礎となるファイル・システムでスパース・コピーがサポートされている場合、PDBレベルのスナップショットはスパースとなります。最初のPDB管理PDBスナップショットのみがフルになります。それ以外の場合、PDBスナップショットにはデータ・ファイルの完全コピーが含まれます。スナップショットには、そのスナップショットからPDBを作成するために必要な他のファイルが含まれています。

スナップショット・ディレクトリ

すべてのPDBに、専用のスナップショット・ディレクトリがあります。このディレクトリ内には、各スナップショットの取得時のSCNに関する名前が付いた専用のサブディレクトリがあります。次の問合せにより、DBIDが1389467754のhrpdbのスパースPDBスナッ

プショットが示されます。

```
SET LINESIZE 200
SET PAGESIZE 50000
```

```
COL SNAPSHOT_NAME FORMAT a27
COL FULL_SNAPSHOT_PATH FORMAT a65
```

```
SELECT SNAPSHOT_NAME, SNAPSHOT_SCN, FULL_SNAPSHOT_PATH FROM DBA_PDB_SNAPSHOTS;
```

SNAPSHOT_NAME	SNAPSHOT_SCN	FULL_SNAPSHOT_PATH
SNAP_1389467754_993556301	2925293	/d1/snapshots/pdb_1389467754/2925293/
SNAP_1389467754_993556306	2925679	/d1/snapshots/pdb_1389467754/2925679/
SNAP_1389467754_993556309	2925698	/d1/snapshots/pdb_1389467754/2925698/

ノート:



スナップショットがスペースではなくフルである場合、フル・スナップショット・パスでは接尾辞が、pdb のアーカイブを指定します。

/d1/snapshots/pdb_1389467754/2925698/のディレクトリには、次のファイルが格納されています。

```
archpar log_1_63_52d1986a_993552590. arc
o1_mf_salestbs_g03341t2_. dbf
o1_mf_sysext_g0333vqw_. dbf
o1_mf_undo_1_g033gd2j_. dbf
o1_mf_sysaux_g0333vqv_. dbf
o1_mf_system_g0333vqt_. dbf
HRPDB.xml
```

このセットには、データファイル、アーカイブREDOログ・ファイルおよびPDBスナップショットに関するメタデータを格納しているXMLファイルが含まれます。次のduコマンドにより、スペースのスナップショット・データ・ファイルのサイズが、データ・ファイルのサイズと比較して小さいことが示されます。

```
% du -h *dbf
16K    o1_mf_salestbs_g03341t2_. dbf
16K    o1_mf_sysaux_g0333vqv_. dbf
16K    o1_mf_sysext_g0333vqw_. dbf
16K    o1_mf_system_g0333vqt_. dbf
16K    o1_mf_undo_1_g033gd2j_. dbf
```

次のデータ・ディクショナリ結合により、スナップショット2925698のスナップショット・ファイル名とタイプが示されます。

```
SELECT f. SNAPSHOT_FILENAME, f. SNAPSHOT_FILETYPE
FROM   DBA_PDB_SNAPSHOTS s, DBA_PDB_SNAPSHOTFILE f
WHERE  s. SNAPSHOT_SCN=f. SNAPSHOT_SCN
AND    s. CON_ID=f. CON_ID
ORDER BY s. SNAPSHOT_SCN DESC;
```

SNAPSHOT_FILENAME	SNAPSHOT
/d1/snapshots/pdb_1389467754/2925698/o1_mf_sysaux_g0333vqv_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925698/o1_mf_system_g0333vqt_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925698/HRPDB.xml	XML

/d1/snapshots/pdb_1389467754/2925698/o1_mf_sysext_g0333vqw_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925698/o1_mf_salestbs_g03341t2_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925698/o1_mf_undo_1_g033gd2j_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925698/archpar log_1_63_52d1986a_993552590. arc	ARCH
/d1/snapshots/pdb_1389467754/2925679/o1_mf_sysext_g0333vqw_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925679/o1_mf_salestbs_g03341t2_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925679/o1_mf_undo_1_g033gd2j_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925679/o1_mf_sysaux_g0333vqv_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925679/archpar log_1_63_52d1986a_993552590. arc	ARCH
/d1/snapshots/pdb_1389467754/2925679/HRPDB. xml	XML
/d1/snapshots/pdb_1389467754/2925679/o1_mf_system_g0333vqt_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925293/HRPDB. xml	XML
/d1/snapshots/pdb_1389467754/2925293/o1_mf_system_g0333vqt_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925293/o1_mf_sysaux_g0333vqv_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925293/o1_mf_undo_1_g033gd2j_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925293/o1_mf_salestbs_g03341t2_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925293/o1_mf_sysext_g0333vqw_. dbf	DATA
/d1/snapshots/pdb_1389467754/2925293/archpar log_1_63_52d1986a_993552590. arc	ARCH

親トピック: [PDBスナップショット・カルーセルの仕組み](#)

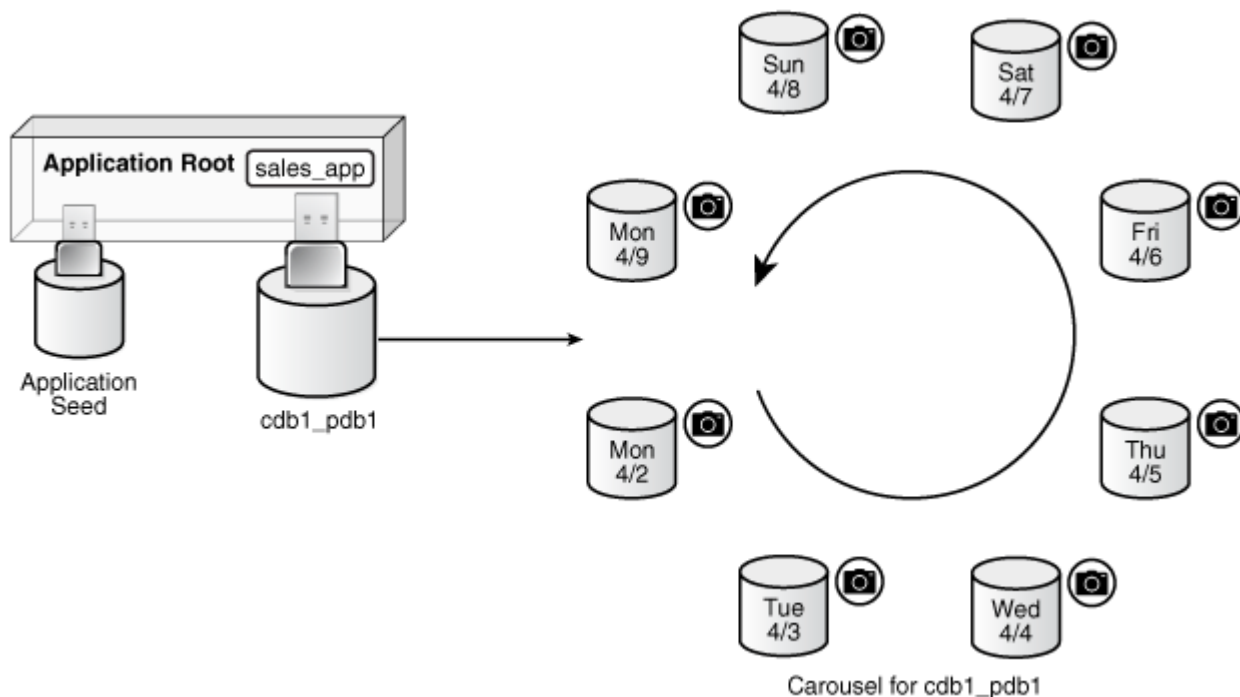
PDBスナップショット・カルーセルの内容

PDBスナップショット・カルーセルは、PDBのすべての既存のスナップショットのセットです。

MAX_PDB_SNAPSHOTSプロパティは、カルーセルで許可されるスナップショットの最大数を指定します。現在の設定は、CDB_PROPERTIESビューに表示されます。

次の図は、cdb1_pdb1のカルーセルを示しています。この例では、データベースはPDBスナップショットを毎日自動的に取得して、8個のスナップショットのセットを維持します。最初の8個のスナップショットが作成された後は、最新のスナップショットが作成されるたびに最も古いスナップショットと置き換えられます。たとえば、4月10日火曜日のスナップショットは4月2日月曜日のスナップショットと置き換えられ、4月11日水曜日のスナップショットは4月3日火曜日のスナップショットと置き換えられます。

図16-3 PDBスナップショット・カルーセル



ファイル・システムでスペース・ファイルがサポートされている場合、カルーセル内の、最初のスナップショットを除くすべてのPDBスナップショットはスペースとなります。ソースPDBは、読取り/書込みモードのままにできます。スペース・ファイルにより、カルーセルの記憶領域が大幅に削減されます。

関連項目:

各種エディションおよびサービスでサポートされる機能の詳細は、[『Oracle Databaseライセンス情報ユーザー・マニュアル』](#)を参照

親トピック: [PDBスナップショット・カルーセルの仕組み](#)

PDBスナップショット・カルーセルのユーザー・インタフェース

SNAPSHOT MODE句を使用してスナップショットの作成を制御し、作成が手動、自動または無効かを決定します。

ALTER PLUGGABLE DATABASE ... SNAPSHOT文

PDBのスナップショット・モードを設定するには、ALTER PLUGGABLE DATABASEまたはCREATE PLUGGABLE DATABASEのSNAPSHOT MODE句で次のいずれかの値を使用します。

- MANUAL

この句(デフォルト)は、PDBの手動スナップショットの作成を有効にします。必要に応じてスナップショットを作成するには、ALTER PLUGGABLE DATABASEまたはCREATE PLUGGABLE DATABASE文でSNAPSHOT snapshot_name句を指定します。

- EVERY snapshot_interval [MINUTES|HOURS]

この句は、時間間隔ごとのスナップショットの自動作成を有効にします。指定する間隔には次の制限があります。

- 分の値は3000より小さくする必要があります。
- 時間の値は2000より小さくする必要があります。

データベースでは、各自動スナップショットにシステム生成の名前が割り当てられます。EVERYが指定されている場合は、PDBに対して手動スナップショットもサポートされていることに注意してください。

- NONE

この句は、PDBのスナップショット作成を無効にします。

関連項目:

- [「PDBスナップショットからのPDBのクローニングについて」](#)
- SNAPSHOT句の構文およびセマンティクスについては、[『Oracle Database SQL言語リファレンス』](#)を参照してください

MAX_PDB_SNAPSHOTSデータベース・プロパティ

PDBのスナップショットの最大数を設定するには、ALTER PLUGGABLE DATABASEまたはCREATE PLUGGABLE DATABASEでMAX_PDB_SNAPSHOTSプロパティを指定します。プロパティのデフォルトは8で、これは最大値でもあります。スナップショットの最大許容数がすでに作成されている場合、データベースでは最も古いスナップショットが削除されます。CDB_PROPERTIESビューに、MAX_PDB_SNAPSHOTSの設定が表示されます。

関連項目:

ALTER PLUGGABLE DATABASE文の構文については、[『Oracle Database SQL言語リファレンス』](#)を参照してください

スナップショット関連のデータ・ディクショナリ・ビュー

次のデータ・ディクショナリ・ビューは、スナップショット情報を提供します。

- DBA_PDB_SNAPSHOTSビューには、スナップショットに関するメタデータ(名前、作成SCN、作成時間およびファイル名を含む)が記録されます。
- DBA_PDB_SNAPSHOTFILEビューには、PDBスナップショット内のファイルの名前とタイプがリストされます。このビューは、スナップショットがスペースの場合にのみ移入されます。
- DBA_PDBSビューには、SNAPSHOT_MODEおよびSNAPSHOT_INTERVAL列があります。

関連項目:

[DBA_PDB_SNAPSHOTS](#)、[DBA_PDB_SNAPSHOTFILE](#)および[DBA_PDBS](#)について学習するには、『Oracle Database』リファレンス』を参照

親トピック: [PDBスナップショット・カールセルについて](#)

PDBスナップショット・カールセル内のスナップショットの最大数の設定

PDBのPDBスナップショットの最大数を設定できます。

MAX_PDB_SNAPSHOTSデータベース・プロパティは、PDBスナップショット・カールセルでの各PDBのスナップショットの最大数を設定します。デフォルトの最大値は8です。プロパティには8より大きい数値を設定できません。

前提条件

PDBが読み取り/書き込みモードでオープンしている必要があります。

PDBのPDBスナップショットの最大数を設定するには:

1. SQL*Plusで、現在のコンテナが制限を設定するPDBであることを確認します。
2. 必要に応じて、SET MAX_PDB_SNAPSHOTSプロパティの現在の設定に対してCDB_PROPERTIESを問い合わせます。
3. SET MAX_PDB_SNAPSHOTS句を指定して、ALTER PLUGGABLE DATABASEまたはALTER DATABASE文を実行します。

例16-1 PDBのPDBスナップショットの最大数の設定

次の問合せは、cdb1_pdb1のカールセル内の最大値を示しています(出力例も示します)。

```
SET LINESIZE 150
COL ID FORMAT 99
COL PROPERTY_NAME FORMAT a17
COL PDB_NAME FORMAT a9
COL VALUE FORMAT a3
COL DESCRIPTION FORMAT a43

SELECT r.CON_ID AS id, p.PDB_NAME, PROPERTY_NAME,
       PROPERTY_VALUE AS value, DESCRIPTION
FROM   CDB_PROPERTIES r, CDB_PDBS p
WHERE  r.CON_ID = p.CON_ID
AND    PROPERTY_NAME LIKE 'MAX_PDB%'
```

```
ORDER BY PROPERTY_NAME;
```

ID	PDB_NAME	PROPERTY_NAME	VAL	DESCRIPTION
3	CDB1_PDB1	MAX_PDB_SNAPSHOTS	8	maximum number of snapshots for a PDB

次のSQL文は、現在のPDBのPDBスナップショットの最大数を7に設定します。

```
ALTER PLUGGABLE DATABASE SET MAX_PDB_SNAPSHOTS=7;
```

例16-2 PDBスナップショット・カルーセル内のすべてのスナップショットの削除

PDBスナップショット・カルーセル内のすべてのスナップショットを削除するには、次の文に示すように、MAX_PDB_SNAPSHOTSデータベース・プロパティを0 (ゼロ)に設定します。

```
ALTER PLUGGABLE DATABASE SET MAX_PDB_SNAPSHOTS=0;
```

この手法は、スナップショットごとにALTER PLUGGABLE DATABASE ... DROP SNAPSHOT snapshot_nameを実行するよりも高速です。

関連項目:

[「CDB内のコンテナへのアクセスについて」](#)

親トピック: [PDBスナップショット・カルーセルの管理](#)

自動PDBスナップショットの構成

PDBを作成または変更する際にSNAPSHOT MODE EVERY句を使用して、自動スナップショットのPDBを構成します。

デフォルトでは、PDBは手動スナップショットに構成されています。

前提条件

ALTER PLUGGABLE DATABASE SNAPSHOT文の次の前提条件に注意してください。

- CDBはローカルUNDOモードである必要があります。
- 管理者はPDBを作成および削除する権限を持っている必要があります。

PDBの変更時に自動スナップショットを構成するには:

1. SQL*Plusで、スナップショット・モードを構成するPDBに管理者としてログインしていることを確認します。
2. 必要に応じて、DBA_PDBSを問い合せて、現在のスナップショット・モードを確認します。
3. SNAPSHOT MODE EVERY interval句にMINUTESまたはHOURSを指定して、ALTER PLUGGABLE DATABASEを実行します。

PDBの作成時に自動スナップショットを構成するには:

1. SQL*Plusで、CDBルートまたはアプリケーション・ルートに管理者としてログインしていることを確認します。
2. 必要に応じて、DBA_PDBSを問い合せて、現在のスナップショット・モードを確認します。
3. SNAPSHOT MODE EVERY interval句にMINUTESまたはHOURSを指定して、CREATE PLUGGABLE DATABASEを実行します。

例16-3 既存のPDBに対する毎日の自動スナップショットの構成

この例では、スナップショット・モードを変更するPDBにログインしていることを想定しています。データ・ディクショナリを問い合わせ、現在、PDBがMANUALモードであることを確認します(出力例も示します)。

```
SELECT SNAPSHOT_MODE "S_MODE", SNAPSHOT_INTERVAL/60 "SNAP_INT_HRS"  
FROM   DBA_PDBS;
```

```
S_MODE SNAP_INT_HRS  
-----  
MANUAL
```

スナップショット・モードを24時間ごとに変更します。

```
ALTER PLUGGABLE DATABASE SNAPSHOT MODE EVERY 24 HOURS;
```

自動モードへの変更を確認します。

```
SELECT SNAPSHOT_MODE "S_MODE", SNAPSHOT_INTERVAL/60 "SNAP_INT_HRS"  
FROM   DBA_PDBS;
```

```
S_MODE SNAP_INT_HRS  
-----  
AUTO          24
```

例16-4 2時間ごとにスナップショットを取得するPDBの作成

この例では、CDBルートにログインしていることを想定しています。次の文は、cdb1_pdb1という名前の既存のPDBからのcdb1_pdb3を作成し、2時間ごとにスナップショットを自動的に取得するように構成します。

```
CREATE PLUGGABLE DATABASE cdb1_pdb3 FROM cdb1_pdb1  
  FILE_NAME_CONVERT=( ' cdb1_pdb1', ' cdb1_pdb3' )  
  SNAPSHOT MODE EVERY 120 MINUTES;
```

関連項目:

- [「PDBスナップショットからのPDBのクローニング: シナリオ」](#)
- [ローカルUNDOモードを使用するようにCDBを構成する方法](#)

親トピック: [PDBスナップショット・カールセルの管理](#)

PDBスナップショットの手動作成

PDBスナップショットを手動で作成するには、ALTER PLUGGABLE DATABASEまたはCREATE PLUGGABLE DATABASEでSNAPSHOT snapshot_name句を指定します。

前提条件

ALTER PLUGGABLE DATABASE SNAPSHOT文の次の前提条件に注意してください。

- CDBはローカルUNDOモードである必要があります。次の問合せを使用すると、モードを確認できます。ローカルUNDOが有効な場合、この問合せはTRUEを返します。

```
SELECT * FROM DATABASE_PROPERTIES WHERE PROPERTY_NAME=' LOCAL_UNDO_ENABLED' ;
```

- DBAは、PDBの作成および削除の権限を持っている必要があります。
- スナップショットをスパースにする場合は、基礎となるストレージ・システムでスパース・ファイルがサポートされている必要があります。この場合、最初のスナップショットのみがフルになります。

PDBスナップショットを作成するには:

1. SQL*Plusで、スナップショット・モードを作成するPDBに管理者としてログインしていることを確認します。
2. 必要に応じて、DBA_PDBS. SNAPSHOT_MODEを問い合せて、スナップショットのモードがNONEに設定されていないことを確認します。
3. SNAPSHOT句を指定してALTER PLUGGABLE DATABASE文を実行します。

例16-5 ユーザー指定の名前を使用したPDBスナップショットの作成

次のSQL文は、cdb1_pdb1の2つのPDBスナップショット(水曜日のデータ・ロードの前と後に1つずつ)を作成します。

```
ALTER PLUGGABLE DATABASE SNAPSHOT cdb1_pdb1_b4WLOAD;
-- data load
ALTER PLUGGABLE DATABASE SNAPSHOT cdb1_pdb1_afWLOAD;
```

DBA_PDB_SNAPSHOTSの次の問合せは、cdb1_pdb1という名前のPDBの2つのスナップショットの場所を示しています(出力例も示します)。

```
SET LINESIZE 150
COL CON_NAME FORMAT a9
COL ID FORMAT 99
COL SNAPSHOT_NAME FORMAT a17
COL SNAP_SCN FORMAT 9999999
COL FULL_SNAPSHOT_PATH FORMAT a61

SELECT CON_ID AS ID, CON_NAME, SNAPSHOT_NAME,
       SNAPSHOT_SCN AS snap_scn, FULL_SNAPSHOT_PATH
FROM   DBA_PDB_SNAPSHOTS
ORDER BY SNAP_SCN;
```

ID	SNAPSHOT_NAME	SNAP_SCN	FULL_SNAPSHOT_PATH
4	CDB1_PDB1_B4WLOAD	5056465	/ade/b/813544604/oracle/dbs/snapshots/pdb_2935056285/5056465/
4	CDB1_PDB1_AFWLOAD	5056501	/ade/b/813544604/oracle/dbs/snapshots/pdb_2935056285/5056501/

PDBスナップショット名を指定しない場合、データベースでは一意の名前が生成されます。

例16-6 システム指定の名前を使用したPDBスナップショットの作成

次のSQL文は、スナップショットは作成されますが、名前が指定されていません。

```
ALTER PLUGGABLE DATABASE SNAPSHOT;
```

次の問合せ例は、データベースでSNAP_という接頭辞が付いた名前をスナップショットに割り当てることを示しています。

```
SET LINESIZE 150
COL CON_NAME FORMAT a9
COL ID FORMAT 99
COL SNAPSHOT_NAME FORMAT a26
COL SNAP_SCN FORMAT 9999999
COL FULL_SNAPSHOT_PATH FORMAT a61

SELECT CON_ID AS id, CON_NAME, SNAPSHOT_NAME,
       SNAPSHOT_SCN AS snap_scn, FULL_SNAPSHOT_PATH
```

```
FROM DBA_PDB_SNAPSHOTS
ORDER BY SNAP_SCN;
```

ID	SNAPSHOT_NAME	SNAP_SCN	FULL_SNAPSHOT_PATH
4	CDB1_PDB1_B4WLOAD	5056465	/ade/b/813544604/oracle/dbs/snapshots/pdb_2935056285/5056465/
4	CDB1_PDB1_AFWLOAD	5056501	/ade/b/813544604/oracle/dbs/snapshots/pdb_2935056285/5056501/
4	SNAP_2935056285_1031574118	5057389	/ade/b/813544604/oracle/dbs/snapshots/pdb_2935056285/5057389/

関連項目:

- [「CDB内のコンテナへのアクセスについて」](#)
- [ローカルUNDOモードを使用するようにCDBを構成する方法](#)

親トピック: [PDBスナップショット・カルーセルの管理](#)

PDBスナップショットの削除

DROP SNAPSHOT句を指定したALTER PLUGGABLE DATABASE文を実行して、PDBスナップショットを削除できます。

PDBに基づくPDBスナップショットをすべて削除するには、PDBのMAX_PDB_SNAPSHOTSプロパティを0 (ゼロ)に設定します。

PDBスナップショットを削除するには:

1. SQL*Plusで、現在のコンテナがPDBスナップショットを作成したPDBであることを確認します。
2. DROP SNAPSHOT句を指定してALTER PLUGGABLE DATABASE文を実行します。

例16-7 PDBスナップショットの削除

次のSQL文は、sales_snapという名前のPDBスナップショットを削除します。

```
ALTER PLUGGABLE DATABASE DROP SNAPSHOT sales_snap;
```

関連項目:

[「CDB内のコンテナへのアクセスについて」](#)

親トピック: [PDBスナップショット・カルーセルの管理](#)

PDBスナップショットのメタデータの表示

データ・ディクショナリ・ビューのDBA_PDB_SNAPSHOTSとDBA_PDB_SNAPSHOTFILEでは、PDBスナップショットのメタデータが示されます。

DBA_PDB_SNAPSHOTSには、スナップショットに関する一般情報(名前、SCN、時間、パスなど)が含まれています。

DBA_PDB_SNAPSHOTFILEでは、スナップショット内のすべてのファイルのパスとファイル名(データファイル、アーカイブREDOログ・ファイルおよびXMLファイル)が示されます。

ノート:



DBA_PDB_SNAPSHOTFILE では、スパース・クローン PDB のみが示されます。スパース・クローンを作成するには、CLONEDB 初期化パラメータを TRUE に設定する必要があります。

PDBスナップショットのメタデータを表示するには:

1. SQL*Plusで、管理ユーザーとしてデータベースにログインします。
2. DBA_PDB_SNAPSHOTSを問い合わせます。

たとえば、次の問合せを実行します(出力例も示します)。

```
COL SNAPSHOT_NAME FORMAT a30
SELECT SNAPSHOT_NAME, SNAPSHOT_SCN, SNAPSHOT_TIME FROM DBA_PDB_SNAPSHOTS;
```

SNAPSHOT_NAME	SNAPSHOT_SCN	SNAPSHOT_TIME
HRPDB_SNAP_F	3678939	1536262569
HRPDB_SNAP_S	4954803	986473745

3. DBA_PDB_SNAPSHOTFILEを問い合わせます。

たとえば、次の結合問合せを実行します(出力例も示します)。

```
SET LINESIZE 120
COL SNAPSHOT_NAME FORMAT a12
COL SNAPSHOT_FILENAME FORMAT a54

SELECT SNAPSHOT_NAME, SNAPSHOT_FILENAME, SNAPSHOT_FILETYPE AS TYPE
FROM DBA_PDB_SNAPSHOTS s, DBA_PDB_SNAPSHOTFILE f
WHERE s. SNAPSHOT_SCN=f. SNAPSHOT_SCN;
```

SNAPSHOT_NAM	SNAPSHOT_FILENAME	TYPE
HRPDB_SNAP_S	/d1/snapshots/4954803/o1_mf_undo_1_fry115bq_. dbf	DATA
HRPDB_SNAP_S	/d1/snapshots/4954803/o1_mf_salestbs_fry19m6h_. dbf	DATA
HRPDB_SNAP_S	/d1/snapshots/4954803/o1_mf_sysext_fry19d1n_. dbf	DATA
HRPDB_SNAP_S	/d1/snapshots/4954803/o1_mf_sysaux_fry19d1m_. dbf	DATA
HRPDB_SNAP_S	/d1/snapshots/4954803/o1_mf_system_fry19d1k_. dbf	DATA
HRPDB_SNAP_S	/d1/snapshots/4954803/HRPDB. xml	XML
HRPDB_SNAP_S	/d1/snapshots/4954803/archpar log_1_274_b87ca51e_985963 814. arc	ARCH

例16-8 完全PDBスナップショットのメタデータの問合せ

次の問合せは、2つのPDBスナップショットを示します。 .pdb拡張子で示されるように、スナップショットはスパースではなくフルです。

```
SET LINESIZE 200
SET PAGESIZE 50000

COL ID FORMAT 99
COL CON_NAME FORMAT a7
COL SNAPSHOT_NAME FORMAT a25
COL SNAPSHOT_SCN FORMAT a7
COL FULL_SNAPSHOT_PATH FORMAT a65

SELECT CON_ID AS ID, CON_NAME, SNAPSHOT_NAME,
```

```

        SNAPSHOT_SCN, FULL_SNAPSHOT_PATH
FROM    DBA_PDB_SNAPSHOTS;

ID CON_NAM SNAPSHOT_NAME          SNAPSHO FULL_SNAPSHOT_PATH
-----
5 HRPDB    SNAP_3286480866_994766895 3160319 /d1/snap_3286480866_3160319.pdb
5 HRPDB    SNAP_3286480866_994767095 3165758 /d1/snap_3286480866_3165758.pdb

```

次のDBA_PDB_SNAPSHOTFILEの問合せでは、行が返されません。このビューはPDBスナップショットがスペースの場合にのみ移入されるためです。

```
SQL> SELECT COUNT(*) FROM DBA_PDB_SNAPSHOTFILE;
```

```

COUNT(*)
-----
0

```

親トピック: [PDBスナップショット・カルーセルの管理](#)

17 アプリケーション・コンテナの管理

アプリケーション・ルートとアプリケーションPDBを含むアプリケーション・コンテナを管理できます。アプリケーション・コンテナにインストールされているアプリケーションも管理できます。



ノート:

この章のタスクは、SQL*Plus または Oracle SQL Developer を使用して実行できます。

- [アプリケーション・コンテナの管理について](#)
アプリケーション・コンテナの管理の一部の側面はCDBルートおよびCDB全体の管理に似ていますが、その他の側面はPDBの管理に似ています。
- [アプリケーション・ルートの変更について](#)
ALTER DATABASE文でアプリケーション・ルートを変更できます。ALTER PLUGGABLE DATABASE文では、アプリケーションPDBのオープン・モードを変更できます。
- [アプリケーション・コンテナ内のアプリケーションの管理](#)
アプリケーション・コンテナ内のアプリケーションをインストール、アップグレードまたはパッチ適用します。
- [アプリケーション共通オブジェクトの管理](#)
アプリケーション共通オブジェクトは、アプリケーション・コンテナ内の共有されるユーザー作成データベース・オブジェクトです。アプリケーション共通オブジェクトは、アプリケーション・ルートに作成されます。
- [アプリケーション・コンテナ内のコンテナに対するDML文の発行](#)
アプリケーション・ルートで発行されるDML文では、アプリケーション・コンテナ内の1つ以上のコンテナを変更できます。また、DML文に対して1つ以上のデフォルト・コンテナ・ターゲットを指定できます。
- [コンテナ・マップを使用したPDBでのパーティション化](#)
コンテナ・マップにより、データが表レベルで物理的にパーティション化されていない場合にデータをアプリケーションPDBレベルでパーティション化できます。

関連トピック

- [アプリケーション・コンテナおよびシードの作成と削除](#)
- [マルチテナント環境のツール](#)

親トピック: [マルチテナント環境の管理](#)

アプリケーション・コンテナの管理について

アプリケーション・コンテナの管理の一部の側面はCDBルートおよびCDB全体の管理に似ていますが、その他の側面はPDBの管理に似ています。

アプリケーション・コンテナの管理は、アプリケーション・ルートとアプリケーション・ルートに接続されるアプリケーションPDBの両方を管理できるため、CDBの管理と似ています。ただし、アプリケーション・コンテナの管理は、アプリケーション・コンテナに対する変更が他のアプリケーション・コンテナやCDB内のPDBに影響しないため、PDBの管理にも似ています。

次の表では、CDBまたはCDBルートを管理する管理タスクに似たアプリケーション・コンテナの管理タスクについて説明します。

表17-1 CDBの管理に似たアプリケーション・コンテナ管理タスク

管理タスク	説明	詳細情報
-------	----	------

管理タスク	説明	詳細情報
アプリケーション共通ユーザーおよび共通して付与される権限の構成	アプリケーション共通ユーザーおよび権限は、CDB ルートの共通ユーザーおよび共通して付与される権限に似ていますが、アプリケーション・コンテナでは、共通ユーザーおよび共通して付与される権限は、アプリケーション・ルート、アプリケーション・ルートに属するアプリケーション PDB、アプリケーション・ルートに属するオプションのアプリケーション・シードなど、アプリケーション・コンテナのコンテナ内のみ存在します。	「CDB の共通ユーザーおよびローカル・ユーザーの概要」
アプリケーション・コンテナの作成	現在のコンテナが CDB ルートである共通ユーザーは、CREATE PLUGGABLE DATABASE 文に AS APPLICATION CONTAINER 句を指定することで CDB ルートに接続するアプリケーション・コンテナを作成できます。データベース・ファイルは、Oracle Managed Files であることが必要です。	「アプリケーション・コンテナの作成」
アプリケーション PDB の作成	現在のコンテナがアプリケーション・ルートである共通ユーザーは、アプリケーション・ルートに接続されるアプリケーション PDB を作成できます。	「PDB およびアプリケーション・コンテナの作成および削除」
コンテナへの切替え	適切な権限を持つユーザーは、アプリケーション・ルート、アプリケーション・ルートに属するアプリケーション PDB、アプリケーション・ルートに属するオプションのアプリケーション・シードなど、アプリケーション・コンテナ内のコンテナ間を切り替えることができます。	「ALTER SESSION 文を使用したコンテナへの切替え」
ALTER SYSTEM SET 文の発行	ALTER SYSTEM SET 文では、アプリケーション・コンテナ内の 1 つ以上のコンテナの初期化パラメータを動的に設定できます。	「ALTER SYSTEM を使用した CDB の変更」
データ定義言語 (DDL) 文の発行	アプリケーション・コンテナでは、DDL 文をアプリケーション・コンテナ内のすべてのコンテナに適用できる場合も、現在のコンテナにのみ適用できる場合もあります。	「DDL 文でのアプリケーション共通オブジェクトの変更」

次の表では、PDBを管理する管理タスクに似たアプリケーション・コンテナの管理タスクについて説明します。

表17-2 PDBの管理に似たアプリケーション・コンテナ管理タスク

管理タスク	説明	詳細情報
アプリケーション・ルートへの接続	アプリケーション・ルートには固有のサービス名があり、ユーザーは PDB に接続するのと同じ方法でアプリケーション・ルートに接続できます。同様に、各アプリケーション PDB には固有のサービス名があり、アプリケーション・シードには固有のサービス名があります。	「CDB 内のコンテナへのアクセス」
ALTER PLUGGABLE	ALTER PLUGGABLE DATABASE 文では、PDB を変更するの	「CDB ルートへの接続時のコンテ

管理タスク	説明	詳細情報
DATABASE 文の発行	と同じ方法でアプリケーション・ルート、アプリケーション PDB およびアプリケーション・シードを変更できます。たとえば、管理者は ALTER PLUGGABLE DATABASE 文でアプリケーション・ルートをオープンまたはクローズできます。	ナの変更 「データベース・レベルでの PDB の変更」
SQL *Plus の STARTUP および SHUTDOWN コマンドの発行	SQL *Plus の STARTUP および SHUTDOWN コマンドは、PDB を操作するのと同じ方法でアプリケーション・ルート、アプリケーション PDB およびアプリケーション・シードを操作します。	「PDB のオープン・モードの変更」
ALTER SYSTEM 文の発行	ALTER SYSTEM 文は、PDB を操作するのと同じ方法でアプリケーション・ルート、アプリケーション PDB およびアプリケーション・シードを操作します。	「ALTER SYSTEM を使用した CDB の変更」 「システム・レベルでの PDB の変更」
表領域の管理	管理者は、アプリケーション・ルートおよびアプリケーション PDB の表領域を作成、変更および削除できます。各コンテナには固有の表領域があります。	「CDB 内表領域の管理について」
データファイルおよび一時ファイルの管理	管理者は、アプリケーション・ルートおよびアプリケーション PDB のデータファイルおよび一時ファイルを作成、変更および削除できます。各コンテナには固有のファイルがあります。	データファイルおよび一時ファイルの管理の詳細は、『 Oracle Database 管理者ガイド 』を参照してください
スキーマ・オブジェクトの管理	PDB で行うのと同じ方法で、アプリケーション・ルートおよび各アプリケーション PDB 内のスキーマ・オブジェクトを作成、変更および削除できます。特定のアプリケーション・ルートまたはアプリケーション PDB で起動するトリガーを作成することもできます。 ただし、アプリケーション・コンテナではアプリケーション共通オブジェクトがサポートされ、アプリケーション・コンテナ内のコンテナ間で共有できます。アプリケーション共通オブジェクトは PDB に作成できません。	アプリケーション共通オブジェクトの管理

親トピック: [アプリケーション・コンテナの管理](#)

アプリケーション・ルートの変更について

ALTER DATABASE文でアプリケーション・ルートを変更できます。ALTER PLUGGABLE DATABASE文では、アプリケーションPDBのオープン・モードを変更できます。

次の表に、アプリケーション・ルートで発行されるALTER DATABASEおよびALTER PLUGGABLE DATABASE文の句によって変更さ

れるコンテナを示します。表には、アプリケーション・ルートで許可されない文もリストしています。

ノート:



現在のコンテナがアプリケーション・ルートであるときに発行された文は、CDB ルート、または現在のアプリケーション・ルートに属していない PDB には影響しません。

表17-3 アプリケーション・ルートのコンテナを変更する文

アプリケーション・ルートのみの変更	1つ以上のアプリケーションPDBの変更	アプリケーション・ルートで発行できない
<p>現在のコンテナがアプリケーション・ルートであるアプリケーション共通ユーザーとして接続した場合、次の句を含む ALTER DATABASE 文を使用すると、アプリケーション・ルートのみが変更されます。</p>	<p>現在のコンテナがアプリケーション・ルートであるアプリケーション共通ユーザーとして接続した場合、次の句を含む ALTER PLUGGABLE DATABASE 文を使用すると、1 つ以上のアプリケーション PDB のオープン・モードを変更できます。</p>	<p>現在のコンテナがアプリケーション・ルートであるアプリケーション共通ユーザーとして接続した場合、次の句を含む ALTER DATABASE 文は使用できません。</p>
<ul style="list-style-type: none"> ● database_file_clauses ● DEFAULT EDITION 句 ● DEFAULT TABLESPACE 句 ● DEFAULT TEMPORARY TABLESPACE 句 	<ul style="list-style-type: none"> ● pdb_change_state <p>現在のコンテナがアプリケーション PDB である場合、この句を含む ALTER PLUGGABLE DATABASE 文を使用すると、現在のアプリケーション PDB のオープン・モードを変更できます。</p>	<ul style="list-style-type: none"> ● startup_clauses ● recovery_clauses ● logfile_clauses ● controlfile_clauses ● standby_database_clauses ● instance_clauses ● security_clause ● RENAME GLOBAL_NAME 句 ● ENABLE BLOCK CHANGE TRACKING 句 ● DISABLE BLOCK CHANGE TRACKING 句
<p>次の句を含む ALTER DATABASE 文を使用すると、アプリケーション・ルートが変更され、アプリケーション PDB にデフォルト値が設定されます。</p>	<p>現在のコンテナがアプリケーション・ルートであるアプリケーション共通ユーザーとして接続した場合、次の句を含む ALTER PLUGGABLE DATABASE 文を使用すると、CDB の再起動時にアプリケーション PDB のオープン・モードを保持することも破棄することもできます。</p>	
<ul style="list-style-type: none"> ● flashback_mode_clause ● SET DEFAULT {BIGFILE SMALLFILE} TABLESPACE 句 ● set_time_zone_clause 	<ul style="list-style-type: none"> ● pdb_save_or_discard_state 	
<p>これらの句を使用すると、特定のアプリケーション PDB にデフォルト以外の値を設定できます。</p>		

関連項目:

- [「現在のコンテナについて」](#)

- [「データベース・レベルでのPDBの変更」](#)
- [Oracle Database SQL言語リファレンス](#)

親トピック: [アプリケーション・コンテナの管理](#)

アプリケーション・コンテナ内のアプリケーションの管理

アプリケーション・コンテナ内にアプリケーションをインストール、アップグレードまたはパッチ適用します。

アプリケーション・コンテナからアプリケーションをアンインストールすることもできます。アプリケーション・ルートでこれらの操作を実行します。アプリケーションPDBがアプリケーション・ルート内のアプリケーションと同期するときに、アプリケーション・コンテナによってアプリケーションの変更がアプリケーションPDBに伝播されます。

- [アプリケーションの管理について](#)
アプリケーション・コンテナでは、**アプリケーション**は名前付きのバージョン管理されたアプリケーション・メタデータおよび共通データのセットです。アプリケーションはアプリケーション・ルートに格納されます。
- [アプリケーション・コンテナへのアプリケーションのインストール](#)
アプリケーション・コンテナにアプリケーションをインストールできます。
- [アプリケーション・コンテナ内のアプリケーションのアップグレード](#)
アプリケーションに対する大きな変更によってアプリケーション・アップグレードが構成されます。アプリケーション・コンテナのアプリケーションをアップグレードできます。
- [アプリケーション・コンテナ内のアプリケーションのパッチ適用](#)
アプリケーションに対する小さな変更によってアプリケーション・パッチが構成されます。
- [アプリケーション・コンテナへの既存のアプリケーションの移行](#)
PDBにインストールされているアプリケーションをアプリケーション・コンテナに移行できます。
- [アプリケーションPDB内のアプリケーションの同期](#)
アプリケーションの同期により、アプリケーションPDB内のアプリケーションがアプリケーション・ルートの最新バージョンおよびパッチに更新されます。
- [アプリケーション・ルート・レプリカとプロキシPDBの同期](#)
異なるCDB内のアプリケーション・コンテナが同じアプリケーションを持つ場合、それらのアプリケーション・ルートは、マスター・アプリケーション・ルート、レプリカ・アプリケーション・ルートおよびプロキシPDBを作成することで同期を維持できます。
- [アプリケーションの互換バージョンの設定](#)
アプリケーションの互換バージョンは、アプリケーション・コンテナに属するアプリケーションPDBで使用可能なアプリケーションの最も古いバージョンです。
- [アプリケーションのインストール、アップグレードおよびパッチ適用の操作中の一括挿入の実行](#)
SQL*Loaderは、アプリケーションのインストール、アップグレードおよびパッチ適用の操作中に表に一括挿入することがサポートされる唯一のユーティリティです。従来型パス・ロードのみ、アプリケーションのインストール、アップグレードおよびパッチ適用の操作中に一括挿入することがサポートされます。
- [アプリケーション・コンテナからのアプリケーションのアンインストール](#)
アプリケーション・コンテナ内のアプリケーションをアンインストールできます。

関連トピック

- [アプリケーション・コンテナ内のアプリケーションの概要](#)

親トピック: [アプリケーション・コンテナの管理](#)

アプリケーション管理について

アプリケーション・コンテナでは、**アプリケーション**は名前付きのバージョン管理されたアプリケーション・メタデータおよび共通データのセットです。アプリケーションはアプリケーション・ルートに格納されます。

このコンテキストでは、"アプリケーション"という用語は"アプリケーション・バックエンド"を意味します。アプリケーション共通オブジェクトには、ユーザー・アカウント、表、PL/SQLパッケージなどが含まれます。アプリケーションは、アプリケーション・ルートに属するアプリケーションPDBと共有できます。アプリケーション変更を実行した場合、アプリケーションPDBはアプリケーション・ルート内のアプリケーションと同期できます。

- [アプリケーション・メンテナンスの基本的なステップ](#)
アプリケーション・ルート内にアプリケーションをインストール、アップグレードおよびパッチ適用できます。
- [アプリケーションのバージョン](#)
アプリケーション・コンテナは、アプリケーションのバージョンとアプリケーションのパッチも管理します。
- [アプリケーション・モジュール名およびサービス名](#)
アプリケーション・モジュール名は、DBMS_APPLICATION_INFO.SET_MODULEプロシージャによって設定されるか、それに相当するOCI属性の設定に従って設定されます。

親トピック: [アプリケーション・コンテナ内のアプリケーションの管理](#)

アプリケーション・メンテナンスの基本的なステップ

アプリケーション・ルート内にアプリケーションをインストール、アップグレードおよびパッチ適用できます。

ALTER PLUGGABLE DATABASE ... BEGIN文を発行して操作を開始し、ALTER PLUGGABLE DATABASE ... END文を発行して操作を終了する必要があります。これらの文は、同じユーザー・セッションまたは別のユーザー・セッションで発行できます。

次に、アプリケーション・コンテナ内のアプリケーションを作成およびメンテナンスするための標準的なプロセスを示します。

1. アプリケーション・コンテナを作成します。
2. ALTER PLUGGABLE DATABASE ... BEGIN INSTALLを使用してアプリケーション・ルートにアプリケーションをインストールします。

このステップには、アプリケーション・データ・モデルの作成と、アプリケーション共通ユーザーおよびアプリケーション共通オブジェクトの構成が含まれます。



ノート:

SQL*Loader は、アプリケーションのインストール、アップグレードおよびパッチ適用の操作中に表に一括挿入することがサポートされる唯一のユーティリティです。

3. アプリケーション・ルートにアプリケーションPDBを作成します。
4. アプリケーションをインストールする必要がある各アプリケーションPDBをアプリケーション・ルートと同期します。使用する文はALTER PLUGGABLE DATABASE APPLICATION ... SYNCです。
5. 各アプリケーションPDBのデータをロードします。
6. アプリケーションをメンテナンスします。ALTER PLUGGABLE DATABASE ... BEGIN UPGRADEを使用してアップグレードし、ALTER PLUGGABLE DATABASE ... BEGIN PATCHを使用してパッチ適用します。
7. アップグレードとパッチから変更を適用する必要があるアプリケーションPDBを同期します。

8. 必要な場合に新規アプリケーションPDBを追加します。

9. 必要に応じて、ALTER PLUGGABLE DATABASE ... BEGIN UNINSTALLを使用してアプリケーションをアンインストールします。

関連項目:

- [「アプリケーション・コンテナの作成」](#)
- アプリケーション・メンテナンス操作の監査方法を学習するには、『[Oracle Databaseセキュリティ・ガイド](#)』を参照してください。

親トピック: [アプリケーションの管理について](#)

アプリケーションのバージョン

アプリケーション・コンテナは、アプリケーションのバージョンとアプリケーションのパッチも管理します。

アプリケーション・コンテナは、次のようにバージョンを管理します。

- アプリケーションをインストールする場合は、アプリケーション・バージョン番号を指定する必要があります。
- アプリケーションをアップグレードする場合は、古いアプリケーション・バージョン番号と新しいアプリケーション・バージョン番号を指定する必要があります。
- アプリケーションにパッチを適用する場合は、パッチの最小アプリケーション・バージョン番号とパッチ番号を指定する必要があります。

アプリケーションが進化するときに、アプリケーション・コンテナでは適用するすべてのバージョンとパッチ変更が保持されます。

異なるアプリケーションPDBが異なるアプリケーション・バージョンを使用するように、アプリケーション・コンテナを構成することもできます。たとえば、様々な顧客にアプリケーションを提供し、各顧客が独自のアプリケーションPDBを持つ場合、一部の顧客がアプリケーションのアップグレードを待つ時間が長くなることがあります。この場合、一部のアプリケーションPDBは最新バージョンのアプリケーションを使用できますが、他のアプリケーションPDBは古いバージョンのアプリケーションを使用できます。

親トピック: [アプリケーションの管理について](#)

アプリケーション・モジュール名およびサービス名

アプリケーション・モジュール名は、DBMS_APPLICATION_INFO.SET_MODULEプロシージャによって設定されるか、それに相当するOCI属性の設定に従って設定されます。

アプリケーション・メンテナンスの際には、データベースで他のアクティビティが発生している可能性があるため、モジュール名が必要です。たとえば、バックグラウンドプロセスによって発行された文はアプリケーション取得表で取得できません。また、他のユーザーがアプリケーションに関係ない文を実行している可能性もあります。モジュール名のチェックによって、取得される文を取得されない文から区別できます。APPLICATION BEGINが発行されたセッションとモジュール名が一致するセッションのみが取得対象とみなされます。

DBA_APPLICATIONSを問い合せて、APPLICATION BEGINが実行されたセッションのモジュール名を確認します。

```
SELECT app_capture_module FROM dba_applications WHERE app_name='APEX';
```

一部の句(SHARING句など)は、ALTER PLUGGABLE DATABASE ... BEGIN文とALTER PLUGGABLE DATABASE ... END文の間で発行された場合にのみ有効です。これらの句に関してセッションのモジュール名が一致しない場合、そのセッションは

BEGIN文とEND文の間に含まれないため、その句を含む文はORA-65021などのエラーで失敗します。

モジュール名が一致しない最も一般的な原因は、デフォルトのモジュール名です。たとえば、SQL*Plusではデータベースへの接続時にデフォルトのモジュール名が設定されます。SYSDBAユーザーとして接続したときに1つのデフォルト・モジュール名(たとえば、sqlplus@host1 (TNS V1-V3))が生成されますが、非SYSDBAユーザーとして接続したときは別のデフォルト・モジュール名(たとえば、SQL*Plus)が生成されます。SYSDBAユーザーと非SYSDBAユーザーの両方がメンテナンスを実行する場合は、SQL*Plusのデフォルト設定を利用せずに、各セッションで明示的にモジュール名を同じ値に設定する必要があります。

また、文をキャプチャするには、文を実行しているセッションのサービス名が、APPLICATION BEGINが実行されたセッションのサービス名と一致している必要があります。DBA_APPLICATIONSを問い合せて、APPLICATION BEGINが実行されたセッションのサービス名を確認します。

```
SELECT app_capture_service FROM dba_applications WHERE app_name='APEX';
```

例17-1 セッションのモジュール名のチェック

この例は、接続したユーザーにSYSDBA権限があるかどうかによってモジュール名が変わることを示しています。

```
SQL> CONNECT / AS SYSDBA
Connected.

SQL> select module from v$session where auid = SYS_CONTEXT('USERENV', 'sessionid');

MODULE
-----
sqlplus@host1 (TNS V1-V3)

SQL> CONNECT dba1
Password: *****
Connected.

SQL> select module from v$session where auid = SYS_CONTEXT('USERENV', 'sessionid');

MODULE
-----
SQL*Plus
```

関連項目:

アプリケーション・モジュール名の設定方法を学習するには、『[Oracle Database PL/SQLパッケージおよびタイプ・リファレンス](#)』を参照してください。

親トピック: [アプリケーションの管理について](#)

アプリケーション・コンテナへのアプリケーションのインストール

アプリケーション・コンテナにアプリケーションをインストールできます。

- [アプリケーション・コンテナへのアプリケーションのインストールについて](#)
アプリケーション・ルートにアプリケーションをインストールするには、ALTER PLUGGABLE DATABASE APPLICATION文を発行します。
- [自動伝播によるアプリケーション・コンテナ内のアプリケーションのインストール](#)
自動伝播によって、アプリケーションは、アプリケーション・ルート内のアプリケーションと同期するアプリケーションPDBにイ

インストールされます。

親トピック: [アプリケーション・コンテナ内のアプリケーションの管理](#)

アプリケーション・コンテナ内のアプリケーションのインストールについて

アプリケーション・ルートにアプリケーションをインストールするには、ALTER PLUGGABLE DATABASE APPLICATION文を発行します。

アプリケーションはアプリケーション・ルートにのみインストールします。アプリケーションと同期するアプリケーションPDBでは、アプリケーションが自動的にインストールされます。自動による方法では、スクリプト、SQL文およびグラフィカル・ユーザー・インタフェース・ツールから1つ以上の手法を使用してインストールを実行できます。

ALTER PLUGGABLE DATABASE APPLICATION BEGIN INSTALL文でインストールを開始し、ALTER PLUGGABLE DATABASE APPLICATION END INSTALL文でインストールを終了します。各インストールには、ALTER PLUGGABLE DATABASE APPLICATION文で指定されるアプリケーション名とバージョン番号を関連付ける必要があります。

関連トピック

- [CDBでのオラクル社が提供するSQLスクリプトの実行](#)

親トピック: [アプリケーション・コンテナへのアプリケーションのインストール](#)

自動伝播によるアプリケーション・コンテナ内のアプリケーションのインストール

自動伝播によって、アプリケーションは、アプリケーション・ルート内のアプリケーションと同期するアプリケーションPDBにインストールされます。

前提条件

次の前提条件を満たしている必要があります。

- 現在のユーザーにはALTER PLUGGABLE DATABASEシステム権限が必要で、権限は一般にアプリケーション・ルートで付与されている必要があります。
- アプリケーション・ルートは読取り/書込みオープンである必要があります。

自動伝播を使用してアプリケーションをインストールするには:

1. SQL*PlusまたはSQL Developerで、現在のコンテナがPDBであることを確認します。
2. 次の形式でALTER PLUGGABLE DATABASE APPLICATION BEGIN INSTALL文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name BEGIN INSTALL  
'application_version_number';
```

たとえば、application_nameがsalesappでapplication_version_numberが4.2の場合は次の文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp BEGIN INSTALL '4.2';
```

3. スクリプト、SQL文またはグラフィカル・ユーザー・インタフェース・ツールを使用してアプリケーションをインストールします。
4. 次の形式でALTER PLUGGABLE DATABASE APPLICATION END INSTALL文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name END INSTALL  
'application_version_number';
```

たとえば、application_nameがsalesappでapplication_version_numberが4.2の場合は次の文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp END INSTALL '4.2';
```



ノート:

application_name および application_version_number が ALTER PLUGGABLE DATABASE APPLICATION BEGIN INSTALL 文と ALTER PLUGGABLE DATABASE APPLICATION END INSTALL 文で一致していることを確認します。

5. SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATIONを発行して、アプリケーションをインストールする必要のあるすべてのアプリケーションPDBを同期します。

関連トピック

- [CDB内のコンテナへのアクセス](#)
- [アプリケーションPDB内のアプリケーションの同期](#)

親トピック: [アプリケーション・コンテナへのアプリケーションのインストール](#)

アプリケーション・コンテナ内のアプリケーションのアップグレード

アプリケーションに対する大きな変更によってアプリケーション・アップグレードが構成されます。アプリケーション・コンテナのアプリケーションをアップグレードできます。

- [アプリケーション・コンテナ内のアプリケーションのアップグレードについて](#)
アプリケーション・ルート内のアプリケーションをアップグレードするには、ALTER PLUGGABLE DATABASE APPLICATION文を発行します。
- [アプリケーション・コンテナ内のアプリケーションのアップグレード](#)
アップグレード後に、アップグレードによって発生したアプリケーションの変更は、アプリケーション・ルートと同期されるアプリケーションPDBに伝播されます。

親トピック: [アプリケーション・コンテナ内のアプリケーションの管理](#)

アプリケーション・コンテナ内のアプリケーションのアップグレードについて

アプリケーション・ルート内のアプリケーションをアップグレードするには、ALTER PLUGGABLE DATABASE APPLICATION文を発行します。

- [アプリケーション・アップグレードの目的](#)
アプリケーション定義は、アプリケーション・ルート内で1回アップグレードできます。これにより、アップグレードした定義を他のアプリケーションPDBと同期できます。
- [アプリケーションのアップグレードの動作](#)
アプリケーションのアップグレード時に、Oracle Databaseによってアプリケーション・ルートが自動的にクローニングされます。
- [アプリケーション・アップグレードのためのユーザー・インタフェース](#)
アプリケーション・ルート内のアプリケーション定義をアップグレードするには、ALTER PLUGGABLE DATABASE APPLICATION ... UPGRADEコマンドを使用します。

関連トピック

- [CDBでのオラクル社が提供するSQLスクリプトの実行](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションのアップグレード](#)

アプリケーション・アップグレードの目的

アプリケーション定義は、アプリケーション・ルート内で1回アップグレードできます。これにより、アップグレードした定義を他のアプリケーションPDBと同期できます。

アプリケーションPDBは、アプリケーション・ルート内のアップグレードされたアプリケーション定義を自動的に継承しません。SYNC句を指定してALTER PLUGGABLE DATABASE文を手動実行すると、アプリケーションPDBがルート内のアプリケーションと同期されます。スクリプト、SQL文およびグラフィカル・ユーザー・インタフェース・ツールから1つ以上の手法を使用してアップグレードできます。

親トピック: [アプリケーション・コンテナ内のアプリケーションのアップグレードについて](#)

アプリケーションのアップグレードの動作

アプリケーションのアップグレード時に、Oracle Databaseによってアプリケーション・ルートが自動的にクローニングされます。

アップグレード中、アプリケーションPDBはルート・クローンを指しています。アップグレード中、アプリケーションは実行を継続します。アプリケーションPDBは、メタデータリンクおよび拡張データリンクの表およびビューに対してDMLを実行できます。アプリケーションPDBは、メタデータリンク・オブジェクト、拡張データリンク・オブジェクトおよびデータリンク・オブジェクトを問い合わせることができます。

アップグレード後、アプリケーション・ルート・クローンはそのまま残り、ルート・クローン内にあるアップグレード前のバージョンのアプリケーションを使用するアプリケーションPDBを引き続きサポートします。アップグレードしたアプリケーションPDBは、アップグレードされたアプリケーション・ルートを指しています。アップグレードされないアプリケーションPDBは引き続きクローンを使用でき、アプリケーション・ルートに接続されているアプリケーションPDBも、ルート・クローンと同じアプリケーション・バージョンを使用できます。

ノート:

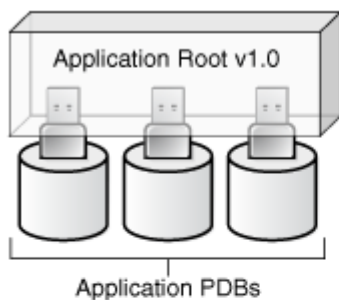


アプリケーションのアップグレードとは異なり、パッチ適用ではアプリケーション・ルート・クローンが作成されません。パッチ適用後にアプリケーション PDB が同期化されていない場合、問合せは、すでにパッチ適用されたアプリケーション・ルートに送信されます。

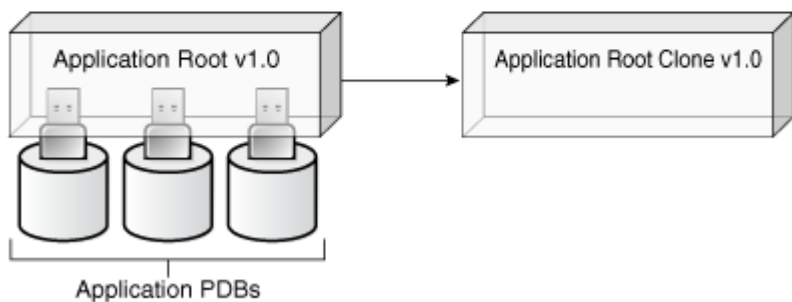
次の図は、アプリケーション・アップグレード・プロセスを示しています。

図17-1 アプリケーション・コンテナ内のアプリケーションのアップグレード

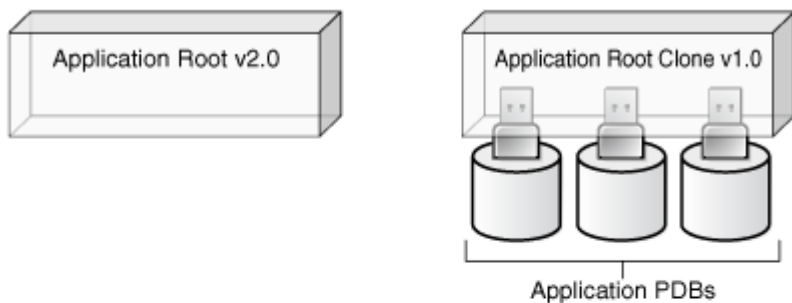
1 Before upgrade



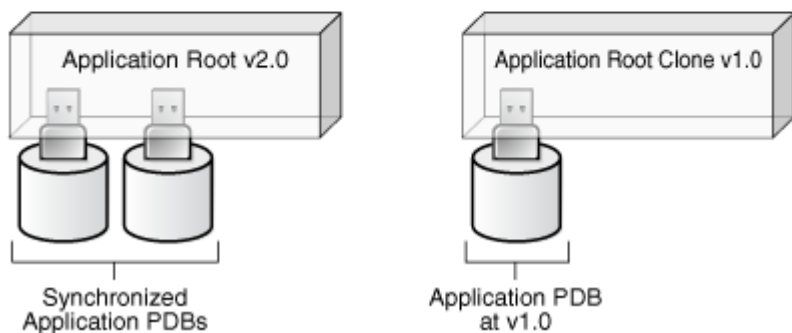
2 Begin upgrade



3 End upgrade



4 After synchronization



ノート:



アプリケーション・ルートがオープン・モードである場合、アプリケーション・ルート・クローンは読み取り専用モードになります。アプリケーション・ルートがクローズされている場合、アプリケーション・ルート・クローンもクローズされます。

親トピック: [アプリケーション・コンテナ内のアプリケーションのアップグレードについて](#)

アプリケーション・アップグレードのためのユーザー・インタフェース

アプリケーション・ルート内のアプリケーション定義をアップグレードするには、ALTER PLUGGABLE DATABASE APPLICATION ... UPGRADEコマンドを使用します。

ALTER PLUGGABLE DATABASE APPLICATION BEGIN UPGRADE文でアップグレードを開始し、ALTER PLUGGABLE DATABASE APPLICATION END UPGRADE文で終了します。各アップグレードには、ALTER PLUGGABLE DATABASE APPLICATION文で指定される、アプリケーション名、開始バージョン番号および終了バージョン番号を関連付ける必要があります。

ノート:



アプリケーション・ルートで透過的データ暗号化が有効になっている場合は、外部パスワード・ストアを構成する必要があります。

親トピック: [アプリケーション・コンテナ内のアプリケーションのアップグレードについて](#)

アプリケーション・コンテナ内のアプリケーションのアップグレード

アップグレード後に、アップグレードによって発生したアプリケーションの変更は、アプリケーション・ルートと同期されるアプリケーションPDBに伝播されます。

前提条件

- CDBはローカルUNDOモードである必要があります。
- 現在のユーザーにはALTER PLUGGABLE DATABASEシステム権限が必要で、権限は一般にアプリケーション・ルートで付与されている必要があります。
- アプリケーション・ルートは読取り/書込みオープンである必要があります。
- アプリケーション・ルートで透過的データ暗号化が有効にされている場合は、外部パスワード・ストアを構成する必要があります。

アプリケーション・コンテナ内のアプリケーションをアップグレードするには:

1. SQL*PlusまたはSQL Developerで、現在のコンテナがアプリケーション・ルートであることを確認します。
2. 次の形式でALTER PLUGGABLE DATABASE APPLICATION BEGIN UPGRADE文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name BEGIN UPGRADE  
'application_start_version_number' TO 'application_end_version_number';
```

たとえば、application_nameがsalesapp、application_start_version_numberが4.2、application_end_version_numberが4.3の場合は、次の文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp BEGIN UPGRADE '4.2' TO '4.3';
```

3. スクリプト、SQL文またはグラフィカル・ユーザー・インタフェース・ツールを使用してアプリケーションをアップグレードします。
4. 次の形式でALTER PLUGGABLE DATABASE APPLICATION END UPGRADE文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name END UPGRADE TO  
'application_end_version_number';
```

たとえば、application_nameがsalesappでapplication_end_version_numberが4.3の場合は、次の文を

実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp END UPGRADE TO '4.3';
```



ノート:

application_name および application_end_version_number が ALTER PLUGGABLE DATABASE APPLICATION BEGIN UPGRADE 文と ALTER PLUGGABLE DATABASE APPLICATION END UPGRADE 文で一致していることを確認します。

5. SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATIONを発行して、アプリケーションをアップグレードする必要があるすべてのアプリケーションPDBを同期します。

関連トピック

- [CDB内のコンテナへのアクセス](#)
- [アプリケーションPDB内のアプリケーションの同期](#)
- [ALTER DATABASEを使用したCDBのUNDOモードの設定](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションのアップグレード](#)

アプリケーション・コンテナ内のアプリケーションへのパッチ適用

アプリケーションに対する小さな変更によってアプリケーション・パッチが構成されます。

小さな変更の例として、バグ修正やセキュリティ・パッチがあります。アプリケーション・コンテナのアプリケーションにパッチ適用できます。

- [アプリケーション・コンテナ内のアプリケーションへのパッチ適用について](#)
アプリケーション・ルート内のアプリケーションにパッチ適用するには、ALTER PLUGGABLE DATABASE APPLICATION文を発行します。
- [自動伝播によるアプリケーション・コンテナ内のアプリケーションへのパッチ適用](#)
パッチ適用に関連するアプリケーションの変更は、アプリケーション・ルート内のアプリケーションと同期されるアプリケーションPDBに伝播されます。

親トピック: [アプリケーション・コンテナ内のアプリケーションの管理](#)

アプリケーション・コンテナ内のアプリケーションへのパッチ適用について

アプリケーション・ルート内のアプリケーションにパッチ適用するには、ALTER PLUGGABLE DATABASE APPLICATION文を発行します。

アプリケーション・ルート内のアプリケーションにのみパッチ適用します。アプリケーションと同期するアプリケーションPDBによって変更が適用されます。スクリプト、SQL文およびグラフィカル・ユーザー・インタフェース・ツールから1つ以上の手法を使用して、パッチ適用を実行できます。

パッチ適用は、小さな操作セットに制限されます。一般に、表の削除などの破壊的な操作はパッチでは許可されていません。アプリケーションにパッチを適用しようとし、操作で「アプリケーション・パッチでサポートされていない操作」エラーが発生した場合、必要な変更を行うにはパッチ適用ではなくアプリケーションをアップグレードします。

ノート:



アプリケーションのアップグレードとは異なり、パッチ適用ではアプリケーション・ルート・クローンが作成されません。パッチ適用後にアプリケーション PDB が同期化されていない場合、問合せは、すでにパッチ適用されたアプリケーション・ルートに送信されます。

ALTER PLUGGABLE DATABASE APPLICATION BEGIN PATCH文でパッチ適用の開始を示し、ALTER PLUGGABLE DATABASE APPLICATION END PATCH文でパッチ適用の終了を示します。各パッチ適用は、アプリケーション名、開始バージョン番号および終了バージョン番号に関連付けられている必要があります。ALTER PLUGGABLE DATABASE APPLICATION文にこれらの値を指定します。

関連トピック

- [CDBでのオラクル社が提供するSQLスクリプトの実行](#)
- [アプリケーション・コンテナ内のアプリケーションのアップグレード](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションのパッチ適用](#)

自動伝播によるアプリケーション・コンテナ内のアプリケーションへのパッチ適用

パッチ適用に関連するアプリケーションの変更は、アプリケーション・ルート内のアプリケーションと同期されるアプリケーションPDBに伝播されます。

前提条件

次の前提条件を満たしている必要があります。

- 現在のユーザーにはALTER PLUGGABLE DATABASEシステム権限が必要で、権限は一般にアプリケーション・ルートで付与されている必要があります。
 - アプリケーション・ルートは読取り/書込みオープン・モードである必要があります。
1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
 2. 次の形式でALTER PLUGGABLE DATABASE APPLICATION BEGIN PATCH文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name
BEGIN PATCH patch_number
MINIMUM VERSION 'minimum_application_version_number';
```

たとえば、application_nameがsalesapp、patch_numberが987654、minimum_application_version_numberが4.2の場合は、次の文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp
BEGIN PATCH 987654 MINIMUM VERSION '4.2';
```

minimum_application_version_numberは、パッチを適用する前にアプリケーションのインストールが必要な最小アプリケーション・バージョンを示します。

3. スクリプト、SQL文またはグラフィカル・ユーザー・インタフェース・ツールを使用してアプリケーションにパッチ適用します。
4. 次の形式でALTER PLUGGABLE DATABASE APPLICATION END PATCH文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name
END PATCH patch_number;
```

たとえば、application_nameがsalesappでpatch_numberが987654の場合は、次の文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp END PATCH 987654;
```

ノート:



application_name および patch_number が ALTER PLUGGABLE DATABASE APPLICATION BEGIN PATCH 文と ALTER PLUGGABLE DATABASE APPLICATION END PATCH 文で一致していることを確認します。

5. SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATIONを発行して、アプリケーションにパッチを適用する必要があるすべてのアプリケーションPDBを同期します。

関連トピック

- [CDB内のテナントへのアクセス](#)
- [アプリケーションPDB内のアプリケーションの同期](#)

親トピック: [アプリケーション・テナント内のアプリケーションのパッチ適用](#)

アプリケーション・テナントへの既存アプリケーションの移行

PDBにインストールされているアプリケーションをアプリケーション・テナントに移行できます。

アプリケーションをアプリケーション・ルートまたはアプリケーションPDBに移行できます。たとえば、Oracle Database 12cリリース2 (12.2) CDBに接続中のPDBにインストールされているアプリケーションを、Oracle Database 18c CDBのアプリケーション・テナントに移行できます。

- [アプリケーション・テナントへの既存のアプリケーションの移行について](#)
既存のPDBを使用してアプリケーション・ルートを作成して、アプリケーションをアプリケーション・ルートに移行できます。
- [既存のPDBを使用したアプリケーション・ルートの作成](#)
PDBをアプリケーション・テナントにコピーすることで、PDBにインストールされているアプリケーションを移行します。
- [既存のPDBを使用したアプリケーションPDBの作成](#)
既存のアプリケーションをアプリケーション・ルートに移行した後で、アプリケーションを使用する既存のPDBを使用して、アプリケーションPDBを作成できます。

親トピック: [アプリケーション・テナント内のアプリケーションの管理](#)

アプリケーション・テナントへの既存アプリケーションの移行について

既存のPDBを使用してアプリケーション・ルートを作成して、アプリケーションをアプリケーション・ルートに移行できます。

アプリケーションが複数のPDBにインストールされる場合、PDBのいずれかを使用して、アプリケーション・ルートを作成できます。PDBのクローニングまたはアプリケーション・ルートとしてのPDBの接続など、PDBをアプリケーション・ルートにコピーするために使用できる方法のいずれかを使用できます。

共通ユーザー、ロールまたはプロファイルがアプリケーション・ルートを作成するために使用されるPDBに存在する場合、DBMS_PDBパッケージのプロシージャを実行して、アプリケーションに関連付ける必要があります。PDBから作成されるアプリケーション・ルートを最初に開くと、各ローカル・ユーザー、ロールおよびプロファイルが共通とマークされます。DBMS_PDBパッケージのプロシージャは、ユーザー、ロールまたはプロファイルをアプリケーションと関連付けます。そのため、ユーザー、ロールまたはプロファイルのすべてのDDL操作は、このアプリケーションのアプリケーションBEGIN... ENDブロック内で後で実行する必要があります。

共有データベース・オブジェクトがアプリケーション・ルートに存在する場合、DBMS_PDBパッケージのプロシージャを実行して、データベース・オブジェクトをアプリケーション共通オブジェクトとしてアプリケーションに関連付ける必要があります。そのため、アプリケーション共通オブジェクトのすべてのDDL操作は、このアプリケーションのアプリケーションBEGIN... ENDブロック内で後で実行する必要があります。

アプリケーション・ルートを配置した後、既存のPDBを使用して、新しいアプリケーション・コンテナにアプリケーションPDBを作成できます。作成されるアプリケーションPDBは、データを含むアプリケーション・オブジェクトを含む必要があります。アプリケーション・バージョンとパッチ番号を同期し、アプリケーションPDBの共有データベース・オブジェクトを確立する追加ステップが必要です。

同じアプリケーションを実行する100個のPDBのシナリオ

現在同じアプリケーションを実行している100個のPDBがあり、これらのPDBをアプリケーション・コンテナに移行するとします。これらのPDBには、アプリケーションが必要とするアプリケーション共通オブジェクトおよび共通ユーザー、ロールおよびプロファイルが含まれます。PDBをアプリケーション・コンテナに移行するには、次のステップに従います。

1. PDBのいずれかを選択し、[「既存のPDBを使用したアプリケーション・ルートの作成」](#)の手順を使用して、このPDBとともにアプリケーション・ルートを作成します。

このステップの一部として、DBMS_PDBパッケージのプロシージャを実行して、データベース・オブジェクト、ユーザー、ロールおよびプロファイルをアプリケーションに関連付けます。
2. [「既存のPDBを使用したアプリケーションPDBの作成」](#)の手順を使用して、アプリケーションを実行しているPDBを使用して100個のアプリケーションPDBを作成します。

関連項目:

- [「アプリケーション・コンテナの作成」](#)
- [「自動伝播によるアプリケーション・コンテナ内のアプリケーションのインストール」](#)
- DBMS_PDBについてさらに学習するには、[『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)を参照してください

親トピック: [アプリケーション・コンテナへの既存アプリケーションの移行](#)

既存のPDBを使用したアプリケーション・ルートの作成

アプリケーション・コンテナにPDBをコピーすることで、PDBにインストールされているアプリケーションを移行します。

前提条件

Oracle Database 12cリリース2 (12.2)以降のCDBが存在する必要があります。

1. CDBで、既存のPDBをクローニングするか、既存のPDBを再配置するか、既存のPDBを切断して接続することにより、アプリケーション・ルートを作成します。
新しいアプリケーション・ルートは、アプリケーションで使用されるすべてのデータベース・オブジェクトを含む必要があります。
2. 現在のコンテナとしてアプリケーション・ルートを使用する場合、ALTER PLUGGABLE DATABASE ... BEGIN INSTALL文を発行して、アプリケーション・インストール操作を開始します。
3. **オプション:** DBA_USERS、DBA_ROLESおよびDBA_PROFILESビューのCOMMON列を問い合せて、共通のユーザー、ロールおよびプロファイルを判断します。
4. DBMS_PDBパッケージの次のプロシージャを実行して、ユーザー、ロールおよびプロファイルをアプリケーションに関連付けます。
 - SET_USER_EXPLICITプロシージャを実行して、アプリケーションの共通ユーザーを設定します。
 - SET_ROLE_EXPLICITプロシージャを実行して、アプリケーションの共通ロールを設定します。
 - SET_PROFILE_EXPLICITプロシージャを実行して、アプリケーションの共通プロファイルを設定します。

DBMS_PDBパッケージに対するEXECUTE権限がない場合は、DBMS_PDB_ALTER_SHARINGパッケージ内のこれらのプロシージャを実行できます。

5. **オプション:** 現在のコンテナとしてアプリケーション・ルートを使用する場合、DBA_OBJECTSビューのSHARING列を問い合わせ、共有するデータベース・オブジェクトを判断します。
 6. DBMS_PDBパッケージの次のプロシージャを実行してデータベース・オブジェクトをアプリケーションに関連付けます。
 - SET_DATA_LINKEDプロシージャを実行して、データリンク・アプリケーションの共通オブジェクトを設定します。
 - SET_METADATA_LINKEDプロシージャを実行して、メタデータリンク・アプリケーションの共通オブジェクトを設定します。
 - SET_EXT_DATA_LINKEDプロシージャを実行して、拡張データリンク・アプリケーションの共通オブジェクトを設定します。
- DBMS_PDBパッケージに対するEXECUTE権限がない場合は、DBMS_PDB_ALTER_SHARINGパッケージ内のこれらのプロシージャを実行できます。
7. ALTER PLUGGABLE DATABASE ... END INSTALL文を発行して、アプリケーションのインストールの操作を終了します。
 8. **オプション:** 以前に実行した問合せを再実行して、データベース・オブジェクトの共有プロパティが正しく、ユーザー、ロールおよびプロファイルの共通プロパティが正しいことを確認します。
 9. **オプション:** 既存のPDBがアプリケーションを使用する場合、これらの既存のPDBを使用して、アプリケーションPDBを作成します。

[「既存のPDBを使用したアプリケーションPDBの作成」](#)を参照してください。

関連トピック

- [アプリケーション・コンテナの作成](#)
- [アプリケーション・コンテナ内のアプリケーションの管理](#)
- [Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)

親トピック: [アプリケーション・コンテナへの既存アプリケーションの移行](#)

既存のPDBを使用したアプリケーションPDBの作成

既存のアプリケーションをアプリケーション・ルートに移行した後で、アプリケーションを使用する既存のPDBを使用して、アプリケーションPDBを作成できます。

前提条件

次の前提条件を満たしている必要があります。

- Oracle Database 12cリリース2 (12.2) CDBが存在する必要があるため、アプリケーションPDBが属するアプリケーション・ルートが存在する必要があります。
 - PDBは、アプリケーションで使用されるすべてのアプリケーション共通オブジェクトを含む必要があります。
 - アプリケーションがアプリケーション・ルートにインストールされている必要があります。
1. アプリケーション・ルートで、既存のPDBをクローニングするか、既存のPDBを切断して接続することにより、アプリケーションPDBを作成します。
PDBの作成中に違反が報告されます。
 2. 必要な権限を持つユーザーとして新規PDBに接続または切替を行います。
 3. ORACLE_HOME/rdbms/adminディレクトリにあるpdb_to_apppdb.sqlスクリプトを実行します。
スクリプトは、アプリケーションPDBとアプリケーション・ルートを自動的に同期します。
 4. **オプション:** DBA_OBJECTSビューのSHARING列を問い合わせ、データベース・オブジェクトの共有プロパティが正しいことを確認します。
 5. **オプション:** DBA_USERS、DBA_ROLESおよびDBA_PROFILESビューのCOMMON列を問い合わせ、ユーザー、ロールおよびプロファイルの共通プロパティが正しいことを確認します。

関連トピック

- [PDBおよびアプリケーション・コンテナの作成および削除](#)

親トピック: [アプリケーション・コンテナへの既存アプリケーションの移行](#)

アプリケーションPDB内のアプリケーションの同期

アプリケーションの同期により、アプリケーションPDB内のアプリケーションがアプリケーション・ルートの最新バージョンおよびパッチに更新されます。

アプリケーション・ルート内のアプリケーションをインストール、アップグレード、パッチ適用またはアンインストールした場合、そのアプリケーションPDBは、同期されるまで変更されません。アプリケーションPDBが現在のコンテナである場合は、次のいずれかの形式のALTER PLUGGABLE DATABASE APPLICATION ... SYNCを使用して手動で同期します。

- 次のように単一のアプリケーションを同期します。ここでのapp1はアプリケーションの名前です。

```
ALTER PLUGGABLE DATABASE APPLICATION app1 SYNC;
```

オプションで、指定したパッチにapp1を同期するにはSYNC TO PATCH patchnoを、指定したバージョンにapp1を同期するにはSYNC TO versionを指定します。

- 次のようにすべてのアプリケーションを同期します。

```
ALTER PLUGGABLE DATABASE APPLICATION ALL SYNC;
```

前提条件と制限

- 現行ユーザーにはALTER PLUGGABLE DATABASEシステム権限が必要です。
- ALLを使用して複数のアプリケーションを指定する場合、SYNC TO句はサポートされません。
- ALLを使用して複数のアプリケーションを指定すると、アプリケーションのBEGINおよびENDブロックが、取得された順序でプレイされます。複数のアプリケーションが相互に依存している場合は、正しく機能させるには単一の文でそれらを同期する必要があります。

1. SQL*Plusで、現在のコンテナがアプリケーションPDBであることを確認します。
2. SYNC句を指定してALTER PLUGGABLE DATABASE APPLICATION文を実行します。

例17-2 アプリケーションPDB内の特定のアプリケーションの同期化

この例では、アプリケーションPDB内のsalesappというアプリケーションに、アプリケーション・ルート内の最新のアプリケーション変更を同期します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp SYNC;
```

例17-3 指定したパッチへのアプリケーションの同期

この例では、アプリケーションPDB内のsalesappというアプリケーションをパッチ100に同期します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp SYNC TO PATCH 100;
```

例17-4 指定したアプリケーション・リリースへのアプリケーションの同期

この例では、アプリケーションPDB内のsalesappというアプリケーションを、そのアプリケーションのリリース2.0に同期します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp SYNC TO '2.0';
```

例17-5 アプリケーションPDB内のすべてのアプリケーションの同期

この例では、アプリケーションPDB内のすべてのアプリケーションをアプリケーション・ルートの最新のアプリケーション変更と同期します。

```
ALTER PLUGGABLE DATABASE APPLICATION ALL SYNC;
```

例17-6 アプリケーションPDB内の暗黙的に作成されたアプリケーションの同期

この例では、アプリケーションPDB内の暗黙的に作成されたすべてのアプリケーションを、アプリケーション・ルート内の暗黙的に作成されたアプリケーションに対する最新のアプリケーション変更と同期します。

```
ALTER PLUGGABLE DATABASE APPLICATION APP$CON SYNC;
```

関連項目:

[「アプリケーションの同期化」](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションの管理](#)

アプリケーション・ルート・レプリカのプロキシPDBとの同期

異なるCDB内のアプリケーション・コンテナが同じアプリケーションを持つ場合、それらのアプリケーション・ルートは、マスター・アプリケーション・ルート、レプリカ・アプリケーション・ルートおよびプロキシPDBを作成することで同期を維持できます。

- [アプリケーション・ルート・レプリカのプロキシPDBとの同期について](#)
プロキシPDBは、アプリケーション・ルートとアプリケーション・ルートのレプリカを同期できます。
- [アプリケーション・ルート・レプリカを参照するプロキシPDBの作成](#)
複数のアプリケーション・コンテナが同じアプリケーションを実行する場合、アプリケーション・コンテナ内のアプリケーションはプロキシPDBを使用して同期を維持できます。

親トピック: [アプリケーション・コンテナ内のアプリケーションの管理](#)

アプリケーション・ルート・レプリカのプロキシPDBとの同期について

プロキシPDBは、アプリケーション・ルートとアプリケーション・ルートのレプリカを同期できます。

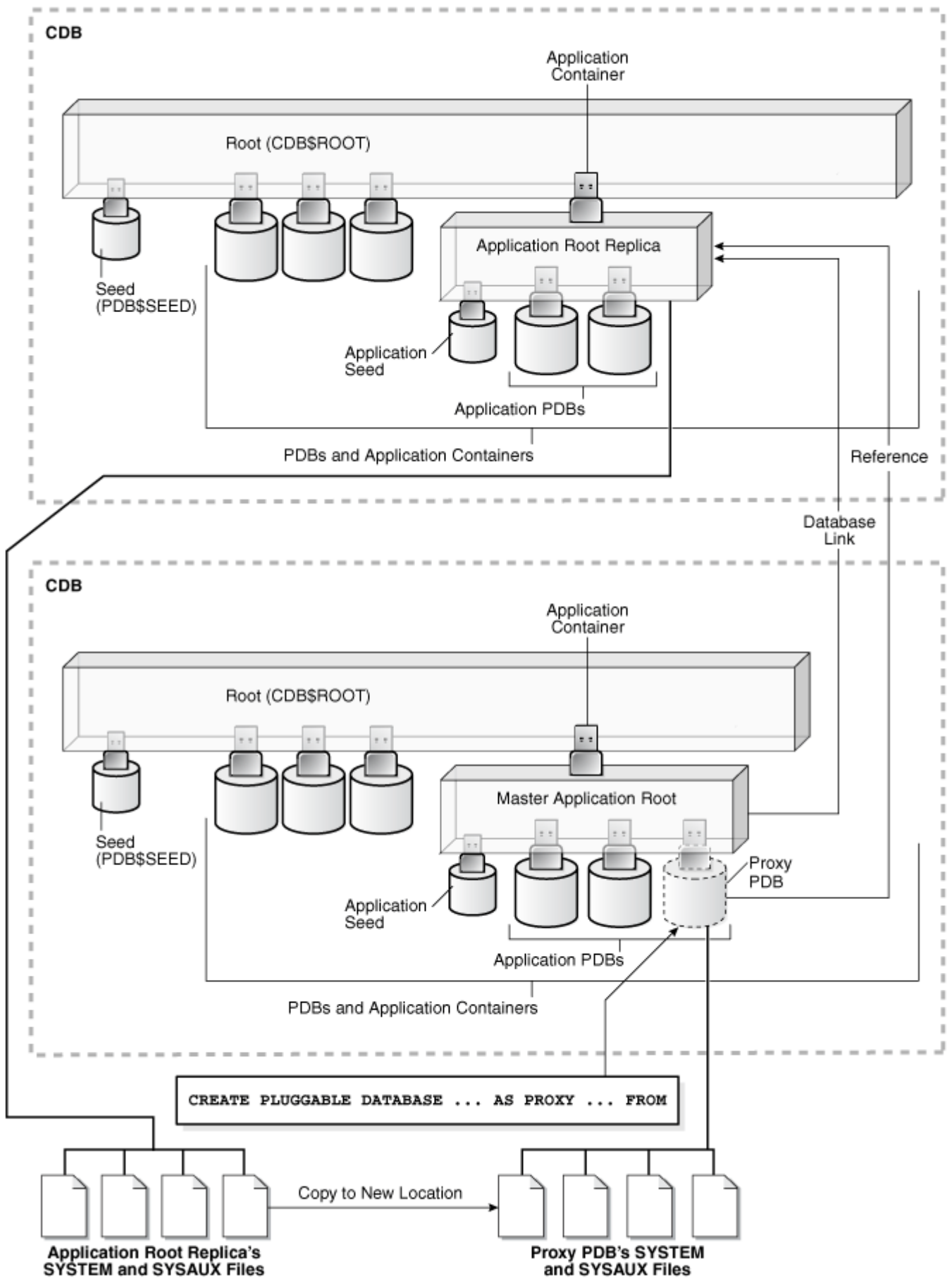
1つのアプリケーションを複数のアプリケーション・コンテナにインストールできます。プロキシPDBを使用すると、アプリケーションをより効率的にインストール、アップグレードおよびパッチ適用できます。

この構成では、1つのアプリケーション・コンテナにマスター・アプリケーション・ルートがあります。マスター・アプリケーション・ルートは、アプリケーションをインストール、アップグレードおよびパッチ適用する場所です。アプリケーション・ルート・レプリカは、マスター・アプリケーション・ルートの完全なコピーです。各アプリケーション・ルート・レプリカは、マスター・アプリケーション・ルート内のプロキシPDBによって参照されます。

プロキシPDBは、マスター・アプリケーション・ルート内のアプリケーション変更と同期されると、その変更を参照先のアプリケーション・ルート・レプリカに伝播します。アプリケーション・ルート・レプリカが同期された後、アプリケーション・ルート・レプリカに接続されているアプリケーションPDBは、レプリカと同期する方法で変更を取得できます。

次の図に、プロキシPDBを使用してアプリケーション・ルート・レプリカを同期する構成を示します。

図17-2 アプリケーション・ルート・レプリカとプロキシPDBの同期



また、アプリケーション・ルート・レプリカが構成され、固有のアプリケーションPDBを持つ場合、マスター・アプリケーション・ルート内のCONTAINERS句を含む問合せは、現在のアプリケーション・コンテナおよびアプリケーション・ルート・レプリカのあるアプリケーション・コンテナからのデータを返すことができます。問合せは、アプリケーション・ルート・レプリカおよびレプリカに接続しているオープンされたアプリケーションPDBからの結果を表示できます。

関連項目:

[「アプリケーションPDB間のアプリケーション共通オブジェクトの問合せ」](#)

親トピック: [アプリケーション・ルート・レプリカのプロキシPDBとの同期](#)

アプリケーション・ルート・レプリカを参照するプロキシPDBの作成

複数のアプリケーション・コンテナが同じアプリケーションを実行する場合、アプリケーション・コンテナ内のアプリケーションはプロキシPDBを使用して同期を維持できます。

1. CREATE PLUGGABLE DATABASE文を使用して、マスター・アプリケーション・ルートを持つアプリケーション・コンテナを作成します。
今すぐまたは後で、アプリケーション・コンテナにアプリケーションをインストールします。
2. 次のいずれかの方法で、アプリケーション・ルート・レプリカを持つアプリケーション・コンテナを作成します。
 - サポートされる方法を使用して空のアプリケーション・コンテナを作成します。
 - マスター・アプリケーション・ルートをクローニングします。アプリケーション・ルート・レプリカで使用されるリスナーのポートが1521でない場合は、作成時にPORT句が必要です。アプリケーション・ルート・レプリカのホストがマスター・アプリケーション・ルートのホストと異なる場合は、作成時にHOST句が必要です。
このアプリケーション・ルート・レプリカは、プロキシPDBによって参照されます。
3. マスター・アプリケーション・ルートで、前のステップで作成したアプリケーション・ルート・レプリカを参照するプロキシPDBを作成します。
4. プロキシPDBをオープンして同期します。
プロキシPDBは同期されると、マスター・アプリケーション・ルート内の変更をアプリケーション・ルート・レプリカに伝播します。
5. **オプション:** マスター・アプリケーション・ルートで、インストール、アップグレードまたはパッチ適用することでアプリケーションを変更します。
6. **オプション:** SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATION文を実行して、プロキシPDBをマスター・アプリケーション・ルート内のアプリケーション変更と同期します。
プロキシPDBは同期されると、マスター・アプリケーション・ルート内の変更をアプリケーション・ルート・レプリカに伝播します。

例17-7 アプリケーション・ルート・レプリカとプロキシPDBの同期

この例では、hqdbおよびdepdbの2つのCDBが存在していることを想定しています。目標は、各CDB内のアプリケーション・コンテナで同じアプリケーションの同期を維持することです。この目標を実現するために、この例では次のアプリケーション・コンテナを構成します。

- hqdb CDBには、msappconというマスター・アプリケーション・ルートがあるアプリケーション・コンテナが含まれます。
 - sampleappというアプリケーションがmsappconマスター・アプリケーション・ルートにインストールされます。
 - msappconアプリケーション・ルートには、mspdb1およびmspdb2という2つのアプリケーションPDBが含まれます。
 - msappconアプリケーション・ルートには、他のCDB内のアプリケーション・ルート・レプリカを参照するprxypdbという名前のプロキシPDBも含まれます。
- depdb CDBには、depappconというアプリケーション・ルート・レプリカがあるアプリケーション・コンテナが含まれます。
 - sampleappというアプリケーションは、msappconマスター・アプリケーション・ルート内のプロキシPDB prxypdbから伝播され、depappconマスター・アプリケーション・ルートにインストールされます。

- depappconアプリケーション・ルートには、deppdb1およびdeppdb2という2つのアプリケーションPDBが含まれません。

この例では、msappconマスター・アプリケーション・ルート内のsampleappアプリケーションに対する変更が、アプリケーションPDBが同期されたときに両方のCDB内のアプリケーションPDBにどのように適用されるかを示します。

1. hqdb CDBにマスター・アプリケーション・ルートを持つアプリケーション・コンテナを作成します。

- SQL*Plusで、現在のコンテナがhqdb CDBルートであることを確認します。
- 次の文でPDBシードからアプリケーション・コンテナを作成します。

```
CREATE PLUGGABLE DATABASE msappcon
AS APPLICATION CONTAINER
ADMIN USER msappconadm IDENTIFIED BY password
STORAGE (MAXSIZE 2G)
DEFAULT TABLESPACE appcontbs
DATAFILE '/disk1/oracle/dbs/mssappcon/msappcon01.dbf' SIZE 250M
AUTOEXTEND ON
FILE_NAME_CONVERT = ('/disk1/oracle/dbs/pdbseed/',
'/disk1/oracle/dbs/msappcon/');
```

- 新規マスター・アプリケーション・ルートを読み取り/書込みモードでオープンします。

```
ALTER PLUGGABLE DATABASE msappcon OPEN;
```

2. マスター・アプリケーション・ルートにアプリケーションをインストールします。

- コンテナをマスター・アプリケーション・ルートに変更します。

```
ALTER SESSION SET CONTAINER=msappcon;
```

- アプリケーションのインストールを開始します。

```
ALTER PLUGGABLE DATABASE APPLICATION sampleapp BEGIN INSTALL '1.0';
```

- アプリケーションをインストールします。

たとえば、データベース・オブジェクトを作成できます。

```
CREATE TABLE apptb SHARING=METADATA
(id NUMBER(6),
widget_name VARCHAR2(20));
```

- アプリケーションのインストールを終了します。

```
ALTER PLUGGABLE DATABASE APPLICATION sampleapp END INSTALL '1.0';
```

3. 1つ以上のアプリケーションPDBをマスター・アプリケーション・ルートに作成し、同期します。

- SQL*Plusで、現在のコンテナがマスター・レプリケーション・ルートであることを確認します。
- マスター・アプリケーション・ルートにアプリケーションPDBを作成します。

たとえば、PDBシードから2つのアプリケーションPDBを作成します。

```
CREATE PLUGGABLE DATABASE mspdb1 ADMIN USER mspdb1admin IDENTIFIED BY password
STORAGE (MAXSIZE 2G)
DEFAULT TABLESPACE mspdb1tbs
DATAFILE '/disk1/oracle/dbs/mspdb1/mspdb101.dbf' SIZE 250M
AUTOEXTEND ON
FILE_NAME_CONVERT = ('/disk1/oracle/dbs/pdbseed/',
```

```

        '/disk1/oracle/dbs/mspdb1/');
CREATE PLUGGABLE DATABASE mspdb2 ADMIN USER mspdb2admin IDENTIFIED BY password
STORAGE (MAXSIZE 2G)
DEFAULT TABLESPACE mspdb2tbs
DATAFILE '/disk1/oracle/dbs/mspdb2/mspdb201.dbf' SIZE 250M
AUTOEXTEND ON
FILE_NAME_CONVERT = ('/disk1/oracle/dbs/pdbseed/',
                    '/disk1/oracle/dbs/mspdb2/');

```

- c. 両方のアプリケーションPDBをオープンします。

```

ALTER PLUGGABLE DATABASE mspdb1 OPEN;
ALTER PLUGGABLE DATABASE mspdb2 OPEN;

```

- d. アプリケーションPDBをマスター・アプリケーション・ルートと同期します。

```

ALTER SESSION SET CONTAINER=mspdb1;
ALTER PLUGGABLE DATABASE APPLICATION sampleapp SYNC;

ALTER SESSION SET CONTAINER=mspdb2;
ALTER PLUGGABLE DATABASE APPLICATION sampleapp SYNC;

```

4. depdb CDBにアプリケーション・ルート・レプリカを持つアプリケーション・コンテナを作成します。

- a. SQL*Plusで、現在のコンテナがdepdb CDBルートであることを確認します。
b. 次の文でPDBシードからアプリケーション・コンテナを作成します。

```

CREATE PLUGGABLE DATABASE depappcon
AS APPLICATION CONTAINER
ADMIN USER depappconadm IDENTIFIED BY password
STORAGE (MAXSIZE 2G)
DEFAULT TABLESPACE appcontbs
DATAFILE '/disk2/oracle/dbs/depappcon/depappcon01.dbf' SIZE 250M
AUTOEXTEND ON
FILE_NAME_CONVERT = ('/disk2/oracle/dbs/pdbseed/',
                    '/disk2/oracle/dbs/depappcon/');

```

ノート:



- アプリケーション・ルート・レプリカで使用されるリスナーのポートが 1521 でない場合は、PORT 句が必要です。
- アプリケーション・ルート・レプリカのホストがマスター・アプリケーション・ルートのホストと異なる場合は、HOST 句が必要です。

- c. 新規アプリケーション・ルート・レプリカを読み取り/書き込みモードでオープンします。

```

ALTER PLUGGABLE DATABASE depappcon OPEN;

```

5. プロキシPDBをマスター・アプリケーション・ルートに作成し、同期します。

- a. SQL*Plusで、現在のコンテナがマスター・レプリケーション・ルートであることを確認します。
b. アプリケーション・ルート・レプリカへのデータベース・リンクを作成します。

```
CREATE PUBLIC DATABASE LINK depappcon
CONNECT TO depappconadm IDENTIFIED BY password USING 'depappcon';
```

- c. プロキシPDBを作成します。

```
CREATE PLUGGABLE DATABASE prxypdb AS PROXY
FROM depappcon@depappcon
FILE_NAME_CONVERT = ('/disk2/oracle/dbs/depsappcon/',
                    '/disk1/oracle/dbs/prxypdb/');
```

- d. プロキシPDBをオープンします。

```
ALTER PLUGGABLE DATABASE prxypdb OPEN;
```

- e. プロキシPDBをマスター・アプリケーション・ルートと同期します。

```
ALTER SESSION SET CONTAINER=prxypdb;
ALTER PLUGGABLE DATABASE APPLICATION sampleapp SYNC;
```

6. 1つ以上のアプリケーションPDBをアプリケーション・ルート・レプリカに作成し、同期します。

- a. コンテナをアプリケーション・ルート・レプリカに変更します。

```
ALTER SESSION SET CONTAINER=depappcon;
```

- b. アプリケーション・ルート・レプリカにアプリケーションPDBを作成します。

たとえば、PDBシードから2つのアプリケーションPDBを作成します。

```
CREATE PLUGGABLE DATABASE deppdb1
ADMIN USER deppdb1admin IDENTIFIED BY password
STORAGE (MAXSIZE 2G)
DEFAULT TABLESPACE deppdb1tbs
DATAFILE '/disk2/oracle/dbs/deppdb1/deppdb101.dbf' SIZE 250M
AUTOEXTEND ON
FILE_NAME_CONVERT = ('/disk2/oracle/dbs/pdbseed/',
                    '/disk2/oracle/dbs/deppdb1/');
```

```
CREATE PLUGGABLE DATABASE deppdb2 ADMIN USER deppdb2admin IDENTIFIED BY password
STORAGE (MAXSIZE 2G)
DEFAULT TABLESPACE deppdb2tbs
DATAFILE '/disk2/oracle/dbs/deppdb2/deppdb201.dbf' SIZE 250M
AUTOEXTEND ON
FILE_NAME_CONVERT = ('/disk2/oracle/dbs/pdbseed/',
                    '/disk2/oracle/dbs/deppdb2/');
```

- c. 両方のアプリケーションPDBをオープンします。

```
ALTER PLUGGABLE DATABASE deppdb1 OPEN;
ALTER PLUGGABLE DATABASE deppdb2 OPEN;
```

- d. アプリケーションPDBをマスター・アプリケーション・ルートと同期します。

```
ALTER SESSION SET CONTAINER=deppdb1;
ALTER PLUGGABLE DATABASE APPLICATION sampleapp SYNC;
```

```
ALTER SESSION SET CONTAINER=deppdb2;
ALTER PLUGGABLE DATABASE APPLICATION sampleapp SYNC;
```

7. アプリケーション・ルート・レプリカのアプリケーションPDB内のapp1tb表の構造を確認します。

- a. アプリケーション・ルート・レプリカから、コンテナをdeppdb1アプリケーションPDBに切り替えます。

```
ALTER SESSION SET CONTAINER=deppdb1;
```

- b. apptb表を記述します。

```
desc apptb
```

出力は次のようになります。

Name	Null?	Type
ID		NUMBER (6)
WIDGET_NAME		VARCHAR2 (20)

8. マスター・アプリケーション・ルートで、アプリケーションをアップグレードします。

- a. コンテナをマスター・アプリケーション・ルートに変更します。

```
ALTER SESSION SET CONTAINER=msappcon;
```

- b. アプリケーションのアップグレードを開始します。

```
ALTER PLUGGABLE DATABASE APPLICATION sampleapp  
BEGIN UPGRADE '1.0' TO '1.1';
```

- c. アプリケーションを変更します。

たとえば、apptb表に行を追加します。

```
ALTER TABLE apptb ADD (widget_type VARCHAR2 (30));
```

- d. アプリケーションのアップグレードを終了します。

```
ALTER PLUGGABLE DATABASE APPLICATION sampleapp END UPGRADE TO '1.1';
```

9. プロキシPDBをマスター・アプリケーション・ルートと同期します。

```
ALTER SESSION SET CONTAINER=prxypdb;  
ALTER PLUGGABLE DATABASE APPLICATION sampleapp SYNC;
```

10. アプリケーション・ルート・レプリカ内のアプリケーションPDBを同期し、アプリケーションのアップグレードを確認します。

- a. アプリケーションPDBを同期します。

```
ALTER SESSION SET CONTAINER=deppdb1;  
ALTER PLUGGABLE DATABASE APPLICATION sampleapp SYNC;  
  
ALTER SESSION SET CONTAINER=deppdb2;  
ALTER PLUGGABLE DATABASE APPLICATION sampleapp SYNC;
```

- b. アプリケーション・ルート・レプリカから、コンテナをdeppdb1アプリケーションPDBに切り替えます。

```
ALTER SESSION SET CONTAINER=deppdb1;
```

- c. apptb表を記述します。

```
desc apptb
```

出力は次のようになります。

Name	Null?	Type
ID		NUMBER (6)
WIDGET_NAME		VARCHAR2 (20)
WIDGET_TYPE		VARCHAR2 (30)

widget_type列がapptb表に追加されたため、アプリケーション・アップグレード内の変更が出力に反映されません。

関連トピック

- [アプリケーション・コンテナの作成](#)
- [プロキシPDBとしてのPDBの作成](#)
- [アプリケーション・コンテナ内のアプリケーションの管理](#)
- [アプリケーションPDB内のアプリケーションの同期](#)

親トピック: [アプリケーション・ルート・レプリカのプロキシPDBとの同期](#)

アプリケーションの互換バージョンの設定

アプリケーションの互換性バージョンは、アプリケーション・コンテナに属するアプリケーションPDBで使用可能なアプリケーション・バージョンのうち、最も古いバージョンです。

互換性バージョンは、互換性バージョンが設定され、アプリケーションPDBが作成されたときに適用されます。アプリケーションのアップグレードによって作成されたアプリケーション・ルート・クローンがある場合、互換性バージョンより古いバージョンに相当するすべてのアプリケーション・ルート・クローンが暗黙的に削除されます。

アプリケーションの互換バージョンを指定するには、アプリケーション・ルートが現在のコンテナであるときに、次のいずれかのSQL文を発行します。

- ALTER PLUGGABLE DATABASE APPLICATION application_name SET COMPATIBILITY VERSION 'application_version_number';

application_nameはアプリケーションの名前であり、application_version_numberは最も古い互換バージョンです。

- ALTER PLUGGABLE DATABASE APPLICATION application_name SET COMPATIBILITY VERSION CURRENT;

application_nameはアプリケーションの名前です。現在のバージョンは、アプリケーション・ルート内のアプリケーションのバージョンです。

ノート:



アプリケーション・コンテナの互換性設定より前のアプリケーション・バージョンを使用するアプリケーション PDB は接続できません。

1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
2. ALTER PLUGGABLE DATABASE APPLICATION SET COMPATIBILITY VERSION文を実行します。

例17-8 互換バージョンに特定のバージョン番号を設定する

この例は、salesappというアプリケーションの互換バージョンにバージョン4.2を設定しています。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp
```

```
SET COMPATIBILITY VERSION '4.2';
```

例17-9 互換バージョンに現在のアプリケーション・バージョンを設定する

この例は、salesappというアプリケーションの互換バージョンに現在のアプリケーション・バージョンを設定しています。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp  
SET COMPATIBILITY VERSION CURRENT;
```

関連項目:

アプリケーション・ルート・クローンの詳細は、[アプリケーション・コンテナ内のアプリケーションのアップグレードについて](#)を参照してください

親トピック: [アプリケーション・コンテナ内のアプリケーションの管理](#)

アプリケーションのインストール、アップグレードおよびパッチ適用の操作中の一括挿入の実行

SQL*Loaderは、アプリケーションのインストール、アップグレードおよびパッチ適用の操作中に表に一括挿入することがサポートされる唯一のユーティリティです。従来型パス・ロードのみ、アプリケーションのインストール、アップグレードおよびパッチ適用の操作中に一括挿入することがサポートされます。

ALTER PLUGGABLE DATABASE APPLICATION BEGINと ALTER PLUGGABLE DATABASE APPLICATION END文の間に正しいSQL*Loaderモジュール名を指定する必要があります。モジュール名はSQL Loader Conventional Path Loadです。

1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
2. 次のプロシージャを実行して、正しいモジュールを設定します。

```
BEGIN  
  DBMS_APPLICATION_INFO.SET_MODULE(  
    'SQL Loader Conventional Path Load', '');  
END;
```

このモジュールは、全体のアプリケーションのインストール、アップグレードまたはパッチ適用の操作に設定したままにする必要があります。

3. アプリケーションのインストール、アップグレードまたはパッチを開始するにはALTER PLUGGABLE DATABASE APPLICATION BEGIN文を実行します。
たとえば、アプリケーション・インストールの一部として一括挿入を実行している場合は、次の形式でALTER PLUGGABLE DATABASE APPLICATION BEGIN INSTALL文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name  
BEGIN INSTALL 'application_version_number';
```

4. SQL*Loaderを使用して、従来型パス・ロードを実行します。
5. アプリケーションのインストール、アップグレードまたはパッチを終了するにはALTER PLUGGABLE DATABASE APPLICATION END文を実行します。
たとえば、アプリケーション・インストールの一部として一括挿入を実行している場合は、次の形式でALTER PLUGGABLE DATABASE APPLICATION END INSTALL文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name  
END INSTALL 'application_version_number';
```


ノート:



application_name および application_version_number が ALTER PLUGGABLE DATABASE APPLICATION BEGIN INSTALL 文と ALTER PLUGGABLE DATABASE APPLICATION END INSTALL 文で一致していることを確認します。

6. SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATIONを発行して、これらのアプリケーションの変更を含める必要のあるすべてのアプリケーションPDBを同期します。

例17-10 アプリケーションのインストール中の従来型パス・ロードの実行

この例では、従来型パス・ロードがアプリケーション・ルートで実行されます。

1. SQL*Plusで、アプリケーション・ルートに切り替えます。

```
ALTER SESSION SET CONTAINER=cdb1_aproot1;
```

2. 正しいモジュールを設定します。

```
BEGIN
  DBMS_APPLICATION_INFO.SET_MODULE(
    'SQL Loader Conventional Path Load', '');
END;
```

3. アプリケーションのインストールを開始します。

```
ALTER PLUGGABLE DATABASE APPLICATION APP1 BEGIN INSTALL '1';
```

4. SQL*Loaderを使用して、従来型パス・ロードを実行します。

```
HOST sqlldr u1/u1@cdb1_aproot1 control=my_bulk_load.ctl -
rows=3 log=my_bulk_load.log
```

5. アプリケーションのインストールを終了します。

```
ALTER PLUGGABLE DATABASE APPLICATION APP1 END INSTALL '1';
```

関連項目:

SQL*Loaderの詳細は、[『Oracle Databaseユーティリティ』](#)を参照してください

親トピック: [アプリケーション・コンテナ内のアプリケーションの管理](#)

アプリケーション・コンテナからのアプリケーションのアンインストール

アプリケーション・コンテナのアプリケーションをアンインストールできます。

- [アプリケーション・コンテナからのアプリケーションのアンインストールについて](#)
アプリケーション・ルートからアプリケーションをアンインストールするには、ALTER PLUGGABLE DATABASE APPLICATION 文を発行します。
- [アプリケーション・コンテナからのアプリケーションのアンインストール](#)
アプリケーション・コンテナからアプリケーションをアンインストールするには、ALTER PLUGGABLE DATABASE APPLICATION BEGIN UNINSTALL文を実行してアンインストールを開始し、ALTER PLUGGABLE DATABASE

APPLICATION END UNINSTALL文を実行して終了します。アプリケーションは、アプリケーション・ルートのアプリケーションと同期されるアプリケーションPDBからアンインストールされます。

親トピック: [アプリケーション・コンテナ内のアプリケーションの管理](#)

アプリケーション・コンテナからのアプリケーションのアンインストールについて

アプリケーション・ルートからアプリケーションをアンインストールするには、ALTER PLUGGABLE DATABASE APPLICATION文を発行します。

アプリケーション・ルートからのみアプリケーションをアンインストールし、アプリケーションと同期するアプリケーションPDBではアプリケーションが自動的にアンインストールされます。アンインストール操作は、スクリプト、SQL文およびグラフィカル・ユーザー・インタフェース・ツールの1つ以上を使用して実行できます。

ALTER PLUGGABLE DATABASE APPLICATION BEGIN UNINSTALL文でアンインストールの開始を示し、ALTER PLUGGABLE DATABASE APPLICATION END UNINSTALL文でアンインストールの終了を示す必要があります。各アンインストールには、ALTER PLUGGABLE DATABASE APPLICATION文で指定されるアプリケーション名とバージョン番号を関連付ける必要があります。

アプリケーションをアンインストールしても、データ・ディクショナリからアプリケーションは削除されません。アプリケーションのアップグレード、パッチ適用およびアンインストールが許可されないように、アプリケーションがUNINSTALLEDとしてマークされます。

アプリケーションのアンインストール中は、アプリケーションへの破壊的な変更が許可されます。アプリケーションPDBで実行されているアプリケーションは、アンインストール中、およびアプリケーションがアプリケーション・ルートからアンインストールされた後も引き続き機能します。アプリケーションがアプリケーションPDBで引き続き機能できるのは、ALTER PLUGGABLE DATABASE APPLICATION BEGIN UNINSTALL文によって、アプリケーション・ルート・クローンと呼ばれるアプリケーション・ルートのクローンが作成されるためです。アプリケーション・ルート・クローンは古いバージョンのアプリケーション・オブジェクトのメタデータ・リポジトリとして機能し、最新バージョンのアプリケーションと同期されていないアプリケーションPDBが引き続き機能できるようにします。クローンはアプリケーションPDBがオープンされているときに作成されるため、アプリケーションをアンインストールする前に、ローカルUNDOをCDBレベルで構成する必要があります。

ノート:



アプリケーションのアップグレードでも、アプリケーション・ルート・クローンが作成されます。

関連項目:

- アプリケーション・ルート・クローンの詳細は、[アプリケーション・コンテナ内のアプリケーションのアップグレードについて](#)を参照してください
- [CDBでのオラクル社が提供するSQLスクリプトの実行](#)

親トピック: [アプリケーション・コンテナからのアプリケーションのアンインストール](#)

アプリケーション・コンテナからのアプリケーションのアンインストール

アプリケーション・コンテナからアプリケーションをアンインストールするには、ALTER PLUGGABLE DATABASE APPLICATION BEGIN UNINSTALL文を実行してアンインストールを開始し、ALTER PLUGGABLE DATABASE APPLICATION END UNINSTALL文を実行して終了します。アプリケーションは、アプリケーション・ルートのアプリケーションと同期されるアプリケーションPDBからアンインス

トールされます。

次の前提条件を満たしている必要があります。

- CDBはローカルUNDOモードである必要があります。
 - 現在のユーザーにはALTER PLUGGABLE DATABASEシステム権限が必要で、権限は一般にアプリケーション・ルートで付与されている必要があります。
 - アプリケーション・ルートは読取り/書込みオープン・モードである必要があります。
1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
 2. 次の形式でALTER PLUGGABLE DATABASE APPLICATION BEGIN UNINSTALL文を実行します。
ALTER PLUGGABLE DATABASE APPLICATION application_name BEGIN UNINSTALL;
たとえば、application_nameがsalesappである場合は、次の文を実行します。
ALTER PLUGGABLE DATABASE APPLICATION salesapp BEGIN UNINSTALL;
 3. スクリプト、SQL文またはグラフィカル・ユーザー・インタフェース・ツールを使用してアプリケーションをアンインストールします。
 4. 次の形式でALTER PLUGGABLE DATABASE APPLICATION END UNINSTALL文を実行します。
ALTER PLUGGABLE DATABASE APPLICATION application_name END UNINSTALL;
たとえば、application_nameがsalesappである場合は、次の文を実行します。
ALTER PLUGGABLE DATABASE APPLICATION salesapp END UNINSTALL;

ノート:



application_name が ALTER PLUGGABLE DATABASE APPLICATION BEGIN UNINSTALL 文と ALTER PLUGGABLE DATABASE APPLICATION END UNINSTALL 文で一致していることを確認します。

5. SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATIONを発行して、アプリケーションをアンインストールする必要があるすべてのアプリケーションPDBを同期します。

関連項目:

- [「CDB内のコンテナへのアクセス」](#)
- [「アプリケーションPDB内のアプリケーションの同期」](#)
- [「ALTER DATABASEを使用したCDBのUNDOモードの設定」](#)

親トピック: [アプリケーション・コンテナからのアプリケーションのアンインストール](#)

アプリケーション共通オブジェクトの管理

アプリケーション共通オブジェクトは、アプリケーション・コンテナで共有されている、ユーザーにより作成されたデータベース・オブジェクトです。アプリケーション共通オブジェクトは、アプリケーション・ルートに作成されます。

- [アプリケーション共通オブジェクトについて](#)
アプリケーション共通オブジェクトは、アプリケーション・ルートに作成され、アプリケーション・ルートに属するアプリケーションPDBと共有されます。
- [アプリケーション共通オブジェクトの制限](#)
いくつかの制限がアプリケーション共通オブジェクトに適用されます。
- [アプリケーション共通オブジェクトの作成](#)
DEFAULT_SHARING初期化パラメータが適切な値に設定されていることを確認するか、SHARING句をCREATE SQL文に

含めることで、アプリケーション共通オブジェクトをアプリケーション・ルートに作成します。

- [アプリケーション共通オブジェクトに対するDML文の発行](#)
メタデータリンク、データリンクおよび拡張データリンクのアプリケーション共通オブジェクトに対するDML文の発行のルールは異なります。
- [DDL文でのアプリケーション共通オブジェクトの変更](#)
特定のDDL文でアプリケーション・ルート内のアプリケーション共通オブジェクトを変更する場合は、ALTER PLUGGABLE DATABASE APPLICATION BEGIN文とALTER PLUGGABLE DATABASE APPLICATION END文の間でオブジェクトを変更する必要があり、変更を適用するためにアプリケーションPDBがアプリケーションと同期する必要があります。

親トピック: [アプリケーション・コンテナの管理](#)

アプリケーション共通オブジェクトについて

アプリケーション共通オブジェクトは、アプリケーション・ルートに作成され、アプリケーション・ルートに属するアプリケーションPDBと共有されています。

アプリケーション共通オブジェクトには、メタデータリンク、データリンクおよび拡張データリンクの3種類があります。次のタイプのデータベース・オブジェクトをアプリケーション共通オブジェクトにすることができます。

- 分析ビュー
- 属性ディメンション
- ディレクトリ
- 外部プロシージャ・ライブラリ
- 階層
- Javaクラス、JavaリソースおよびJavaソース
- オブジェクト表、オブジェクト・タイプおよびオブジェクト・ビュー
- 順序
- パッケージ、ストアド・ファンクションおよびストアド・プロシージャ
- シノニム
- 表(グローバル一時表を含む)
- トリガー
- ビュー
- [アプリケーション共通オブジェクトの作成](#)
現在のコンテナがアプリケーション・ルートのときにCREATE文を発行し、SHARING句を指定することで、アプリケーション共通オブジェクトを作成します。
- [メタデータリンク・アプリケーション共通オブジェクトについて](#)
メタデータリンク・アプリケーション共通オブジェクトの場合、オブジェクトのメタデータはアプリケーション・ルートに1回格納されます。
- [拡張データリンク・アプリケーション共通オブジェクトについて](#)
データリンク・アプリケーション共通オブジェクトの場合は、オブジェクトのメタデータとデータの両方がアプリケーション・ルートに1回格納されます。アプリケーション・ルートに属する各アプリケーションPDB内のデータ・リンクにより、アプリケーションPDBはオブジェクトのメタデータとデータを共有できます。
- [拡張データリンク・アプリケーション共通オブジェクトについて](#)

拡張データリンク・オブジェクトの場合、各アプリケーションPDBは、アプリケーション・ルートの共通データを共有しながら独自データを作成できます。アプリケーション・ルートに格納されるデータのみがすべてのアプリケーションPDBに共通です。

関連項目:

[「アプリケーション共通オブジェクト」](#)

親トピック: [アプリケーション共通オブジェクトの管理](#)

アプリケーション共通オブジェクトの作成

現在のコンテナがアプリケーション・ルートのときにCREATE文を発行し、SHARING句を指定することで、アプリケーション共通オブジェクトを作成します。

共有属性は、CREATE文にSHARING句を含めるか、アプリケーション・ルートでDEFAULT_SHARING初期化パラメータを設定することで指定できます。DEFAULT_SHARING初期化パラメータを設定する場合、設定は、アプリケーション・ルートに作成されたサポートされるタイプのすべてのデータベース・オブジェクトのデフォルト共有属性です。ただし、SHARING句がCREATE文に含まれる場合、その設定はDEFAULT_SHARING初期化パラメータの設定をオーバーライドします。

共有属性に対して次のいずれかを指定できます。

- **METADATA:** メタデータ・リンクはデータベース・オブジェクトのメタデータを共有しますが、データは各コンテナに固有です。これらのデータベース・オブジェクトは、メタデータリンク・アプリケーション共通オブジェクトと呼ばれます。この設定がデフォルトです。
- **DATA:** データ・リンクはデータベース・オブジェクトを共有し、そのデータはアプリケーション・コンテナ内のすべてのコンテナに対して同じです。そのデータはアプリケーション・ルートにのみ格納されます。これらのデータベース・オブジェクトは、データリンク・アプリケーション共通オブジェクトと呼ばれます。
- **EXTENDED DATA:** 拡張データ・リンクはデータベース・オブジェクトを共有し、アプリケーション・ルートのそのデータはアプリケーション・コンテナ内のすべてのコンテナに対して同じです。ただし、アプリケーション・コンテナの各アプリケーションPDBには、アプリケーションPDBに一意のデータを格納できます。このタイプのデータベース・オブジェクトの場合、データはアプリケーション・ルートに格納され、必要に応じて各アプリケーションPDBに格納されます。これらのデータベース・オブジェクトは、拡張データリンク・アプリケーション共通オブジェクトと呼ばれます。
- **NONE:** データベース・オブジェクトは共有されません。

ほとんどのタイプのアプリケーション共通オブジェクトの場合、SHARING句の有効な設定はMETADATAおよびNONEのみです。次のタイプのアプリケーション共通オブジェクトでは、SHARING句に追加の設定を行うことができます。

- 表(オブジェクト表を除く)の場合、SHARING句にはMETADATA、DATA、EXTENDED DATAまたはNONEを設定できます。オブジェクト表の場合は、METADATAまたはNONEのみが有効です。
- ビュー(オブジェクト・ビューを除く)の場合、SHARING句にはMETADATA、DATA、EXTENDED DATAまたはNONEを設定できます。オブジェクト・ビューの場合は、METADATAまたはNONEのみが有効です。
- 順序の場合、SHARING句にはMETADATA、DATAまたはNONEを設定できます。

メタデータ・リンク順序を使用する場合、各アプリケーションPDBには固有の順序があります。1つのアプリケーションPDBのNEXTVAL疑似列を使用して、メタデータ・リンク順序を増やす場合、アプリケーション・コンテナの他のアプリケーションPDBの順序の値に影響しません。

データ・リンク順序を使用する場合、各アプリケーションPDBはアプリケーション・ルートで同じ順序を共有します。1つの

アプリケーションPDBのNEXTVAL疑似列を使用して、メタデータ・リンク順序を増やす場合、同じアプリケーション・コンテナの他のすべてのアプリケーションPDBも変更されます。

アプリケーション共通オブジェクトは、アプリケーションのインストール、アップグレードまたはパッチの一部としてのみ作成または変更できます。アプリケーションPDBは、変更を行ったアプリケーションと同期する場合にアプリケーション共通オブジェクトに変更を適用します。アプリケーション共通オブジェクトの作成、削除または変更時にアプリケーションPDBがクローズされている場合、アプリケーションPDBがオープンされてアプリケーションと同期したときに、適切な変更がアプリケーションPDBに適用されます。

アプリケーション共通オブジェクトの名前は、アプリケーション・ルートに属するアプリケーションPDB内のローカル・データベース・オブジェクトまたはCDBルート内のOracleが提供する共通オブジェクトの名前と競合することはできません。新規にオープンされたアプリケーションPDBに、名前がアプリケーション共通オブジェクトの名前と競合するローカル・データベース・オブジェクトが含まれている場合、アプリケーションPDBはRESTRICTEDモードでオープンされます。この場合は、アプリケーションPDBを通常モードでオープンする前に名前の競合を解決する必要があります。

親トピック: [アプリケーション共通オブジェクトについて](#)

メタデータリンク・アプリケーション共通オブジェクトについて

メタデータリンク・アプリケーション共通オブジェクトの場合、オブジェクトのメタデータはアプリケーション・ルートに1回格納されます。

アプリケーション・ルートに属する各アプリケーションPDB内のメタデータ・リンクにより、アプリケーションPDBはオブジェクト名と構造を含むオブジェクトのメタデータを共有できます。アプリケーション・ルートおよびアプリケーション・ルートに属する各アプリケーションPDBを含め、オブジェクトのデータは各コンテナに固有です。

メタデータリンク・アプリケーション共通オブジェクトに対するデータ定義言語 (DDL) 操作は、アプリケーションのインストール、アップグレードまたはパッチの一部としてのみアプリケーション・ルートで実行できます。ただし、データは通常の実行言語 (DML) の操作を使用してアプリケーションPDBで変更できます。

たとえば、複数の地域オフィスがある会社について考えます。その会社は従業員に関する情報の構造に一貫性を持たせる必要がありますが、各オフィスの従業員は異なります。この会社がアプリケーション・コンテナに人事アプリケーションを持つ場合は、地域オフィスごとに異なるアプリケーションPDBを作成し、メタデータリンク表を使用して従業員情報を格納できます。列などの表のデータ構造は、各地域オフィスのアプリケーションPDBで同じですが、従業員データは異なります。

別の例として、複数の異なるビジネスで使用されているセールス・アプリケーションを構築およびメンテナンスする会社があります。各ビジネスは、同じセールス・アプリケーションを使用しますが、各ビジネスのデータは異なります。たとえば、各ビジネスには異なる顧客がいるため、異なる顧客データがあります。各クライアントがそのアプリケーションに同じデータ構造を使用するようにするために、会社はメタデータリンク・アプリケーション共通オブジェクトを含むアプリケーション・コンテナを作成できます。セールス・アプリケーションを使用する各ビジネスには固有のアプリケーションPDBがあり、データ構造は各アプリケーションPDBで同じですが、データは異なります。

親トピック: [アプリケーション共通オブジェクトについて](#)

拡張データリンク・アプリケーション共通オブジェクトについて

データリンク・アプリケーション共通オブジェクトの場合は、オブジェクトのメタデータとデータの両方がアプリケーション・ルートに1回格納されます。アプリケーション・ルートに属する各アプリケーションPDB内のデータ・リンクにより、アプリケーションPDBはオブジェクトのメタデータとデータを共有できます。


データリンク・アプリケーション共通オブジェクトに対するDDL操作は、アプリケーションのインストール、アップグレードまたはパッチの一部としてのみアプリケーション・ルートで実行できます。また、データはアプリケーション・ルートでのみ通常の実行言語 (DML) 操作を使用して変更できます。データはアプリケーションPDBで変更できません。

たとえば、複数の地域オフィスがある会社について考えます。会社では、製品名や説明など、販売する製品に関する情報をす

すべての地域オフィスで一貫させる必要があります。この会社がアプリケーション・コンテナにセールス・アプリケーションを持つ場合は、地域オフィスごとに異なるアプリケーションPDBを作成し、データリンク表を使用して製品情報を格納できます。各アプリケーションPDBは製品情報を問い合わせることができ、製品情報は各地域オフィスで一貫します。

データリンク・アプリケーション共通オブジェクトは、標準で変更のないデータにも役立ちます。たとえば、国の郵便番号を格納する表は、アプリケーション・コンテナ内のデータリンク・アプリケーション共通オブジェクトにすることができます。すべてのアプリケーションPDBは、アプリケーション・ルート内の同じ郵便番号データにアクセスします。

ノート:



データリンク・アプリケーション共通オブジェクトがプロキシ PDB を使用してアプリケーション・ルート・レプリカを同期する構成の一部である場合、アプリケーション・ルートのデータリンク・オブジェクトに対する DML 操作はアプリケーションのアクションの外側で行うことができますが、その DML 操作はプロキシ PDB を介してアプリケーション・ルート・レプリケーションに自動的に伝播されません。DML 操作をアプリケーション・ルート・レプリカに伝播する場合、アプリケーション・ルート内のデータリンク・オブジェクトに対する DML 操作は、アプリケーションのインストール、アップグレードまたはパッチ適用内で行う必要があります。

親トピック: [アプリケーション共通オブジェクトについて](#)


拡張データリンク・アプリケーション共通オブジェクトについて

拡張データリンク・オブジェクトの場合、各アプリケーションPDBは、アプリケーション・ルートの共通データを共有しながら独自データを作成できます。アプリケーション・ルートに格納されるデータのみがすべてのアプリケーションPDBに共通です。

拡張データリンク・アプリケーション共通オブジェクトに対するDDL操作は、アプリケーションのインストール、アップグレードまたはパッチの一部としてのみアプリケーション・ルートで実行できます。ただし、データは通常のDMLの操作を使用してアプリケーション・ルートまたはアプリケーションPDBで変更できます。

たとえば、アプリケーション・コンテナ内のセールス・アプリケーションは複数のアプリケーションPDBをサポートできますが、すべてのアプリケーションPDBに出荷を目的として米国の郵便番号が必要です。この場合、郵便番号は、すべてのアプリケーションPDBからアクセスできるようにアプリケーション・ルートに格納できます。ただし、1つのアプリケーションPDBはカナダでもセールスを行い、このアプリケーションPDBは米国とカナダの郵便番号を必要とします。この1つのアプリケーションPDBは、アプリケーション・ルートではなくアプリケーションPDB内の拡張データリンク・オブジェクトにカナダの郵便番号を格納できます。

ノート:

- 
- 拡張データリンク・オブジェクトにすることができるデータベース・オブジェクトのタイプは、表およびビューのみです。
 - 拡張データリンク・アプリケーション共通オブジェクトがプロキシ PDB を使用してアプリケーション・ルート・レプリカを同期する構成の一部である場合、アプリケーション・ルートの拡張データリンク・オブジェクトに対する DML 操作はアプリケーションのアクションの外側で行うことができますが、その DML 操作はプロキシ PDB を介してアプリケーション・ルート・レプリケーションに自動的に伝播されません。DML 操作をアプリケーション・ルート・レプリカに伝播する場合、アプリケーション・ルート内の拡張データリンク・オブジェクトに対する DML 操作は、アプリケーションのインストール、アップグレードまたはパッチ適用内で行う必要があります。

親トピック: [アプリケーション共通オブジェクトについて](#)

アプリケーション共通オブジェクトの制限

いくつかの制限がアプリケーション共通オブジェクトに適用されます。

アプリケーション共通オブジェクトに対する問合せでは、現在のコンテナではないコンテナからデータを返すことができます。たとえば、現在のコンテナがアプリケーション・ルートである場合、CONTAINERS句が含まれている問合せでは、アプリケーションPDBからメタデータリンク・アプリケーション共通オブジェクトのデータを返すことができます。また、現在のコンテナがアプリケーションPDBである場合、データリンク・アプリケーション共通オブジェクトおよび拡張データリンク・アプリケーション共通オブジェクトに対する問合せでは、アプリケーション・ルートにあるデータが返されます。

次のタイプの列では、データが返される問合せで、現在のコンテナではないコンテナからデータが返されません。

- ユーザー定義型: オブジェクト型、VARRAY、REFおよびネストした表
- Oracle提供型: ANYTYPE、ANYDATASET、URI型、SDO_TOPO_GEOMETRY、SDO_GEORASTERおよびExpression

また、オブジェクト表およびオブジェクト・ビューに対する問合せでは、現在のコンテナではないコンテナからはデータが返されません。

関連トピック

- [アプリケーションPDB間のアプリケーション共通オブジェクトの問合せ](#)

親トピック: [アプリケーション共通オブジェクトの管理](#)

アプリケーション共通オブジェクトの作成

DEFAULT_SHARING初期化パラメータが適切な値に設定されていることを確認するか、SHARING句をCREATE SQL文に含めることで、アプリケーション共通オブジェクトをアプリケーション・ルートに作成します。

アプリケーションのインストール、アップグレードまたはパッチの一部としてメタデータリンク・オブジェクト、拡張データリンクまたはデータリンク・オブジェクトをアプリケーション・ルートに作成できます。アプリケーションPDBは、アプリケーション・ルート内のアプリケーションと同期する場合にアプリケーション共通オブジェクトに変更を適用します。

1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
現在のユーザーは、データベース・オブジェクトを作成するために必要な権限を持っている必要があります。
2. アプリケーションのインストール、アップグレードまたはパッチを開始するにはALTER PLUGGABLE DATABASE APPLICATION BEGIN文を実行します。

たとえば、アプリケーション・インストールの一部としてアプリケーション共通オブジェクトを作成している場合は、次の形式でALTER PLUGGABLE DATABASE APPLICATION BEGIN INSTALL文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name
BEGIN INSTALL 'application_version_number';
```

3. アプリケーション共通オブジェクトを作成し、次のいずれかの方法でその共有属性を指定します。
 - DEFAULT_SHARING初期化パラメータがアプリケーション・ルートの目的の共有属性に設定されていることを確認し、CREATE SQL文を発行してデータベース・オブジェクトを作成します。
 - CREATE SQL文を発行し、SHARING句を含めてMETADATA、DATAまたはEXTENDED DATAを設定します。
SHARING句がSQL文に含まれる場合、これはDEFAULT_SHARING初期化パラメータで指定された値よりも優先されます。たとえば、DEFAULT_SHARING初期化パラメータがアプリケーション・ルートでMETADATAに設定され、データベース・オブジェクトがSHARINGをDATAに設定して作成された場合、データベース・オブジェクトはデータリンク・データベース・オブジェクトとして作成されます。



ノート:

データベース・オブジェクトが作成された後、その共有属性は変更できません。

4. アプリケーションのインストール、アップグレードまたはパッチを終了するにはALTER PLUGGABLE DATABASE APPLICATION END文を実行します。

たとえば、アプリケーション・インストールの一部としてアプリケーション共通オブジェクトを作成している場合は、次の形式でALTER PLUGGABLE DATABASE APPLICATION END INSTALL文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name
END INSTALL 'application_version_number';
```



ノート:

application_name および application_version_number が ALTER PLUGGABLE DATABASE APPLICATION BEGIN INSTALL 文と ALTER PLUGGABLE DATABASE APPLICATION END INSTALL 文で一致していることを確認します。

5. アプリケーションPDBを現在のコンテナとして使用し、SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATIONを発行して、これらの変更を適用する必要があるすべてのアプリケーションPDBを同期します。

例17-11 DEFAULT_SHARING初期化パラメータの設定

この例では、DEFAULT_SHARING初期化パラメータをメモリとSPFILEの両方でDATAに設定します。共有をサポートするデータベース・オブジェクトがアプリケーション・ルートに作成されており、SHARING句がCREATE SQL文に含まれない場合、データベース・オブジェクトはDEFAULT_SHARING初期化パラメータで指定された共有属性を使用します。

```
ALTER SYSTEM SET DEFAULT_SHARING=DATA SCOPE=BOTH;
```

例17-12 メタデータリンク・オブジェクトの作成

この例では、SHARING=METADATA句を含めることでemployees_mdメタデータリンク表を作成します。application_nameはsalesappであり、application_version_numberは4.2で、オブジェクトはアプリケーションのインストール時に作成されます。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp BEGIN INSTALL '4.2';
CREATE TABLE employees_md SHARING=METADATA
(employee_id    NUMBER(6),
 first_name     VARCHAR2(20),
 last_name     VARCHAR2(25) CONSTRAINT emp_last_name_nn_demo NOT NULL,
 email         VARCHAR2(25) CONSTRAINT emp_email_nn_demo    NOT NULL,
 phone_number  VARCHAR2(20),
 hire_date     DATE DEFAULT SYSDATE
              CONSTRAINT emp_hire_date_nn_demo NOT NULL,
 job_id        VARCHAR2(10) CONSTRAINT emp_job_nn_demo NOT NULL,
 salary       NUMBER(8,2)  CONSTRAINT emp_salary_nn_demo NOT NULL,
 commission_pct NUMBER(2,2),
 manager_id   NUMBER(6),
 department_id NUMBER(4),
 dn          VARCHAR2(300),
 CONSTRAINT emp_salary_min_demo CHECK (salary > 0),
 CONSTRAINT emp_email_uk_demo UNIQUE (email));
ALTER PLUGGABLE DATABASE APPLICATION salesapp END INSTALL '4.2';
```

例17-13 データリンク・オブジェクトの作成

この例では、SHARING=DATA句を含めることでproduct_descriptions_obデータリンク表を作成します。application_nameはsalesappであり、application_version_numberは4.2で、オブジェクトはアプリケーションのインストール時に作成されます。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp BEGIN INSTALL '4.2';
CREATE TABLE product_descriptions_ob SHARING=DATA (
  product_id          NUMBER(6),
  language_id         VARCHAR2(3),
  translated_name     NVARCHAR2(50)
    CONSTRAINT translated_name_nn NOT NULL,
  translated_description NVARCHAR2(2000)
    CONSTRAINT translated_desc_nn NOT NULL);
ALTER PLUGGABLE DATABASE APPLICATION salesapp END INSTALL '4.2';
```

例17-14 拡張データリンク・オブジェクトの作成

この例では、EXTENDEDキーワードとSHARING句を含めることでpostalcodes拡張データリンク表を作成します。application_nameはsalesappであり、application_version_numberは4.2で、オブジェクトはアプリケーションのインストール時に作成されます。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp BEGIN INSTALL '4.2';
CREATE TABLE postalcodes SHARING=EXTENDED DATA
  (code          VARCHAR2(7),
   country_id   NUMBER,
   place_name   VARCHAR2(20));
ALTER PLUGGABLE DATABASE APPLICATION salesapp END INSTALL '4.2';
```

例17-15 アプリケーション・ルートで共有されないオブジェクトの作成

この例では、departments_ns表を作成し、SHARING=NONE句を含めることで共有共通アプリケーション・オブジェクトではないことを指定します。作成後に、このデータベース・オブジェクトはアプリケーション・ルートでのみアクセスできます。

```
CREATE TABLE departments_ns SHARING=NONE
  (department_id   NUMBER(4),
   department_name VARCHAR2(30) CONSTRAINT dept_name_nn NOT NULL,
   manager_id      NUMBER(6),
   location_id     NUMBER(4),
   dn              VARCHAR2(300));
```

ノート:



ALTER PLUGGABLE DATABASE APPLICATION BEGIN および END 文は、共有共通オブジェクトではないオブジェクトの作成時には不要です。ただし、ALTER PLUGGABLE DATABASE APPLICATION BEGIN 文と END 文の間で共有されないオブジェクトを作成する場合、オブジェクトはアプリケーションと同期するアプリケーション PDB に作成されます。

関連トピック

- [アプリケーション・コンテナ内のアプリケーションの管理](#)
- [アプリケーションPDB内のアプリケーションの同期](#)

親トピック: [アプリケーション共通オブジェクトの管理](#)

アプリケーション共通オブジェクトに対するDML文の発行

メタデータリンク、データリンクおよび拡張データリンクのアプリケーション共通オブジェクトに対するDML文の発行のルールは異なります。

- [メタデータリンク共通オブジェクトに対するDMLの発行](#)
メタデータリンク・アプリケーション・オブジェクトには、通常どおりDMLを発行できます。
- [データリンク共通オブジェクトに対するDMLの発行](#)
データリンク・アプリケーション・オブジェクトの場合は、アプリケーション・ルートで通常どおりDMLを発行します。拡張データリンク・アプリケーション・オブジェクトの場合は、アプリケーション・ルートおよびアプリケーションPDBで通常どおりDMLを発行します。

関連トピック

- [アプリケーション・コンテナ内のアプリケーションの管理](#)
- [アプリケーションPDB内のアプリケーションの同期](#)
- [アプリケーション・ルート・レプリカのプロキシPDBとの同期](#)

親トピック: [アプリケーション共通オブジェクトの管理](#)

メタデータリンク共通オブジェクトに対するDMLの発行

メタデータリンク・アプリケーション・オブジェクトには、通常どおりDMLを発行できます。

メタデータリンク・アプリケーション共通オブジェクトの場合、オブジェクトの定義はすべてのアプリケーションPDBで同じですが、データは異なります。ユーザーとアプリケーションは、通常のデータベース・オブジェクトと同じ方法でこれらのオブジェクトに対してDML文を発行できます。DMLは現在のコンテナにのみ作用します。

- [CONTAINERS句を使用した問合せ](#)
メタデータリンク・オブジェクトの場合は、CONTAINERS句を使用してアプリケーション・コンテナ内のすべてのPDBに対して表またはビューを問い合わせることができます。
- [デフォルトのコンテナまたはDMLの設定](#)
メタデータリンク・オブジェクトのCONTAINERS_DEFAULT属性を設定すると、アプリケーション・ルートで発行されるDMLがデフォルトでCONTAINERS句でラップされます。

親トピック: [アプリケーション共通オブジェクトに対するDML文の発行](#)

CONTAINERS句を使用した問合せ

メタデータリンク・オブジェクトの場合は、CONTAINERS句を使用してアプリケーション・コンテナ内のすべてのPDBに対して表またはビューを問い合わせることができます。

メタデータリンク・オブジェクトの場合は、アプリケーション・ルートでDMLを実行するときにCONTAINERS句が役に立ちます。問合せによってUNION ALLが実行されると、ルートとオープンしているすべてのアプリケーションPDB (RESTRICTEDモードのPDBを除く)のオブジェクトからすべての行が戻されます。

PDBのサブセットを問い合わせるには、述語にCON_IDまたはCON\$NAMEを指定します。問合せ対象の表またはビューにCON_ID列が含まれていない場合、問合せはCON_ID列を問合せ結果に追加し、与えられた行が表すデータがあるコンテナを識別します。

前提条件

次の前提条件に注意してください。

- アプリケーション・コンテナのデータを問い合わせるには、共通ユーザーとしてアプリケーション・ルートに接続する必要があります。

- 表またはビューは、アプリケーション・コンテナのアプリケーション・ルートとすべてのPDBに存在する必要があります。
- 表またはビューは、自分のスキーマ内に存在している必要があります。schemaを指定する必要はありませんが、指定する場合は自分のスキーマを指定する必要があります。

アプリケーション・コンテナのメタデータリンク・オブジェクトを問い合わせるには:

1. アプリケーション共通ユーザーとしてアプリケーション・ルートにログインします。
2. SELECT文にCONTAINERS句を指定します。

たとえば、次の文はルートとすべてのアプリケーションPDBのsh. customers表の行数をカウントします(出力例を含む)。

```
SELECT c. CON_ID, COUNT(*)
FROM   CONTAINERS(sh. customers) c
GROUP BY c. CON_ID
ORDER BY 1;
```

CON_ID	COUNT (*)
3	20002
6	426
8	7232

親トピック: [メタデータリンク共通オブジェクトに対するDMLの発行](#)

デフォルトのコンテナまたはDMLの設定

メタデータリンク・オブジェクトのCONTAINERS_DEFAULT属性を設定すると、アプリケーション・ルートで発行されるDMLがデフォルトでCONTAINERS句でラップされます。

ALTER TABLE文またはALTER VIEW文にENABLE CONTAINERS_DEFAULTを設定します。DBA_TABLESおよびDBA_VIEWSビューのCONTAINERS_DEFAULT列は、デフォルトでCONTAINERS句に対してデータベース・オブジェクトが有効になるかどうかを示します。

メタデータリンク表またはビューに関するDMLのデフォルト・コンテナを設定するには:

1. アプリケーション共通ユーザーとしてアプリケーション・ルートにログインします。
2. アプリケーション・ルートでENABLE CONTAINERS_DEFAULT句を指定したALTER TABLE文またはALTER VIEW文を発行します。

次の文は、sh. customersのデフォルト・コンテナを設定します。

```
ALTER TABLE sh. customers ENABLE CONTAINERS_DEFAULT;
```

この属性を設定した後、アプリケーション・ルートで発行される問合せとDML文はsh. customersに対してデフォルトでCONTAINERS句を使用します。

親トピック: [メタデータリンク共通オブジェクトに対するDMLの発行](#)

データリンク共通オブジェクトに対するDMLの発行

データリンク・アプリケーション・オブジェクトの場合は、アプリケーション・ルートで通常どおりDMLを発行します。拡張データリンク・アプリケーション・オブジェクトの場合は、アプリケーション・ルートおよびアプリケーションPDBで通常どおりDMLを発行します。

データリンク・アプリケーション・オブジェクトの場合、アプリケーション・ルートのDMLはアプリケーション・コンテナのすべてのPDBがアクセスできるデータに作用します。アプリケーションPDBのデータリンク・アプリケーション・オブジェクトに対してDMLを発行することはできません。

拡張データリンク・アプリケーション・オブジェクトの場合、アプリケーション・ルートのDMLはアプリケーション・コンテナのすべての

PDBがアクセスできるデータに作用します。アプリケーションPDBのDMLは、そのアプリケーションPDBに固有のデータにのみ作用します。

データリンクまたは拡張データリンク・オブジェクトを含むアプリケーション・ルートがあるとして、また、このルートはプロキシPDBと同期されたアプリケーション・ルート・レプリカのマスターであるとして、この場合、アプリケーションのインストール、アップグレードまたはパッチ適用中にDMLが実行されると、DMLはレプリカとのみ同期します。特に、ルートでのDMLはALTER PLUGGABLE DATABASE APPLICATION ... {BEGIN|END}文の間で実行される必要があります。他のDMLは、現在のルートにのみ適用され、ルート・レプリカとは同期されません。

アプリケーション・ルート・レプリカ構成の一部ではないアプリケーション共通オブジェクトのDMLを発行するには：

1. データベース・オブジェクトに対してDML文を発行するために必要な権限を持つユーザーとして、アプリケーション・コンテナ内の該当するコンテナに接続します。
2. DML文を通常どおりに発行します。

アプリケーション・ルート・レプリカ構成の一部ではないデータリンク・オブジェクトまたは拡張データリンク・オブジェクトのDMLを発行するには：

1. SQL*Plusで、現在のコンテナが構成のアプリケーション・ルート・レプリカのマスター・アプリケーション・ルートであることを確認します。

現在のユーザーは、データベース・オブジェクトに対してDML文を発行するために必要な権限を持っている必要があります。

2. アプリケーションのインストール、アップグレードまたはパッチを開始するためにALTER PLUGGABLE DATABASE APPLICATION ... BEGIN文を実行します。

アプリケーション・アップグレードの一部としてアプリケーション共通オブジェクトを変更する場合は、次の形式でアップグレード文を発行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name BEGIN UPGRADE
'application_start_version_number' TO
'application_end_version_number';
```

たとえば、application_nameがsalesapp、application_start_version_numberが4.2、application_end_version_numberが4.3の場合は、次の文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp
BEGIN UPGRADE '4.2' TO '4.3';
```

3. データリンク・アプリケーション共通オブジェクトに対してDML文を発行します。
4. ALTER PLUGGABLE DATABASE APPLICATION ... END文を実行します。

たとえば、アプリケーション・アップグレードの一部としてアプリケーション共通オブジェクトを変更する場合は、次の形式で文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name END UPGRADE
TO 'application_end_version_number';
```

たとえば、application_nameがsalesapp、application_start_version_numberが4.2、application_end_version_numberが4.3の場合は、次の文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp END UPGRADE TO '4.3';
```


ノート:



application_name および application_end_version_number が ALTER PLUGGABLE DATABASE APPLICATION BEGIN UPGRADE 文と ALTER PLUGGABLE DATABASE APPLICATION END UPGRADE 文で一致していることを確認します。

5. これらの変更を適用する必要があるすべてのアプリケーションPDBを同期するには、アプリケーションPDBが現在のコンテナであるときに、SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATION文を発行します。

親トピック: [アプリケーション共通オブジェクトに対するDML文の発行](#)

DDL文を使用したアプリケーション共通オブジェクトの変更

特定のDDL文でアプリケーション・ルート内のアプリケーション共通オブジェクトを変更する場合は、ALTER PLUGGABLE DATABASE APPLICATION BEGIN文とALTER PLUGGABLE DATABASE APPLICATION END文の間でオブジェクトを変更する必要があります。変更を適用するためにアプリケーションPDBがアプリケーションと同期する必要があります。

アプリケーション・ルート内のメタデータリンク・オブジェクトまたはデータリンク・オブジェクトを変更できます。ALTER、RENAMEまたはDROP SQL文をデータベース・オブジェクトに対して実行してDDL変更を実行します。

1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
現在のユーザーは、データベース・オブジェクトに対する計画的な変更を行うために必要な権限を持っている必要があります。
2. アプリケーションのインストール、アップグレードまたはパッチを開始するにはALTER PLUGGABLE DATABASE APPLICATION BEGIN文を実行します。
たとえば、アプリケーション・アップグレードの一部としてアプリケーション共通オブジェクトを変更している場合は、次の形式でALTER PLUGGABLE DATABASE APPLICATION BEGIN UPGRADE文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name BEGIN UPGRADE  
'application_start_version_number' TO 'application_end_version_number';
```

たとえば、application_nameがsalesapp、application_start_version_numberが4.2、application_end_version_numberが4.3の場合は、次の文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp BEGIN UPGRADE  
'4.2' TO '4.3';
```

3. DDL文でアプリケーション共通オブジェクトを変更します。
たとえば、ALTER TABLE文は列を表に追加できます。
4. アプリケーションのインストール、アップグレードまたはパッチを終了するにはALTER PLUGGABLE DATABASE APPLICATION END文を実行します。
たとえば、アプリケーション・アップグレードの一部としてアプリケーション共通オブジェクトを変更している場合は、次の形式でALTER PLUGGABLE DATABASE APPLICATION END UPGRADE文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name END UPGRADE  
TO 'application_end_version_number';
```

たとえば、application_nameがsalesappでapplication_end_version_numberが4.3の場合は、次の文を実行します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp END UPGRADE TO '4.3';
```



ノート:

application_name および application_end_version_number が ALTER PLUGGABLE DATABASE APPLICATION BEGIN UPGRADE 文と ALTER PLUGGABLE DATABASE APPLICATION END UPGRADE 文で一致していることを確認します。

- アプリケーションPDBを現在のコンテナとして使用し、SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATION を発行して、これらの変更を適用する必要があるすべてのアプリケーションPDBを同期します。

関連トピック

- [アプリケーション・コンテナ内のアプリケーションの管理](#)
- [アプリケーションPDB内のアプリケーションの同期](#)

親トピック: [アプリケーション共通オブジェクトの管理](#)

アプリケーション・コンテナ内のコンテナに対するDML文の発行

アプリケーション・ルートで発行されるDML文では、アプリケーション・コンテナ内の1つ以上のコンテナを変更できます。また、DML文に対して1つ以上のデフォルト・コンテナ・ターゲットを指定できます。

- [アプリケーション・コンテナ内のコンテナに対するDML文の発行について](#)
DML文は、アプリケーション・コンテナ内の複数のコンテナ内のデータベース・オブジェクトに影響することがあります。
- [アプリケーション・コンテナ内のDML文のデフォルト・コンテナの指定](#)
アプリケーション・コンテナ内のDML文のデフォルト・コンテナを指定するには、CONTAINERS DEFAULT TARGET句を指定したALTER PLUGGABLE DATABASE文を発行します。

親トピック: [アプリケーション・コンテナの管理](#)

アプリケーション・コンテナ内のコンテナに対するDML文の発行について

DML文は、アプリケーション・コンテナ内の複数のコンテナ内のデータベース・オブジェクトに影響することがあります。

アプリケーション・ルートで、CONTAINERS句を含む単一のDML文は、アプリケーション・コンテナ内の1つ以上のコンテナ内の表またはビューを変更できます。CONTAINERS句を使用するには、変更される表またはビューをCONTAINERS句に指定し、コンテナをWHERE句に指定します。INSERT VALUES文でターゲット・コンテナを指定するには、VALUES句にCON_ID値を指定します。また、UPDATE文またはDELETE文でターゲット・コンテナを指定するには、WHERE句にCON_ID述語を指定します。

たとえば、次のDML文はコンテナ内のsales.customers表を7または8のCON_IDで更新します。

```
UPDATE CONTAINERS(sales.customers) ctab
SET ctab.city_name=' MIAMI'
WHERE ctab.CON_ID IN(7,8) AND
CUSTOMER_ID=3425;
```

WHERE句のCON_IDに指定する値は、現在のアプリケーション・コンテナ内のコンテナに対するものである必要があります。

DML操作のデフォルト・ターゲット・コンテナを指定できます。DML文でWHERE句にCON_IDの値が指定されていない場合、DML操作のターゲット・コンテナはアプリケーション・ルートのデータベース・プロパティCONTAINERS_DEFAULT_TARGETで指定されたコンテナです。アプリケーション・ルートで発行した場合、次のDML文はアプリケーション・コンテナのデフォルト・ターゲット・コンテナを変更します。

```
UPDATE CONTAINERS(sales.customers) ctab
SET ctab.city_name=' MIAMI'
WHERE CUSTOMER_ID=3425;
```

デフォルトでは、アプリケーション・コンテナ内のデフォルト・ターゲット・コンテナにはすべてのアプリケーションPDBが含まれますが、そのアプリケーション・ルートまたはアプリケーション・シードは含まれません。次の問合せを実行することで、アプリケーション・コンテナのデフォルト・ターゲット・コンテナを決定できます。

```
SELECT PROPERTY_VALUE
FROM   DATABASE_PROPERTIES
WHERE  PROPERTY_NAME='CONTAINERS_DEFAULT_TARGET';
```

また、アプリケーション・ルートの表またはビューに対してCONTAINERS_DEFAULT属性を有効にできます。この属性が有効になっている場合、CONTAINERS句はデフォルトでデータベース・オブジェクトに対する問合せおよびDML文に使用され、CONTAINERS句はSQL文で指定する必要がありません。アプリケーション・ルートの表またはビューに対してCONTAINERS_DEFAULT属性を有効にするには、ENABLE CONTAINERS_DEFAULTを指定したALTER TABLEまたはALTER VIEW文を実行します。

CONTAINERS句には、次の制限事項が適用されます。

- CONTAINERS DEFAULT TARGET句はSELECT文に影響しません。
- INSERTのターゲットがCONTAINERS () 内にあるSELECT文を使用したINSERT文はサポートされません。
- INSERTのターゲットがCONTAINERS () 内にある、複数表に対するINSERT文はサポートされません。
- CONTAINERS句を使用するDML文では、データベース・リスナーがTCP (IPCではなく)を使用して構成され、PORT値およびHOST値がPORT句およびHOST句を使用して各ターゲットPDBに指定されている必要があります。

関連トピック

- [アプリケーション共通オブジェクトについて](#)

親トピック: [アプリケーション・コンテナ内のコンテナに対するDML文の発行](#)

アプリケーション・コンテナにおけるDML文のデフォルト・コンテナの指定

アプリケーション・コンテナ内のDML文のデフォルト・コンテナを指定するには、CONTAINERS DEFAULT TARGET句を指定したALTER PLUGGABLE DATABASE文を発行します。

WHERE句でコンテナを指定せずにDML文がアプリケーション・ルートで発行されている場合、DML文はアプリケーション・コンテナのデフォルト・コンテナに影響します。デフォルト・コンテナには、アプリケーション・ルートまたはアプリケーションPDBを含む、アプリケーション・コンテナ内の任意のコンテナを指定できます。1つのデフォルトのコンテナのみを使用できます。

1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
現在のユーザーには、ALTER PLUGGABLE DATABASE権限が共通して付与されている必要があります。
2. CONTAINERS DEFAULT TARGET句を指定してALTER PLUGGABLE DATABASE文を実行します。

例17-16 アプリケーション・コンテナ内のDML文のデフォルト・コンテナの指定

この例では、APDB1がアプリケーション・コンテナ内のDML文のデフォルト・コンテナであることを指定します。

```
ALTER PLUGGABLE DATABASE CONTAINERS DEFAULT TARGET = (APDB1);
```

例17-17 デフォルト・コンテナのクリア

この例は、デフォルト・コンテナ設定をクリアしています。設定されていない場合、デフォルト・コンテナはアプリケーション・ルートです。

```
ALTER PLUGGABLE DATABASE CONTAINERS DEFAULT TARGET = NONE;
```

親トピック: [アプリケーション・コンテナ内のコンテナに対するDML文の発行](#)

コンテナ・マップを使用したPDBでのパーティション化

コンテナ・マップにより、データが表レベルで物理的にパーティション化されていない場合にデータをアプリケーションPDBレベルでパーティション化できます。

- [コンテナ・マップについて](#)
コンテナ・マップは、アプリケーション・ルートで定義されているパーティション化されたマップ表を指定するデータベース・プロパティです。
- [コンテナ・マップの作成](#)
マップ・オブジェクトを作成し、CONTAINER_MAPデータベース・プロパティをマップ・オブジェクトに設定することで、コンテナ・マップを作成します。

親トピック: [アプリケーション・コンテナの管理](#)

コンテナ・マップについて

コンテナ・マップは、アプリケーション・ルートで定義されているパーティション化されたマップ表を指定するデータベース・プロパティです。

コンテナ・マップを使用して、メタデータリンク・オブジェクトのデータをパーティション化します。コンテナ・マップは、よく使用される列に基づいてアプリケーションPDB内のデータをパーティション化します。

たとえば、各アプリケーションPDBに異なるデータを格納する、countries_mlt (cname列を使用)というメタデータリンク表を作成する場合があります。pdb_map_tblというマップ表がcname列でリスト別にパーティション化します。amer_pdb、euro_pdbおよびasia_pdbの各パーティションがアプリケーションPDBの名前に対応します。

コンテナ・マップは、共通オブジェクトの列に論理パーティション・キーを定義できます。コンテナ・マップに基づいてコンテナが内部的に解決されるため、このマッピングでは、CON_ID述語を使用して問合せを定義するか、または問合せでCONTAINERS句を使用するための要件が削除されます。

一部のタイプの行ベース統合では、複数のテナントを含む単一のPDBでテナントIDを使用します。コンテナ・マップは、テナントごとに異なるPDBを使用する構成に移行する場合に役立ちます。

- [マップ・オブジェクト](#)
マップ・オブジェクトはパーティション化された表です。
- [リスト・パーティション化されたコンテナ・マップ: 例](#)
この例では、地理的地域のデータを格納するPDBに問合せをルーティングするコンテナ・マップが使用されます。
- [レンジ・パーティション化されたコンテナ・マップ: 例](#)
この例では、特定の部門のデータを格納するPDBに問合せをルーティングするコンテナ・マップが使用されます。

親トピック: [コンテナ・マップを使用したPDBでのパーティション化](#)

マップ・オブジェクト

マップ・オブジェクトはパーティション化された表です。

マップ表内のパーティションの名前は、アプリケーション・コンテナ内のアプリケーションPDBの名前と一致します。メタデータリンク・オブジェクトは表レベルで物理的にパーティション化されませんが、コンテナ・マップで使用されるパーティション化戦略を使用して問い合わせることができます。

マップ表をメタデータリンク表に関連付けるには、アプリケーション・ルートに接続しているときに、ALTER PLUGGABLE DATABASE ... CONTAINER_MAPでマップ表を指定します。1つのアプリケーション・コンテナに複数のコンテナ・マップは作成できません。コンテナ・マップはCDBルートで作成できません。

ノート:



- データは、マップ・オブジェクトに定義されているパーティションと整合性のある方法で、PDB 表にロードされる必要があります。
- アプリケーション・コンテナ内のアプリケーション PDB が変更された場合、マップ・オブジェクトはそれらの変更に対応して自動的に同期されません。たとえば、マップ・オブジェクトで参照されているアプリケーション PDB が、切断、名前変更または削除されていることがあります。そのような変更を反映するには、マップ・オブジェクトを手動で更新する必要があります。

Oracle Database 18c以降は、CONTAINERS () 問合せでマップを使用するために、マップ表のパーティション化列がメタデータリンク表の列と一致している必要はありません。表 sh. sales がコンテナ・マップ pdb_map_tbl に対して有効で、cname がマップ表のパーティション化列であるとして。sh. sales に cname 列が含まれていない場合でも、マップ表は次の問合せを適切な PDB にルーティングします: `SELECT * FROM CONTAINERS(sh. sales) WHERE cname = 'US' ORDER BY time_id`。

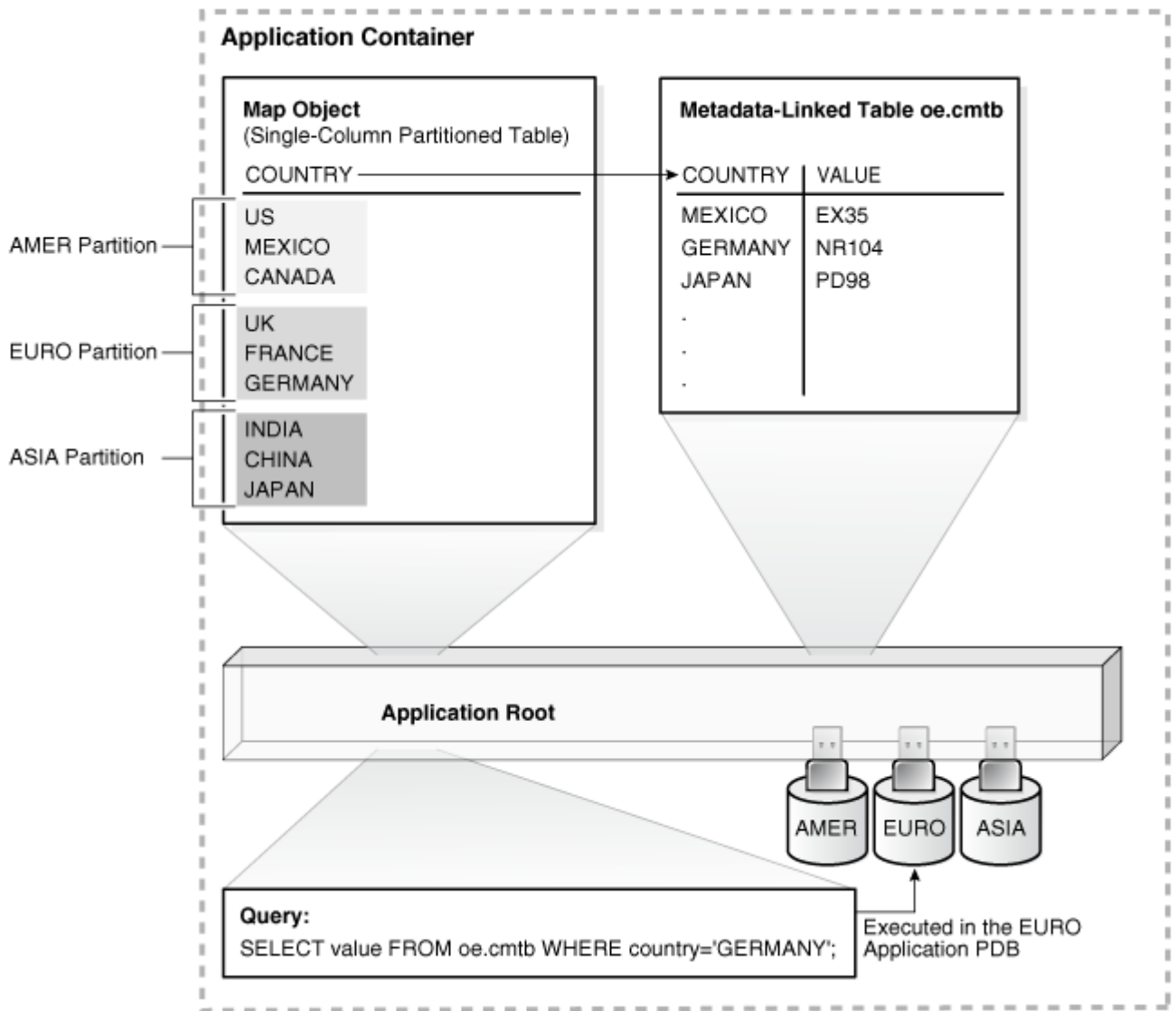
親トピック: [コンテナ・マップについて](#)

リスト・パーティション化されたコンテナ・マップ: 例

この例では、地理的地域のデータを格納する PDB に問合せをルーティングするコンテナ・マップが使用されます。

次のアプリケーション・ルート図は、マップ・オブジェクト、メタデータリンク表、およびメタデータリンク表に対する問合せを示しています。問合せは適切なアプリケーション PDB で実行されます。

図17-3 コンテナ・マップ



図には、アプリケーション・コンテナにAMER、EUROおよびASIAという3つのアプリケーションPDBがあることが示されています。PDBには、対応する地域のデータが格納されます。oe.cmtbというメタデータリンク表は、アプリケーションに関する情報を格納します。この表にはCOUNTRY列があります。このパーティション化戦略のために、リスト別のパーティションを使用して、各地域のパーティションを作成するマップ・オブジェクトを作成します。国の値(図に示されている問合せではGERMANY)によって、地域(ここではEURO)が決定されます。

関連項目:

この例の詳細は説明は、[「コンテナ・マップの作成」](#)を参照してください

親トピック: [コンテナ・マップについて](#)

レンジ・パーティション化されたコンテナ・マップ: 例

この例では、特定の部門のデータを格納するPDBに問合せをルーティングするコンテナ・マップが使用されます。

レンジ・パーティション化された表をマップ・オブジェクトに使用する別の例について考えます。次のSQL文はアプリケーション・ルート内にマップ・オブジェクトを作成します。

```
CREATE TABLE app_con_admin.conmap (
  department_id NUMBER NOT NULL)
PARTITION BY RANGE (department_id) (
PARTITION apppdb1 VALUES LESS THAN (100),
```

```
PARTITION apppdb2 VALUES LESS THAN (200),
PARTITION apppdb3 VALUES LESS THAN (300));
```

このマップ・オブジェクトは、よく使用される列department_idに基づいてアプリケーションPDB apppdb1、apppdb2および apppdb3のデータをパーティション化します。次のSQL文はCONTAINER_MAPデータベース・プロパティをアプリケーション・ルートの app_con_admin.conmap表に設定します。

```
ALTER PLUGGABLE DATABASE SET CONTAINER_MAP=' app_con_admin.conmap' ;
```

コンテナ・マップを使用する問合せは、CONTAINERS句を使用する問合せと同様の結果を生成します。たとえば、次の問合せは同様の結果を返します。

```
SELECT employee_id
FROM   CONTAINERS(hr.employees)
WHERE  department_id = 10
AND    CON_ID IN (44);
```

```
SELECT employee_id
FROM   hr.employees
WHERE  department_id = 10;
```

CONTAINERSを指定した最初の実行例に示されているように、問合せが単一のアプリケーションPDBにのみ関連する場合、問合せではこのアプリケーションPDBのコンテナIDをWHERE句に指定する必要があります。この要件によって、アプリケーション変更が生じることがあります。

2番目の実行例では、CONTAINERS句のかわりにコンテナ・マップを使用しています。2番目の実行例では、コンテナ・マップが問合せを適切なアプリケーションPDBに送信するためコンテナを指定しません。通常、コンテナ・マップを使用する問合せはCONTAINERS句を使用する問合せよりも効率的です。

コンテナ・マップは、ALTER DATABASEシステム権限を持つ共通ユーザーが作成する必要があります。問合せは、コンテナ・マップで有効にされているオブジェクトに対して実行されます。問合せ権限は、オブジェクトに付与されている権限によって決定されます。

親トピック: [コンテナ・マップについて](#)

コンテナ・マップの作成

マップ・オブジェクトを作成し、CONTAINER_MAPデータベース・プロパティをマップ・オブジェクトに設定することで、コンテナ・マップを作成します。

マップ・オブジェクトは、各パーティション名がアプリケーション・コンテナ内のアプリケーションPDBの名前と一致するパーティション化された表です。

前提条件

コンテナ・マップを作成するには、次の前提条件を満たしている必要があります。

- コンテナ・マップを作成する前に、適切なPDBのあるアプリケーション・コンテナがCDBに存在している必要があります。
- アプリケーション・コンテナには、少なくとも1つのアプリケーションがインストールされている必要があります。

コンテナ・マップを作成するには:

1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
2. CONTAINER_MAPデータベース・プロパティをマップ・オブジェクトに設定します。

次の文で、map_table_schemaを表の所有者で置換し、map_table_nameを表の名前で置換します。


```
ALTER DATABASE SET CONTAINER_MAP = 'map_table_schema.map_table_name';
```

3. アプリケーションのインストール、アップグレードまたはパッチ適用を開始します。
4. コンテナ・マップで使用されるメタデータリンク表が存在しない場合、それを作成します。
5. ALTER TABLE ... ENABLE CONTAINER_MAP文を発行することで、問い合わせる表のコンテナ・マップを有効にします。
6. ALTER TABLE ... ENABLE CONTAINERS_DEFAULT文を発行することで、CONTAINERS句に対して問い合わせる表が有効になっていることを確認します。
7. 前に開始したアプリケーションのインストール、アップグレードまたはパッチ適用が終了します。

例17-18 コンテナ・マップの作成と使用

この例では、コンテナ・マップを使用する単純なアプリケーションを作成します。アプリケーション・コンテナにAMER、EUROおよびASIAという3つのアプリケーションPDBがあることを想定します。アプリケーションPDBは異なる地域(それぞれアメリカ、ヨーロッパおよびアジア)のデータを格納します。メタデータリンク表はアプリケーションに関する情報を格納し、COUNTRY列があります。このパーティション化戦略では、リストによるパーティションを使用して、各地域のパーティションを作成するマップ・オブジェクトを作成し、国値を使用して地域を決定します。

1. SQL*Plusで、現在のコンテナがアプリケーション・ルートであることを確認します。
2. マップ・オブジェクトを作成します。

```
CREATE TABLE salesadm.conmap (country VARCHAR2(30) NOT NULL)
PARTITION BY LIST (country) (
  PARTITION AMER VALUES ('US', 'MEXICO', 'CANADA'),
  PARTITION EURO VALUES ('UK', 'FRANCE', 'GERMANY'),
  PARTITION ASIA VALUES ('INDIA', 'CHINA', 'JAPAN')
);
```

3. CONTAINER_MAPデータベース・プロパティをマップ・オブジェクトに設定します。

```
ALTER PLUGGABLE DATABASE SET CONTAINER_MAP='salesadm.conmap';
```

4. アプリケーションのインストールを開始します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp BEGIN INSTALL '1.0';
```

5. コンテナ・マップを使用して問い合わせられるメタデータリンク表を作成します。

```
CREATE TABLE oe.cmtb SHARING=METADATA (
  value VARCHAR2(30),
  country VARCHAR2(30));
```

6. 問い合わせる表に対してコンテナ・マップを有効にします。

```
ALTER TABLE oe.cmtb ENABLE CONTAINER_MAP;
```

7. CONTAINERS句に対して問い合わせる表が有効になっていることを確認します。

```
ALTER TABLE oe.cmtb ENABLE CONTAINERS_DEFAULT;
```

8. アプリケーションのインストールを終了します。

```
ALTER PLUGGABLE DATABASE APPLICATION salesapp END INSTALL '1.0';
```

9. セッションを各アプリケーションPDBに切り替えて同期します。

```
ALTER SESSION SET CONTAINER=amer;
```



```
ALTER PLUGGABLE DATABASE APPLICATION salesapp SYNC;

ALTER SESSION SET CONTAINER=euro;
ALTER PLUGGABLE DATABASE APPLICATION salesapp SYNC;

ALTER SESSION SET CONTAINER=asia;
ALTER PLUGGABLE DATABASE APPLICATION salesapp SYNC;
```

10. パーティション化戦略に基づいて、各アプリケーションPDBのoe.cmtb表に値を挿入します。

```
ALTER SESSION SET CONTAINER=amer;
INSERT INTO oe.cmtb VALUES ('AMER VALUE', 'US');
INSERT INTO oe.cmtb VALUES ('AMER VALUE', 'MEXICO');
INSERT INTO oe.cmtb VALUES ('AMER VALUE', 'CANADA');
COMMIT;

ALTER SESSION SET CONTAINER=euro;
INSERT INTO oe.cmtb VALUES ('EURO VALUE', 'UK');
INSERT INTO oe.cmtb VALUES ('EURO VALUE', 'FRANCE');
INSERT INTO oe.cmtb VALUES ('EURO VALUE', 'GERMANY');
COMMIT;

ALTER SESSION SET CONTAINER=asia;
INSERT INTO oe.cmtb VALUES ('ASIA VALUE', 'INDIA');
INSERT INTO oe.cmtb VALUES ('ASIA VALUE', 'CHINA');
INSERT INTO oe.cmtb VALUES ('ASIA VALUE', 'JAPAN');
COMMIT;
```

11. セッションをアプリケーション・ルートに切り替えて、コンテナ・マップを使用してデータを問い合わせます。

```
ALTER SESSION SET CONTAINER=sales;

SELECT value FROM oe.cmtb WHERE country='MEXICO';

SELECT value FROM oe.cmtb WHERE country='GERMANY';

SELECT value FROM oe.cmtb WHERE country='JAPAN';
```

最初の問合せの出力はAMER VALUE、2番目の問合せの出力はEURO VALUE、3番目の問合せの出力はASIA VALUEである必要があります。これらの値は、コンテナ・マップが正常に動作していることを示します。

親トピック: [コンテナ・マップを使用したPDBでのパーティション化](#)

18 マルチテナント環境のセキュリティの管理

SQL*PlusおよびOracle Enterprise Managerを使用して、マルチテナント環境の共通およびローカルのユーザーとロールを管理できます。

- [共通およびローカルに付与される権限の管理](#)
マルチテナント環境では、CDB全体またはアプリケーション・コンテナに対する共通の権限を付与したり、特定のPDBに対してローカルで権限を付与したりできます。
- [共通ロールおよびローカル・ロールの管理](#)
共通ロールはルートで作成されるロールであり、ローカル・ロールはPDBで作成されます。
- [PDBロックダウン・プロファイルを使用したPDBでの操作の制限](#)
マルチテナント環境でPDBロックダウン・プロファイルを使用して、プラグブル・データベース(PDB)の一連のユーザー操作を制限できます。
- [PDBのオペレーティング・システム・ユーザーの構成](#)
DBMS_CREDENTIAL.CREATE_CREDENTIALプロシージャで、ユーザー・アカウントを、プラグブル・データベース(PDB)のオペレーティング・システム・ユーザーになるように構成します。
- [マルチテナント環境でのアプリケーション・コンテキストの使用](#)
アプリケーション・コンテキストにはユーザーIDが格納されており、これに基づいてデータベースのデータにユーザーがアクセスできるかを否かを決定できます。
- [マルチテナント環境でのOracle Virtual Private Databaseの使用](#)
Oracle Virtual Private Database (VPD)を使用すると、データにアクセスするユーザーをフィルタ処理できます。
- [マルチテナント環境でのTransport Layer Securityの使用](#)
Transport Layer Security (TLS)はアプリケーション・コンテナのマルチテナント環境で使用できます。
- [マルチテナント環境におけるOracle Data Redaction](#)
マルチテナント環境では、Oracle Data Redactionポリシーは、現在のプラグブル・データベース(PDB)内のオブジェクトのみに適用されます。
- [マルチテナント環境での監査の概要](#)
監査とは、構成済のデータベース・アクション(データベース・ユーザーと非データベース・ユーザーの両方からのアクション)を監視して記録することです。非データベース・ユーザーとは、CLIENT_IDENTIFIER属性を使用してデータベースで認識されるアプリケーション・ユーザーのことです。

親トピック: [マルチテナント環境の管理](#)

共通およびローカルに付与される権限の管理

マルチテナント環境では、CDB全体またはアプリケーション・コンテナに対する共通の権限を付与するか、または特定のPDBに対してローカルで権限を付与できます。

- [Oracleマルチテナント・オプションが権限に影響を与えるしくみ](#)
マルチテナント環境では、共通ユーザーを含むすべてのユーザーは、現在のコンテナ内でのみ権限を実行できます。
- [共通およびローカルに付与される権限について](#)
マルチテナント環境では、共通ユーザーとローカル・ユーザーの両方は、相互に権限を付与できます。
- [共通に付与されるシステム権限の使用方法](#)
ユーザーは、システム権限が付与されているPDB内でのみシステム権限を実行できます。
- [共通に付与されるオブジェクト権限の使用方法](#)
共通オブジェクトのオブジェクト権限は、オブジェクト自体とそのオブジェクト上の関連するすべてのリンクに適用されます。

- [PDBへのアクセス権限の付与または取消し](#)
マルチテナント環境では、PDBアクセスに対する権限の付与および取消を実行できます。
- [例: マルチテナント環境での権限の付与](#)
マルチテナント環境で権限を付与するには、GRANT文を使用できます。
- [共通ユーザーによるCONTAINER_DATAオブジェクトの情報の表示](#)
共通ユーザーは、ルート内のCONTAINER_DATAオブジェクトや特定のPDB内のデータに関する情報を表示できます。

親トピック: [マルチテナント環境のセキュリティの管理](#)

Oracleマルチテナント・オプションが権限に影響を与えるしくみ

マルチテナント環境では、共通ユーザーを含むすべてのユーザーは、現在のコンテナ内でのみ権限を実行できます。

ただし、ルートに接続されているユーザーは、他のプラグブル・データベース(PDB)に影響を与える特定の操作を実行できます。これらの操作には、ALTER PLUGGABLE DATABASE、CREATE USER、CREATE ROLEおよびALTER USERが含まれます。共通ユーザーは、これらの操作を可能にする、共通に付与される権限を持つ必要があります。ルートに接続されている共通ユーザーは、ビューにアクセスするために必要な権限を付与されていて、様々なPDBに関するデータを表示できるようにCONTAINER_DATA属性が設定されている場合、ルートのコンテナ・データ・オブジェクト(たとえば、マルチテナント・コンテナ・データベース(CDB)・ビューやV\$ビューなど)を介してPDBに関するメタデータを確認できます。共通ユーザーは、PDBの表またはビューに問合せできません。

共通ユーザーは、他のPDBの権限を実行できません。必要なPDBに最初に切り替えて、そこから権限を実行する必要があります。異なるコンテナに切り替えるには、共通ユーザーにSET CONTAINER権限が必要です。SET CONTAINER権限は、共通に付与するか、ユーザーが切替えを試みるコンテナに付与する必要があります。また、共通ユーザーは、そのPDBのCREATE SESSION権限に応じて、現在の初期コンテナがこのユーザーが必要とするコンテナである新しいデータベース・セッションを開始できます。

共通に付与される権限が個々のPDBに構成されたセキュリティを妨げる場合があるので注意してください。たとえば、アプリケーションPDBのデータベース管理者がPDBのいずれのユーザーも特定のアプリケーション共通オブジェクトを変更できないようにします。PUBLICまたはオブジェクトの共通ユーザーもしくは共通ロールに共通に付与された権限(UPDATEなど)は、PDBのデータベース管理者の意図に反した動作をします。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [共通およびローカルに付与される権限の管理](#)

共通およびローカルに付与される権限について

マルチテナント環境では、共通ユーザーとローカル・ユーザーの両方は、相互に権限を付与できます。

権限自体は、共通でもローカルでもありません。権限がどのように適用されるかは、権限が共通に付与されるか、ローカルに付与されるかによって異なります。

共通に付与される権限の場合:

- 共通に付与される権限は、既存および将来のすべてのコンテナで使用できます。
- 権限を共通に付与できるのは共通ユーザーのみで、権限受領者が共通の場合のみです。
- 共通ユーザーは、他の共通ユーザーまたは共通ロールに権限を付与できます。
- 権限付与者は、ルートに接続して、GRANT文のCONTAINER=ALLを指定する必要があります。
- システム権限とオブジェクト権限は、どちらも共通に付与できます。(オブジェクト権限は、指定したオブジェクトに関して)

のみ実現化します。)

- 共通ユーザーを指定されたコンテナに接続または切り替える場合、様々なアクティビティ(表の作成など)を実行するユーザーの機能は、共通に付与された権限および特定のコンテナでローカルに付与された権限によって制御されます。
- PUBLICには権限を共通に付与しないでください。

ローカルに付与される権限

- ローカルに付与された権限は、それが付与されたコンテナでのみ使用できます。権限がルートで付与されている場合は、ルートにのみ適用されます。
- 共通ユーザーおよびローカル・ユーザーは、どちらも権限をローカルに付与できます。
- 共通ユーザーおよびローカル・ユーザーは、他の共通ロールまたはローカル・ロールに権限を付与できます。
- 権限付与者は、コンテナに接続して、GRANT文のCONTAINER=CURRENTを指定する必要があります。
- ユーザーは、他のユーザーまたはロール(共通およびローカルの両方)あるいはPUBLICロールにローカルに権限を付与できます。

関連トピック

- [CDBでの権限およびロール付与の概要](#)
- [マルチテナント環境でPUBLICロールが機能するしくみ](#)

親トピック: [共通およびローカルに付与される権限の管理](#)

共通に付与されるシステム権限の使用法

ユーザーは、システム権限が付与されているPDB内でのみシステム権限を実行できます。

たとえば、PDB B内の共通ユーザーAにシステム権限がローカルに付与されている場合、ユーザーAは、PDB Bに接続されている間のみ、その権限を実行できます。

システム権限は、ルートでのみ適用可能で、次の要件を満たしている場合は、既存および将来のすべてのPDBで適用できます。

- システム権限付与者が共通ユーザーで、権限受領者が共通ユーザー、共通ロールまたはPUBLICロールです。事実上すべてのユーザーがシステム権限を使用できるようになるため、システム権限をPUBLICロールに共通に付与しないでください。
- システム権限付与者は、共通に付与される権限に対してADMIN OPTIONを所有しています。
- GRANT文には、CONTAINER=ALL句を含める必要があります。

次の例は、共通ユーザーc##hr_adminに権限を共通に付与する方法を示しています。

```
CONNECT SYSTEM
Enter password: password
Connected.

GRANT CREATE ANY TABLE TO c##hr_admin CONTAINER=ALL;
```

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [共通およびローカルに付与される権限の管理](#)

共通に付与されるオブジェクト権限の使用方法

共通オブジェクトのオブジェクト権限は、オブジェクト自体とそのオブジェクト上の関連するすべてのリンクに適用されます。

このリンクには、すべてのメタデータ・リンクおよびデータ・リンク(旧称オブジェクト・リンク)のほか、ルートやコンテナに属するすべてのPDB(将来のPDBを含む)内で特定の要件を満たしたときに関連付けが行われる拡張データ・リンクが含まれます。

この要件を次に示します。

- オブジェクト権限付与者が共通ユーザーで、権限受領者が共通ユーザー、共通ロールまたはPUBLICロールです。
- オブジェクト権限付与者は、権限に対して共通に付与されるGRANT OPTIONを所有しています
- GRANT文には、CONTAINER=ALL句が含まれています。

次の例は、共通ユーザーc##hr_adminにオブジェクト権限を付与して、CDBルートまたは関連付けられているアクセス可能なPDBのいずれかでDBA_PDBビューから選択できるようにする方法を示しています。

```
CONNECT SYSTEM
Enter password: password
Connected.

GRANT SELECT ON DBA_OBJECTS TO c##hr_admin
CONTAINER=ALL;
```

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)
- [CDBのデータ・ディクショナリ・アーキテクチャ](#)
- [CDBのネームスペース](#)
- [マルチテナント環境でPUBLICロールが機能するしくみ](#)

親トピック: [共通およびローカルに付与される権限の管理](#)

PDBへのアクセス権限の付与または取消し

マルチテナント環境では、PDBアクセスに対する権限の付与および取消を実行できます。

マルチテナント環境で権限を付与するには:

- GRANTまたはREVOKE文にCONTAINER句を含めます。

CONTAINERをALLに設定すると、既存および将来のすべてのコンテナに権限が適用され、CURRENTに設定すると、ローカル・コンテナのみに権限が適用されます。CONTAINER句を省略すると、ローカル・コンテナに権限が適用されます。ルートからGRANT文を発行してCONTAINER句を省略すると、権限がローカルに適用されます。

関連トピック

- [Oracle Database SQL言語リファレンス](#)

親トピック: [共通およびローカルに付与される権限の管理](#)

例: マルチテナント環境での権限の付与

マルチテナント環境で権限を付与するには、GRANT文を使用できます。

[例18-1](#)は、既存および将来のすべてのコンテナでこの権限を使用できるように、共通ユーザーc##hr_adminにCREATE TABLE権限を共通に付与する方法を示しています。

例18-1 マルチテナント環境での権限の付与

```
CONNECT SYSTEM
Enter password: password
Connected.
```

```
GRANT CREATE TABLE TO c##hr_admin CONTAINER=ALL;
```

親トピック: [共通およびローカルに付与される権限の管理](#)

共通ユーザーによるCONTAINER_DATAオブジェクトの情報の表示

共通ユーザーは、ルート内のCONTAINER_DATAオブジェクトや特定のPDB内のデータに関する情報を表示できます。

- [ルートに接続中のルート、CDBおよびPDBに関するデータの表示](#)
共通ユーザーが問合せを実行した場合に、X\$表ならびにV\$、GV\$およびCDB_*ビューの表示情報を制限できます。
- [特定のPDBのデータを問い合わせる共通ユーザーの有効化](#)
特定のPDBに関するデータへのアクセスを共通ユーザーに許可するには、ユーザーのCONTAINER_DATA属性を調整します。

親トピック: [共通およびローカルに付与される権限の管理](#)

ルートに接続中のルート、CDBおよびPDBに関するデータの表示

共通ユーザーが問合せを実行した場合に、X\$表ならびにV\$、GV\$およびCDB_*ビューの表示情報を制限できます。

X\$表およびこれらのビューには、アプリケーション・ルートおよび関連付けられたアプリケーションPDBに関する情報(CDBルートに接続している場合は、CDB全体に関する情報)が含まれます。

この情報の制限は、他のPDBに関する機密情報を公開しない場合に役立ちます。この機能を有効にするために、Oracle Databaseでは、これらの表およびビューをコンテナ・データ・オブジェクトとして提供します。特定の表またはビューがコンテナ・データ・オブジェクトかどうかを確認するには、USER_|DBA_|ALL_VIEWS|TABLESディクショナリ・ビューのTABLE_NAME、VIEW_NAMEおよびCONTAINER_DATA 列を問い合わせます。

デフォルト(ユーザー・レベル)およびオブジェクト固有のCONTAINER_DATA属性に関する情報を検索するには:

1. SQL*PlusまたはSQL Developerで、rootとしてログインします。
2. CDB_CONTAINER_DATAデータ・ディクショナリ・ビューを問い合わせます。

次に例を示します。

```
COLUMN USERNAME FORMAT A13
COLUMN DEFAULT_ATTR FORMAT A7
COLUMN OWNER FORMAT A11
COLUMN OBJECT_NAME FORMAT A11
COLUMN ALL_CONTAINERS FORMAT A3
COLUMN CONTAINER_NAME FORMAT A10
COLUMN CON_ID FORMAT A6

SELECT USERNAME, DEFAULT_ATTR, OWNER, OBJECT_NAME,
       ALL_CONTAINERS, CONTAINER_NAME, CON_ID
FROM   CDB_CONTAINER_DATA
ORDER BY OBJECT_NAME;
```

USERNAME	DEFAULT_ATTR	OWNER	OBJECT_NAME	ALL_CONTAINERS	CONTAINER_NAME	CON_ID
C##HR_ADMIN	N	SYS	V\$SESSION	N	CDB\$ROOT	1
C##HR_ADMIN	N	SYS	V\$SESSION	N	SALESPDB	1

C##HR_ADMIN	Y	N	HRPDB	1
C##HR_ADMIN	Y	N	CDB\$ROOT	1
DBSNMP	Y	Y		1
SYSTEM	Y	Y		1

関連トピック

- [CDBのコンテナ・データ・オブジェクト](#)
- [『Oracle Databaseリファレンス』](#)

親トピック: [共通ユーザーによるCONTAINER_DATAオブジェクトの情報の表示](#)

特定のPDBのデータを問い合わせる共通ユーザーの有効化

特定のPDBに関するデータへのアクセスを共通ユーザーに許可するには、ユーザーのCONTAINER_DATA属性を調整します。

共通ユーザーによる特定のPDBのデータのアクセスを有効化するには:

- ルートでALTER USER文を発行します。

例18-2 CONTAINER_DATA属性の設定

この例は、ALTER USER文を発行して、共通ユーザーc##hr_adminがV\$SESSIONビュー(このユーザーがこのビューを問い合わせることができるものと仮定します)のCDB\$ROOT、SALES_PDBおよびHRPDBコンテナに関する情報を表示できるようにする方法を示しています。

```
CONNECT SYSTEM
Enter password: password
Connected.

ALTER USER c##hr_admin
SET CONTAINER_DATA = (CDB$ROOT, SALESPDB, HRPDB)
FOR V$SESSION CONTAINER=CURRENT;
```

詳細は、次のとおりです。

- SET CONTAINER_DATAは、コンテナのほか、ユーザーがアクセスできる対象に関するデータをリストします。
- FOR V\$SESSIONは、共通ユーザーc##hr_adminが問い合わせるCONTAINER_DATA動的ビューを指定します。
- ルートに接続する場合にCONTAINER=ALLがALTER USER文のデフォルトのため、CONTAINER = CURRENTを指定する必要がありますが、CONTAINER_DATA属性の変更はルートに制限する必要があります。

ユーザーc##hr_adminが自身がアクセス可能なすべてのCONTAINER_DATAオブジェクト内のCDB\$ROOT、SALES_PDB、HRPDBコンテナに関連する情報を表示できるようにするには、FOR V\$SESSIONを省略します。次に例を示します。

```
ALTER USER c##hr_admin
SET CONTAINER_DATA = (CDB$ROOT, SALESPDB, HRPDB)
CONTAINER=CURRENT;
```

関連トピック

- [Oracle Database SQL言語リファレンス](#)

親トピック: [共通ユーザーによるCONTAINER_DATAオブジェクトの情報の表示](#)

共通ロールおよびローカル・ロールの管理

共通ロールはルートで作成されるロールであり、ローカル・ロールはPDBで作成されます。

- [共通ロールおよびローカル・ロールの管理について](#)
マルチテナント環境では、データベース・ロールをPDBに固有にすることも、システム・コンテナまたはアプリケーション・コンテナ全体で使用することもできます。
- [共通ロールの使用方法](#)
共通ロールは、ルートのほか、マルチテナント環境でそれらのロールが定義されているコンテナの各PDBで表示できます。
- [マルチテナント環境でPUBLICロールが機能するしくみ](#)
OracleによってPUBLICロールに付与されるすべての権限はローカルに付与されます。
- [共通ロールの作成、変更または削除に必要な権限](#)
共通に付与されるCREATE ROLE、ALTER ROLEおよびDROP ROLE権限を持つ共通ユーザーのみが、共通ロールの作成、変更または削除ができます。
- [共通ロールの作成の規則](#)
共通ロールを作成する場合は、特別な規則に従う必要があります。
- [共通ロールの作成](#)
CREATE ROLE文を使用して、共通ロールを作成できます。
- [ローカル・ロールの作成の規則](#)
ローカル・ロールを作成するには、特別な規則に従う必要があります。
- [ローカル・ロールの作成](#)
CREATE ROLE文を使用して、ロールを作成できます。
- [共通ユーザーとローカル・ユーザーに対するロールの付与と取消し](#)
ロールの付与と取消しは、共通ユーザーまたはローカル・ユーザーのアクセス範囲にのみ適用されます。

親トピック: [マルチテナント環境のセキュリティの管理](#)

共通ロールおよびローカル・ロールの管理について

マルチテナント環境では、データベース・ロールをPDBに固有にすることも、システム・コンテナまたはアプリケーション・コンテナ全体で使用することもできます。

共通ロールとは、IDと(オプションの)パスワードがコンテナのルートで作成され、ルートのほか、そのコンテナに属する既存および将来のすべてのPDBで認識されるロールです。

ローカル・ロールは、1つのPDBにのみ存在し、このPDB内でのみ使用できます。共通に付与される権限は持ちません。

次の点に注意してください。

- 共通ユーザーは、共通ロールを作成して、他の共通ユーザーおよびローカル・ユーザーに付与できます。
- ロール(ローカルまたは共通)は、ローカル・ユーザーまたはロールに対してローカルに付与できます。
- 共通ロールをローカルに付与する場合、その共通ロールの権限は、ロールが付与されるコンテナ内にのみ適用されます。
- ローカル・ユーザーは共通ロールを作成できませんが、共通ユーザーおよび他のローカル・ユーザーに共通ロールを付与できます。
- 共通ロールをCDBルートまたはアプリケーション・ルートで作成する場合、CONTAINER = ALL句がデフォルトです。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [共通ロールおよびローカル・ロールの管理](#)

共通ロールの使用方法

共通ロールは、ルートのほか、マルチテナント環境でそれらのロールが定義されているコンテナの各PDBで表示できます。

次の場合、権限は共通ロールに対して共通に付与されます。

- 付与者は、共通ユーザーである。
- 付与者は、付与される権限に対して、共通に付与されるADMIN OPTIONを所有している。
- GRANT文には、CONTAINER=ALL句が含まれています。

共通ロールがローカルに付与された権限を含む場合、これらの権限は、共通ロールに付与されたPDB内にも適用されます。ローカル・ロールは共通に付与できません。

たとえば、CDB共通ユーザーc##hr_mgrに、DBAロールが共通付与されているとします。これは、ユーザーc##hr_mgrは、マルチテナント環境のルートおよび各PDBでDBAロールに関連付けられている権限を使用できることを意味します。一方、CDB共通ユーザーc##hr_mgrに、hr_pdb PDBに対するDBAロールがローカルで付与されているのみであれば、このユーザーは、hr_pdb PDBでのみDBAロールの権限を使用できます。

親トピック: [共通ロールおよびローカル・ロールの管理](#)

マルチテナント環境でPUBLICロールが機能するしくみ

OracleによってPUBLICロールに付与されるすべての権限はローカルに付与されます。

この機能により、各PDBで個別にPUBLICロールに付与された権限およびロールを必要に応じて取り消すことができます。権限をPUBLICロールに付与する必要がある場合は、ローカルに付与します。PUBLICには権限を共通に付与しないでください。

関連トピック

- [共通およびローカルに付与される権限について](#)

親トピック: [共通ロールおよびローカル・ロールの管理](#)

共通ロールの作成、変更または削除に必要な権限

共通に付与されるCREATE ROLE、ALTER ROLEおよびDROP ROLE権限を持つ共通ユーザーのみが、共通ロールの作成、変更または削除ができます。

共通ユーザーはローカル・ロールも作成できますが、作成されたPDBでのみそれらのロールを使用できます。

親トピック: [共通ロールおよびローカル・ロールの管理](#)

共通ロールの作成の規則

共通ロールを作成する場合は、特別な規則に従う必要があります。

この規則は次のとおりです。

- 正しいルートにいることを確認します。共通ロールを作成するには、正しいルート(CDBルートまたはアプリケーション・ルート)にいる必要があります。PDBから共通ロールを作成することはできません。正しいルートにいることを確認するには、次のいずれかを実行します。
 - CDBルートにいることを確認するには、show_con_nameコマンドを発行します。CDB\$ROOTと表示される必要があります。
 - アプリケーション・ルートにいることを確認するには、次の問合せにYESが戻されることを確認します。

```
SELECT APPLICATION_ROOT FROM V$PDBS WHERE CON_ID=SYS_CONTEXT('USERENV', 'CON_ID');
```

- 共通ロールに付ける名前がCOMMON_USER_PREFIXパラメータの値(デフォルトではC##)で始まるようにします。この要件はDBAやRESOURCEなど、Oracle Databaseによって提供される既存のロールの名前に適用されないことに注意してください。
- オプションで、CONTAINER句をALLに設定します。ルートにいるかぎり、CONTAINER = ALL句を省略しても、ロールは、デフォルトでCDBルートまたはアプリケーション・ルートの共通ロールとして作成されます。

親トピック: [共通ロールおよびローカル・ロールの管理](#)

共通ロールの作成

CREATE ROLE文を使用して、共通ロールを作成できます。

1. 共通ロールを作成するCDBまたはアプリケーション・コンテナのルートに接続します。

次に例を示します。

```
CONNECT SYSTEM
Enter password: password
Connected.
```

2. CONTAINER句をALLに設定してCREATE ROLE文を実行します。

次に例を示します。

```
CREATE ROLE c##sec_admin IDENTIFIED BY password CONTAINER=ALL;
```

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [共通ロールおよびローカル・ロールの管理](#)

ローカル・ロールの作成の規則

ローカル・ロールを作成するには、次の特別な規則に従う必要があります。

これらの規則は次のとおりです。

- ロールを作成するPDBに接続する必要があり、CREATE ROLE権限がある必要があります。
- ローカル・ロールに付ける名前をCOMMON_USER_PREFIXパラメータの値(デフォルトではC##)で始めることはできません。
- CREATE ROLE文にCONTAINER=CURRENTを含め、ローカル・ロールとしてロールを指定できます。PDBに接続しており、この句を省略すると、CONTAINER=CURRENT句が含まれます。
- 共通ロールとローカル・ロールの名前を同じにすることはできません。ただし、異なるPDBのローカル・ロールに同じ名前を使用できます。既存のロールの名前を検索するには、CDB_ROLESおよびDBA_ROLESデータ・ディクショナリ・ビューを問い合わせます。

親トピック: [共通ロールおよびローカル・ロールの管理](#)

ローカル・ロールの作成

CREATE ROLE文を使用して、ロールを作成できます。

1. ローカル・ロールを作成するPDBに接続します。

次に例を示します。

```
CONNECT SYSTEM@hrpdb
Enter password: password
Connected.
```

2. CONTAINER句をCURRENTに設定してCREATE ROLE文を実行します。

次に例を示します。

```
CREATE ROLE sec_admin CONTAINER=CURRENT;
```

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [共通ロールおよびローカル・ロールの管理](#)

共通ユーザーとローカル・ユーザーに対するロールの付与と取消し

ロールの付与と取消は、共通ユーザーまたはローカル・ユーザーのアクセス範囲にのみ適用されます。

共通ユーザーは、他の共通ユーザーへの共通ロールの付与および取消しを行うことができます。ローカル・ユーザーは、共通ロールを共通ユーザーを含むPDBのユーザーに付与できますが、これは、PDB内のみで適用されます。

次の例は、共通ユーザーc##sec_adminへのすべてのコンテナで使用するAUDIT_ADMIN共通ロールの付与方法を示しています。

```
CONNECT SYSTEM
Enter password: password
Connected.

GRANT AUDIT_ADMIN TO c##sec_admin CONTAINER=ALL;
```

同様に、次の例は、ローカル・ユーザーaud_adminによる共通ユーザーc##sec_adminへのhrpdb PDB内で使用するAUDIT_ADMIN共通ロールの付与方法を示しています。

```
CONNECT aud_admin@hrpdb
Enter password: password
Connected.

GRANT AUDIT_ADMIN TO c##sec_admin CONTAINER=CURRENT;
```

この例は、ローカル・ユーザーaud_adminがPDBの別のユーザーからロールを取り消す方法を示しています。CONTAINER句を省略すると、CURRENTが暗黙のうちに入れられます。

```
CONNECT aud_admin@hrpdb
Enter password: password
Connected.

REVOKE sec_admin FROM psmith CONTAINER=CURRENT;
```

親トピック: [共通ロールおよびローカル・ロールの管理](#)

PDBロックダウン・プロファイルを使用したPDBでの操作の制限

マルチテナント環境でPDBロックダウン・プロファイルを使用して、プラガブル・データベース(PDB)の一連のユーザー操作を制限できます。

- [PDBロックダウン・プロファイルについて](#)

PDBロックダウン・プロファイルは、操作グループを制御する名前付きの機能セットです。

- [デフォルトのPDBロックダウン・プロファイル](#)

Oracle Databaseには、サイトの要件にあわせてカスタマイズが可能なデフォルトのPDBロックダウン・プロファイルのセットが用意されています。

- [PDBロックダウン・プロファイルの作成](#)

PDBロックダウン・プロファイルを作成するには、CREATE LOCKDOWN PROFILEシステム権限が必要です。

- [PDBロックダウン・プロファイルの有効化または無効化](#)

PDBロックダウン・プロファイルを有効または無効にするには、PDB_LOCKDOWN初期化パラメータを使用します。

- [PDBロックダウン・プロファイルの削除](#)

PDBロックダウン・プロファイルを削除するには、DROP LOCKDOWN PROFILEシステム権限があり、CDBまたはアプリケーション・ルートにログインすることが必要です。

親トピック: [マルチテナント環境のセキュリティの管理](#)

PDBロックダウン・プロファイルについて

PDBロックダウン・プロファイルは、操作グループを制御する名前付きの機能セットです。

場合によっては、操作を個別に有効または無効にできます。たとえば、PDBロックダウン・プロファイルに、ALTER SYSTEM文で使用する特定の句を無効にする設定を含めることができます。

PDBロックダウン・プロファイルは、ユーザーに対して定義されるリソース制限と同様、機能が提供する機能性へのユーザー・アクセスを制限します。名前が示すように、CDBでは、アプリケーション・コンテナ、PDBまたはアプリケーションPDBに対してPDBロックダウン・プロファイルを使用します。カスタム・プロファイルを作成して、サイトの要件に対応することができます。PDBプロファイルを使用すると、アプリケーションのカスタム・セキュリティ・ポリシーを定義できます。さらに、**基本プロファイル**と呼ばれる別のプロファイルに基づいて、ロックダウン・プロファイルを作成できます。このプロファイルは、基本プロファイルが変更されたときに動的に更新されるように構成するか、または基本プロファイルが更新されたときに静的になる(変更しない)ように構成できます。ロックダウン・プロファイルは、Oracle Cloudとオンプレミスの両方の環境用に設計されています。

IDがPDB間で共有される場合、昇格する権限が存在することがあります。ロックダウン・プロファイルを使用すると、この権限の昇格を防ぐことができます。IDは、次の状況で共有できます。

- オペレーティング・システム・レベルでは、データベースがファイルやプロセスなどのオペレーティング・システム・リソースと対話するとき
- ネットワーク・レベルでは、データベースが他のシステムと通信し、ネットワークIDが重要なとき
- データベースの内部では、PDBが共通オブジェクトへのアクセスまたは作成を行うとき、またはデータベース・リンクなどの機能を使用してコンテナ境界を越えて通信するとき

共有IDを使用し、PDBロックダウン・プロファイルのメリットを受ける機能は、次のカテゴリに分かれます。

- ネットワーク・アクセス機能。これはネットワークを使用してPDB外部と通信する操作です。たとえば、PL/SQLパッケージUTL_TCP、UTL_HTTP、UTL_MAIL、UTL_SNMP、UTL_INADDRおよびDBMS_DEBUG_JDWPは、この種の操作を実行します。現在、ネットワークIDを共有してこの種のアクセスを制御するためにACLが使用されています。
- 共通ユーザーまたは共通オブジェクトのアクセス。これは、PDBのローカル・ユーザーが共通ユーザー・アカウントを介して通信したり、共通スキーマのオブジェクトにアクセスする操作です。この種の操作には、共通スキーマでのオブジェクトの追加または置換、共通オブジェクトへの権限の付与、共通ディレクトリ・オブジェクトへのアクセス、共通ユーザーへのINHERIT PRIVILEGESロールの付与、および共通ユーザーに対するユーザー・プロキシの操作が含まれます。
- オペレーティング・システム・アクセス。たとえば、UTL_FILEまたはDBMS_FILE_TRANSFER PL/SQLパッケージへのアクセスを制限できます。

- 接続。たとえば、共通ユーザーによるPDBへの接続を制限したり、SYSOPER管理権限を持つローカル・ユーザーによる制限モードでオープンしたPDBへの接続を制限できます。

PDBロックダウン・プロファイルを作成する一般的な手順では、最初にCREATE LOCKDOWN PROFILE文を使用してCDBルートまたはアプリケーション・ルートにプロファイルを作成した後、ALTER LOCKDOWN PROFILE文を使用してそれに制限を加えます。

PDBロックダウン・プロファイルを有効にするには、ALTER SYSTEM文を使用して、PDB_LOCKDOWNパラメータを設定できます。既存のPDBロックダウン・プロファイルに関する情報を確認するには、CDBまたはアプリケーション・ルートに接続してDBA_LOCKDOWN_PROFILESデータ・ディクショナリ・ビューを問い合わせます。ローカル・ユーザーは、V\$LOCKDOWN_RULES動的データ・ディクショナリ・ビューを問い合わせ、PDBロックダウン・パラメータの内容を確認できます。

親トピック: [PDBロックダウン・プロファイルを使用したPDBでの操作の制限](#)

デフォルトのPDBロックダウン・プロファイル

Oracle Databaseには、サイトの要件にあわせてカスタマイズが可能なデフォルトのPDBロックダウン・プロファイルのセットが用意されています。

デフォルトでは、これらのプロファイルの大部分は空です。これらは、デプロイメント要件に応じて、構成するプレースホルダまたはテンプレートとして設計されています。

これらのプロファイルに関する詳細情報は次のとおりです。

- PRIVATE_DBAASには、プライベート・クラウドDatabase-as-a-Service (DBaaS)のデプロイメントに適した制限が組み込まれています。これらの制限は次のとおりです。
 - 各PDBのデータベース管理者が同じである必要があります
 - 異なるユーザーによるデータベースへの接続が許可されます
 - 異なるアプリケーションが許可されます

PRIVATE_DBAASでは、ユーザーによるPDBへの接続が許可されますが、Oracle Databaseの管理機能は使用できません。

- SAASには、Software-as-a-Service(SaaS)デプロイメントに適した制限が組み込まれています。これらの制限は次のとおりです。
 - 各PDBのデータベース管理者が同じである必要があります
 - 異なるユーザーによるデータベースへの接続が許可されます
 - 同じアプリケーションを使用する必要があります

SAASロックダウン・プロファイルにはPRIVATE_DBAASプロファイルよりも多くの制限があります。ユーザーは異なってもかまいませんが、アプリケーション・コードは同じです。ユーザーは、直接接続できずアプリケーションを介してのみ接続する必要があります。ユーザーは、すべての管理機能を実行する権限を与えられていません。

- PUBLIC_DBAASには、パブリック・クラウドDatabase-as-a-Service (DBaaS)のデプロイメントに適した制限が組み込まれています。制限事項は次のとおりです。
 - 各PDBに異なるDBA
 - 異なるユーザー
 - 異なるアプリケーション

PUBLIC_DBAASロックダウン・プロファイルは、最も制限が厳しいロックダウン・プロファイルです。

PDBロックダウン・プロファイルの作成

PDBロックダウン・プロファイルを作成するには、CREATE LOCKDOWN PROFILEシステム権限が必要です。

作成したロックダウン・プロファイルに制限を加えた後、有効にできます。

1. CREATE LOCKDOWN PROFILEシステム権限を持つユーザーとしてCDBルートまたはアプリケーション・ルートに接続します。たとえば、次のようにCDBルートに接続します。

```
CONNECT c##sec_admin
Enter password: password
```

2. 次の構文を使用してCREATE LOCKDOWN PROFILE文を実行し、プロファイルを作成します。

```
CREATE LOCKDOWN PROFILE profile_name
[FROM static_base_profile | INCLUDING dynamic_base_profile];
```

詳細は、次のとおりです。

- profile_nameはロックダウン・プロファイルを割り当てる名前です。DBA_LOCKDOWN_PROFILESデータ・ディクショナリ・ビューのPROFILE_NAMES列を問い合せて、既存の名前を検索できます。
- FROM static_base_profileは、既存のプロファイルの値を使用して、新しいロックダウン・プロファイルを作成します。基本プロファイルへの以降の変更は、新しいプロファイルには影響しません。
- INCLUDING dynamic_base_profileも既存の基本プロファイルの値を使用して新しいロックダウン・プロファイルを作成しますが、この新しいロックダウン・プロファイルは、基本プロファイルを構成するDISABLE STATEMENTルール、および基本プロファイルへの以降の変更を継承するという点が異なります。新規プロファイルに明示的に追加されたルールが基本プロファイル内のルールと競合する場合は、基本プロファイル内のルールが優先されます。たとえば、基本プロファイルのOPTION_VALUE句は、新しいプロファイルのOPTION_VALUE句より優先されます。

次の2つのPDBロックダウン・プロファイル文では、継承がどのように機能するかを示しています。

```
CREATE LOCKDOWN PROFILE hr_prof INCLUDING PRIVATE_DBAAS;
CREATE LOCKDOWN PROFILE hr_prof2 FROM hr_prof;
```

最初の文では、hr_profはPRIVATE_DBAAS基本プロファイルへの変更内容を継承します。新しい文がPRIVATE_DBAASに対して有効になった後、hr_profに対して有効になります。2番目の文では、これとは対照的に、hr_profが変更された場合、hr_prof2はその基本プロファイルに依存しないため、変更されません。

3. ALTER LOCKDOWN PROFILE文を実行して、プロファイルに制限を加えます。

次に例を示します。

```
ALTER LOCKDOWN PROFILE hr_prof DISABLE STATEMENT = ('ALTER SYSTEM');
ALTER LOCKDOWN PROFILE hr_prof ENABLE STATEMENT = ('ALTER SYSTEM') clause = ('flush
shared_pool');
ALTER LOCKDOWN PROFILE hr_prof DISABLE FEATURE = ('XDB_PROTOCOLS');
```

前述の例は、次のとおりです。

- DISABLE STATEMENT = ('ALTER SYSTEM')は、PDBに対するALTER SYSTEM文の使用をすべて無効にします。
- ENABLE STATEMENT = ('ALTER SYSTEM') clause = ('flush shared_pool')は、ALTER SYSTEMの

FLUSH_SHARED_POOL句の使用のみを有効にします。

- DISABLE FEATURE = ('XDB_PROTOCOLS')は、このPDBによるXDBプロトコル(FTP、HTTP、HTTPS)の使用を禁止します

PDBロックダウン・プロファイルを作成すると、ALTER SYSTEM SET PDB_LOCKDOWN SQL文を使用してプロファイルを有効にする準備が整います。

関連トピック

- [Oracle Database SQL言語リファレンス](#)

親トピック: [PDBロックダウン・プロファイルを使用したPDBでの操作の制限](#)

PDBロックダウン・プロファイルの有効化または無効化

PDBロックダウン・プロファイルを有効または無効にするには、PDB_LOCKDOWN初期化パラメータを使用します。

ALTER SYSTEM SET PDB_LOCKDOWNを使用して、次のコンテキストでロックダウン・プロファイルを有効にできます。

- CDB (すべてのPDBに影響します)
- アプリケーション・ルート(コンテナ内のすべてのアプリケーションPDBに影響します)
- アプリケーションPDB
- PDB

ノート:



プロファイルを有効にするためにインスタンスを再起動する必要はありません。ALTER SYSTEM SET PDB_LOCKDOWN 文が完了すると、プロファイル・ルールは即時に有効になります。

CDBルートでPDB_LOCKDOWNを設定すると、PDB_LOCKDOWNがコンテナ・レベルで設定されていないかぎり、すべてのPDBおよびアプリケーション・ルートでこの設定が継承されます。ロックダウン・プロファイルを無効にするには、PDB_LOCKDOWNをnullに設定します。CDBルートでこのパラメータをnullに設定すると、ロックダウン・プロファイルは、PDB内でプロファイルを明示的に設定したものの以外のすべてのPDBに対して無効になります。

SYSDBA管理権限またはALTER SYSTEMシステム権限が共通に付与されたCDB共通ユーザーは、CDBルートで作成されたロックダウン・プロファイルにのみPDB_LOCKDOWNを設定できます。アプリケーション共通SYSDBA管理権限またはALTER SYSTEMシステム権限を持つアプリケーションの共通ユーザーは、アプリケーション・ルートで作成されたロックダウン・プロファイルにのみPDB_LOCKDOWNを設定できます。

1. 共通に付与されるALTER SYSTEMまたは共通に付与されるSYSDBA権限を持つユーザーとして目的のコンテナにログインします。
たとえば、すべてのPDBに対してプロファイルを有効にするには、CDBルートにログインします。

```
CONNECT c##sec_admin
Enter password: password
```

2. ALTER SYSTEM SET PDB_LOCKDOWN文を実行します。
たとえば、次の文は、hr_profという名前のロックダウン・プロファイルをすべてのPDBに対して有効化します。

```
ALTER SYSTEM SET PDB_LOCKDOWN = hr_prof;
```

次の文は、PDB_LOCKDOWNパラメータをリセットします。

```
ALTER SYSTEM RESET PDB_LOCKDOWN;
```

前述の文のバリエーションには、SCOPE句が含まれます。

```
ALTER SYSTEM RESET PDB_LOCKDOWN SCOPE = BOTH;
```

次の文は、PDBレベルで明示的に設定されたものを除く、CDB内のすべてのロックダウン・プロファイルを無効化します。

```
ALTER SYSTEM SET PDB_LOCKDOWN = '' SCOPE = BOTH;
```

PDBロックダウン・プロファイルの名前を確認するには、DBA_LOCKDOWN_PROFILESデータ・ディクショナリ・ビューのPROFILE_NAME列を問い合わせます。

- 必要に応じて、DBA_LOCKDOWN_PROFILESを問い合わせ、プロファイルに関する情報を確認します。
たとえば、次の問合せを実行します。

```
SET LINESIZE 150
COL PROFILE_NAME FORMAT a20
COL RULE FORMAT a20
COL CLAUSE FORMAT a25

SELECT PROFILE_NAME, RULE, CLAUSE, STATUS FROM CDB_LOCKDOWN_PROFILES;
```

出力例は次のように表示されます。

PROFILE_NAME	RULE	CLAUSE	STATUS
HR_PROF	XDB_PROTOCOLS		DISABLE
HR_PROF	ALTER SYSTEM		DISABLE
HR_PROF	ALTER SYSTEM	FLUSH SHARED_POOL	ENABLE
HR_PROF2			EMPTY
PRIVATE_DBAAS			EMPTY
PUBLIC_DBAAS			EMPTY
SAAS			EMPTY

親トピック: [PDBロックダウン・プロファイルを使用したPDBでの操作の制限](#)

PDBロックダウン・プロファイルの削除

PDBロックダウン・プロファイルを削除するには、DROP LOCKDOWN PROFILEシステム権限があり、CDBまたはアプリケーション・ルートにログインする必要があります。

DBA_LOCKDOWN_PROFILESデータ・ディクショナリ・ビューを問い合わせ、既存のPDBロックダウン・プロファイルの名前を検索できます。

- DROP LOCKDOWN PROFILEシステム権限を持つユーザーとしてCDBルートまたはアプリケーション・ルートに接続します。
たとえば、次のようにCDBルートに接続します。

```
CONNECT c##sec_admin
Enter password: password
```

- DROP LOCKDOWN_PROFILE文を実行します。
次に例を示します。

```
DROP LOCKDOWN PROFILE hr_prof2;
```

- 必要に応じて、DBA_LOCKDOWN_PROFILESを問い合わせることでプロファイルの現在のリストを確認します。

たとえば、次の問合せを実行します。

```
SET LINESIZE 150
COL PROFILE_NAME FORMAT a20
COL RULE FORMAT a20
COL CLAUSE FORMAT a25

SELECT PROFILE_NAME, RULE, CLAUSE, STATUS FROM CDB_LOCKDOWN_PROFILES;
```

出力例は次のように表示されます。

PROFILE_NAME	RULE	CLAUSE	STATUS
HR_PROF	XDB_PROTOCOLS		DISABLE
HR_PROF	ALTER SYSTEM		DISABLE
HR_PROF	ALTER SYSTEM	FLUSH SHARED_POOL	ENABLE
PRIVATE_DBAAS			EMPTY
PUBLIC_DBAAS			EMPTY
SAAS			EMPTY

親トピック: [PDBロックダウン・プロファイルを使用したPDBでの操作の制限](#)

PDBのオペレーティング・システム・ユーザーの構成

DBMS_CREDENTIAL.CREATE_CREDENTIALプロシージャは、ユーザー・アカウントを、プラグブル・データベース(PDB)のオペレーティング・システム・ユーザーになるように構成します。

- [PDBのオペレーティング・システム・ユーザーの構成について](#)
oracleオペレーティング・システム・ユーザーのかわりに、特定のユーザー・アカウントをそのPDBのオペレーティング・システム・ユーザーに設定できます。
- [PDBのオペレーティング・システム・ユーザーの構成](#)
DBMS_CREDENTIAL.CREATE_CREDENTIALプロシージャで、PDBのオペレーティング・システム・ユーザーを設定できます。
- [PDBでのデフォルトの資格証明の設定](#)
指定したPDBのデータベース・プロパティDEFAULT_CREDENTIALを設定できます。

親トピック: [マルチテナント環境のセキュリティの管理](#)

PDBのオペレーティング・システム・ユーザーの構成について

oracleオペレーティング・システム・ユーザーのかわりに、特定のユーザー・アカウントをそのPDBのオペレーティング・システム・ユーザーに設定できます。

特定のユーザーをPDBのオペレーティング・システム・ユーザーとして設定しない場合、PDBではデフォルトでoracleオペレーティング・システム・ユーザーが使用されます。ルートについては、オペレーティング・システムと対話する必要がある場合、oracleオペレーティング・システム・ユーザーを使用できます。

セキュリティ向上のため、マルチテナント環境の各PDBに対して一意のオペレーティング・システム・ユーザーを設定することをお勧めします。そうすることで、oracleオペレーティング・システム・ユーザーより権限の低いユーザーとしてオペレーティング・システムとの対話を行えるほか、PDBに属するデータを、他のPDBに接続しているユーザーのアクセスから保護することにも役立ちます。

親トピック: [PDBのオペレーティング・システム・ユーザーの構成](#)

PDBのオペレーティング・システム・ユーザーの構成

DBMS_CREDENTIAL.CREATE_CREDENTIALプロシージャで、PDBのオペレーティング・システム・ユーザーを設定できます。

1. DBMS_CREDENTIAL PL/SQLパッケージに対するEXECUTE権限およびALTER SYSTEMシステム権限を持つユーザーとして、データベース・インスタンスのルートにログインします。

次に例を示します。

```
sqlplus c##sec_admin
Enter password: password
```

2. DBMS_CREDENTIAL.CREATE_CREDENTIALプロシージャを実行して、オペレーティング・システム・ユーザーのOracle資格証明を作成します。

たとえば、os_adminという名前のユーザーの資格証明を設定するには、次のようにします。

```
BEGIN
DBMS_CREDENTIAL.CREATE_CREDENTIAL (
  credential_name => 'PDB1_OS_USER',
  username        => 'os_admin',
  password        => 'password');
END;
/
```

3. オペレーティング・システム・ユーザーが使用されるPDBに接続します。

次に例を示します。

```
CONNECT cc##sec_admin@hrpdb
Enter password: password
```

使用可能なPDBを検索するには、show pdbsコマンドを実行します。現在のPDBを確認するには、show con_nameコマンドを実行します。

4. ステップ2で資格証明を設定したユーザーに対して、PDB_OS_CREDENTIAL初期化パラメータを設定します。

次に例を示します。

```
ALTER SYSTEM SET PDB_OS_CREDENTIAL = PDB1_OS_USER SCOPE = SPFILE;
```

PDB_OS_CREDENTIALパラメータは静的なパラメータであるため、SCOPE = SPFILE句を使用して設定する必要があります。

5. データベース・インスタンスを再起動します。

```
SHUTDOWN IMMEDIATE
STARTUP
```

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)
- [『Oracle Databaseリファレンス』](#)
- [Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)
- [PDBでのデフォルトの資格証明の設定](#)

親トピック: [PDBのオペレーティング・システム・ユーザーの構成](#)

PDBでのデフォルトの資格証明の設定

指定したPDBのデータベース・プロパティDEFAULT_CREDENTIALを設定できます。

デフォルトの資格証明は、オブジェクトをオブジェクト・ストアからPDBにインポートする場合に便利です。impdpの使用時に資格証明名を指定しない場合、Oracle Data Pumpとオブジェクト・ストア・モジュールは、DEFAULT_CREDENTIALオブジェクトを使

用してユーザー名とパスワードを取得できます。資格証明を指定しないで impdp を実行する場合は、ダンプ・ファイル名の前に DEFAULT_CREDENTIAL: を付ける必要があります。

デフォルトの資格証明を設定するには:

1. 管理者権限で PDB にログインします。
2. ALTER DATABASE 文を使用して、デフォルトの資格証明を設定します。
たとえば、次の文を入力して、資格証明を SYSTEM. HR_CRED に設定します。

```
ALTER DATABASE PROPERTY SET DEFAULT_CREDENTIAL = 'SYSTEM. HR_CRED';
```

例18-3 デフォルトの資格証明を使用したデータの PDB へのインポート

この例では、デフォルトの資格証明が存在することを前提としています。次のコマンドはオブジェクト・ストアからデータをインポートし、文字列 DEFAULT_CREDENTIAL を使用して URL を事前に設定します。

```
impdp hr@pdb1 table_exists_action=replace ¥  
dumpfile=DEFAULT_CREDENTIAL:https://example.com/ostore/obucket/myt.dmp
```

関連項目:

- [「データ・ポンプを使用した CDB へのデータの移動」](#)
- データ・ポンプ・インポートについて学習するには、[『Oracle Database ユーティリティ』](#)を参照

親トピック: [PDB のオペレーティング・システム・ユーザーの構成](#)

マルチテナント環境でのアプリケーション・コンテキストの使用

アプリケーション・コンテキストにはユーザー ID が格納されており、これに基づいてデータベースのデータにユーザーがアクセスできるかを否かを決定できます。

- [アプリケーション・コンテキストとは](#)
アプリケーション・コンテキストとは、Oracle Database がメモリーに格納する名前と値のペアです。
- [マルチテナント環境でのアプリケーション・コンテキスト](#)
マルチテナント環境のどこでアプリケーションを作成するかによって、アプリケーション・コンテキストを作成する場所が決まります。

親トピック: [マルチテナント環境のセキュリティの管理](#)

アプリケーション・コンテキストとは

アプリケーション・コンテキストとは、Oracle Database がメモリーに格納する名前と値のペアです。

コンテキストには、**ネームスペース**と呼ばれるラベル(たとえば、従業員 ID を取得するアプリケーション・コンテキストは empno_ctx)があります。このコンテキストにより、Oracle Database は認証中にデータベース・ユーザーと非データベース・ユーザーに関する情報を入手できます。

コンテキスト内は名前と値のペア(結合配列)です。名前は、値を保持するメモリー内の場所を指定します。アプリケーションはアプリケーション・コンテキストを使用して、ユーザーに関するセッション情報(ユーザー ID または他のユーザー固有の情報など)またはクライアント ID にアクセスして、その情報をデータベースに安全に引き渡すことができます。

この情報を使用して、ユーザーがアプリケーションを通じてデータにアクセスできるようにしたりアクセスできないようにできます。アプリケーション・コンテキストを使用して、データベース・ユーザーと非データベース・ユーザーの両方を認証できます。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [マルチテナント環境でのアプリケーション・コンテキストの使用](#)

マルチテナント環境でのアプリケーション・コンテキスト

マルチテナント環境のどこでアプリケーションを作成するかによって、アプリケーション・コンテキストを作成する場所が決まります。

アプリケーション・ルートまたはCDBルートにアプリケーションがインストールされると、そのアプリケーションはアプリケーション・コンテナまたはシステム・コンテナおよび関連付けられたアプリケーションPDBからアクセスできるようになります。そのルートで共通アプリケーション・コンテキストを作成する必要があります。

アプリケーション・コンテナで使用する共通アプリケーション・コンテキストを作成する場合は、次の点に注意してください。

- マルチテナント環境でアプリケーション・コンテキストを作成するには、CREATE CONTEXT SQL文でCONTAINER句を設定します。たとえば、アプリケーション・ルートで共通アプリケーション・コンテキストを作成するには、CONTAINERをALLに設定してCREATE CONTEXTを実行する必要があります。PDBでアプリケーション・コンテキストを作成するには、CONTAINERをCURRENTに設定します。
- ローカル・アプリケーション・コンテキストと共通アプリケーション・コンテキストに同じ名前を使用することはできません。既存のアプリケーション・コンテキストの名前は、次の問合せを実行して検索できます。

```
SELECT OBJECT_NAME FROM DBA_OBJECTS WHERE OBJECT_TYPE = 'CONTEXT' ;
```

- 共通アプリケーション・コンテキストを管理するために作成するPL/SQLパッケージは、共通PL/SQLパッケージであることが必要です。つまり、それがアプリケーション・ルートまたはCDBルートに存在する必要があります。特定のPDBのアプリケーション・コンテキストを作成する場合、関連付けられたPL/SQLパッケージをそのPDBに格納する必要があります。
- 共通アプリケーション・コンテキストのアプリケーション・コンテナまたはシステム・コンテナから共通セッション・アプリケーション・コンテキストの下に設定した名前と値のペアは、共通ユーザーが異なるコンテナにアクセスする場合、他のアプリケーション・コンテナまたはシステム・コンテナからアクセスできません。
- アプリケーション・コンテナまたはシステム・コンテナから共通グローバル・アプリケーション・コンテキストの下に設定した名前と値のペアは、同じユーザー・セッションの同じコンテナ内の場合にかぎりアクセスできます。
- アプリケーションは、アプリケーション・ルート、CDBルートまたはPDBのいずれに存在していても、アプリケーション・コンテキストの値を取得できます。
- CDBまたはアプリケーション・コンテナへのPDBの接続操作中、共通アプリケーション・コンテキストの名前がPDBのローカル・アプリケーション・コンテキストの名前と競合する場合は、PDBを制限モードでオープンする必要があります。データベース管理者は、PDBを通常モードでオープンする前に、競合を修正する必要があります。
- 切断操作中、共通アプリケーション・コンテキストはその共通セマンティクスを維持することで、後にそのPDBが同じ名前前の共通アプリケーション・コンテキストがある別のCDBに接続する場合、引き続き共通オブジェクトのように動作します。PDBが同じ共通アプリケーション・コンテキストが存在しないアプリケーション・コンテナまたはシステム・コンテナに接続する場合、そのPDBはローカル・オブジェクトのように動作します。

アプリケーション・コンテキストがローカル・アプリケーション・コンテキストであるかアプリケーションの共通アプリケーション・コンテキストであるかを確認するには、DBA_CONTEXTまたはALL_CONTEXTデータ・ディクショナリ・ビューのSCOPE列を問い合わせます。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [マルチテナント環境でのアプリケーション・コンテキストの使用](#)

マルチテナント環境でのOracle Virtual Private Databaseの使用

Oracle Virtual Private Database (VPD)を使用すると、データにアクセスするユーザーをフィルタ処理できます。

- [Oracle Virtual Private Database](#)
Oracle Virtual Private Database (VPD)で、行および列レベルでデータベース・アクセスを制御するセキュリティ・ポリシーを作成します。
- [マルチテナント環境でのOracle Virtual Private Database](#)
アプリケーション・ルートで仮想プライベート・データベース・ポリシーを作成して、関連付けられたすべてのアプリケーション PDBで使用できます。

親トピック: [マルチテナント環境のセキュリティの管理](#)

Oracle Virtual Private Database

Oracle Virtual Private Database (VPD)で、行および列レベルでデータベース・アクセスを制御するセキュリティ・ポリシーを作成します。

ノート:



Oracle Database リリース 12c では、VPD にかわって Real Application Security (RAS)が採用されました。アプリケーションで行レベルおよび列レベルのアクセス制御が必要な新規プロジェクトには RAS を使用することをお勧めします。

基本的には、Oracle Virtual Private Databaseのセキュリティ・ポリシーが適用された表、ビューまたはシノニムに対して発行されるSQL文に、動的なWHERE句が追加されます。

Oracle Virtual Private Databaseを使用すると、データベース表、ビューまたはシノニムに対するセキュリティを直接詳細なレベルまで規定できます。これらのデータベース・オブジェクトにセキュリティ・ポリシーを直接付加すると、ユーザーがデータにアクセスするたびにポリシーが自動的に適用されるため、セキュリティを回避できません。

ユーザーがOracle Virtual Private Databaseポリシーで保護されている表、ビューまたはシノニムに直接的または間接的にアクセスすると、Oracle DatabaseはユーザーのSQL文を動的に変更します。この変更は、セキュリティ・ポリシーを実装する関数によって戻されたWHERE条件(述語)に基づいて行われます。Oracle Databaseでは、関数内に記述された条件、または関数が戻す条件を使用して、動的かつユーザーに対して透過的に文が変更されます。Oracle Virtual Private Databaseポリシーは、SELECT、INSERT、UPDATE、INDEXおよびDELETE文に適用できます。

たとえば、ユーザーが次の問合せを実行するとします。

```
SELECT * FROM OE. ORDERS;
```

Oracle Virtual Private Databaseポリシーにより、WHERE句の文が動的に追加されます。次に例を示します。

```
SELECT * FROM OE. ORDERS  
WHERE SALES_REP_ID = 159;
```

この例では、ユーザーは営業担当者159の受注のみを表示できます。

このユーザーのセッション情報(ユーザーのIDなど)に基づいてユーザーをフィルタ処理する場合は、WHERE句を作成してアプリケーション・コンテキストを使用できます。次に例を示します。

```
SELECT * FROM OE. ORDERS
WHERE SALES_REP_ID = SYS_CONTEXT('USERENV', 'SESSION_USER');
```



ノート:

Oracle Virtual Private Database では、DDL(TRUNCATE 文や ALTER TABLE 文)のフィルタ処理はサポートされていません。

関連トピック

- [『Oracle Database Real Application Security管理者および開発者ガイド』](#)
- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [マルチテナント環境でのOracle Virtual Private Databaseの使用](#)

マルチテナント環境でのOracle Virtual Private Database

アプリケーション・ルートで仮想プライベート・データベース・ポリシーを作成して、関連付けられたすべてのアプリケーションPDBで使用できます。

CDBの制限は、共有の状況依存ポリシーのほか、仮想プライベート・データベース・ポリシー関連のビューにも適用されます。マルチテナント環境全体に仮想プライベート・データベース・ポリシーを作成することはできません。

アプリケーション・コンテナについては、仮想プライベート・データベース・ポリシーを作成して、アプリケーション・ルートに属するすべてのPDBに共通ポリシーを適用することで、アプリケーション共通オブジェクトを保護できます。つまり、アプリケーション・ルートにアプリケーションをインストールすると、共通オブジェクトを保護するすべての共通仮想プライベート・データベース・ポリシーは、アプリケーション・コンテナのすべてのPDBに適用され、即座に実施されます。

次の点に注意してください。

- アプリケーション・ルートでは共通仮想プライベート・データベース・ポリシーおよびそれに関連するPL/SQLファンクションのみを作成し、それをアプリケーション共通オブジェクトに付加することができます。ファンクションがポリシーと同じ場所にない場合、実行時にエラーが発生します。
- 共通オブジェクトに適用される仮想プライベート・データベース・ポリシーは、アプリケーションPDBからアプリケーション共通オブジェクトにアクセスする場合、アプリケーション・コンテナに属するPDBで自動的に強制される共通ポリシーと見なされます。
- アプリケーション共通仮想プライベート・データベース・ポリシーは、アプリケーション共通オブジェクトのみを保護できます。
- アプリケーション・ルートのアプリケーション共通オブジェクトに適用され、すべてのアプリケーションPDBに適用される仮想プライベート・データベース・ポリシーは、共通仮想プライベート・データベース・ポリシーと見なされます。ローカル・データベース表に適用され、1つのPDBで実施されるポリシーは、ローカル仮想プライベート・データベース・ポリシーと見なされます。

たとえば、ポリシーVPD_P1がアプリケーション・ルートのアプリケーション共通表T1に適用される場合は、共通ポリシーと見なされます。このポリシーは各アプリケーションPDBに強制されます。VPD_P1というポリシーがPDB1のT1というローカル表に適用される場合は、ローカル・ポリシーと見なされ、PDB1のみがその影響を受けます。VPD_P1というポリシーがアプリケー

ション・ルートのT1というローカル表に適用される場合も、アプリケーション・ルートのみが影響を受けるため、ローカル・ポリシーと見なされます。この概念は、仮想プライベート・データベース・ポリシーの有効化、無効化および削除などの他の操作にも適用されます。

- アプリケーション共通仮想プライベート・データベース・ポリシーはアプリケーション共通オブジェクトのみを保護し、ローカル仮想プライベート・データベース・ポリシーはローカル・オブジェクトのみを保護します。
- アプリケーション・コンテキストを使用している場合は、共通データベース・セッション・ベースのアプリケーション・コンテキストおよび共通グローバル・アプリケーション・コンテキスト・オブジェクトを共通仮想プライベート・データベース構成で使用するようにしてください。
- アプリケーション・コンテナの仮想プライベート・データベース・ポリシーは、アプリケーション・ルートに格納されます。PDBにはローカル・ポリシーのみが格納されます。PDBをアプリケーション・コンテナに接続する場合、共通ポリシーはローカル・ポリシーに変換されません。かわりに、Oracle Databaseがアプリケーション・ルートからそれらをロードし、ポリシーからローカルPDBの共通オブジェクトにアクセスがあると、ローカルPDBでそれらを強制します。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [マルチテナント環境でのOracle Virtual Private Databaseの使用](#)

マルチテナント環境でのTransport Layer Securityの使用

Transport Layer Security (TLS)はアプリケーション・コンテナのマルチテナント環境で使用できます。

アプリケーション・コンテナのマルチテナント環境でTransport Layer Security (TLS)を使用する場合、各PDBで独自のウォレットと独自のTLS認証用の証明書を使用できることを確認する必要があります。

- 各PDBに対して個別のsqlnet.oraファイルはないため、walletディレクトリのサブディレクトリにウォレットを置きます。サブディレクトリの名前は、ウォレットを使用するPDBのGUIDです。たとえば、sqlnet.oraのWALLET_LOCATIONパラメータが次のように設定されているとします。

```
(SOURCE=(METHOD=FILE)(METHOD_DATA=
(DIRECTORY=/home/oracle/wallet)))
```

各PDBのウォレットを/home/oracle/wallet/PDB_GUIDディレクトリに置きます。既存のPDBおよびそのGUIDを確認するには、DBA_PDBSデータ・ディクショナリ・ビューを問い合わせます。

WALLET_LOCATIONパラメータが指定されていない場合、デフォルトのウォレット・パスのリーフ・サブディレクトリにPDBウォレットを置く必要があります。サブディレクトリの名前はPDBのGUID、リーフ・サブディレクトリの名前はwalletです。次に例を示します。

```
$ORACLE_BASE/admin/db_unique_name/PDB_GUID/wallet
```

または、ORACLE_BASE環境変数が設定されていない場合は、Oracleホームを使用できます。

```
$ORACLE_HOME/admin/db_unique_name/PDB_GUID/wallet
```

これらのデフォルトの場所は、LDAPの認証用にウォレットを特定するためにOracle Enterprise User Securityによって使用されるデフォルトに対応します。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [マルチテナント環境のセキュリティの管理](#)

マルチテナント環境におけるOracle Data Redaction

マルチテナント環境では、Oracle Data Redactionポリシーは、現在のプラガブル・データベース(PDB)内のオブジェクトのみに適用されます。

データ・リダクション・ポリシーは、マルチテナント・コンテナ・データベース(CDB)には作成できません。これは、一般的にデータ・リダクション・ポリシーを作成するオブジェクトがPDBにあるためです。SYSDBA権限を持っている場合は、SHOW PDBSコマンドを実行して、CDB内のすべてのPDBをリストできます。

CDBルートと同様、アプリケーション・ルートにデータ・リダクション・ポリシーを作成することはできません。

親トピック: [マルチテナント環境のセキュリティの管理](#)

マルチテナント環境での監査の概要

監査とは、構成済のデータベース・アクション(データベース・ユーザーと非データベース・ユーザーの両方からのアクション)を監視して記録することです。非データベース・ユーザーとは、CLIENT_IDENTIFIER属性を使用してデータベースで認識されるアプリケーション・ユーザーのことです。

実行されたSQL文のタイプなどの個々のアクション、またはユーザー名、アプリケーション、時間などの様々なデータの組合せをベースとして使用できます。成功したアクティビティと失敗したアクティビティの両方の監査が可能で、特定のユーザーを監査対象に含めたり、除外したりできます。マルチテナント環境では、プラガブル・データベース(PDB)の個々のアクション、またはマルチテナント・コンテナ・データベース(CDB)全体の個々のアクションを監査できます。

監査はデフォルトで有効です。監査レコードは、同一の形式で統合監査証跡に書き込まれ、UNIFIED_AUDIT_TRAILビューから使用できます。

- [マルチテナント環境での統合監査](#)
マルチテナント環境では統合監査を使用できます。
- [例: マルチテナント環境でのDBAロールの監査](#)
マルチテナント環境で、CREATE AUDIT POLICY文を使用してロールを監査できます。
- [マルチテナント環境での統合監査ポリシーまたはAUDIT設定](#)
マルチテナント環境では、個々のPDBおよびルートに統合監査ポリシーを作成できます。
- [マルチテナント環境でのファイングレイン監査](#)
ファイングレイン監査ポリシーは、CDBルート、アプリケーション・ルート、CDB PDBおよびアプリケーションPDBで作成できます。

親トピック: [マルチテナント環境のセキュリティの管理](#)

マルチテナント環境での統合監査

マルチテナント環境では、統合監査を使用できます。

ポリシーのタイプに応じて、各PDBに監査設定を適用したり、CDBに監査設定を適用できます。マルチテナント環境では、ルートを含むPDBにそれぞれ独自の統合監査証跡があります。

マルチテナント環境の監査設定は、次の領域に影響を与えます。

- CREATE AUDIT POLICYおよびAUDIT文で作成された統合監査ポリシー: ルートと個々のPDBの両方のポリシーを作成できます。
- SYSLOGに書き込まれた監査レコード: UNIXプラットフォームでは、CDBルートでUNIFIED_AUDIT_COMMON_SYSTEMLOG初期化パラメータを設定して、特定の統合監査証跡列がSYSLOGに書き込

まれるようにできます。WindowsとUNIXの両方で、UNIFIED_AUDIT_SYSTEMLOGパラメータをルートとPDBレベルの両方で設定できます。

- ファイングレイン監査ポリシー：ルートではなく、個々のPDBのポリシーのみを作成できます。
- 監査証跡の削除：ルートと個々のPDBの両方に対して、削除操作を実行できます。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [マルチテナント環境での監査の概要](#)

例: マルチテナント環境でのDBAロールの監査

マルチテナント環境で、CREATE AUDIT POLICY文を使用してロールを監査できます。

次の例は、マルチテナント環境で事前定義済の共通ロールDBAを監査する方法を示しています。

例18-4 マルチテナント環境でのDBAロールの監査

```
CREATE AUDIT POLICY role_dba_audit_pol
ROLES DBA
CONTAINER = ALL;

AUDIT POLICY role_dba_audit_pol;
```

親トピック: [マルチテナント環境での監査の概要](#)

マルチテナント環境での統合監査ポリシーまたはAUDIT設定

マルチテナント環境では、個々のPDBおよびルートに統合監査ポリシーを作成できます。

- [ローカル、CDB共通およびアプリケーション共通監査ポリシーについて](#)
監査ポリシーは、ローカル、CDB共通またはアプリケーション共通のいずれかにすることができます。
- [マルチテナント環境での従来の監査](#)
従来型の監査(統合監査ではない)では、AUDITおよびNOAUDIT文で、マルチテナント環境の文と権限を監査できます。
- [ローカル統合監査ポリシーまたは共通統合監査ポリシーの構成](#)
CONTAINER句の使用はマルチテナント環境限定で、CREATE AUDIT POLICY文で使用します。
- [例: ローカル統合監査ポリシー](#)
CREATE AUDIT POLICY文で、ルートまたはPDBにローカル統合監査ポリシーを作成できます。
- [例: CDB共通統合監査ポリシー](#)
CREATE AUDIT POLICY文で、CDB共通統合監査ポリシーを作成できます。
- [例: アプリケーション共通統合監査ポリシー](#)
アプリケーション・コンテナ共通統合監査ポリシーの場合、アクション・オプションとシステム権限オプションを監査して、共通オブジェクトおよびロールを参照できます。
- [監査証跡でのローカルまたは共通監査ポリシーまたは設定の表示方法](#)
ルートまたはアクションが発生したPDBから、統合監査ポリシー・ビューを問い合わせることができます。

親トピック: [マルチテナント環境での監査の概要](#)

ローカル、CDB共通およびアプリケーション共通監査ポリシーについて

監査ポリシーは、ローカル、CDB共通またはアプリケーション共通のいずれかにすることができます。

これは統合監査ポリシーと、AUDIT SQL文を使用して作成されたポリシーの両方に適用されます。

- ローカル監査ポリシー。このタイプのポリシーは、ルート(CDBまたはアプリケーション)またはPDB (CDBまたはアプリケーション)に存在できます。ルートに存在するローカル監査ポリシーには、ローカル・オブジェクトと共通オブジェクトの両方用のオブジェクト監査オプションを含めることができます。AUDIT_ADMINロールが付与されているローカル・ユーザーおよび共通ユーザーは、両方ともローカル・ポリシーを有効にできます(ローカル・ユーザーはPDBから、共通ユーザーは権限のあるルートまたはPDBから有効にできます)。ローカル監査ポリシーは、ローカル・ユーザーおよびロールと共通ユーザーおよびロールの両方に対して有効にすることができます。

ローカル監査ポリシーは、アプリケーション・ローカル・オブジェクトおよびアプリケーション・ローカル・ロールのほか、システム・アクション・オプションおよびシステム権限オプションに対して作成できます。ローカル監査ポリシーをすべてのコンテナにわたり共通ユーザーに対して実行することや、共通監査ポリシーをローカル・ユーザーに対して実施することはできません。

- CDB共通監査ポリシー。このタイプのポリシーは、マルチテナント環境のすべてのPDBに使用できます。共通監査ポリシーを作成および保持できるのは、AUDIT_ADMINロールが付与されている共通ユーザーのみです。共通監査ポリシーは、共通ユーザーのみに対して有効にできます。共通監査ポリシーは、ルートにのみ作成する必要があります。このタイプのポリシーには、共通オブジェクトのみのオブジェクト監査オプションを含めることができ、共通ユーザーのみに対して有効にできます。共通監査ポリシーは、共通ユーザーおよびロールに対してのみ有効にできます。

共通監査ポリシーをすべてのコンテナにわたりローカル・ユーザーに対して実施することはできません。

- アプリケーション共通監査ポリシー。CDB共通監査ポリシーと同様、このタイプのポリシーは、マルチテナント環境のすべてのPDBに使用できます。共通監査ポリシーは、アプリケーション共通オブジェクトおよびアプリケーション共通ロールのほか、システム・アクション・オプションおよびシステム権限オプションに対して作成できます。このタイプのポリシーはアプリケーション・ルート・コンテナでのみ作成できますが、アプリケーション共通ユーザーとCDB共通ユーザーの両方で有効にできます。オブジェクトを監査する場合は、これらのオブジェクトがアプリケーション共通オブジェクトであることを確認してください。DBA_OBJECTSデータ・ディクショナリ・ビューのSHARING列を問い合わせることで、オブジェクトがアプリケーション共通オブジェクトであるかどうかを判断できます。

デフォルトでは、CDBとアプリケーションの両方のシナリオにおいて、監査ポリシーは現在のPDBに対してローカルです。

次の表では、異なるマルチテナント環境における監査ポリシーの適用方法について説明します。

表18-1 CDBルート、アプリケーション・ルートおよび個々のPDBへの監査ポリシーの適用方法

監査オプション・タイプ	CDBルート	アプリケーション・ルート	個々のPDB
共通監査文または監査ポリシー	CDB 共通ユーザーに適用されます	CDB 共通ユーザーに適用されます	CDB 共通ユーザーに適用されます
アプリケーション・コンテナの共通監査文または監査ポリシー	該当なし	<ul style="list-style-type: none"> ● CDB 共通ユーザーに適用され、現在のアプリケーション・コンテナに対してのみ有効です ● アプリケーション・コンテナ共通ユーザーに適用されます 	<ul style="list-style-type: none"> ● CDB 共通ユーザーに適用され、このアプリケーション・コンテナに対してのみ有効です ● アプリケーション共通ユーザーに適用されます
ローカル監査文または監査ポリシー	ローカル構成は許可されません	ローカル構成は許可されません	<ul style="list-style-type: none"> ● CDB 共通ユーザーに適用されます ● アプリケーション共通ユーザー

に適用されます

親トピック: [マルチテナント環境での統合監査ポリシーまたはAUDIT設定](#)

マルチテナント環境での従来の監査

従来型の監査(統合監査ではない)では、AUDITおよびNOAUDIT文で、マルチテナント環境の文と権限を監査できます。

ローカル監査ポリシーまたは共通監査ポリシーになるように監査ポリシーを構成するには、作成または変更の他のSQL文に対して一般的に行うように、CONTAINER句を含める必要があります。アプリケーション・コンテナを監査する場合、ローカル・ユーザーおよびロールならびに共通ユーザーおよびロールによって実行されたSQL文およびシステム権限を監査できます。監査レコードは、アクションが実行されたコンテナに作成されます。

- AUDITまたはNOAUDIT文を現在のCDBまたはアプリケーションPDBに適用する場合は、このPDBでCONTAINERをCURRENTに設定する必要があります。次に例を示します。

```
AUDIT DROP ANY TABLE BY SYSTEM BY ACCESS CONTAINER = CURRENT;
```

- AUDITまたはNOAUDIT文をマルチテナント環境全体に適用する場合は、CDBルートでCONTAINERをALLに設定する必要があります。アプリケーション・ルートの場合は、アプリケーション・ルート内に設定します。次に例を示します：

```
AUDIT DROP ANY TABLE BY SYSTEM BY ACCESS CONTAINER = ALL;
```

従来の監査オプションがアプリケーション・コンテナでの使用を考慮して設計されているかどうかを確認するには、DBA_OBJ_AUDIT_OPTSおよびDBA_OBJECTSデータ・ディクショナリ・ビューの結合問合せを実行します。具体的には、両方のビューでOWNERおよびOBJECT_NAME列を使用し、DBA_OBJECTSでAPPLICATION列を使用します。

関連トピック

- [Oracle Database SQL言語リファレンス](#)

関連項目:

従来の[AUDIT](#)および[NOAUDIT](#) SQL文の詳細は、*Oracle Database SQL言語リファレンス*を参照してください

親トピック: [マルチテナント環境での統合監査ポリシーまたはAUDIT設定](#)

ローカル統合監査ポリシーまたは共通統合監査ポリシーの構成

CONTAINER句の使用はマルチテナント環境限定で、CREATE AUDIT POLICY文で使用します。

CDB環境またはアプリケーション・コンテナ環境でローカルまたは共通(CDBまたはアプリケーション)統合監査ポリシーを作成するには、CREATE AUDIT POLICY文にCONTAINER句を含めます。

- 次の構文を使用して、ローカル統合監査ポリシーまたは共通統合監査ポリシーを作成します。

```
CREATE AUDIT POLICY policy_name  
  action1 [, action2 ]  
  [CONTAINER = {CURRENT | ALL}];
```

詳細は、次のとおりです。

- CURRENTは、監査ポリシーを現在のPDBにローカルになるように設定します。
- ALLは、監査ポリシーを共通監査ポリシー(マルチテナント環境全体で使用可能にする)にします。

たとえば、共通統合監査ポリシーの場合は次のようになります。

```
CREATE AUDIT POLICY dict_updates
ACTIONS UPDATE ON SYS.USER$,
DELETE ON SYS.USER$,
UPDATE ON SYS.LINK$,
DELETE ON SYS.LINK$
CONTAINER = ALL;
```

次の点に注意してください。

- CONTAINER句はCREATE AUDIT POLICY文に設定できますが、ALTER AUDIT POLICYまたはDROP AUDIT POLICYには設定できません。この設定を使用するように既存の統合監査ポリシーの範囲を変更する場合は、ポリシーを削除してから再作成します。
- AUDIT文の場合は、リリース12.x以降の監査機能にまだ移行していないOracleデータベースがある場合など、監査設定のみにCONTAINER句を設定します。統合監査ポリシーを有効にするために使用するAUDIT文には、CONTAINER句を使用できません。
- PDBにいる場合、CONTAINER句に設定できるのはALLではなくCURRENTのみです。PDBにいる場合に設定を省略すると、デフォルトはCONTAINER = CURRENTになります。
- ルートにいる場合は、CONTAINER句を、ポリシーをルートのみ適用する場合はCURRENTに、ポリシーをCDB全体に適用する場合はALLに設定できます。CONTAINER句を省略すると、デフォルトはCONTAINER = CURRENTになります。
- オブジェクトの場合：
 - 共通監査ポリシーには共通オブジェクトのみ、ローカル監査ポリシーにはローカル・オブジェクトと共通オブジェクトの両方を含めることができます。
 - 関係するオブジェクトがローカルの場合は、CONTAINERをALLに設定することはできません。共通オブジェクトにする必要があります。
- 権限の場合：
 - 関係するユーザー・アカウントがローカル・アカウントと共通アカウントの混在の場合は、CONTAINERをCURRENTに設定(またはCONTAINER句を省略)できます。これにより、現在のPDBのみに適用されるローカル監査構成が作成されます。
 - 関係するユーザーがローカル・ユーザーの場合は、CONTAINERをALLに設定することはできません。共通ユーザーにする必要があります。
 - CONTAINERをALLに設定し、ユーザー・リストを指定しない場合(BY句をAUDIT文に使用)、構成が各PDBのすべての共通ユーザーに適用されます。
- アプリケーション・コンテナの場合、アプリケーションのインストール、アップグレード、パッチ適用およびアンインストールに使用されるアプリケーション・コンテナ・スクリプトから共通統合監査ポリシーを実行できます。そのように行うには：
 - アプリケーション・コンテナ・ルートに共通統合監査ポリシーを作成し、このポリシーをCONTAINER = ALLに設定します。または、このポリシーを次のステップで説明するスクリプトに含めることもできます。
 - Oracle Databaseのインストール、アップグレード、パッチ適用またはアンインストールに通常使用するスクリプトのカスタム・バージョンを作成します。

- このスクリプト内の次の行に、監査するSQL文を含めます。

```
ALTER PLUGGABLE DATABASE APPLICATION BEGIN INSTALL
List SQL statements here. Separate each statement with a semi-colon.
ALTER PLUGGABLE DATABASE APPLICATION END INSTALL
```

スクリプトに統合監査ポリシーを含める場合は、CREATE AUDIT POLICYとAUDIT POLICYの両方の文を含めるようにします。

監査ポリシーを作成して有効にした後、監査ポリシーがデータベースで定義されたものであるか、スクリプトからのものであるかにかかわらず、アプリケーション共通オブジェクトへのすべてのユーザー・アクセスが監査されます。

- アプリケーションのインストール、アップグレード、パッチ適用およびアンインストール操作をアプリケーション・ルートまたはアプリケーションPDBでローカルに監査するには、共通統合監査ポリシーに関する前の手順と同様の手順に従いますが、後からアプリケーションPDBを同期します。次に例を示します。

```
ALTER PLUGGABLE DATABASE APPLICATION application_name SYNC;
```

関連トピック

- [アプリケーション・コンテナ内のアプリケーションの管理](#)

親トピック: [マルチテナント環境での統合監査ポリシーまたはAUDIT設定](#)

例: ローカル統合監査ポリシー

CREATE AUDIT POLICY文で、ルートまたはPDBにローカル統合監査ポリシーを作成できます。

ルートでローカル統合監査ポリシーを作成すると、マルチテナント環境全体ではなく、ルートにのみ適用されます。

次の例は、共通ユーザーc##sec_adminによってPDBから作成され、共通ユーザーc##hr_adminに適用されているローカル統合監査ポリシーを示しています。

例18-5 ローカル統合監査ポリシー

```
CONNECT c##sec_admin@hrpdb
Enter password: password
Connected.

CREATE AUDIT POLICY table_privs
PRIVILEGES CREATE ANY TABLE, DROP ANY TABLE
CONTAINER = CURRENT;

AUDIT POLICY table_privs BY c##hr_admin;
```

親トピック: [マルチテナント環境での統合監査ポリシーまたはAUDIT設定](#)

例: CDB共通統合監査ポリシー

CREATE AUDIT POLICY文で、CDB共通統合監査ポリシーを作成できます。

[例18-6](#)に、共通ユーザーc##sec_adminによってルートから作成され、共通ユーザーc##hr_adminに適用されている共通統合監査ポリシーを示します。

例18-6 共通統合監査ポリシー

```
CONNECT c##sec_admin
Enter password: password
Connected.
```

```
CREATE AUDIT POLICY admin_pol
ACTIONS CREATE TABLE, ALTER TABLE, DROP TABLE
ROLES c##hr_mgr, c##hr_sup
CONTAINER = ALL;
```

```
AUDIT POLICY admin_pol BY c##hr_admin;
```

親トピック: [マルチテナント環境での統合監査ポリシーまたはAUDIT設定](#)

例: アプリケーション共通統合監査ポリシー

アプリケーション・コンテナ共通統合監査ポリシーの場合、アクション・オプションとシステム権限オプションを監査して、共通オブジェクトおよびロールを参照できます。

アプリケーション共通監査ポリシーの作成はアプリケーション・ルートからに限定されますが、このポリシーをアプリケーション共通ユーザーとCDB共通ユーザーの両方に対して有効にできます。

次の例は、アプリケーション・コンテナapp_pdbでアプリケーション共通ユーザーSYSTEMを監査するポリシーを作成する方法を示しています。この監査ポリシーは、SYSTEM. utils_tab表に対するSELECTアクション、およびコンテナ・データベース(CDBルートを含む)内の任意のPDBに対するDROP TABLEアクションを監査します。このポリシーは、すべてのコンテナでSELECT ANY TABLEシステム権限の使用も監査します。

例18-7 アプリケーション共通統合監査ポリシー

```
CONNECT c##sec_admin@app_pdb
Enter password: password
Connected.
```

```
CREATE AUDIT POLICY app_pdb_admin_pol
ACTIONS SELECT ON hr_app_cdb.utils_tab, DROP TABLE
PRIVILEGES SELECT ANY TABLE
CONTAINER = ALL;
```

```
AUDIT POLICY app_pdb_admin_pol by SYSTEM, c##hr_admin;
```

前述の例では、CONTAINERをALLに設定することで、ポリシーは、アプリケーション・ルートおよびアプリケーション・ルートに属するすべてのアプリケーションPDBのすべての関連オブジェクトへのアクセスのみに適用されます。この範囲外にポリシーは適用されません。

親トピック: [マルチテナント環境での統合監査ポリシーまたはAUDIT設定](#)

監査証跡でのローカルまたは共通監査ポリシーまたは設定の表示方法

ルートまたはアクションが発生したPDBから、統合監査ポリシー・ビューを問い合わせることができます。

次のタイプの間合せを実行できます。

- すべてのPDBからのレコードを監査する。監査証跡には、PDBで実行された監査アクションが反映されます。たとえば、PDB1のユーザーlbrownが、共通監査ポリシーまたはローカル監査ポリシーのいずれかによって監査されたアクションを実行すると、監査証跡によってこのアクションが取得されます。UNIFIED_AUDIT_TRAILデータ・ディクショナリ・ビューのDBID列は、監査アクションが実行され、ポリシーが適用されるPDBを示します。すべてのPDBからの監査レコードを確認する場合は、ルートからCDB_UNIFIED_AUDIT_TRAILデータ・ディクショナリ・ビューを問い合わせる必要があります。
- 共通監査ポリシーからのレコードを監査する。この場所は、共通監査ポリシーが監査レコードになる場所です。アクションが実際に発生した場所に依りて、マルチテナント環境(ルートまたはPDB)の任意の場所で監査レコードを生成できます。たとえば、共通監査ポリシーfga_polは、DBMS_FGA PL/SQLパッケージのEXECUTE権限を監査し、このアクション

ンがPDB1で発生すると、監査レコードはルートではなく、PDB1に生成されます。このため、監査レコードはPDB1に表示できます。

ポリシー名にWHERE句を使用している場合は(例: WHERE UNIFIED_AUDIT_POLICIES = 'FGA_POL'), ポリシーについて、ルートまたはPDBのいずれかからUNIFIED_AUDIT_TRAILデータ・ディクショナリ・ビューを問い合わせることができます。

次の例は、共通統合監査ポリシーの結果を検索する方法を示します。

```
CONNECT c##sec_admin
Enter password: password
Connected.

SELECT DBID, ACTION_NAME, OBJECT_SCHEMA, OBJECT_NAME FROM CDB_UNIFIED_AUDIT_TRAIL WHERE DBUSERNAME =
'c##hr_admin';
46892-1
```

DBID	ACTION_NAME	OBJECT_SCHEMA	OBJECT_NAME
653916017	UPDATE	HR	EMPLOYEES
653916018	UPDATE	HR	JOB_HISTORY
653916017	UPDATE	HR	JOBS

親トピック: [マルチテナント環境での統合監査ポリシーまたはAUDIT設定](#)

マルチテナント環境でのファイングレイン監査

ファイングレイン監査ポリシーは、CDBルート、アプリケーション・ルート、CDB PDBおよびアプリケーションPDBで作成できます。

マルチテナント環境におけるファイングレイン監査ポリシーには、次のような一般的なルールがあります。

- ファイングレイン監査ポリシーは、SYSオブジェクトに対して作成できません。
- ファイングレイン監査ポリシーは(ローカルまたはアプリケーション共通を問わず)、拡張データ・リンク・オブジェクトに対して作成できません。
- CDBルートでファイングレイン監査ポリシーを作成する場合、すべてのPDBにポリシーを適用することはできません。ポリシーはCDBルート内のオブジェクトに適用されます。(つまり、CDBルートに対する共通のファイングレイン監査ポリシーは存在しません。)すべてのPDBで共通オブジェクトのアクセスを監査するようにファイングレイン監査ポリシーを作成する場合は、監査ポリシーを各PDBで明示的に作成し、PDBでアクセス可能にする共通オブジェクトに対してそのポリシーを有効化する必要があります。
- PDBでファイングレイン監査ポリシーを作成する場合、ポリシーはPDB内のオブジェクトにのみ適用されます。
- アプリケーション共通ファイングレイン監査ポリシーは、アプリケーション・ルートに接続し、BEGIN/ENDブロック内にある場合にのみ作成できます。アプリケーション・ルートに接続し、BEGIN/ENDブロック外でファイングレイン監査ポリシーを作成すると、ファイングレイン監査ポリシーはアプリケーション・ルートに作成されます。
- アプリケーション共通ファイングレイン監査ポリシーは、ローカルPDBオブジェクトに対して作成できません。
- アプリケーション共通ファイングレイン監査ポリシーにハンドラがある場合、このハンドラはアプリケーション共通ユーザーまたはCDB共通ユーザーによって所有されている必要があります。
- アプリケーション・ファイングレイン監査ポリシーは、ローカル(PDB)オブジェクトおよびCDB共通オブジェクトに対して作成できます。ポリシーはそのコンテナに対してローカルであるため、ポリシーが定義されたオブジェクトは、ポリシーが定義された特定のコンテナ内でのみ監査されます。たとえば、ファイングレイン監査ポリシーをhr_pdb PDBで作成する場合、このポリシーを作成する対象のオブジェクトは、hr_pdb PDB内に存在する必要があります。
- ローカル・ファイングレイン監査ポリシーは、アプリケーションPDB内のオブジェクト・リンク・オブジェクトおよび拡張データ・リ

リンク・オブジェクトに対して作成できません。メタデータリンク・オブジェクトは、ファイングレイン監査ポリシーで使用できます。

- アプリケーション・ルート・ローカル・ポリシーは、アプリケーション共通オブジェクトに対して使用できます。
- ファイングレイン監査ポリシーを共通監査ポリシーとしてアプリケーション・ルートで作成する場合、このアプリケーション・ルートに属する各PDBで有効になります。したがって、アプリケーションPDBのアプリケーション共通オブジェクトおよびCDB共通オブジェクト(アプリケーション共通ファイングレイン監査ポリシーが定義されたもの)は、そのアプリケーションPDB内のファイングレイン監査証跡において監査されます。
- アプリケーションのインストール、アップグレード、パッチ適用またはアンインストール操作のスクリプトを作成する際、ALTER PLUGGABLE DATABASE app_name BEGIN INSTALLおよびALTER PLUGGABLE DATABASE app_name END INSTALLブロック内にSQL文を含めて、様々な操作を実行できます。ファイングレイン監査ポリシー文は、これらのブロック内にのみ含めることができます。
- アプリケーション共通ファイングレイン監査ポリシーの有効化、無効化または削除の実行は、アプリケーション・ルートから、およびスクリプト内のALTER PLUGGABLE DATABASE app_name BEGIN INSTALLおよびALTER PLUGGABLE DATABASE app_name END INSTALLブロック内からに限定されます。

関連トピック

- [『Oracle Databaseセキュリティ・ガイド』](#)

親トピック: [マルチテナント環境での監査の概要](#)

19 CDBおよびPDBの監視

SQL*PlusまたはSQL Developerを使用して、CDBおよびPDBに関する情報を表示できます。

- [ビュー内のCDBおよびコンテナ情報について](#)
CDBでは、データ・ディクショナリ表およびビュー定義のメタデータがルートのみで格納されます。
- [データベースがCDBであるかどうかの判断](#)
V\$DATABASEビューのCDB列を問い合わせ、データベースがCDBであるか非CDBであるかを判断できます。CDB列は、現在のデータベースがCDBの場合はYESを返し、現在のデータベースが非CDBの場合はNOを返します。
- [CDB内のコンテナに関する情報の表示](#)
V\$CONTAINERSビューは、ルートおよびすべてのPDBを含む、CDB内のすべてのコンテナに関する情報を提供します。
- [PDBに関する情報の表示](#)
CDB_PDBSビューおよびDBA_PDBSビューは、各PDBのステータスなど、CDBに関連付けられたPDBに関する情報を提供します。
- [各PDBのオープン・モードの表示](#)
V\$PDBSビューは、現在のデータベース・インスタンスに関連付けられたPDBに関する情報を提供します。
- [コンテナ・データ・オブジェクトの問合せ](#)
ルートでは、コンテナ・データ・オブジェクトは、ルートおよびそのPDBに含まれるデータベース・オブジェクト(表やユーザーなど)に関する情報を表示できます。PDB情報へのアクセスは、共通ユーザーのCONTAINER_DATA属性によって制御されます。
- [CONTAINERS句を使用したコンテナ間の問合せ](#)
CONTAINERS句を使用すると、CDB内のすべてのコンテナ間で表およびビューを問い合わせることができます。アプリケーション・コンテナ内のすべてのコンテナでアプリケーション共通オブジェクトを問い合わせることもできます。
- [現在のコンテナのIDまたは名前の判別](#)
CDB内の現在のコンテナIDまたはコンテナ名を判別できます。
- [PDBの変更可能な初期化パラメータをリスト](#)
CDBでは、一部の初期化パラメータがルートおよびすべてのPDBに適用されます。このような初期化パラメータが変更されると、CDB全体に影響を与えます。他の初期化パラメータは、各コンテナで異なる値に設定できます。
- [PDBの履歴の表示](#)
CDB_PDB_HISTORYビューには、CDB内のPDBの履歴が表示されます。各PDBがいつどのように作成されたかに関する情報、および各PDBの履歴に関するその他の情報を提供します。
- [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)
複数のビューが、CDB内のアプリケーション・コンテナ内のアプリケーションに関する情報を提供します。

関連トピック

- [マルチテナント環境のツール](#)

親トピック: [マルチテナント環境の管理](#)

ビュー内のCDBおよびコンテナの情報について

CDBでは、データ・ディクショナリ表およびビュー定義のメタデータがルートのみで格納されます。

各コンテナ(各PDB、アプリケーション・ルート、アプリケーションPDBを含む)には、そのコンテナに含まれているオブジェクトに関する、独自のデータ・ディクショナリ表とビューのセットがあります。各コンテナには異なるデータ・オブジェクトとスキーマ・オブジェクトを含めることができるため、各コンテナで同じビューを問い合わせても、コンテナによってデータ・ディクショナリ・ビューに異なるメタデータが表示される場合があります。たとえば、2つの異なるコンテナでは、DBA_TABLESビューに表示される、表に関するメタデータが異なる

場合があり、これは、コンテナごとに異なる表が含まれていることがあるためです。コンテナは、[メタデータ・リンク](#)と呼ばれる内部メカニズムを使用して、ルート内のこれらのビューのメタデータにアクセスできます。

ディクショナリ表に、各コンテナではなくCDB全体に関連する情報が格納される場合は、メタデータと、データ・ディクショナリ・ビューに表示されるデータの両方がルートに格納されます。たとえば、自動ワークロード・リポジトリ(AWR)データはルートに格納でき、DBA_HIST_ACTIVE_SESS_HISTORYビューなどの一部のデータ・ディクショナリ・ビューに表示されます。コンテナは、[データ・リンク](#)と呼ばれる内部メカニズムを使用して、ルート内のこれらのタイプのビューのメタデータとデータの両方にアクセスできます。

- [現在のコンテナがCDBルートではない場合の情報の表示について](#)
現在のコンテナがPDB、アプリケーション・ルートまたはアプリケーションPDBである場合、データ・ディクショナリには現在のコンテナのメタデータのみが表示されます。
- [現在のコンテナがCDBルートである場合の情報の表示について](#)
現在のコンテナがCDBルートである場合、共通ユーザーは、コンテナ・データ・オブジェクトを問い合せて、CDBルート、PDB、アプリケーション・ルートおよびアプリケーションPDBのデータ・ディクショナリ情報を表示できます。
- [CDBのビュー](#)
CDBおよびそのPDBに関する情報をビューのセットに問い合せることができます。

関連項目:

コンテナ、メタデータ・リンクおよびデータ・リンクでのディクショナリ・アクセスの詳細は、[「CDBのデータ・ディクショナリ・アーキテクチャ」](#)を参照してください

親トピック: [CDBおよびPDBの監視](#)

現在のコンテナがCDBルートではない場合の情報の表示について

現在のコンテナがPDB、アプリケーション・ルートまたはアプリケーションPDBの場合、データ・ディクショナリには現在のコンテナのみのメタデータが表示されます。

PDB、アプリケーション・ルートまたはアプリケーションPDBに接続されているアプリケーションからは、データ・ディクショナリは非CDB用であるかのように見えます。データ・ディクショナリには、現在のコンテナに関連するメタデータのみが表示されます。また、CDBルートではないコンテナでは、CDBビューには、対応するDBAビューで表示されるデータベース・オブジェクトに関する情報のみが示されます。

親トピック: [ビュー内のCDBおよびコンテナの情報について](#)

現在のコンテナがCDBルートである場合の情報の表示について

現在のコンテナがCDBルートである場合、共通ユーザーは、コンテナ・データ・オブジェクトを問い合せて、CDBルート、PDB、アプリケーション・ルートおよびアプリケーションPDBのデータ・ディクショナリ情報を表示できます。

コンテナ・データ・オブジェクトは、次の内容に関連するデータを含めることができる表またはビューです。

- 1つ以上のコンテナ
- CDB全体
- 1つ以上のコンテナおよびCDB全体

コンテナ・データ・オブジェクトには、V\$, GV\$, CDB_およびいくつかの自動ワークロード・リポジトリDBA_HIST*のビューが含まれています。共通ユーザーのCONTAINER_DATA属性は、コンテナ・データ・オブジェクトに表示できるコンテナを決定します。

CDBでは、すべてのDBA_ビューに、対応するCDB_ビューがあります。すべてのCDB_ビューはコンテナ・データ・オブジェクトですが、ほとんどのDBA_ビューはそうではありません。

各コンテナ・データ・オブジェクトには、返された各行のコンテナを識別するCON_ID列が含まれています。[表19-1](#)では、CON_ID列の値の意味について説明します。

表19-1 コンテナ・データ・オブジェクトのCON_ID列

CON_ID列の値	説明
0	CDB 全体に関連するデータ
1	CDB ルートに関連するデータ
2	PDB シードに関連するデータ
3 - 4,098	PDB、アプリケーション・ルートまたはアプリケーション PDB に関連するデータ 各コンテナには独自のコンテナ ID があります。

次のビューは、他の[G]V\$ビューとは動作が異なります。

- [G]V\$SYSSTAT
- [G]V\$SYS_TIME_MODEL
- [G]V\$SYSTEM_EVENT
- [G]V\$SYSTEM_WAIT_CLASS

CDBルートからの問合せにおいて、これらのビューは、返される各行のCON_ID列に0を含めてインスタンス全体のデータを返します。ただし、他のコンテナ・データ・オブジェクトと同じように動作する同等のビューを問い合わせできます。[G]V\$CON_SYSSTAT、[G]V\$CON_SYS_TIME_MODEL、[G]V\$CON_SYSTEM_EVENTおよび[G]V\$CON_SYSTEM_WAIT_CLASSの各ビューは、CDB内の各コンテナに固有のデータを返すことができます。

ノート:

- コンテナ・データ・オブジェクトの問合せにおいて返されるデータは、コンテナがオープンであるかどうか、および問合せを実行するユーザーに付与された権限によって異なります。
- Oracle Real Application Clusters (Oracle RAC)環境では、コンテナ・データ・オブジェクトによって返されるデータは、セッションの接続先インスタンスによって変わる場合があります。
- 非 CDB では、コンテナ・データ・オブジェクト内のすべての CON_ID 列は 0 (ゼロ)です。
- コンテナが制限モードでオープンされている場合は、CDB_ビューに対する問合せで無視されます。

関連項目:

- [「現在のコンテナについて」](#)
- [「CDBのコンテナ・データ・オブジェクト」](#)
- コンテナ・データ・オブジェクトの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

親トピック: [ビュー内のCDBおよびコンテナの情報について](#)

CDBのビュー

CDBおよびそのPDBに関する情報をビューのセットに問い合わせることができます。

[表19-2](#)は、CDBおよびそのPDBの監視に役立つデータ・ディクショナリ・ビューについて説明しています。

表19-2 CDBのビュー

ビュー	説明	詳細情報
次を含むコンテナ・データ・オブジェクト <ul style="list-style-type: none"> ● V\$ビュー ● GV\$ビュー ● CDB_ビュー ● DBA_HIST*ビュー 	コンテナ・データ・オブジェクトによって、複数の PDB に関する情報を表示できます。各コンテナ・データ・オブジェクトには、コンテナを識別するための CON_ID 列が含まれています。	「コンテナ・データ・オブジェクトの問合せ」 『Oracle Database セキュリティ・ガイド』
{CDB DBA}_PDBS	各 PDB のステータスなど、CDB に関連付けられた PDB に関する情報が表示されます。	「PDB に関する情報の表示」 『Oracle Database リファレンス』
CDB_PROPERTIES	CDB 内の各コンテナの永続的なプロパティが表示されます。	『Oracle Database リファレンス』
{CDB DBA}_PDB_HISTORY	各 PDB の履歴が表示されます。	『Oracle Database リファレンス』
{CDB DBA}_CONTAINER_DATA	CDB に指定されているユーザー・レベルおよびオブジェクト・レベルの CONTAINER_DATA 属性に関する情報が表示されます。	『Oracle Database リファレンス』
{CDB DBA}_HIST_PDB_INSTANCE	ワークロード・リポジトリの PDB およびインスタンスが表示されます。	『Oracle Database リファレンス』
{CDB DBA}_PDB_SAVED_STATES	CDB の現在保存されている PDB の状態に関する情報が表示されます。	『Oracle Database リファレンス』 「CDB 再起動時の PDB のオープン」

ビュー	説明	詳細情報
		モードの保持または破棄
{CDB DBA}_APPLICATIONS	アプリケーション・コンテナ内のすべてのアプリケーションを記述します。	アプリケーションに関する情報の表示
{CDB DBA}_APP_STATEMENTS	アプリケーション・コンテナ内でのアプリケーション・インストール、アップグレードおよびパッチ操作からのすべての文を記述します。	アプリケーション文に関する情報の表示
{CDB DBA}_APP_PATCHES	アプリケーション・コンテナ内のすべてのアプリケーション・パッチを記述します。	アプリケーション・パッチに関する情報の表示
{CDB DBA}_APP_ERRORS	アプリケーション・コンテナ内に生成されたすべてのアプリケーション・エラー・メッセージを記述します。	アプリケーション・エラーに関する情報の表示
{CDB DBA}_CDB_RSRC_PLANS	すべての CDB リソース・プランに関する情報が表示されます。	『 Oracle Database リファレンス 』 「CDB リソース・プランの表示」
{CDB DBA}_CDB_RSRC_PLAN_DIRECTIVES	すべての CDB リソース・プラン・ディレクティブに関する情報が表示されます。	『 Oracle Database リファレンス 』 「CDB リソース・プラン・ディレクティブの表示」
PDB_ALERTS	PDB アラートの理由の説明が示されません。	『 Oracle Database リファレンス 』
PDB_PLUG_IN_VIOLATIONS	PDB と、それが属する CDB 間の非互換性に関する情報が表示されます。このビューは、DBMS_PDB.CHECK_PLUG_COMPATIBILITY を実行して生成される情報を表示するためにも使用されます。	『 Oracle Database リファレンス 』 「切断された PDB の接続」
{USER ALL DBA CDB}_OBJECTS	データベース・オブジェクトに関する情報が表示され、SHARING 列に、データベース・オブジェクトがメタデータ・リンク・オブジェクト、データ・リンク・オブジェクト、拡張データ・リンク・オブジェクト、または別のオブジェクトにリンクされていないスタンド	『 Oracle Database リファレンス 』

ビュー	説明	詳細情報
	アロン・オブジェクトのいずれであるかが示されます。	
{ALL DBA CDB}_SERVICES	データベース・サービスに関する情報が表示され、PDB 列に、各サービスに関連付けられている PDB の名前が示されます。	『Oracle Database リファレンス』
{USER ALL DBA CDB}_VIEWS {USER ALL DBA CDB}_TABLES	CONTAINER_DATA 列に、ビューまたは表がコンテナ・データ・オブジェクトであるかどうかを示されます。	『Oracle Database リファレンス』
{USER ALL DBA CDB}_USERS	COMMON 列に、ユーザーが共通ユーザーであるかローカル・ユーザーであるかが示されます。	『Oracle Database リファレンス』
{USER ALL DBA CDB}_ROLES {USER ALL DBA CDB}_COL_PRIVS {USER ALL}_COL_PRIVS_MADE {USER ALL}_COL_PRIVS_REC'D {USER ALL}_TAB_PRIVS_MADE {USER ALL}_TAB_PRIVS_REC'D {USER DBA CDB}_SYS_PRIVS {USER DBA CDB}_ROLE_PRIVS ROLE_TAB_PRIVS ROLE_SYS_PRIVS	COMMON 列に、ロールまたは権限が共通に付与されているか、またはローカルに付与されているかが示されます。	『Oracle Database リファレンス』
{USER ALL DBA CDB}_ARGUMENTS {USER ALL DBA CDB}_CLUSTERS {USER ALL DBA CDB}_CONSTRAINTS {ALL DBA CDB}_DIRECTORIES {USER ALL DBA CDB}_IDENTIFIERS {USER ALL DBA CDB}_LIBRARIES	ORIGIN_CON_ID 列は、行の元のコンテナ ID を示します。	『Oracle Database リファレンス』

ビュー	説明	詳細情報
{USER ALL DBA CDB}_PROCEDURES		
{USER ALL DBA CDB}_SOURCE		
{USER ALL DBA CDB}_SYNONYMS		
{USER ALL DBA CDB}_VIEWS		
[G]V\$DATABASE	制御ファイルからの、データベースに関する情報が表示されます。データベースが CDB である場合は、CDB 関連の情報が含まれます。	「データベースが CDB であるかどうかの判断」 『Oracle Database リファレンス』
[G]V\$CONTAINERS	ルートおよびすべての PDB を含む、現在の CDB に関連付けられているコンテナに関する情報が表示されます。	「CDB 内のコンテナに関する情報の表示」 『Oracle Database リファレンス』
[G]V\$PDBS	各 PDB のオープン・モードなど、現在の CDB に関連付けられている PDB に関する情報が表示されます。	「各 PDB のオープン・モードの表示」 『Oracle Database リファレンス』
[G]V\$PDB_INCARNATION	すべての PDB インカーネーションに関する情報を示します。RESETLOGS オプションを指定して PDB をオープンすると、新規の PDB インカーネーションが作成されます。	『Oracle Database リファレンス』
[G]V\$SYSTEM_PARAMETER	初期化パラメータに関する情報が表示され、ISPDB_MODIFIABLE 列に PDB のパラメータを変更できるかどうかを示されます。	「PDB の変更可能な初期化パラメータをリスト」 『Oracle Database リファレンス』
[G]V\$PARAMETER		
V\$DIAG_ALERT_EXT	CDB の現在のコンテナのトレース・ファイルおよびアラート・ファイルのデータを表示します。	Oracle Database SQL チューニング・ガイド
[G]V\$DIAG_APP_TRACE_FILE		
[G]V\$DIAG_OPT_TRACE_RECORDS		
V\$DIAG_SESS_OPT_TRACE_RECORDS		
V\$DIAG_SESS_SQL_TRACE_RECORDS		
[G]V\$DIAG_SQL_TRACE_RECORDS		

ビュー	説明	詳細情報
[G]V\$DIAG_TRACE_FILE		
[G]V\$DIAG_TRACE_FILE_CONTENTS		
V\$DIAG_INCIDENT	CDB 内の現在のコンテナの問題および	『Oracle Database リファレンス』
V\$DIAG_PROBLEM	インシデントに関する情報を表示します。	

親トピック: [ビュー内のCDBおよびコンテナの情報について](#)

データベースがCDBであるかどうかの判断

V\$DATABASEビューのCDB列を問い合わせ、データベースがCDBであるか非CDBであるかを判断できます。CDB列は、現在のデータベースがCDBの場合はYESを返し、現在のデータベースが非CDBの場合はNOを返します。

データベースがCDBであるかどうかを判断するには:

1. SQL*Plusで、管理ユーザーとしてデータベースに接続します。
2. V\$DATABASEビューを問い合わせます。

例19-1 データベースがCDBであるかどうかの判断

```
SELECT CDB FROM V$DATABASE;
```

出力例:

```
CDB
---
YES
```

関連項目:

[『Oracle Databaseリファレンス』](#)

親トピック: [CDBおよびPDBの監視](#)

CDB内のコンテナに関する情報の表示

V\$CONTAINERSビューは、ルートおよびすべてのPDBを含む、CDB内のすべてのコンテナに関する情報を提供します。

この情報を表示するには、現在のコンテナがルートである共通ユーザーが問合せを実行する必要があります。現在のコンテナがPDBである場合、このビューには現在のPDBに関する情報のみが表示されます。

CDB内のコンテナに関する情報を表示するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
[「CDB内のコンテナへのアクセスについて」](#)を参照してください。
2. V\$CONTAINERSビューを問い合わせます。

例19-2 CDB内の各コンテナに関する識別情報の表示

```
COLUMN NAME FORMAT A8
```

```
SELECT NAME, CON_ID, DBID, CON_UID, GUID FROM V$CONTAINERS ORDER BY CON_ID;
```

出力例:

NAME	CON_ID	DBID	CON_UID	GUID
CDB\$ROOT	1	659189539	1	C091A6F89C7572A1E0436797E40AC78D
PDB\$SEED	2	4026479912	4026479912	C091AE9C00377591E0436797E40AC138
HRPDB	3	3718888687	3718888687	C091B6B3B53E7834E0436797E40A9040
SALESPDB	4	2228741407	2228741407	C091FA64EF8F0577E0436797E40ABE9F

関連項目:

- [「マルチテナント環境の構成と管理について」](#)
- [「現在のコンテナについて」](#)
- [「現在のコンテナのIDまたは名前の判別」](#)
- [『Oracle Databaseリファレンス』](#)

親トピック: [CDBおよびPDBの監視](#)

PDBに関する情報の表示

CDB_PDBSビューおよびDBA_PDBSビューは、各PDBのステータスなど、CDBに関連付けられたPDBに関する情報を提供します。

この情報を表示するには、現在のコンテナがルートである共通ユーザーが問合せを実行する必要があります。現在のコンテナがPDBである場合は、これらのビューに対するすべての問合せで結果が返されません。

PDBに関する情報を表示するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。

[「SQL*Plusを使用したCDB内のコンテナへのアクセス」](#)を参照してください。

2. CDB_PDBSまたはDBA_PDBSビューを問い合わせます。

例19-3 各PDBのコンテナID、名前およびステータスの表示

```
COLUMN PDB_NAME FORMAT A15
```

```
SELECT PDB_ID, PDB_NAME, STATUS FROM DBA_PDBS ORDER BY PDB_ID;
```

出力例:

PDB_ID	PDB_NAME	STATUS
2	PDB\$SEED	NORMAL
3	HRPDB	NORMAL
4	SALESPDB	NORMAL

関連項目:

[「現在のコンテナについて」](#)

親トピック: [CDBおよびPDBの監視](#)

各PDBのオープン・モードの表示

V\$PDBSビューは、現在のデータベース・インスタンスに関連付けられたPDBに関する情報を提供します。

このビューを問い合わせ、各PDBのオープン・モードを判断します。オープンしている各PDBについて、このビューでは、PDBが最後にオープンされたかのがいつであるかを表示することもできます。共通ユーザーは、現在のコンテナがルートまたはPDBである場合にこのビューを問い合わせることができます。現在のコンテナがPDBである場合、このビューには現在のPDBに関する情報のみが表示されます。

各PDBのオープン・ステータスを表示するには:

1. SQL*Plusでコンテナにアクセスします。

[「SQL*Plusを使用したCDB内のコンテナへのアクセス」](#)を参照してください。

2. V\$PDBSビューを問い合わせます。

例19-4 各PDBの名前およびオープン・モードの表示

```
COLUMN NAME FORMAT A15
COLUMN RESTRICTED FORMAT A10
COLUMN OPEN_TIME FORMAT A30

SELECT NAME, OPEN_MODE, RESTRICTED, OPEN_TIME FROM V$PDBS;
```

出力例:

NAME	OPEN_MODE	RESTRICTED	OPEN_TIME
PDB\$SEED	READ ONLY	NO	21-MAY-12 12.19.54.465 PM
HRPDB	READ WRITE	NO	21-MAY-12 12.34.05.078 PM
SALESPDB	MOUNTED	NO	22-MAY-12 10.37.20.534 AM

関連項目:

- [「ALTER PLUGGABLE DATABASEを使用したPDBのオープン・モードの変更」](#)
- [「PDBのオープン・モードの変更」](#)
- [「ALTER PLUGGABLE DATABASE文を使用したPDBの変更」](#)
- [「現在のコンテナについて」](#)

親トピック: [CDBおよびPDBの監視](#)

コンテナ・データ・オブジェクトの問合せ

ルートでは、コンテナ・データ・オブジェクトは、ルートおよびそのPDBに含まれるデータベース・オブジェクト(表やユーザーなど)に関

する情報を表示できます。PDB情報へのアクセスは、共通ユーザーのCONTAINER_DATA属性によって制御されます。

たとえば、CDB_ビューはコンテナ・データ・オブジェクトです。コンテナ・データ・オブジェクトの詳細は、[「現在のコンテナがCDBルートである場合の情報の表示について」](#)および『[Oracle Databaseセキュリティ・ガイド](#)』を参照してください。

各コンテナ・データ・オブジェクトには、問合せ結果内の各PDBのコンテナIDが表示されたCON_ID列が含まれています。DBA_PDBSビューを問い合せて、コンテナIDのPDB名を表示できます。

コンテナ・データ・オブジェクトを使用して複数のPDBに関する情報を表示するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
[「CDB内のコンテナへのアクセスについて」](#)を参照してください。
2. コンテナ・データ・オブジェクトを問い合せて、目的の情報を表示します。

ノート:



問合せにコンテナ・データ・オブジェクトおよび非コンテナ・データ・オブジェクトの結合が含まれ、現在のコンテナがルートである場合、問合せは CDB 全体のデータのみを返します(CON_ID = 0)。

例19-5 複数のPDBの特定のスキーマが所有する表の表示

この例では、ルートからDBA_PDBSビューおよびCDB_TABLESビューを問い合せて、CDBに関連付けられているPDBのhrユーザーおよびoeユーザーが所有する表を表示します。この問合せは、PDBのIDが2より大きい(p. PDB_ID > 2)行のみを返すことで、CDBルートおよびPDBシードのユーザーを表示しないようにしています。

```
COLUMN PDB_NAME FORMAT A15
COLUMN OWNER FORMAT A15
COLUMN TABLE_NAME FORMAT A30

SELECT p. PDB_ID, p. PDB_NAME, t. OWNER, t. TABLE_NAME
FROM DBA_PDBS p, CDB_TABLES t
WHERE p. PDB_ID > 2 AND
      t. OWNER IN ('HR', 'OE') AND
      p. PDB_ID = t. CON_ID
ORDER BY p. PDB_ID;
```

出力例:

PDB_ID	PDB_NAME	OWNER	TABLE_NAME
3	HRPDB	HR	COUNTRIES
3	HRPDB	HR	JOB_HISTORY
3	HRPDB	HR	EMPLOYEES
3	HRPDB	HR	JOBS
3	HRPDB	HR	DEPARTMENTS
3	HRPDB	HR	LOCATIONS
3	HRPDB	HR	REGIONS
4	SALESPDB	OE	PRODUCT_INFORMATION
4	SALESPDB	OE	INVENTORIES
4	SALESPDB	OE	ORDERS
4	SALESPDB	OE	ORDER_ITEMS
4	SALESPDB	OE	WAREHOUSES
4	SALESPDB	OE	CUSTOMERS
4	SALESPDB	OE	SUBCATEGORY_REF_LIST_NESTEDTAB
4	SALESPDB	OE	PRODUCT_REF_LIST_NESTEDTAB
4	SALESPDB	OE	PROMOTIONS

この出力例では、PDB hrpdbにhrスキーマの表があり、PDB salespdbにoeスキーマの表があることを示しています。

例19-6 複数のPDBのユーザーの表示

この例では、ルートからDBA_PDBSビューおよびCDB_USERSビューを問い合わせ、各PDBのユーザーを表示します。この問合せでは、p. PDB_ID > 2を使用して、CDBルートおよびPDBシードのユーザーを表示しないようにしています。

```
COLUMN PDB_NAME FORMAT A15
COLUMN USERNAME FORMAT A30

SELECT p. PDB_ID, p. PDB_NAME, u. USERNAME
FROM DBA_PDBS p, CDB_USERS u
WHERE p. PDB_ID > 2 AND
      p. PDB_ID = u. CON_ID
ORDER BY p. PDB_ID;
```

出力例:

PDB_ID	PDB_NAME	USERNAME
.		
.		
.		
3	HRPDB	HR
3	HRPDB	OLAPSYS
3	HRPDB	MDSYS
3	HRPDB	ORDSYS
.		
.		
.		
4	SALESPDB	OE
4	SALESPDB	CTXSYS
4	SALESPDB	MDSYS
4	SALESPDB	EXFSYS
4	SALESPDB	OLAPSYS
.		
.		
.		

例19-7 CDB内の各PDBのデータファイルの表示

この例では、DBA_PDBSおよびCDB_DATA_FILESビューを問い合わせ、PDBシードなど、CDB内のすべてのPDBの各データファイルの名前と場所を表示します。

```
COLUMN PID FORMAT 999
COLUMN PDB_NAME FORMAT A8
COLUMN FILE_ID FORMAT 9999
COLUMN TABLESPACE_NAME FORMAT A10
COLUMN FILE_NAME FORMAT A45

SELECT p. PDB_ID AS PID, p. PDB_NAME, d. FILE_ID, d. TABLESPACE_NAME, d. FILE_NAME
FROM DBA_PDBS p, CDB_DATA_FILES d
WHERE p. PDB_ID = d. CON_ID
ORDER BY p. PDB_ID;
```

出力例:

PID	PDB_NAME	FILE_ID	TABLESPACE	FILE_NAME
-----	----------	---------	------------	-----------

```

-----
2 PDB$SEED      6 SYSAUX      /disk1/oracle/dbs/pdbseed/cdb1_ax. f
2 PDB$SEED      5 SYSTEM      /disk1/oracle/dbs/pdbseed/cdb1_db. f
3 HRPDB         9 SYSAUX      /disk1/oracle/dbs/hrpdb/hrpdb_ax. f
3 HRPDB         8 SYSTEM      /disk1/oracle/dbs/hrpdb/hrpdb_db. f
3 HRPDB        13 USER        /disk1/oracle/dbs/hrpdb/hrpdb_usr. dbf
4 SALESPDB     15 SYSTEM      /disk1/oracle/dbs/salespdb/salespdb_db. f
4 SALESPDB     16 SYSAUX      /disk1/oracle/dbs/salespdb/salespdb_ax. f
4 SALESPDB     18 USER        /disk1/oracle/dbs/salespdb/salespdb_usr. dbf

```

例19-8 CDB内の一時ファイルの表示

この例では、CDB_TEMP_FILESビューを問い合わせ、CDB内の各一時ファイルの名前と場所、および一時ファイルを使用する表領域を表示します。

```

COLUMN CON_ID FORMAT 999
COLUMN FILE_ID FORMAT 9999
COLUMN TABLESPACE_NAME FORMAT A15
COLUMN FILE_NAME FORMAT A45

SELECT CON_ID, FILE_ID, TABLESPACE_NAME, FILE_NAME
FROM CDB_TEMP_FILES
ORDER BY CON_ID;

```

出力例:

```

CON_ID FILE_ID TABLESPACE_NAM FILE_NAME
-----
1      1 TEMP          /disk1/oracle/dbs/t_tmp1. f
2      2 TEMP          /disk1/oracle/dbs/pdbseed/t_tmp1. f
3      3 TEMP          /disk1/oracle/dbs/hrpdb/t_hrpdb_tmp1. f
4      4 TEMP          /disk1/oracle/dbs/salespdb/t_salespdb_tmp1. f

```

例19-9 PDBに関連付けられているサービスの表示

この例では、CDB_SERVICESビューを問い合わせ、PDB名、ネットワーク名、およびPDBに関連付けられている各サービスのテナIDを表示します。

```

COLUMN NETWORK_NAME FORMAT A30
COLUMN PDB FORMAT A15
COLUMN CON_ID FORMAT 999

SELECT PDB, NETWORK_NAME, CON_ID FROM CDB_SERVICES
WHERE PDB IS NOT NULL AND
      CON_ID > 2
ORDER BY PDB;

```

出力例:

```

PDB          NETWORK_NAME          CON_ID
-----
HRPDB        hrpdb.example.com      3
SALESPDB    salespdb.example.com    4

```

関連項目:

- [「現在のテナについて」](#)

- [「CDBのコンテナ・データ・オブジェクト」](#)
- コンテナ・データ・オブジェクトの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください
- [『Oracle Databaseリファレンス』](#)

親トピック: [CDBおよびPDBの監視](#)

CONTAINERS句を使用したコンテナ間の問合せ

CONTAINERS句を使用すると、CDB内のすべてのコンテナ間で表およびビューを問い合わせることができます。アプリケーション・コンテナ内のすべてのコンテナでアプリケーション共通オブジェクトを問い合わせることもできます。

- [CONTAINERS句を使用したコンテナ間の問合せについて](#)
CONTAINERS句を使用すると、CDB内のコンテナ間で問い合わせることができます。
- [すべてのコンテナ間でのユーザーが作成した表およびビューの問合せ](#)
CONTAINERS句を使用すると、すべてのコンテナ間でユーザーが作成した表およびビューを問い合わせることができます。この句により、CDB内のすべてのオープンPDBに存在する表またはビューのデータを表示するために、CDBルートからの問合せが可能になります。
- [アプリケーションPDBでのアプリケーション共通オブジェクトの問合せ](#)
CONTAINERS句では、アプリケーション・コンテナ内のすべてのPDBでアプリケーション共通オブジェクトを問い合わせることができます。アプリケーション・ルートからの問合せによって、コンテナ内のオープンしているすべてのPDBに存在するオブジェクトのデータが表示されます。

親トピック: [CDBおよびPDBの監視](#)

CONTAINERS句を使用したコンテナ間の問合せについて

CONTAINERS句を使用すると、CDB内のコンテナ間で問い合わせることができます。

CONTAINERS句を使用すると、CDB内のすべてのコンテナ間でユーザーが作成した表およびビューを問い合わせることができます。この句により、CDB内のすべてのオープン・コンテナに存在する表またはビューのデータを表示するために、CDBルートからの問合せが可能になります。

また、CONTAINERS句では、アプリケーション・コンテナ内のすべてのアプリケーションPDBで表やビューなどのアプリケーション共通オブジェクトを問い合わせることができます。この句では、アプリケーション・ルートから問い合わせ、アプリケーション・コンテナ内のオープンしているすべてのアプリケーションPDBに存在する表またはビューのデータを表示できます。

CONTAINERS句は、暗黙的に生成された次の3つの列を公開します。

- CON_ID: 行の取得元のコンテナのID。
- CON\$NAME: 行の取得元のコンテナの名前。これは非表示の列です。
- CDB\$NAME: 行の取得元のCDBの名前。プロキシPDBまたはCDBフリートがない場合、すべての行のCDB\$NAMEの値は同じになります。これは非表示の列です。

CONTAINERS句が評価される際、各コンテナはパーティションとして扱われます。したがって、CONTAINERS句を使用した問合せの計画出力にはパーティション・イテレータが含まれます。パーティション・プルーニングを使用すると、問合せの実行中にアクセスされるコンテナのセットを制限できます。プルーニング述語は、CON_ID列またはCON\$NAME列のいずれかに指定でき、どちらもCONTAINERS句に対して暗黙的に生成されます。

CONTAINERS句の評価では、パラレル実行プロセスを使用します。各コンテナがパラレル実行プロセス(P00 *)に割り当てられ、プロセスはコンテナに切り替わって実表またはビューで再帰SQL文を実行します。実表またはビューは、その名前が引数として

CONTAINERS句に渡されるオブジェクトです。

CONTAINERS_PARALLEL_DEGREE初期化パラメータでは、CONTAINERS句を含む問合せの並列度を制御できます。

CONTAINERS_PARALLEL_DEGREEの値が65535 (デフォルト)未満の場合は、指定した値が使用されます。

CONTAINERS_PARALLEL_DEGREE初期化パラメータにデフォルト値(65535)が設定されている場合、CONTAINERS句を使用する問合せはデフォルトでパラレル化されます。デフォルトの並列度は、次の式を使用して計算します。

```
max (min (cpu_count, number_of_open_containers), #instances)
```

また、DEFAULT_PDB_HINTヒントをCONTAINERS句に渡すことができます。ヒントは、各コンテナで実行される問合せで渡されます。

再帰的SQL文によってアクセスされる列は、問合せでアクセスされるCONTAINERS句の列によって決まります。CONTAINERS句を使用する問合せの述語は、再帰的SQLにプッシュ・ダウンして各コンテナ内で評価できるため、CONTAINERS句でポスト・フィルタとして処理する必要がある行数が大幅に削減されます。

CONTAINERS句を含む問合せから生成される再帰的SQLが並列化されるように強制するには、CONTAINERS句のDEFAULT_PDB_HINTヒントを使用するか、自動並列度を使用します。ただし、CONTAINERS句を含む問合せから生成される再帰的SQLには、パラレル文のキューイングは使用できません。

次のタイプの列がCONTAINERS句に指定された表に存在する場合は、削除されます。

- 次のユーザー定義型: オブジェクト型、VARRAY、REFおよびネストした表
- 次のOracle提供のタイプ: ANYTYPE、ANYDATASET、URIタイプ、SDO_TOPO_GEOMETRY、SDO_GEORASTERおよびExpression

ノート:

- コンテナが制限モードでオープンされている場合は、CONTAINERS句で無視されます。
- CONTAINERS句が使用され、エラーがコンテナで戻される場合、問合せではエラーが発生したコンテナから結果を戻さず、エラーは戻りません。たとえば、ローカル変数にリモート表からのBFILE列は選択できません。これを実行する問合せがCONTAINERS句を使用し、ローカルおよびリモート・コンテナを含む場合、問合せはローカル・コンテナの結果を戻しますが、リモート・コンテナは戻さず、エラーは戻りません。

関連項目:

- [「現在のコンテナについて」](#)
- CONTAINERS句およびCONTAINERSヒントの詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください
- コンテナ・データ・オブジェクトの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください
- CONTAINERS_PARALLEL_DEGREE初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください
- 自動並列度およびパラレル文のキューイングの詳細は、[Oracle Databaseデータ・ウェアハウス・ガイド](#)を参照してください

親トピック: [CONTAINERS句を使用したコンテナ間の問合せの概要](#)

すべてのコンテナ間でのユーザーが作成した表およびビューの問合せ

CONTAINERS句を使用すると、すべてのコンテナ間でユーザーが作成した表およびビューを問い合わせることができます。この句により、CDB内のすべてのオープンPDBに存在する表またはビューのデータを表示するために、CDBルートからの問合せが可能になります。

前提条件

CONTAINERS句に指定した表およびビュー、またはそのシノニムは、CDBルートおよび他のすべてのコンテナに存在する必要があります。

CONTAINERS句を使用してすべてのコンテナ間で表およびビューを問い合わせるには:

1. SQL*Plusでコンテナにアクセスします。
複数のコンテナのデータを表示する場合は、現在のコンテナがCDBルートであることを確認します。
[「CDB内のコンテナへのアクセスについて」](#)を参照してください。
2. CONTAINERS句を含む問合せを実行します。

例19-10 すべてのコンテナ間での共通ユーザーが所有する表の問合せ

この例では、次のことを想定しています。

- 組織に複数のPDBがあり、各PDBは組織内のそれぞれ異なる部門のものです。
- 各PDBには、部門の従業員を追跡管理するemployees表がありますが、各PDBの表には別個の従業員が含まれます。
- また、CDBルートには空のemployees表があります。
- 各コンテナのemployees表は、同じ共通ユーザーが所有します。

CDBルートを現在のコンテナとし、表を所有する共通ユーザーを現行ユーザーとして、CONTAINERS句を指定した次の問合せを実行し、すべてのPDBのemployees表の全従業員を返します。

```
SELECT * FROM CONTAINERS (employees);
```

例19-11 すべてのコンテナ間でのローカル・ユーザーが所有する表の問合せ

この例では、次のことを想定しています。

- 組織に複数のPDBがあり、各PDBは組織内のそれぞれ異なる部門のものです。
- 各PDBには、部門の従業員を追跡管理するhr. employees表がありますが、各PDBの表には別個の従業員が含まれます。
- また、CDBルートには、共通ユーザーが所有する空のemployees表があります。

すべてのPDBの全従業員を返す問合せを実行するには、最初に共通ユーザーとして各PDBに接続し、次の文を使用してビューを作成します。

```
CREATE OR REPLACE VIEW employees AS SELECT * FROM hr. employees;
```

ビューを所有する共通ユーザーは、CDBルートのemployees表を所有する共通ユーザーと同じである必要があります。各PDBでこの文を実行すると、共通ユーザーは各PDBにemployeesという名前のビューを保持します。

CDBルートを現在のコンテナとし、共通ユーザーを現行ユーザーとして、CONTAINERS句を指定した次の問合せを実行し、すべてのPDBのhr. employees表の全従業員を返します。

```
SELECT * FROM CONTAINERS (employees);
```

特定のコンテナのビューを問い合わせることもできます。たとえば、次のSQL文では、CON_IDが3および4のコンテナのビューを問い合わせます。

```
SELECT * FROM CONTAINERS (employees) WHERE CON_ID IN (3, 4);
```

ノート:



CONTAINERS 句を使用してOracle社提供の表およびビューを問い合わせることもできます。この問合せを実行する際、現行ユーザーが表またはビューの所有者であることを確認するか、CONTAINERS 句を使用してビューを作成し、そのビューに対する SELECT 権限を適切なユーザーに付与します。

関連項目:

- [「現在のコンテナについて」](#)
- [「CDBのコンテナ・データ・オブジェクト」](#)
- CONTAINERS句の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください
- コンテナ・データ・オブジェクトの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください

親トピック: [CONTAINERS句を使用したコンテナ間の問合せの概要](#)

アプリケーションPDB間のアプリケーション共通オブジェクトの問合せ

CONTAINERS句では、アプリケーション・コンテナ内のすべてのPDBでアプリケーション共通オブジェクトを問い合わせることができます。アプリケーション・ルートからの問合せによって、コンテナ内のオープンしているすべてのPDBに存在するオブジェクトのデータが表示されます。

CONTAINERS句は、メタデータリンク・アプリケーション共通オブジェクトに最も役立ちます。メタデータリンク・アプリケーション共通オブジェクトでは、構造はアプリケーション・コンテナのすべてのコンテナで同じですが、データは異なります。CONTAINERS句を使用して、複数のアプリケーションPDB内のメタデータリンク・アプリケーション共通オブジェクトのデータを表示できます。拡張データリンク・オブジェクトについてのメリットも同様です。CONTAINERS句は、パラレル実行を使用して、アプリケーション・ルートでホストされる各アプリケーションPDB間で問合せを実行します。

CONTAINERS句を使用してすべてのアプリケーションPDB間で表およびビューを問い合わせるには:

1. SQL*Plusで、アプリケーション・ルートにアクセスします。
[「CDB内のコンテナへのアクセスについて」](#)を参照してください。
2. CONTAINERS句を含む問合せを実行します。

ノート:



アプリケーション・ルートの表またはビューに対して CONTAINERS_DEFAULT 属性を有効にできます。この属性が有効になっている場合、CONTAINERS 句はデフォルトでデータベース・オブジェクトに対する問合せおよび DML 文に使用さ

れ、SQL 文に CONTAINERS 句は必要ありません。アプリケーション・ルートの表またはビューに対して CONTAINERS_DEFAULT 属性を有効にするには、ENABLE CONTAINERS_DEFAULT 句を指定した ALTER VIEW または CREATE OR REPLACE VIEW 文を実行します。

例19-12 すべてのアプリケーションPDBでのアプリケーション共通オブジェクトの問合せ

この例では、次のことを想定しています。

- 組織に複数のアプリケーションPDBがあり、各アプリケーションPDBは組織内のそれぞれ異なる部門のものです。
- 各アプリケーションPDBには、部門の従業員を追跡管理するemployees表がありますが、各アプリケーションPDBの表には別個の従業員が含まれます。
- また、アプリケーション・ルートには空のemployees表があります。
- 各コンテナのemployees表は、同じ共通ユーザーが所有します。
- 会社には、アプリケーション・コンテナ内のアプリケーションを使用する複数のテナントがあり、各テナントに固有のアプリケーションPDBがあります。
- 会社は、メタデータリンク・アプリケーション共通オブジェクトを使用して、すべてのアプリケーションPDB内で同じデータの構造を維持しますが、データはアプリケーションPDBごとに異なります。
- 各アプリケーションPDBには、各テナントの顧客に関する情報を格納するメタデータリンクsales.customers表がありません。

アプリケーション・ルートを現在のコンテナとし、表を所有するアプリケーション共通ユーザーを現行ユーザーとして、CONTAINERS句を指定した次の問合せを実行し、すべてのアプリケーションPDBのsales.customers表の全顧客を返します。

```
SELECT * FROM CONTAINERS(sales.customers);
```

関連項目:

- [「アプリケーション共通オブジェクトについて」](#)
- [「現在のコンテナについて」](#)
- CONTAINERS句の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください
- コンテナ・データ・オブジェクトの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください

親トピック: [CONTAINERS句を使用したコンテナ間の問合せの概要](#)

現在のコンテナのIDまたは名前の判別

CDB内の現在のコンテナIDまたはコンテナ名を判別できます。

現在のコンテナIDを判別するには:

- 次のSQL*Plusコマンドを実行します。

```
SHOW CON_ID
```

現在のコンテナ名を判別するには:

- 次のSQL*Plusコマンドを実行します。

```
SHOW CON_NAME
```

また、[表19-3](#)に示されているファンクションを使用して、コンテナのコンテナID、コンテナ名およびDBIDを判別することもできます。

表19-3 コンテナ情報を返すファンクション

ファンクション	説明
CON_NAME_TO_ID('container_name')	コンテナの名前に基づいてコンテナ ID を返します。
CON_DBID_TO_ID(container_dbid)	コンテナの DBID に基づいてコンテナ ID を返します。
CON_UID_TO_ID(container_uid)	コンテナの一意識別子(UID)に基づいてコンテナ ID を返します。
CON_GUID_TO_ID(container_guid)	コンテナのグローバル一意識別子(GUID)に基づいてコンテナ ID を返します。
CON_ID_TO_CON_NAME(container_id)	コンテナ ID に基づいてコンテナの名前を返します。
CON_ID_TO_DBID(container_id)	コンテナ ID に基づいてコンテナの DBID を返します。

V\$CONTAINERSビューには、CDB内の各コンテナの名前、DBID、UIDおよびGUIDが表示されます。

例19-13 コンテナの名前に基づいたコンテナIDの戻し

```
SELECT CON_NAME_TO_ID('HRPDB') FROM DUAL;
```

例19-14 コンテナDBIDに基づいたコンテナIDの戻し

```
SELECT CON_DBID_TO_ID(2226957846) FROM DUAL;
```

例19-15 コンテナIDに基づいたコンテナの名前の戻し

```
SELECT CON_ID_TO_CON_NAME(4) FROM DUAL;
```

関連項目:

- [「マルチテナント環境について」](#)
- [「現在のコンテナについて」](#)
- [「CDB内のコンテナに関する情報の表示」](#)
- V\$CONTAINERSビューの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

親トピック: [CDBおよびPDBの監視](#)

PDBの変更可能な初期化パラメータをリスト

CDBでは、一部の初期化パラメータがルートおよびすべてのPDBに適用されます。このような初期化パラメータが変更されると、CDB全体に影響を与えます。他の初期化パラメータは、各コンテナで異なる値に設定できます。

たとえば、1つのパラメータについて、ルートで1つの値を設定し、あるPDBで別の値を設定し、2番目のPDBではさらに別の値を設定できます。

この項で示す問合せによって、各PDBで個別に設定できる初期化パラメータがリストされます。

各コンテナで変更可能な初期化パラメータをリストするには:

1. SQL*Plusでコンテナにアクセスします。

[「CDB内のコンテナへのアクセスについて」](#)を参照してください。

2. 次の問合せを実行します。

```
SELECT NAME FROM V$SYSTEM_PARAMETER
WHERE ISPDB_MODIFIABLE = 'TRUE'
ORDER BY NAME;
```

この問合せでリストされた初期化パラメータがPDBに個別に設定されていない場合、PDBではルートのパラメータ値が継承されます。

関連項目:

- [「ALTER SYSTEMを使用したCDBの変更」](#)
- [「システム・レベルでのPDBの変更」](#)

親トピック: [CDBおよびPDBの監視](#)

PDBの履歴の表示

CDB_PDB_HISTORYビューには、CDB内のPDBの履歴が表示されます。各PDBがいつどのように作成されたかに関する情報、および各PDBの履歴に関するその他の情報を提供します。

各PDBの履歴を表示するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。

[「SQL*Plusを使用したCDB内のコンテナへのアクセス」](#)を参照してください。

2. CDB_PDB_HISTORYビューを問い合わせます。

例19-16 PDBの履歴の表示

この例では、各PDBの履歴に関する次の情報を表示します。

- DB_NAMEフィールドには、PDBが含まれているCDBが表示されます。
- CON_IDフィールドには、PDBのコンテナIDが表示されます。
- PDB_NAMEフィールドには、PDBのいずれかのインカネーションでのPDBの名前が表示されます。
- OPERATIONフィールドには、PDBの履歴で実行された操作が表示されます。
- OP_TIMESTAMPフィールドには、操作が実行された日付が表示されます。
- PDBが操作でクローニングされた場合、CLONED_FROM_PDBフィールドにはPDBのクローニング元となるPDBが表示されます。

```
COLUMN DB_NAME FORMAT A10
COLUMN CON_ID FORMAT 999
COLUMN PDB_NAME FORMAT A15
COLUMN OPERATION FORMAT A16
```

```
COLUMN OP_TIMESTAMP FORMAT A10  
COLUMN CLONED_FROM_PDB_NAME FORMAT A15
```

```
SELECT DB_NAME, CON_ID, PDB_NAME, OPERATION, OP_TIMESTAMP, CLONED_FROM_PDB_NAME  
FROM CDB_PDB_HISTORY  
WHERE CON_ID > 2  
ORDER BY CON_ID;
```

出力例:

DB_NAME	CON_ID	PDB_NAME	OPERATION	OP_TIMESTAMP	CLONED_FROM_PDB
NEWCDB	3	HRPDB	CREATE	10-APR-12	PDB\$SEED
NEWCDB	4	SALESPDB	CREATE	17-APR-12	PDB\$SEED
NEWCDB	5	TESTPDB	CLONE	30-APR-12	SALESPDB

ノート:



現在のコンテナが PDB である場合、CDB_PDB_HISTORY ビューには、現在の PDB の履歴のみが表示されます。現在のコンテナが PDB であるローカル・ユーザーは、DBA_PDB_HISTORY ビューを問い合わせ、問合せから CON_ID 列を除外して、現在の PDB の履歴を表示できます。

関連項目:

[「現在のコンテナについて」](#)

親トピック: [CDBおよびPDBの監視](#)

アプリケーション・コンテナ内のアプリケーションに関する情報の表示

複数のビューが、CDB内のアプリケーション・コンテナ内のアプリケーションに関する情報を提供します。

- [アプリケーションに関する情報の表示](#)
DBA_APPLICATIONSビューは、アプリケーション・コンテナ内のアプリケーションに関する情報を提供します。
- [アプリケーション・ステータスに関する情報の表示](#)
DBA_APP_PDB_STATUSビューは、アプリケーション・コンテナ内のアプリケーションのステータスに関する情報を提供します。各アプリケーションPDB内の各アプリケーションのステータスを表示できます。
- [アプリケーション文に関する情報の表示](#)
DBA_APP_STATEMENTSビューは、アプリケーションのインストール、アップグレードおよびパッチ操作中に発行されたSQL文に関する情報を提供します。
- [アプリケーションのバージョンに関する情報の表示](#)
DBA_APP_VERSIONSビューは、アプリケーション・コンテナ内のアプリケーションのバージョンに関する情報を提供します。
- [アプリケーション・パッチに関する情報の表示](#)
DBA_APP_PATCHESビューは、アプリケーション・コンテナ内のアプリケーション・パッチに関する情報を提供します。
- [アプリケーション・エラーに関する情報の表示](#)
DBA_APP_ERRORSビューは、アプリケーションPDBがアプリケーション・ルート内のアプリケーションと同期する際に発生した情報エラーを提供します。

- [アプリケーション・コンテナ内の共有データベース・オブジェクトのリスト](#)
DBA_OBJECTSビューは、アプリケーション・コンテナ内の共有データベース・オブジェクトをリストできます。
- [アプリケーション・コンテナ内の拡張データリンク・オブジェクトのリスト](#)
DBA_TABLESおよびDBA_VIEWSビューは、アプリケーション・コンテナ内の拡張データリンク・オブジェクトをリストできます。

関連トピック

- [アプリケーション・コンテナの作成および削除](#)
- [アプリケーション・コンテナの管理](#)

親トピック: [CDBおよびPDBの監視](#)

アプリケーションに関する情報の表示

DBA_APPLICATIONSビューは、アプリケーション・コンテナ内のアプリケーションに関する情報を提供します。

ノート:



DBA_APPLICATIONS ビューは、現在のコンテナのみのアプリケーションに関する情報を提供します。現在のアプリケーション・コンテナ内のすべてのアプリケーション PDB 内のアプリケーションに関する情報を表示するには、アプリケーション・ルートが現在のコンテナであるときに、DBA_APP_PDB_STATUS に問い合わせます。

アプリケーション・コンテナ内のアプリケーションに関する情報を提供するには:

1. SQL*Plusで、アプリケーション・コンテナのアプリケーション・ルートにアクセスします。
2. DBA_APPLICATIONSビューを問い合わせます。

例19-17 アプリケーション・コンテナのアプリケーションに関する詳細の表示

この問合せは、アプリケーション・コンテナ内のユーザーが作成した各アプリケーションの名前、最新バージョンおよびステータスを表示します。

```
COLUMN APP_NAME FORMAT A15
COLUMN APP_VERSION FORMAT A15
COLUMN APP_STATUS FORMAT A15

SELECT APP_NAME, APP_VERSION, APP_STATUS
FROM DBA_APPLICATIONS
WHERE APP_IMPLICIT=' N' ;
```

次のサンプル出力は、salesappアプリケーションを示しています。

APP_NAME	APP_VERSION	APP_STATUS
SALESAPP	1.2	NORMAL

ノート:



Oracle Database は、アプリケーション共通ユーザー操作が ALTER PLUGGABLE DATABASE APPLICATION BEGIN/END 文の外部の CONTAINER=ALL 句で発行されたときにいくつかのアプリケーションを暗黙的に作成します。サンプル問合せでは、WHERE 句に APP_IMPLICIT=' N' を指定することで暗黙的に作成されたアプリケーションが除外されます。

関連トピック

- [アプリケーション・コンテナの管理](#)
- [アプリケーションPDB内のアプリケーションの同期](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)

アプリケーション・ステータスに関する情報の表示

DBA_APP_PDB_STATUSビューは、アプリケーション・コンテナ内のアプリケーションのステータスに関する情報を提供します。各アプリケーションPDB内の各アプリケーションのステータスを表示できます。

このビューは、アプリケーションPDBがクローズされている場合でも、アプリケーションPDB内のアプリケーションのステータスを表示できます。

ノート:



アプリケーション・ルートから問い合わせた場合、DBA_APP_PDB_STATUS ビューは現在のアプリケーション・コンテナ内のすべてのアプリケーション PDB 内のアプリケーションに関する情報を提供します。現在のコンテナのアプリケーションに関する情報のみを表示するには、DBA_APPLICATIONS ビューを問い合わせます。

アプリケーション・コンテナ内のアプリケーション・ステータスに関する情報を表示するには:

1. SQL*Plusで、アプリケーション・コンテナのアプリケーション・ルートにアクセスします。

[「CDB内のコンテナへのアクセスについて」](#)を参照してください。

2. DBA_APP_PDB_STATUSビューを問い合わせます。

例19-18 アプリケーション・ステータスに関する情報の表示

この問合せは、アプリケーションPDBの名前、アプリケーションの名前、アプリケーションのバージョン番号およびアプリケーションのステータスを表示します。

```
COLUMN PDB_NAME FORMAT A15
COLUMN APP_NAME FORMAT A15
COLUMN APP_VERSION FORMAT A20
COLUMN APP_STATUS FORMAT A12

SELECT p. PDB_NAME, s. APP_NAME, s. APP_VERSION, s. APP_STATUS
FROM DBA_PDBS p, DBA_APP_PDB_STATUS s
WHERE p. CON_UID = s. CON_UID;
```

出力は次のようになります。

PDB_NAME	APP_NAME	APP_VERSION	APP_STATUS
SALES1	SALESAPP	4.2	NORMAL

ノート:



アプリケーションのステータスは、アプリケーションが最新のバージョンと同期されていなくても、アプリケーション PDB 内

で NORMAL であることがあります。他のステータスは、操作が進行中であるか、操作で問題が発生したことを示していることがあります。たとえば、UPGRADING ステータスは、アプリケーションのアップグレードがアプリケーション PDB で進行中であるか、アプリケーション PDB がアプリケーションをアップグレードしようとしたときにエラーが発生したことを示している可能性があります。

関連項目:

[「アプリケーション・コンテナの管理」](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)

アプリケーション文に関する情報の表示

DBA_APP_STATEMENTSビューは、アプリケーションのインストール、アップグレードおよびパッチ操作中に発行されたSQL文に関する情報を提供します

Oracle Databaseでは、アプリケーションのインストール、アップグレードおよびパッチ操作中に発行されたすべてのSQL文が記録され、DBA_APP_STATEMENTSビューを問い合わせることでこれらの文の履歴を表示できます。

アプリケーション操作中に発行されたSQL文に関する情報を表示するには:

1. SQL*Plusで、アプリケーション・コンテナのアプリケーション・ルートにアクセスします。

[「CDB内のコンテナへのアクセスについて」](#)を参照してください。

2. DBA_APP_STATEMENTSビューを問い合わせます。

例19-19 アプリケーション文に関する情報の表示

この問合せは、アプリケーション・コンテナ内のアプリケーションに対するSQL文の文ID、取得時間、SQL文およびアプリケーション名を表示します。

```
SET LONG 8000
SET PAGES 8000
COLUMN STATEM_ID FORMAT NNNNN
COLUMN CAPTURE_TIME FORMAT A12
COLUMN APP_STATEMENT FORMAT A36
COLUMN APP_NAME FORMAT A15

SELECT STATEMENT_ID AS STATEM_ID, CAPTURE_TIME, APP_STATEMENT, APP_NAME
FROM DBA_APP_STATEMENTS
ORDER BY STATEMENT_ID;
```

出力は次のようになります。

STATEM_ID	CAPTURE_TIME	APP_STATEMENT	APP_NAME
1	30-AUG-15	SYS	APP\$1E87C094764 1142FE0534018F8 0AA6C5
2	30-AUG-15	ALTER PLUGGABLE DATABASE APPLICATION APP\$CON BEGIN INSTALL '1.0'	APP\$1E87C094764 1142FE0534018F8 0AA6C5
3	30-AUG-15	ALTER PLUGGABLE DATABASE APPLICATION APP\$CON END INSTALL '1.0'	APP\$1E87C094764 1142FE0534018F8


```

OAA6C5
4 30-AUG-15  SYS                                SALESAPP
5 30-AUG-15  ALTER PLUGGABLE DATABASE APPLICATION SALESAPP
   salesapp BEGIN INSTALL '1.0'
6 30-AUG-15  CREATE TABLE oe.cmtb SHARING=METADAT SALESAPP
   A (
       value VARCHAR2(30),
       country VARCHAR2(30))
7 30-AUG-15  CREATE TABLE conmap (                                SALESAPP
       country VARCHAR2(30) NOT NULL)
   PARTITION BY LIST (country) (
   PARTITION AMER VALUES ('US', 'MEXICO'
   , 'CANADA'),
   PARTITION EURO VALUES ('UK', 'FRANCE'
   , 'GERMANY'),
   PARTITION ASIA VALUES ('INDIA', 'CHIN
   A', 'JAPAN'))
8 30-AUG-15  ALTER TABLE oe.cmtb ENABLE CONTAINER SALESAPP
   _MAP
9 30-AUG-15  ALTER PLUGGABLE DATABASE APPLICATION SALESAPP
   salesapp END INSTALL '1.0'
.
.
.

```

ノート:



Oracle Database は、アプリケーション共通ユーザー操作が ALTER PLUGGABLE DATABASE APPLICATION BEGIN/END 文の外部の CONTAINER=ALL 句で発行されたときにいくつかのアプリケーションを暗黙的に作成します。これらのアプリケーションの名前は APP\$ で開始し、サンプル出力はこれらのアプリケーションを表示します。

関連項目:

- [「アプリケーション・コンテナの管理」](#)
- [「アプリケーションPDB内のアプリケーションの同期」](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)

アプリケーション・バージョンに関する情報の表示

DBA_APP_VERSIONSビューは、アプリケーション・コンテナ内のアプリケーションのバージョンに関する情報を提供します。

Oracle Databaseは、アプリケーション・コンテナ内の各アプリケーションのバージョンを記録します。

アプリケーション・コンテナ内のアプリケーション・バージョンに関する情報を表示するには:

1. SQL*Plusで、アプリケーション・コンテナのアプリケーション・ルートにアクセスします。

[「CDB内のコンテナへのアクセスについて」](#)を参照してください。

2. DBA_APP_VERSIONSビューを問い合わせます。

例19-20 アプリケーション・バージョンに関する情報の表示

この問合せは、バージョン化されているアプリケーションの名前、バージョン番号およびバージョンのコメントを表示します。

```
COLUMN APP_NAME FORMAT A15
COLUMN APP_VERSION FORMAT A20
COLUMN APP_VERSION_COMMENT FORMAT A25

SELECT APP_NAME, APP_VERSION, APP_VERSION_COMMENT
FROM DBA_APP_VERSIONS;
```

出力は次のようになります。

APP_NAME	APP_VERSION	APP_VERSION_COMMENT
SALESAPP	1.0	Sales Application

関連項目:

[「アプリケーション・コンテナの管理」](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)

アプリケーション・パッチに関する情報の表示

DBA_APP_PATCHESビューは、アプリケーション・コンテナ内のアプリケーション・パッチに関する情報を提供します。

Oracle Databaseは、アプリケーション・コンテナ内の各アプリケーションのパッチを記録します。

アプリケーション・コンテナ内のアプリケーション・パッチに関する情報を表示するには:

1. SQL*Plusで、アプリケーション・コンテナのアプリケーション・ルートにアクセスします。

[「CDB内のコンテナへのアクセスについて」](#)を参照してください。

2. DBA_APP_PATCHESビューを問い合わせます。

例19-21 アプリケーション・パッチに関する情報の表示

この問合せは、アプリケーション・コンテナ内の各パッチについて、パッチが適用されたアプリケーションの名前、パッチ番号、パッチの最小アプリケーション・バージョンおよびパッチのステータスを表示します。

```
COLUMN APP_NAME FORMAT A15
COLUMN PATCH_NUMBER FORMAT NNNNNNNN
COLUMN PATCH_MIN_VERSION FORMAT A10
COLUMN PATCH_STATUS FORMAT A15

SELECT APP_NAME, PATCH_NUMBER, PATCH_MIN_VERSION, PATCH_STATUS
FROM DBA_APP_PATCHES;
```

出力は次のようになります。

APP_NAME	PATCH_NUMBER	PATCH_MIN_	PATCH_STATUS
SALESAPP	1	1.2	INSTALLED

関連項目:

親トピック: [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)

アプリケーション・エラーに関する情報の表示

DBA_APP_ERRORSビューは、アプリケーションPDBがアプリケーション・ルート内のアプリケーションと同期する際に発生した情報エラーを提供します。

アプリケーションPDBは、SYNC句を指定したALTER PLUGGABLE DATABASE APPLICATION文を発行します。DBA_APP_ERRORSビューを問い合わせることで、アプリケーション同期中のアプリケーション・エラーの履歴を表示できます。

アプリケーションの同期中に発生したエラーに関する情報を表示するには:

1. SQL*Plusで、アプリケーション・コンテナのアプリケーション・ルートにアクセスします。
[「CDB内のコンテナへのアクセスについて」](#)を参照してください。
2. DBA_APP_ERRORSビューを問い合わせます。

例19-22 アプリケーション同期中に発生したエラーに関する詳細の表示

この問合せは、アプリケーション同期中に発生したエラーについて、アプリケーション名、エラーを生成したSQL文、エラー番号およびエラー・メッセージを表示します。

```
SET LONG 8000
SET PAGES 8000
COLUMN APP_NAME FORMAT A15
COLUMN APP_STATEMENT FORMAT A36
COLUMN ERRORNUM FORMAT NNNNNNNN
COLUMN ERRORMSG FORMAT A20

SELECT APP_NAME, APP_STATEMENT, ERRORNUM, ERRORMSG
FROM DBA_APP_ERRORS;
```

関連項目:

[「アプリケーション・コンテナの管理」](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)

アプリケーション・コンテナ内の共有データベース・オブジェクトのリスト

DBA_OBJECTSビューは、アプリケーション・コンテナ内の共有データベース・オブジェクトをリストできます。

共有データベース・オブジェクトは、メタデータリンク・アプリケーション共通オブジェクト、データリンク・アプリケーション共通オブジェクトおよび拡張データリンク・アプリケーション共通オブジェクトです。

アプリケーション・コンテナ内の共有データベース・オブジェクトをリストするには:

1. SQL*Plusで、アプリケーション・コンテナのアプリケーション・ルートにアクセスします。
[「CDB内のコンテナへのアクセスについて」](#)を参照してください。
2. DBA_OBJECTSビューを問い合わせ、SHARING列を指定します。

例19-23 アプリケーション・コンテナ内のユーザーが作成した共有データベース・オブジェクトのリスト

この問合せは、アプリケーション・コンテナ内のユーザーが作成した共有データベース・オブジェクトの所有者と名前を表示します。各共有データベース・オブジェクトがメタデータリンク・アプリケーション共通オブジェクトとデータリンク・アプリケーション共通オブジェクトのどちらであるかも表示します。問合せでは、オラクル社が提供する共有データベース・オブジェクトは除外されます。

```
COLUMN OWNER FORMAT A15
COLUMN OBJECT_NAME FORMAT A25
COLUMN SHARING FORMAT A13

SELECT OWNER, OBJECT_NAME, SHARING
       FROM DBA_OBJECTS WHERE SHARING != 'NONE'
       AND ORACLE_MAINTAINED = 'N' ;
```

出力は次のようになります。

OWNER	OBJECT_NAME	SHARING
SALESADM	CONMAP	METADATA LINK
OE	PRODUCT_DESCRIPTIONS_OB	DATA LINK
OE	CMTB	METADATA LINK

関連項目:

[アプリケーション共通オブジェクトの管理](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)

アプリケーション・コンテナ内の拡張データリンク・オブジェクトのリスト

DBA_TABLESおよびDBA_VIEWSビューは、アプリケーション・コンテナ内の拡張データリンク・オブジェクトをリストできます。

拡張データリンク・オブジェクトは、各アプリケーションPDBがアプリケーション・ルートの共通データを共有しながら特定の独自データを作成できる特殊なタイプのデータリンク・オブジェクトです。アプリケーション・ルートに格納されるデータのみがすべてのアプリケーションPDBに共通です

アプリケーション・コンテナ内の拡張データリンク・オブジェクトをリストするには:

1. SQL*Plusで、アプリケーション・コンテナのアプリケーション・ルートにアクセスします。

[「CDB内のコンテナへのアクセスについて」](#)を参照してください。

2. DBA_TABLESまたはDBA_VIEWSビューを問い合わせ、WHERE句にEXTENDED_DATA_LINK='YES'を指定します。

例19-24 アプリケーション・コンテナ内の拡張データリンク表の表示

この問合せは、アプリケーション・コンテナ内の拡張データリンク表の所有者と名前を表示します。

```
COLUMN OWNER FORMAT A20
COLUMN TABLE_NAME FORMAT A30

SELECT OWNER, TABLE_NAME FROM DBA_TABLES WHERE EXTENDED_DATA_LINK='YES' ;
```

出力は次のようになります。

OWNER	TABLE_NAME
-------	------------

関連項目:

[アプリケーション共通オブジェクトの管理](#)

親トピック: [アプリケーション・コンテナ内のアプリケーションに関する情報の表示](#)

第V部 マルチテナント環境でのOracle機能の使用

マルチテナント環境でOracle Databaseの機能を使用できます。

- [CDBとPDBのバックアップおよびリカバリ](#)
マルチテナント・コンテナ・データベース(CDB)およびプラグブル・データベース(PDB)をバックアップおよびリカバリできます。
- [マルチテナント環境でのデータベース・ユーティリティの使用](#)
マルチテナント環境で、Oracle Data Pump、DBNEWID、Oracle LogMinerなどのユーティリティを使用できます。
- [PDBに対するOracle Resource Managerの使用](#)
PL/SQLパッケージ・プロシージャを使用して、Oracle Resource Manager (リソース・マネージャ)によるマルチテナント・コンテナ・データベース(CDB)内のプラグブル・データベース(PDB)へのリソース割当てを管理します。
- [CDBでのOracle Schedulerの使用](#)
Oracle Schedulerを使用してマルチテナント・コンテナ・データベース(CDB)のジョブをスケジュールできます。
- [CDBでのOracle Database Vaultの使用](#)
マルチテナント・コンテナ・データベース(CDB)でOracle Database Vaultを使用できます。
- [CDBでのXStreamの使用](#)
マルチテナント・コンテナ・データベース(CDB)でOracle Database XStreamを使用できます。

20 CDBとPDBのバックアップおよびリカバリ

マルチテナント・コンテナ・データベース(CDB)およびプラグブル・データベース(PDB)をバックアップおよびリカバリできます。

- [CDBとPDBのバックアップおよびリカバリの概要](#)
マルチテナント・アーキテクチャを使用すると、マルチテナント・コンテナ・データベース(CDB)全体、ルート、または1つ以上のプラグブル・データベース(PDB)に対してバックアップおよびリカバリ操作を実行できます。
- [CDBのバックアップおよび完全リカバリ](#)
マルチテナント・コンテナ・データベース(CDB)全体に対するバックアップと完全リカバリ操作を実行するには、TARGETとしてルートに接続します。
- [PDBのバックアップおよび完全リカバリ](#)
バックアップと完全リカバリ操作は、単一のプラグブル・データベース(PDB)または複数のPDBに対して実行できます。
- [マルチテナント環境でのPoint-in-Timeリカバリ](#)
マルチテナント・コンテナ・データベース(CDB)全体または特定のプラグブル・データベース(PDB)のPoint-in-Timeリカバリを実行できます。
- [マルチテナント環境でのフラッシュバック・データベース](#)
マルチテナント・コンテナ・データベース(CDB)全体または特定のプラグブル(PDB)に対して、フラッシュバック・データベース操作を実行できます。

親トピック: [マルチテナント環境でのOracle機能の使用](#)

CDBとPDBのバックアップおよびリカバリの概要

マルチテナント・アーキテクチャを使用すると、マルチテナント・コンテナ・データベース(CDB)全体、ルート、または1つ以上のプラグブル・データベース(PDB)に対してバックアップおよびリカバリ操作を実行できます。

CDBとPDBのバックアップおよびリカバリで使用するOracle Recovery Manager (RMAN)コマンドは、非CDBで使用するコマンドと同じです(ただし、構文にわずかな違いがあります)。非CDBで実行するバックアップおよびリカバリ操作は、CDBとPDBでも実行可能です。次のものが必要となります。

- 全体バックアップと増分バックアップ
- 完全なPoint-in-Timeリカバリ(PITR)
- フラッシュバック・データベース
- レポート作成操作(バックアップのリスト作成およびバックアップの照合チェック)

CDBおよびPDBへの接続について

ルートには、次のいずれかの方法で接続できます。

- オペレーティング・システム認証を使用した接続
SYSDBA権限を持つSYSユーザーとしてルートに接続されます。
- 共通ユーザーとしてのローカル接続
- 共通ユーザーとしてのOracle Net Servicesを介した接続

TARGETとしてPDBに接続するには、次のいずれかの手法を使用します。

- PDBのデータベース・サービスに変換されるネット・サービス名を使用して接続します
- SYSDBA権限またはSYSBACKUP権限を持つ共通ユーザーまたはローカル・ユーザーとしてローカルに接続します

ノート:



PDB に直接接続している場合には、一部の操作は使用できません。これらの操作のリストは、[『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』](#)を参照してください。

関連項目:

PDBのバックアップおよびリカバリの詳細は、『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』で次の項を参照してください。

- [ターゲットとしてのrootへの接続](#)
- [ターゲットとしてのPDBへの接続](#)

親トピック: [CDBとPDBのバックアップおよびリカバリ](#)

CDBのバックアップおよび完全リカバリ

マルチテナント・コンテナ・データベース(CDB)全体に対するバックアップと完全リカバリ操作を実行するには、TARGETとしてルートに接続します。

SYSDBA権限またはSYSBACKUP権限を持つ共通ユーザーとして接続を確立する必要があります。

ルートに接続したら、非CDBに対する操作の実行に使用するものと同じコマンドを使用して、CDB全体のバックアップと完全リカバリを実行します。

関連項目:

CDBのバックアップと完全リカバリの実行の詳細は、『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』の次の項を参照してください。

- [「CDB全体のバックアップ」](#)
- [「CDB全体の完全リカバリの実行」](#)
- [「CDB全体の検証」](#)
- [「CDBのレポート」](#)

親トピック: [CDBとPDBのバックアップおよびリカバリ](#)

PDBのバックアップおよび完全リカバリ

バックアップと完全リカバリ操作は、単一のプラガブル・データベース(PDB)または複数のPDBに対して実行できます。

PDBのバックアップ

PDBを再配置したり非CDBをPDBとしてクローニングする場合、プリプラグイン・バックアップ使用の維持が必要になる場合があ

ります。宛先CDBでプリプラグイン・バックアップを使用可能にするには、プリプラグイン・バックアップに関するメタデータが宛先CDBのRMANリポジトリにエクスポートされる必要があります。

バックアップを使用可能にする方法は、操作のタイプによって異なります。

- 非CDBのクローニングによるPDBの作成

非CDBが読み取り/書き込みモードでオープンされている場合は、クローニングの前の最後のステップとして DBMS_PDB.EXPORTRMANBACKUP プロシージャを実行する必要があります。非CDBをPDBとして宛先CDBに接続するとき、ソース非CDBのバックアップ・メタデータが、宛先CDBのデータ・ディクショナリにコピーされます。

- 別のCDBへのPDBの再配置

ソースPDBを切断するとき、バックアップ・メタデータは自動的にエクスポートされます。したがって、DBMS_PDB.EXPORTRMANBACKUPを実行する必要はありません。

プリプラグイン・バックアップは、ソース非CDBまたはPDBを接続する宛先CDBでのみ使用できます。

ノート:



- プリプラグイン・バックアップについて学習するには、[『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』](#)を参照してください
- データベース全体のプリプラグイン・バックアップを作成する方法を学習するには、[『Oracle Database バックアップおよびリカバリ・ユーザーズ・ガイド』](#)を参照してください
- DBMS_PDB.EXPORTRMANBACKUP プロシージャについてさらに学習するには、[『Oracle Database PL/SQL パッケージおよびタイプ・リファレンス』](#)を参照してください

バックアップ・コマンドの構文

Oracle Recovery Manager (RMAN)コマンドは同じですが、複数のPDBに対する操作の実行で使用する構文は一部変更します。

単一のPDBに対するバックアップと完全リカバリ操作を実行するには、TARGETとして次のいずれかのコンテナに接続します。

- PDB

この場合は、非CDBのバックアップまたはリカバリで使用するのと同じコマンドを使用します。たとえば、PDBをバックアップするには、BACKUP DATABASEコマンドを使用します。

- CDB\$ROOT

この場合は、RMANコマンドでPLUGGABLE DATABASE句を使用します。次のコマンドでは、ルートに接続されている場合にhrpdbというPDBをバックアップします。

```
BACKUP PLUGGABLE DATABASE hrpdb;
```

単一のコマンドを使用して複数のPDBに対するバックアップと完全リカバリ操作を実行するには、ルートに接続する必要があります。PLUGGABLE DATABASE句の後に、操作を実行する対象のPDBのリストを指定して使用します。次の例では、ルートに接続されている場合にhrpdb、salespdbおよびinvpdbの各PDBをバックアップします。

```
BACKUP PLUGGABLE DATABASE hrpdb, salespdb, invpdb;
```

関連項目:

PDBのバックアップおよびリカバリの詳細は、『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』で次の項を参照してください。

- [「ターゲットとしてのPDBへの接続」](#)
- [「ターゲットとしてのrootへの接続」](#)
- [「RMANを使用したPDBのバックアップ」](#)
- [「RMANによるPDBの完全リカバリの実行」](#)
- [「PDBの検証」](#)
- [「PDBのレポート」](#)

親トピック: [CDBとPDBのバックアップおよびリカバリ](#)

マルチテナント環境でのPoint-in-Timeリカバリ

マルチテナント・コンテナ・データベース(CDB)全体または特定のプラグブル・データベース(PDB)のPoint-in-Timeリカバリを実行できます。

CDBのPoint-in-Timeリカバリ

CDBのPoint-in-Timeリカバリを実行するには、次の前提条件を満たしている必要があります。

- SYSDBAまたはSYSBACKUP権限を持つ共通ユーザーとしてルート・コンテナにログインする必要があります。
- CDBはマウントされている必要があります。

リカバリ操作を実行するときは、非CDBで使用するのと同じコマンドを使用します。

PDBのPoint-in-Timeリカバリ

オープンまたはクローズのCDBでPDBがクローズされたときには、PDBを過去の時点にリカバリできます。その手法は、CDBのUNDOモードによって異なります。次の表で、その違いを説明します。

表20-1 Point-in-Timeリカバリ方法の違い

UNDOモード	補助インスタンスを使用するか	TARGETとして接続する場所	リカバリに使用するRMANコマンド
共有	はい	CDB ルート	リカバリする必要がある PDB を指定するための PLUGGABLE DATABASE 句を含めます。 RMAN は、リカバリ時に作成される一時ファイルを格納するために、補助の宛先を使用します。高速リカバリ領域が構成されている場合は、その領域が補助の宛先として使用されます。補助の

UNDOモード	補助インスタンスを使用するか	TARGETとして接続する場所	リカバリに使用するRMANコマンド
			宛先は、RECOVER コマンドに AUXILIARY DESTINATION 句を使用することで、明示的に指定できます。
ローカル	いいえ	CDB ルートまたは PDB	PDB に接続したときには、非 CDB に使用するのと同じコマンドを使用します。ルートに接続したときには、リカバリする必要がある PDB を指定するための PLUGGABLE DATABASE 句を含めます。

関連項目:

Point-in-Timeリカバリの詳細は、『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』の次の項を参照してください。

- [「マルチテナント環境でのリストア・ポイントの概要」](#)
- [「CDBリストア・ポイントの作成」](#)
- [「PDBリストア・ポイントの作成」](#)
- [「CDBおよびPDBのPoint-in-Timeリカバリの実行」](#)

親トピック: [CDBとPDBのバックアップおよびリカバリ](#)

マルチテナント環境でのフラッシュバック・データベース

マルチテナント・コンテナ・データベース(CDB)全体または特定のプラグブル(PDB)に対して、フラッシュバック・データベース操作を実行できます。

RMANは、Point-in-Timeリカバリ時に作成される一時ファイルを格納するために、補助の宛先を使用します。デフォルトでは、高速リカバリ領域が補助の宛先として使用されます。補助の宛先は、RECOVERコマンドにAUXILIARY DESTINATION句を使用することで、明示的に指定できます。

CDBのフラッシュバック

CDBのフラッシュバック・データベースを実行するには、次の前提条件を満たしている必要があります。

- SYSDBAまたはSYSBACKUP権限を持つ共通ユーザーとしてルートに接続する必要があります。
- CDBはマウントされている必要があります。

フラッシュバック操作の目標時点は、CDBリストア・ポイント、時間式、またはSCNを使用して指定します。CDBリストア・ポイントは、そのCDB内のどのPDBからでもアクセスできます。ただし、リストア・ポイントは、そのPDBのPDBサブインカンネーションを反映

しません。

PDBのフラッシュバック

PDBがクローズされていてCDBがオープンしている場合は、FLASHBACK DATABASEコマンドを使用することで、このPDBに対するフラッシュバック・データベース操作を実行できます。特定のPDBでフラッシュバック・データベース操作を実行すると、そのPDBに関連するデータファイルのみを変更します。CDB内のその他のPDBは影響を受けず、使用できます。PDBリストア・ポイントは、そのリストア・ポイントが定義されたPDBからのみアクセス可能で、このPDBに対する操作にのみ使用できることに注意してください。

表20-2 フラッシュバック方法の違い

CDBのUNDOモード	補助インスタンスを使用するか	TARGETとして接続する	
		場所	コマンド
共有	はい	CDB ルート	FLASHBACK PLUGGABLE DATABASE コマンドを使用します。クリーン PDB リストア・ポイントにのみフラッシュバックできます。 RMAN は、フラッシュバック時に作成される一時ファイルを格納するために、補助の宛先を使用します。高速リカバリ領域が構成されている場合は、その領域が補助の宛先として使用されます。補助の宛先は、FLASHBACK PLUGGABLE DATABASE コマンドで AUXILIARY DESTINATION 句を使用して、明示的に指定できます。
ローカル	いいえ	CDB ルートまたは PDB	FLASHBACK PLUGGABLE DATABASE コマンドを使用します。フラッシュバック操作の目標時点は、CDB リストア・ポイント、PDB リストア・ポイント、時間式、または目的の SCN を使用して指定できます。

関連項目:

CDBおよびPDBのフラッシュバックの詳細は、『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』の次の項を参照してください。

- [「マルチテナント環境でのリストア・ポイントの概要」](#)
- [「CDBリストア・ポイントの作成」](#)

- [「PDBリストア・ポイントの作成」](#)
- [「CDB全体に対するフラッシュバック・データベース操作の実行」](#)
- [「PDBに対するフラッシュバック・データベース操作の実行」](#)

親トピック: [CDBとPDBのバックアップおよびリカバリ](#)

21 マルチテナント環境でのデータベース・ユーティリティの使用

マルチテナント環境で、Oracle Data Pump、DBNEWID、Oracle LogMinerなどのユーティリティを使用できます。

- [CDBでのデータのインポートおよびエクスポート](#)
Oracle Data Pumpは、データベース間でデータとメタデータの高速移動を可能にするテクノロジーです。
- [CDBでのLogMinerの使用](#)
マルチテナント・コンテナ・データベース(CDB)でLogMinerを使用できます。
- [DBNEWIDのCDBおよびPDBに関する考慮事項](#)
DBNEWIDパラメータであるPDBを使用すると、プラグブル・データベース(PDB)のDBIDを変更できます。

親トピック: [マルチテナント環境でのOracle機能の使用](#)

CDBでのデータのインポートおよびエクスポート

Oracle Data Pumpテクノロジーを使用すると、データおよびメタデータをデータベース間で非常に高速に移動できます。

- [マルチテナント環境でのデータ・ポンプの使用について](#)
一般的に、PDBでのデータ・ポンプの使用は、非CDBでのデータ・ポンプの使用と同じです。
- [データ・ポンプを使用したCDBへのデータの移動](#)
空のPDBを作成した後、Oracle Data Pumpの全体モード・エクスポートおよびインポート操作を使用して、データをPDBに移動できます。
- [データ・ポンプを使用したCDB内またはCDB間のPDBの移動](#)
PDBに対するデータ・ポンプ・エクスポートおよびインポート操作は、共通ユーザーの処理方法を除き、非CDBに対する操作と同じです。

親トピック: [マルチテナント環境でのデータベース・ユーティリティの使用](#)

マルチテナント環境でのデータ・ポンプの使用について

一般的に、PDBでデータ・ポンプを使用することは、非CDBでデータ・ポンプを使用することと同じです。

マルチテナント・コンテナ・データベース(CDB)は、0、1または多数のユーザー作成のプラグブル・データベース(PDB)を含むOracleのデータベースです。PDBは、スキーマ、スキーマ・オブジェクトおよび非スキーマ・オブジェクトのポータブル・セットで、Oracle Netクライアントからは、非CDBとして表示されます。非CDBとは、CDBではないOracle Databaseです。

データ・ポンプを使用して、次のシナリオでデータベースの一部またはすべてを移行できます。

- 非CDBからPDBへ
- 同じCDBまたは異なるCDB内のPDB間
- PDBから非CDBへ

ノート:



データ・ポンプでは、CDB全体にわたる操作はサポートされていません。CDBのルートまたはシード・データベースに接続して

ORA-39357: Warning: Oracle Data Pump operations are not typically needed when connected to the root

データ・ポンプを使用したCDBへのデータの移動

空のPDBを作成した後、Oracle Data Pumpの全体モード・エクスポートおよびインポート操作を使用して、データをPDBに移動できます。

データはトランスポートابل・オプションの有無にかかわらずインポートできます。トランスポートابل・オプションを全体モード・エクスポートまたはインポートで使用する場合、全体トランスポートابل・エクスポート/インポートと呼ばれます。

トランスポートابل・オプションが使用される場合、エクスポートおよびインポートでは、トランスポートابل表領域のデータ移動と従来のデータ移動の両方を使用しますが、後者はSYSTEMやSYS_AUXなどの非トランスポートابل表領域に存在する表が対象です。トランスポートابل・オプションを使用すると、表データをアンロードおよび再ロードする必要と、ユーザー表領域の索引構造を再作成する必要がなくなるため、エクスポート時間と(特に)インポート時間を短縮できます。

データ・ポンプを使用してCDBにデータを移動する場合は、次の要件に注意してください。

- マルチテナント環境を管理するには、CDB_DBAロールを持っている必要があります。
- Oracle Database 11.2.0.2以前からの全体データベース・エクスポートは、Oracle Database 12c (CDBまたは非CDB)にインポートされる場合があります。ただし、登録したオプションおよびコンポーネントの情報がエクスポートに含まれるように、まずソース・データベースをOracle Database 11gリリース2 (11.2.0.3以降)にアップグレードすることをお勧めします。
- 全体データベース・エクスポートまたは全体トランスポートابل・データベース・エクスポートのいずれかを使用して、Oracle Database 11gリリース2 (11.2.0.3以上)をCDB(または非CDB)に移行する場合、Oracle Database 12cにインポートできるダンプ・ファイルを生成するためにデータ・ポンプ・エクスポート・パラメータVERSION=12を設定する必要があります。VERSION=12を設定しない場合、生成したエクスポート・ファイルには、登録したデータベース・オプションおよびコンポーネントの完全な情報は含まれません。
- ネットワーク・ベースの全体トランスポートابل・インポートでは、FULL=YES、TRANSPORTABLE=ALWAYSおよびTRANSPORT_DATAFILES=datafile_nameパラメータを使用する必要があります。ソース・データベースがOracle Database 11gリリース11.2.0.3以上で、Oracle Database 12cリリース1 (12.1)より前の場合、VERSION=12パラメータも必要です。
- ファイル・ベースの全体トランスポートابل・インポートでは、TRANSPORT_DATAFILES=datafile_nameパラメータのみを使用する必要があります。データ・ポンプ・インポートでは、TRANSPORTABLE=ALWAYSおよびFULL=YESパラメータの存在が推測されます。
- Oracle Database 12cリリース2 (12.2)以降、マルチテナント・コンテナ・データベース(CDB)環境では、CDBの各PDBに対する一意のパスとして、デフォルトのデータ・ポンプ・ディレクトリ・オブジェクトDATA_PUMP_DIRが定義されます。この一意のパスは、CREATE PLUGGABLE DATABASE文のPATH_PREFIX句が相対パスに定義されているかどうかに関係なく、定義されます。
- Oracle Database 19c以降では、impdpのcredentialパラメータに、オブジェクト・ストア・バケットへのアクセスに必要なユーザー名とパスワードを含む資格証明オブジェクトの名前を指定します。DEFAULT_CREDENTIALという名前のPDBプロパティを使用して、デフォルトの資格証明を指定することもできます。デフォルトの資格証明を使用してimpdpを実行する場合は、ダンプ・ファイル名の前にDEFAULT_CREDENTIAL:を付け、credentialパラメータは指定しません。

例21-1 PDBへの表のインポート

エクスポート/インポート操作に特定のPDBを指定するには、データ・ポンプの開始時に、接続識別子を接続文字列に指定します。たとえば、pdb1というPDBにデータをインポートするには、データ・ポンプ・コマンドラインで次のように入力します。

```
impdp hr@pdb1 DIRECTORY=dpump_dir1 DUMPFILE=hr.dmp TABLES=employees
```

例21-2 データのインポート時の資格証明の指定

この例では、次に示すように、DBMS_CREDENTIAL.CREATE_CREDENTIALを使用してHR_CREDという名前の資格証明を作成していることを前提としています。

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL (
    credential_name => 'HR_CRED',
    username => 'atpc_user@oracle.com',
    password => 'password'
  );
END;
/
```

次のコマンドは、資格証明HR_CREDを指定し、オブジェクト・ストアに格納されているファイルを指定します。ファイルのURLはhttps://example.com/ostore/dnfs/myt.dmpです。

```
impdp hr@pdb1 ¥
  table_exists_action=replace ¥
  credential=HR_CRED ¥
  parallel=16 ¥
  dumpfile=https://example.com/ostore/dnfs/myt.dmp
```

例21-3 デフォルトの資格証明を使用したデータのインポート

1. 次に示すように、DBMS_CREDENTIAL.CREATE_CREDENTIALを使用して、HR_CREDという名前の資格証明を作成します。

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL (
    credential_name => 'HR_CRED',
    username => 'atpc_user@oracle.com',
    password => 'password'
  );
END;
/
```

2. PDBプロパティDEFAULT_CREDENTIALを次のように設定します。

```
ALTER DATABASE PROPERTY SET DEFAULT_CREDENTIAL = 'ADMIN.HR_CRED'
```

3. 次のコマンドは、ダンプ・ファイルの場所https://example.com/ostore/dnfs/myt.dmpの接頭辞として、デフォルトの資格証明を指定します。

```
impdp hr@pdb1 ¥
  table_exists_action=replace ¥
  parallel=16 ¥
  dumpfile=default_credential:https://example.com/ostore/dnfs/myt.dmp
```

credentialパラメータは指定しません。

関連項目:

- オブジェクト・ストアへのアクセスに必要なSSL認証の構成方法を学習するには、[Oracle『Oracle Databaseセキュリティ・ガイド』](#)を参照

- データ・ポンプ・インポートを使用してファイルをオブジェクト・ストアにロードする方法について学習するには、[『Oracle Databaseユーティリティ』](#)を参照

親トピック: [CDBでのデータのインポートおよびエクスポート](#)

データ・ポンプを使用したCDB内またはCDB間のPDBの移動

PDBに対するデータ・ポンプ・エクスポートおよびインポート操作は、共通ユーザーの処理方法を除き、非CDBに対する操作と同じです。

CDBで共通ユーザーを作成した場合、CDBのPDB内からそのユーザーの全データベースまたは特権スキーマをエクスポートすると、標準のCREATE USER C##common name DDL文がインポート時に実行されます。ユーザー名の共通ユーザー接頭辞C##が原因で、文は失敗します。次のエラー・メッセージが返されます。

```
ORA-65094:invalid local user or role name
```

エクスポート対象のPDBで、そのユーザーのスキーマにローカル・オブジェクトを作成してそれらをインポートする場合、同じ名前の共通ユーザーがすでにターゲットCDBインスタンスに存在することを確認するか、次のようにimpdpコマンドでデータ・ポンプ・インポートのREMAP_SCHEMAパラメータを使用します。

```
REMAP_SCHEMA=C##common name:local user name
```

関連トピック

- [『Oracle Databaseユーティリティ』](#)
- [『Oracle Databaseユーティリティ』](#)
- [『Oracle Databaseユーティリティ』](#)

親トピック: [CDBでのデータのインポートおよびエクスポート](#)

CDBのLogMinerの使用

LogMinerは、マルチテナント・コンテナ・データベース(CDB)で使用できます。

次の項では、CDBと非CDBでLogMinerの使用時に注意する必要がある相違点について説明します。

ルート・コンテナにすべてのPDBのスーパーセットの文字セットが含まれている場合、LogMinerでは異なる文字セットのPDBを含むCDBがサポートされます。

マルチテナント環境を管理するには、CDB_DBAロールを持っている必要があります。

- [CDBでのLogMiner V\\$ビューおよびDBAビュー](#)
CDBでは、システムで実行されているLogMinerセッションの情報を表示するためにLogMinerによって使用されるビューに、CON_IDという追加の列が含まれます。
- [CDBでのV\\$LOGMNR_CONTENTSビュー](#)
CDBでは、V\$LOGMNR_CONTENTSビューとその関連機能はルート・データベースに制限されています。V\$LOGMNR_CONTENTSには、CDBを支援するいくつかの新しい列が存在します。
- [CDBでのサブリメンタル・ロギングの有効化](#)
CDBで、ALTER DATABASEコマンドを使用してデータベース全体のサブリメンタル・ロギングを有効または無効にするための構文。
- [CDBでのフラット・ファイル・ディクショナリの使用](#)
単一のフラット・ファイルにCDB全体のディクショナリ・スナップショットを取得することはできません。個々のPDBに接続す

る必要があり、そのPDB専用のスナップショットをフラット・ファイルに取得できます。

親トピック: [マルチテナント環境でのデータベース・ユーティリティの使用](#)

CDBでのLogMiner V\$ビューおよびDBAビュー

CDBでは、システムで実行されているLogMinerセッションの情報を表示するためにLogMinerによって使用されるビューに、CON_IDという追加の列が含まれます。

CON_ID列によって、情報が表示されるセッションに関連付けられたコンテナIDを識別します。プラグブル・データベース(PDB)のビューを問い合わせると、データベースに関連付けられた情報のみが表示されます。この新しい動作の影響を受けるビューは次のとおりです。

- V\$LOGMNR_DICTIONARY_LOAD
- V\$LOGMNR_LATCH
- V\$LOGMNR_PROCESS
- V\$LOGMNR_SESSION
- V\$LOGMNR_STATS

ノート:



CDBをサポートするため、V\$LOGMNR_CONTENTS ビューには、CON_ID 以外にもいくつかの新しい列が含まれます。

次のDBAビューには、名前がCDBで始まる類似のCDBビューがあります。

ログ・ビューのタイプ	DBAビュー	CDBビュー
LogMiner ログ・ビュー	DBA_LOGMNR_LOG	CDB_LOGMNR_LOG
LogMiner パージ済ログ・ビュー	DBA_LOGMNR_PURGED_LOG	CDB_LOGMNR_PURGED_LOG
LogMiner セッション・ログ・ビュー	DBA_LOGMNR_SESSION	CDB_LOGMNR_SESSION

DBAビューには、問合せが行われたコンテナで定義されているセッションに関連する情報のみが表示されます。

CDBビューには、特定の行が表すデータを含むコンテナを識別する追加のCON_ID列が含まれます。ルートからCDBビューを問い合わせると、それらのビューを使用してすべてのコンテナに関する情報を参照できます。

親トピック: [CDBのLogMinerの使用](#)

CDBでのV\$LOGMNR_CONTENTSビュー

CDBでは、V\$LOGMNR_CONTENTSビューとその関連機能はルート・データベースに制限されています。V\$LOGMNR_CONTENTSには、CDBを支援するいくつかの新しい列が存在します。

- CON_ID: 問合せの実行元のコンテナに関連付けられたIDが含まれます。V\$LOGMNR_CONTENTSはルート・データベースに制限されるため、問合せがCDBで実行されると、この列は値1を返します。
- SRC_CON_NAME: PDBの名前。この情報は、マイニングがLogMinerディクショナリで実行される場合にのみ使用できま

す。

- SRC_CON_ID: REDOレコードを生成したPDBのコンテナID。この情報は、マイニングがLogMinerディクショナリで実行される場合にのみ使用できます。
- SRC_CON_DBID: PDBの識別子。この情報は、マイニングが現在のLogMinerディクショナリで実行される場合にのみ使用できます。
- SRC_CON_GUID: PDBに関連付けられたGUIDが含まれます。この情報は、マイニングが現在のLogMinerディクショナリで実行される場合にのみ使用できます。

親トピック: [CDBのLogMinerの使用](#)

CDBでのサプリメンタル・ロギングの有効化

CDBで、ALTER DATABASEコマンドを使用してデータベース全体のサプリメンタル・ロギングを有効または無効にするための構文。たとえば、サプリメンタル・ログ・データを追加または削除する場合は、次の構文を使用します。

```
ALTER DATABASE [ADD|DROP] SUPPLEMENTAL LOG DATA ...
```

次の点に注意してください。

- CDBでは、CDB\$ROOTから有効にしたサプリメンタル・ロギング・レベルは、CDB全体で有効になります。
- 少なくとも最小サプリメンタル・ロギングがCDB\$ROOTで有効になっている場合、追加のサプリメンタル・ロギング・レベルをPDBレベルで有効にできます。
- CDB\$ROOTからCDBレベルで有効にしたサプリメンタル・ロギング・レベルは、PDBレベルでは無効にできません。
- CDB\$ROOTからすべてのサプリメンタル・ロギングを削除した場合、以前のPDBレベルの設定に関係なく、CDB全体ですべてのサプリメンタル・ロギングが無効になります。

CREATE TABLE文とALTER TABLE文で開始するサプリメンタル・ロギング操作は、CDBルートかPDBから実行できます。これらは適用先の表にのみ影響します。

親トピック: [CDBのLogMinerの使用](#)

CDBでのフラット・ファイル・ディクショナリの使用

単一のフラット・ファイルにCDB全体のディクショナリ・スナップショットを取得することはできません。個々のPDBに接続する必要があり、そのPDB専用のスナップショットをフラット・ファイルに取得できます。

したがって、フラット・ファイル・ディクショナリを使用する場合、データ・ディクショナリがフラット・ファイル内に含まれるPDBに関連付けられた変更を対象とするREDOログのみをマイニングできます。

親トピック: [CDBのLogMinerの使用](#)

DBNEWIDのCDBおよびPDBに関する考慮事項

DBNEWIDパラメータであるPDBを使用すると、プラグブル・データベース(PDB)のDBIDを変更できます。

デフォルトでは、コンテナ・データベース(CDB)に対してDBNEWIDユーティリティを実行すると、そのCDBのみのDBIDが変更され、そのCDBに含まれるプラグブル・データベース(PDB)のDBIDは変更されません。このため、CDBをクローニングした場合には、PDBのDBIDが重複するという問題が発生することがあります。

Oracle Database 12c リリース2 (12.2)からは、新しいDBNEWID PDBパラメータを使用して、PDBのDBIDを変更でき

ます。特定のPDBを指定することはできません。すべてのPDBのDBIDを新しくするか、どのPDBのDBIDも変更しないかのどちらかです。PDBパラメータは、CDB環境にのみ適用可能です。このファイルは次のような形式です。

```
PDB=[ALL | NONE]
```

- ALLを指定した場合、CDBのDBIDが変更されるだけでなく、そのCDBに含まれるすべてのPDBのDBIDも変更されます。
- NONE (デフォルト)を指定すると、CDBのDBIDが変更されてもPDBのDBIDは変更されません。

PDB=ALLの使用をお勧めしますが、下位互換性の理由からPDB=NONEがデフォルトになっています。

親トピック: [マルチテナント環境でのデータベース・ユーティリティの使用](#)

22 PDBに対するOracle Resource Managerの使用

PL/SQLパッケージ・プロシージャを使用して、Oracle Resource Manager (リソース・マネージャ)によるマルチテナント・コンテナ・データベース(CDB)内のプラガブル・データベース(PDB)へのリソース割当てを管理します。

この章では、次の前提条件を満たしていることを前提としています。

- CDBを構成および管理する方法について理解しています。
- リソース・マネージャを使用して非CDBのリソースを割り当てる方法について理解しています。

ノート:



- この章のタスクは、SQL*Plus または Oracle SQL Developer を使用して実行できます。
- リソース・マネージャは、Oracle Enterprise Manager Cloud Control (Cloud Control)のグラフィカル・ユーザー・インタフェースで管理することもできます。
- 単純にするために、この章では PDB、アプリケーション・ルートおよびアプリケーション PDB を「PDB」と呼びます。

- [マルチテナント環境でのOracle Resource Managerの概要](#)

CDBでは、複数のPDB内のワークロードがシステム・リソースおよびCDBリソースについて競合する場合があります。リソース・プランはこの問題を解決します。

- [CDBリソース・プランの管理](#)

CDBでは、PDBの優先度のレベルが異なる場合があります。CDBリソース・プランを作成し、これらの優先度に従ってPDBごとにリソースを分配できます。

- [PDBリソース・プランの管理](#)

個々のPDBのリソース・プランを作成し、有効化および変更できます。

- [Oracle Database Resource Managerで管理されるPDBの監視](#)

動的パフォーマンス・ビューのセットにより、PDBに対するOracle Database Resource Managerの設定結果を監視できます。

関連項目:

- [「マルチテナント環境の管理」](#)
- Resource Managerについてさらに学習するには、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [マルチテナント環境でのOracle機能の使用](#)

マルチテナント環境でのOracle Resource Managerの概要

CDBでは、複数のPDB内のワークロードがシステム・リソースおよびCDBリソースについて競合する場合があります。リソース・プランはこの問題を解決します。

マルチテナント環境では、リソース・マネージャは2つのレベルで動作します。

- CDBレベル

リソース・マネージャを使用して、システム・リソースおよびCDBリソースを巡って競合関係にある複数のPDBワークロードを管理できます。PDBへのリソースの割当て方法を指定し、特定のPDBのリソース使用率を制限できます。主要なツールはCDBリソース・プランです。

- PDBレベル

リソース・マネージャを使用して、各PDB内のワークロードを管理できます。主要なツールはPDBリソース・プランです。

リソース・マネージャにより、次の2つのステップでリソースを割り当てます。

1. システム・リソースの一部を各PDBに割り当てます。
2. 特定のPDBでは、前述のステップで取得したシステム・リソースの一部をPDBに接続された各セッションに割り当てます。

ノート:



リソース・マネージャは、ルート内のアクティビティを自動的に管理します。

マルチテナント環境でリソース・マネージャを使用するには、次の前提条件を満たす必要があります。

- CDBが存在し、PDBを含んでいる必要があります。
- DBMS_RESOURCE_MANAGERパッケージを使用するタスクを完了するには、ADMINISTER_RESOURCE_MANAGERシステム権限が必要です。
- [マルチテナント環境でのリソース管理の目的](#)
リソース・マネージャは、CDBのリソースのより効率的な使用方法を提供できます。
- [リソース・プラン・ディレクティブの概要](#)
CDBリソース・プランにより、そのリソース・プラン・ディレクティブ(ディレクティブ)のセットに従って、リソースがそのPDBに割り当てられます。
- [バックグラウンド・タスクと管理タスクおよびコンシューマ・グループ](#)
CDBでは、バックグラウンド・タスクおよび管理タスクが、これらのタスクを最適に実行するリソース・マネージャのコンシューマ・グループにマップします。
- [PDBレベルのリソースに関する初期化パラメータ](#)
PDBのCPU、メモリー、セッションおよびI/Oを制御するには、初期化パラメータを使用します。

関連項目:

- [「CDBの作成および構成」](#)および[「PDBおよびアプリケーション・コンテナの作成および削除」](#)を参照してください
- DBMS_RESOURCE_MANAGERパッケージについてさらに学習するには、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [PDBに対するOracle Resource Managerの使用](#)

マルチテナント環境でのリソース管理の目的

リソース・マネージャは、CDBのリソースのより効率的な使用方法を提供できます。

CDBに対するリソースの割当てがオペレーティング・システムによって決定される場合は、次のようなワークロード管理に関する問題が生じることがあります。

- PDB間の不適当なリソースの割当て

オペレーティング・システムはタスク間の優先度付けができないため、すべてのアクティブなプロセスの間で均等にリソースを分配します。このため、1つ以上のPDBで過大なシステム・リソースが使用され、他のPDBでリソースが不足する場合があります。

- 1つのPDB内における不適当なリソースの割当て

1つのPDBに接続された1つ以上のセッションで過大なシステム・リソースが使用され、同じPDBに接続された他のセッションでリソースが不足する場合があります。

- PDBの一貫性のないパフォーマンス

他のPDBとの間で競合するシステム・リソースの量が一定でない場合は、1つのPDBのパフォーマンスは必ずしも一定になりません。

- PDBのリソース使用率データの不足

PDBの監視およびチューニングにおいてはリソース使用率データが重要になります。非CDBのリソース使用率データを収集するときに、この非CDBがシステム上で実行されている唯一のデータベースである場合、オペレーティング・システム・モニタリング・ツールを使用できることがあります。ただし、CDBの場合、システム上で複数のPDBが実行されるため、オペレーティング・システム・モニタリング・ツールの有用性は低くなります。

リソース・マネージャは、PDB間およびPDB内でのハードウェア・リソースの割当て方法をCDBで厳密に制御することによって、これらの問題を克服するのに役立ちます。

複数のPDBを含むCDBでは、通常、一部のPDBの重要度が他のPDBより高くなります。リソース・マネージャを使用すると、特定のPDBのリソース使用率を優先度付けしたり制限できます。リソース・マネージャを使用すると、次のことが可能になります。

- PDBごとにシステム・リソースの共有が異なるよう指定して、より多くのリソースが重要度の高いPDBに割り当てられるようにします。
- 特定のPDBのCPU使用率を制限します。
- 特定のPDBで使用できるパラレル実行サーバーの数を制限します。
- 特定のPDBのメモリー使用率を制限します
- 特定のPDBで保証されるメモリー量を指定します
- 特定のPDBで使用可能なメモリーの最大値を指定します
- 異なるPDBのセットにPDBパフォーマンス・プロファイルを使用します

PDBのセットのパフォーマンス・プロファイルでは、システム・リソースの共有、CPU使用率、およびパラレル実行サーバーの数を指定できます。PDBパフォーマンス・プロファイルにより、個々のPDBではなくプロファイルのリソース・マネージャ・ディレクティブを指定することで、大量のPDBのリソースを管理できます。

- 1つのPDBに接続された様々なセッションのリソース使用率を制限します。
- 特定のPDBによって生成されるI/Oを制限します
- PDBのリソース使用率を監視します。

親トピック: [マルチテナント環境でのOracle Resource Managerの概要](#)

リソース・プラン・ディレクティブの概要

CDBリソース・プランにより、そのリソース・プラン・ディレクティブ(ディレクティブ)のセットに従って、リソースがそのPDBに割り当てられます。

CDBリソース・プランとそのリソース・プラン・ディレクティブは親子関係にあります。各ディレクティブは、パフォーマンス・プロファイル内の一連のPDBまたは単一のPDBのいずれかを参照します。

同じCDB内の個々のPDBおよびPDBパフォーマンス・プロファイルの両方にディレクティブを指定できます。現在アクティブなプランの2つのディレクティブで同じPDBまたは同じPDBパフォーマンス・プロファイルを参照することはできません。

- [PDBパフォーマンス・プロファイル](#)

PDBパフォーマンス・プロファイルは、同じ優先度およびリソース制御が指定された一連のPDBにリソース・プラン・ディレクティブを構成します。

- [リソース・プラン・ディレクティブ](#)

ディレクティブは、CPUとパラレル実行サーバーの割当てを制御します。

親トピック: [マルチテナント環境でのOracle Resource Managerの概要](#)

PDBパフォーマンス・プロファイル

PDBパフォーマンス・プロファイルは、同じ優先度およびリソース制御が指定された一連のPDBにリソース・プラン・ディレクティブを構成します。

たとえば、Gold、SilverおよびBronzeというパフォーマンス・プロファイルを作成できます。各プロファイルには、PDBのタイプの重要性に基づいて一連の様々なディレクティブを指定します。Gold PDBはSilver PDBよりミッション・クリティカルで、Silver PDBはBronze PDBよりミッション・クリティカルです。PDBでは、DB_PERFORMANCE_PROFILE初期化パラメータを使用してパフォーマンス・プロファイルを指定します。

PDBロックダウン・プロファイルを使用して、SGA_TARGET、PGA_AGGREGATE_LIMITなど、リソースを制御するPDB初期化パラメータを指定できます。ロックダウン・プロファイルによって、PDB管理者による設定の変更が防止されます。

パフォーマンス・プロファイルとロックダウン・プロファイルには、一致した名前を使用することをお勧めします。PDBの所有者がプロファイルを切り替えることを防ぐために、PDBパフォーマンス・プロファイルをPDBロックダウン・プロファイルに配置することをお勧めします。

親トピック: [リソース・プラン・ディレクティブの概要](#)

リソース・プラン・ディレクティブ

ディレクティブは、CPUとパラレル実行サーバーの割当てを制御します。

ディレクティブにより、PDBまたはPDBパフォーマンス・プロファイルごとに指定する共有値に基づいてPDBへのリソースの割当てを制御できます。共有値が高いほど、リソースは増加します。たとえば、あるPDBに、もう1つのPDBに割り当てられた2倍のリソースを割り当てることを指定でき、このことを行うには、最初のPDBに2番目のPDBの2倍高い共有値を設定します。同様に、最初のPDBパフォーマンス・プロファイルに2番目のPDBパフォーマンス・プロファイルの2倍高い共有値を設定することで、最初のPDBパフォーマンス・プロファイルに、2番目のPDBパフォーマンス・プロファイルに割り当てられたリソースの2倍のリソースを割り当てることを指定できます。設定は、各プロファイルを使用するPDBのセットに適用されます。

PDBおよびPDBパフォーマンス・プロファイルに使用率制限を指定することもできます。制限はPDBまたはパフォーマンス・プロファイルへの割当てを制御します。たとえば、制限を使用して、CDBに使用可能なCPU全体の割合として、PDBで確保できるCPUの量を制御できます。

CDB内の各PDBおよびPDBパフォーマンス・プロファイルに割り当てるリソースを正確に制御するために、共有と使用率制限の両方を一緒に使用できます。

関連項目:

PDBロックダウン・プロファイルの詳細は、[セキュリティを強化するためのPDBユーザーの制限について](#)を参照してください

親トピック: [リソース・プラン・ディレクティブの概要](#)

バックグラウンド・タスクと管理タスクおよびコンシューマ・グループ

CDBでは、バックグラウンド・タスクおよび管理タスクが、これらのタスクを最適に実行するリソース・マネージャのコンシューマ・グループにマップします。

リソース・マネージャでは、次のルールを使用して、コンシューマ・グループにタスクをマップします。

- タスクは、このタスクを開始するコンテナ内のコンシューマ・グループにマップされます。
タスクがCDBルートで開始される場合、このタスクはCDBルート内のコンシューマ・グループにマップします。タスクがPDBで開始される場合、このタスクはPDB内のコンシューマ・グループにマップします。
- 多くのメンテナンス・タスクおよび管理タスクは、コンシューマ・グループに自動的にマップします。
たとえば、自動化されたメンテナンス・タスクはORA\$AUTOTASKにマップします。特定の状況では、タスクはコンシューマ・グループにマップしますが、マッピングは変更可能です。このようなタスクとしては、RMANバックアップ、RMANイメージ・コピー、Oracle Data Pumpおよびメモリー内移入があります。

ノート:



事前定義のコンシューマ・グループのマッピング・ルールについてさらに学習するには、『[Oracle Database 管理者ガイド](#)』を参照してください

親トピック: [マルチテナント環境でのOracle Resource Managerの概要](#)

PDBレベルのリソースの初期化パラメータ

初期化パラメータを使用して、PDBのCPU、メモリー、セッションおよびI/Oを制御します。

- [PDBのCPU関連の初期化パラメータ](#)
CPU_COUNT初期化パラメータは、Oracle Databaseで使用する使用可能なCPUの数を指定します。
- [PDBのメモリー関連の初期化パラメータ](#)
複数の初期化パラメータでPDBのメモリー使用を制御します。
- [PDBのセッション関連の初期化パラメータ](#)
PDBでセッションがリソースを消費する方法は、複数の初期化パラメータによって制御されます。
- [PDBのI/O関連の初期化パラメータ](#)
MAX_IOPSおよびMAX_MBPS初期化パラメータは、PDBで生成されるディスクI/Oを制限します。

親トピック: [マルチテナント環境でのOracle Resource Managerの概要](#)

PDBのCPU関連の初期化パラメータ

CPU_COUNT初期化パラメータには、Oracle Databaseが使用できるCPUの数を指定します。

インスタンス・ケーシングは、初期化パラメータを使用して、インスタンスが同時に使用できるCPU数を制限する手法です。PDBレベルでCPU_COUNTを設定できます。リソース・マネージャが有効化されている場合、PDBはCPU_COUNTで指定されたCPU数に拘束(制限)されます。

CPU_COUNTは、そのCDBプランでutilization_limitディレクティブと同じように機能します。ただし、CPU_COUNT制限は、利用率ではなくCPUの数の観点で表されます。utilization_limitおよびCPU_COUNTの両方を指定した場合は、より低い制限が適用されます。

CPU_COUNTは、PDBが新しいコンテナに接続された場合にCPU_COUNT設定は接続されているPDBに残るため便利です。また、Oracle Databaseは、パラレル実行用のパラメータなど、その他多くのPDBパラメータを導出するために、PDBのCPU_COUNT設定を使用します。

親トピック: [PDBレベルのリソースに関する初期化パラメータ](#)

PDBのメモリー関連の初期化パラメータ

複数の初期化パラメータでPDBのメモリー使用を制御します。

PDBが現在のコンテナである場合は、次の表の初期化パラメータによって現在のPDBのメモリー使用が制御されます。これらのパラメータの1つ以上がPDBに対して設定されている場合は、そのCDBおよび他のPDBがそれらの操作に対して十分なメモリーを持っていることを確認します。初期化パラメータは、次の条件を満たす場合にのみ、PDBでのメモリーの使用方法を制御します。

- NONCDB_COMPATIBLE初期化パラメータが、CDBルートでfalseに設定されている。
- CDBルートにMEMORY_TARGET初期化パラメータが設定されていないか、0 (ゼロ)が設定されている。

表22-1 PDBのメモリー使用を制御する初期化パラメータ

初期化パラメータ	説明
DB_CACHE_SIZE	<p>PDB に対して保証される最小バッファ・キャッシュ容量を設定します。</p> <p>次の要件が満たされている必要があります。</p> <ul style="list-style-type: none">● CDB ルートの DB_CACHE_SIZE の設定の 50%以下である必要があります。● すべての PDB の DB_CACHE_SIZE 設定の合計が、CDB ルートの DB_CACHE_SIZE の設定の 50%以下である必要があります。 <p>CDB ルートで SGA_TARGET 初期化パラメータが 0(ゼロ)以外の値に設定されている場合、これらの要件は適用されません。</p> <p>SGA_TARGET 初期化パラメータが 0(ゼロ)以外の値に設定されている場合は、次の要件が満たされている必要があります。</p>

初期化パラメータ	説明
	<ul style="list-style-type: none"> ● PDB 内の DB_CACHE_SIZE および SHARED_POOL_SIZE の値は、PDB の SGA_TARGET 値の 50%以下にする必要があります。 ● PDB 内の DB_CACHE_SIZE および SHARED_POOL_SIZE の値は、CDB レベルの SGA_TARGET 値の 50%以下にする必要があります。 ● CDB 内のすべての PDB における DB_CACHE_SIZE および SHARED_POOL_SIZE の合計は、CDB レベルの SGA_TARGET 値の 50%以下にする必要があります。
SHARED_POOL_SIZE	<p>PDB に対して保証される最小共有プール容量を設定します。</p> <p>次の要件が満たされている必要があります。</p> <ul style="list-style-type: none"> ● CDB ルートの SHARED_POOL_SIZE の設定の 50%以下である必要があります。 ● すべての PDB の SHARED_POOL_SIZE 設定の合計が、CDB ルートの SHARED_POOL_SIZE の設定の 50%以下である必要があります。 <p>CDB ルートで SGA_TARGET 初期化パラメータが 0(ゼロ)以外の値に設定されている場合、これらの要件は適用されません。</p> <p>SGA_TARGET 初期化パラメータが 0(ゼロ)以外の値に設定されている場合は、次の要件が満たされている必要があります。</p> <ul style="list-style-type: none"> ● PDB 内の DB_CACHE_SIZE および SHARED_POOL_SIZE の値は、PDB の SGA_TARGET 値の 50%以下にする必要があります。 ● PDB 内の DB_CACHE_SIZE および SHARED_POOL_SIZE の値は、CDB レベルの SGA_TARGET 値の 50%以下にする必要があります。 ● CDB 内のすべての PDB における DB_CACHE_SIZE および SHARED_POOL_SIZE の合計は、CDB レベルの SGA_TARGET 値の 50%以下にする必要があります。
SGA_MIN_SIZE	<p>PDB の最小 SGA サイズを設定します。</p> <p>次の要件が満たされている必要があります。</p>

初期化パラメータ	説明
SGA_TARGET	<ul style="list-style-type: none"> ● CDB ルートの SGA_TARGET の設定の 50%以下である必要があります。 ● PDB の SGA_TARGET の設定の 50%以下である必要があります。 ● すべての PDB の SGA_MIN_SIZE 設定の合計が、CDB ルートの SGA_TARGET の設定の 50%以下である必要があります。 <p>CDB ルートで SGA_TARGET 初期化パラメータが設定されていないか、0 (ゼロ)に設定されている場合、これらの要件は適用されません。</p>
PGA_AGGREGATE_LIMIT	<p>PDB の最大 PGA サイズを設定します。</p> <p>PDB の SGA_TARGET 設定は、CDB ルートで SGA_TARGET 初期化パラメータが 0 (ゼロ)以外の値に設定されている場合にのみ強制されます。PDB の SGA_TARGET 設定は、CDB ルートの SGA_TARGET 設定以下である必要があります。</p> <p>PDB の最大 PGA サイズを設定します。</p> <p>次の要件が満たされている必要があります。</p> <ul style="list-style-type: none"> ● CDB ルートの PGA_AGGREGATE_LIMIT の設定以下である必要があります。 ● PDB の PGA_AGGREGATE_TARGET の設定の 2 倍以上である必要があります。
PGA_AGGREGATE_TARGET	<p>PDB のターゲット集計 PGA サイズを設定します。</p> <p>次の要件が満たされている必要があります。</p> <ul style="list-style-type: none"> ● CDB レベルで設定されている PGA_AGGREGATE_TARGET 値以下である必要があります。 ● CDB レベルで設定されている PGA_AGGREGATE_LIMIT 初期化パラメータ値の 50%以下である必要があります。 ● PDB に設定されている PGA_AGGREGATE_LIMIT 値の

初期化パラメータ**説明**

50%以下である必要があります。

例22-1 PDBで使用可能な最大平均PGAメモリーの設定

PDBを現在のコンテナとして、次のSQL文を実行して、メモリーとSPFILEの両方でPGA_AGGREGATE_LIMIT初期化パラメータを90MBに設定します。

```
ALTER SYSTEM SET PGA_AGGREGATE_LIMIT = 90M SCOPE = BOTH;
```

例22-2 PDBの最大SGAサイズの設定

i

PDBを現在のコンテナとして、次のSQL文を実行して、メモリーとSPFILEの両方でSGA_MIN_SIZE初期化パラメータを500MBに設定します。

```
ALTER SYSTEM SET SGA_MIN_SIZE = 500M SCOPE = BOTH;
```

親トピック: [PDBレベルのリソースに関する初期化パラメータ](#)

PDBのセッション関連の初期化パラメータ

いくつかの初期化パラメータによって、PDBにおいてセッションでリソースがどのように消費されるかを制御します。

表22-2 PDBのセッション使用を制御する初期化パラメータ

初期化パラメータ	説明(PDBレベルで設定する場合)	PDBレベルでのデフォルト
SESSIONS	<p>PDB で使用できる最大セッション数を設定します。</p> <p>PDB で、その SESSIONS パラメータで構成されている数を超えてセッションを使用しようとすると、ORA-00018 エラー・メッセージが生成されます。PDB の場合、SESSIONS パラメータでは再帰セッションはカウントされないため、10%の調整は必要になりません。</p> <p>PDB の SESSIONS パラメータを変更できるのはその PDB のみです。CDB レベルで設定されている SESSIONS 値より大きい値は設定できません。</p>	CDB レベルでの SESSIONS に、PDB レベルの CPU_COUNT/CDB レベルの CPU_COUNT の比率を乗じた値
MAX_IDLE_TIME	セッションをアイドル状態にしておける最長時間(分数)を指定します。その最大値に達すると、Oracle Database によって自動的にセッションが終了されます。	0 (未設定)

初期化パラメータ	説明(PDBレベルで設定する場合)	PDBレベルでのデフォルト
MAX_IDLE_BLOCKER_TIME	<p>セッションがアイドル状態になってから終了候補になるまでの時間(分数)を設定します。</p> <p>このパラメータを使用すると、アイドル状態のセッションは、それによって別のセッションがブロックされている場合は終了されます。Oracle Database では、次のいずれかの状況でブロックされているセッションが対象とみなされます。</p> <ul style="list-style-type: none"> ● そのセッションによって、別のセッションに必要なロックが保持されている。 ● そのセッションがパラレル操作であり、そのコンシューマ・グループ、PDB または CDB でそのパラレル・サーバー上限に達するかパラレル操作がキューに入れている。 ● そのセッションの PDB またはデータベース・インスタンスが、そのセッション制限またはプロセス制限に到達しようとしている。 <p>MAX_IDLE_TIME と異なり、MAX_IDLE_BLOCKER_TIME では、リソースはそれらが必要になったときにのみ終了されます。</p>	0 (未設定)

親トピック: [PDBレベルのリソースに関する初期化パラメータ](#)

PDBのI/O関連の初期化パラメータ

MAX_IOPSおよびMAX_MBPS初期化パラメータは、PDBで生成されるディスクI/Oを制限します。

大量のディスクI/Oはパフォーマンス低下の原因となることがあります。不適切に設計されたSQL、大量トランザクションでの索引や表のスキャンなど、複数の要因によりディスクI/Oが過剰になることがあります。1つのPDBが大量のディスクI/Oを生成している場合、それは同じCDB内の他のPDBのパフォーマンスを低下させることがあります。

次の初期化パラメータの一方または両方を使用して、特定のPDBで生成されるI/Oを制限します。

- MAX_IOPS初期化パラメータは、毎秒のI/O操作の数を制限します。
- MAX_MBPS初期化パラメータは、毎秒のI/O操作のMB数を制限します。

単一のPDBに対して前述の両方の初期化パラメータを設定した場合、Oracle Databaseは両方の制限を強制します。I/Oリソース管理(IORM)を使用してPDB間のI/Oを管理するOracle Exadataには、これらの制限が強制されないことに注意してください。

CDBルートが現在のコンテナであるときに、これらの初期化パラメータを設定した場合、値はCDB内のすべてのコンテナのデフォルト値になります。アプリケーション・ルートが現在のコンテナであるときに、これらの初期化パラメータを設定した場合、値はアプリケーション・コンテナ内のすべてのアプリケーションPDBのデフォルト値になります。PDBまたはアプリケーションPDBが現在のコンテナであるときに、これらの初期化パラメータを設定した場合、その設定はCDBルートまたはアプリケーション・ルートのデフォルト設定より優先されます。これらのパラメータは非CDBには設定できません。

両方の初期化パラメータのデフォルトは0(ゼロ)です。これらの初期化パラメータがPDBで0(ゼロ)に設定されていて、CDBルートで0に設定されている場合、PDBのI/O制限はありません。これらの初期化パラメータがアプリケーションPDBで0(ゼロ)に設定されていて、アプリケーション・ルートで0に設定されている場合、アプリケーションPDBのI/O制限はありません。

制御ファイルやパスワード・ファイルに対するI/O操作などのクリティカルなI/O操作は、制限の対象にならず、制限に達した場合も実行を続けます。ただし、I/O操作数およびI/O操作のMB数の計算では、クリティカルなI/O操作を含むすべてのI/O操作がカウントされます。

DBA_HIST_RSRC_PDB_METRICビューを使用して、PDBに対して合理的なI/O制限を計算できます。制限を計算する際には、IOPS、IOMBPS、IOPS_THROTTLE_EXEMPTおよびIOMBPS_THROTTLE_EXEMPT列の値を考慮します。rsmgr:io rate limit 待機イベントは、制限に達したことを示します。

例22-3 PDBによって生成されるI/Oの制限

PDBを現在のコンテナとして、次のSQL文を実行し、メモリーとSPFILEの両方でMAX_IOPS初期化パラメータを毎秒1,000 I/O操作の制限に設定します。

```
ALTER SYSTEM SET MAX_IOPS = 1000 SCOPE = BOTH;
```

例22-4 PDBによって生成されるI/OのMB数の制限

PDBを現在のコンテナとして、次のSQL文を実行し、メモリーとSPFILEの両方でMAX_MBPS初期化パラメータを毎秒200MBのI/Oの制限に設定します。

```
ALTER SYSTEM SET MAX_MBPS = 200 SCOPE = BOTH;
```

関連項目:

- [「システム・レベルでのPDBの変更」](#)
- MAX_IOPS初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。
- MAX_MBPS初期化パラメータの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

親トピック: [PDBレベルのリソースに関する初期化パラメータ](#)

CDBリソース・プランの管理

CDBでは、PDBの優先度のレベルが異なる場合があります。CDBリソース・プランを作成し、これらの優先度に従ってPDBごとにリソースを分配できます。

- [CDBリソース・プランについて](#)
PDBの共有およびリソース制限を割り当てるCDBリソース・プランを作成します。

- [PDBを管理するためのCDBリソース・プランの作成](#)
個々のPDBにCDBリソース・プランを作成し、プランに対してディレクティブを定義するには、DBMS_RESOURCE_MANAGERパッケージを使用します。
- [PDBを管理するためのCDBリソース・プランの作成：使用例](#)
この使用例では、個々のPDBのCDBリソース・プランの作成に必要な各ステップを示します。
- [PDBパフォーマンス・プロファイルを使用したCDBリソース・プランの作成](#)
DBMS_RESOURCE_MANAGERパッケージを使用してPDBパフォーマンス・プロファイルのCDBリソース・プランを作成し、プランに対してディレクティブを定義します。プロファイルを使用する各PDBには、CDBリソース・プラン・ディレクティブが採用されます。
- [PDBパフォーマンス・プロファイルに対するCDBリソース・プランの作成：使用例](#)
この使用例では、PDBパフォーマンス・プロファイルのCDBリソース・プランの作成に必要な各ステップを示します。
- [CDBリソース・プランの有効化](#)
ルートのRESOURCE_MANAGER_PLAN初期化パラメータを設定して、CDBに対してリソース・マネージャを有効化します。
- [CDBリソース・プランの変更](#)
CDBリソース・プランの変更には、プランの更新、PDBのプラン・ディレクティブの作成、更新または削除、デフォルト・ディレクティブの更新などのタスクが含まれます。
- [CDBリソース・プランの無効化](#)
ルートのRESOURCE_MANAGER_PLAN初期化パラメータを解除して、CDBに対してリソース・マネージャを無効化します。
- [CDBのプランおよびディレクティブに関する情報の表示](#)
CDBリソース・プラン、CDBリソース・プラン・ディレクティブおよびCDBにおける事前定義のリソース・プランに関する情報を表示できます。

親トピック: [PDBに対するOracle Resource Managerの使用](#)

CDBリソース・プランについて

PDBの共有およびリソース制限を割り当てるCDBリソース・プランを作成します。

- [PDBへのリソース割当ての共有](#)
PDB間でリソースを割り当てるには、PDBまたはパフォーマンス・プロファイルごとに共有値を割り当てます。共有値が高いほど、パフォーマンス・プロファイルを使用する1つまたは複数のPDBに保証されるリソースは多くなります。
- [PDBの使用率制限](#)
使用率制限は、特定のPDBまたは特定のPDBパフォーマンス・プロファイルのシステム・リソースの使用を制限します。
- [PDBのデフォルト・ディレクティブ](#)
PDBにディレクティブを明示的に定義しない場合、PDBではPDB用のデフォルト・ディレクティブが使用されます。

親トピック: [CDBリソース・プランの管理](#)

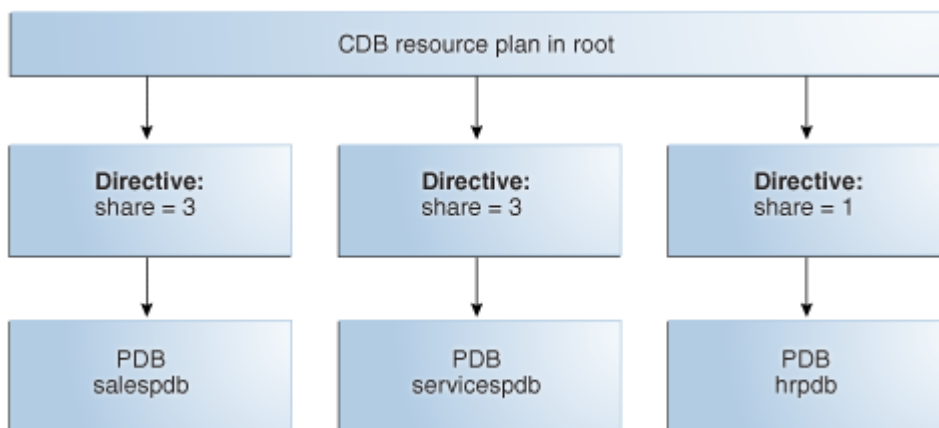
PDBへのリソース割当ての共有

PDB間でリソースを割り当てるには、PDBまたはパフォーマンス・プロファイルごとに共有値を割り当てます。共有値が高いほど、パフォーマンス・プロファイルを使用する1つまたは複数のPDBに保証されるリソースは多くなります。

DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVEプロシージャを使用してPDBの共有値を指定し、DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVEプロシージャを使用してPDBパフォーマンス・プロファイルの共有値を指定します。どちらの場合も、sharesパラメータでPDBの共有値を指定します。複数のPDBが同じPDBパフォーマンス・プロファイルを使用できます。

次の図は、CDBリソース・プランに共有値が指定された3つのPDBの例を示しています。

図22-1 CDBリソース・プランにおける共有



前述の図は、共有の合計数が7 (3+3+1)であることを示しています。PDB salespdbおよびservicespdbは、それぞれリソースの3/7が保証されていますが、PDB hrpdbはリソースの1/7が保証されています。ただし、リソースの競合がない場合は、いずれのPDBでも保証された量より多くのリソースを使用できます。

次の表は、前述の図のPDBに対する、共有値に基づくリソース割当てを示しています。表では、PDBのロードでは、割り当てられたすべてのシステム・リソースを消費することを前提としています。

表22-3 サンプルPDBへのリソース割当て

リソース	リソース割当て	関連項目
CPU	PDB salespdb および servicespdb では、同じ量の CPU リソースを使用できます。PDB salespdb および servicespdb は、それぞれ PDB hrpdb の 3 倍の CPU リソースを保証されています。	このリソースの詳細は、 『Oracle Database 管理者ガイド』 を参照してください
パラレル実行サーバー	PDB salespdb および servicespdb からキューされているパラレル問合せは、同等に選択されます。PDB salespdb および servicespdb からキューされているパラレル問合せは、PDB hrpdb からキューされているパラレル問合せの 3 倍の頻度で選択されます。	このリソースの詳細は、 『Oracle Database 管理者ガイド』 を参照してください

親トピック: [CDBリソース・プランについて](#)

PDBの使用率制限

使用率制限により、特定のPDBまたは特定のPDBパフォーマンス・プロファイルのシステム・リソース使用率が抑制されます。

CPUの使用率制限およびパラレル実行サーバーを指定できます。PDBの使用率制限は、CDBリソース・プランで設定されます。

次の表で、PDBの使用率制限およびPDBが使用率制限に達した場合に実行されるリソース・マネージャの処理について説明します。PDBパフォーマンス・プロファイルで指定した制限の場合、制限はPDBパフォーマンス・プロファイルを使用するすべてのPDBに適用されます。たとえば、pdb1およびpdb20にはパフォーマンス・プロファイルBRONZEがあり、BRONZEには10%に設定されている制限がある場合、pdb1には10%の制限、pdb20には10%の制限が適用されます。

表22-4 PDBの使用率制限

リソース	リソース使用率制限	制限に達した場合のリソース・マネージャの処理
------	-----------	------------------------

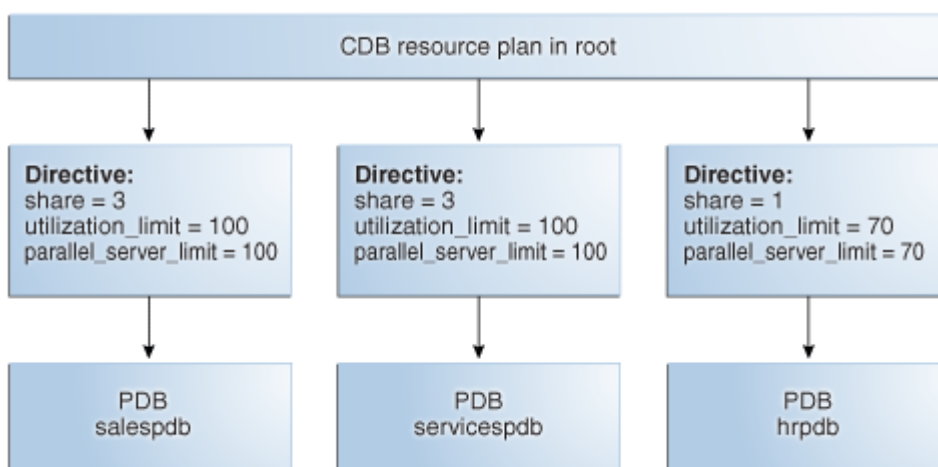
リソース	リソース使用率制限	制限に達した場合のリソース・マネージャの処理
CPU	<p>PDB に接続されているセッションに対する CPU 使用率制限は、DBMS_RESOURCE_MANAGER パッケージのサブプログラムの <code>utilization_limit</code> パラメータで設定されます。<code>utilization_limit</code> パラメータによって、PDB で使用できるシステム・リソースの割合が指定されます。この値の範囲は、0 から 100 までです。</p> <p>また、PDB で CPU を制限するには、<code>CPU_COUNT</code> 初期化パラメータを設定します。たとえば、<code>CPU_COUNT</code> に 8 を設定した場合、PDB は特定の時点で 8 個を超える CPU を使用できません。<code>utilization_limit</code> および <code>CPU_COUNT</code> の両方を指定した場合は、より制限される(低い)値が施行されます。</p>	<p>PDB の CPU 使用率が使用率制限を超えないように、リソース・マネージャによって PDB セッションが制限されます。</p>
<p>パラレル実行サーバー</p>	<p>パラレル文のキューイングを使用して PDB のパラレル実行サーバーの数を制限できます。制限に達した場合は、データベースによってパラレル問合せがキューに追加されるため、制限はキューイング・ポイントです。</p> <p>次のいずれかの方法で制限(キューイング・ポイント)を設定できます。</p> <ul style="list-style-type: none"> ● PDB で設定する <code>PARALLEL_SERVERS_TARGET</code> 初期化パラメータの値 ● CDB リソース・マネージャ計画で PDB に設定された <code>parallel_server_limit</code> ディレクティブの値を乗算した、CDB ルートの <code>PARALLEL_SERVERS_TARGET</code> 初期化パラメータの設定の値 <p>たとえば、CDB ルートで <code>PARALLEL_SERVERS_TARGET</code> 初期化パラメータが 200 に設定され、PDB の <code>parallel_server_limit</code> ディレクティブが 10% に設定されている場合、PDB の使用率制限は 20 パラレル実行サーバー ($200 * .10$)となります。</p> <p>前述の両方の方法で制限が設定されている場合</p>	<p>PDB で使用されるパラレル実行サーバーの数が制限を超えた場合は、リソース・マネージャがパラレル問合せをキューに追加します。</p> <p>ノート: CDB では、パラレル文は PDB および CDB レベルの両方で <code>PARALLEL_SERVERS_TARGET</code> 設定に基づいてキューに追加されます。PDB で使用されるパラレル・サーバーの数が PDB のターゲットを超過したとき、またはすべての PDB によって使用されるパラレル・サーバーの数が CDB のターゲットを超過したときに、文はキューに追加されます。</p>

は、2つのうち低い値が使用されます。
 PARALLEL_SERVERS_TARGET のデフォルト値は、
[『Oracle Database リファレンス』](#)を参照してください。

ノート: CDB プランで parallel_server_limit
 ディレクティブを使用するかわりに、
 PARALLEL_SERVERS_TARGET 初期化パラメータを
 使用することをお勧めします。

次の図は、CDBリソース・プランに共有および使用率制限が指定された3つのPDBの例を示しています。

図22-2 CDBリソース・プランにおける共有および使用率制限



前述の図では、utilization_limitおよびparallel_server_limitが両方ともこれらのPDBについて100%に設定されていることから、PDB salespdbおよびservicespdbに対する使用率制限がないことを示しています。ただし、PDB hrpdbは、utilization_limitおよびparallel_server_limitが両方とも70%に設定されていることから、適用可能なシステム・リソースの70%に制限されています。

ノート:



このシナリオでは、PDB の PARALLEL_SERVERS_TARGET 初期化パラメータに低い制限が指定されていないことを前提としています。PDB でパラレル実行サーバーの PARALLEL_SERVERS_TARGET 初期化パラメータに低い制限が指定されている場合は、より低い制限が使用されます。

関連項目:

- [Oracle Database管理者ガイド](#)
- CPU_COUNTについて学習するには、[『Oracle Databaseリファレンス』](#)を参照してください

親トピック: [CDBリソース・プランについて](#)

PDBのデフォルト・ディレクティブ

PDBにディレクティブを明示的に定義しない場合、PDBではPDB用のデフォルト・ディレクティブが使用されます。

次の表に、PDB用の初期デフォルト・ディレクティブの属性を示します。

表22-5 PDB用の初期デフォルト・ディレクティブ属性

ディレクティブ属性	値
shares	1
utilization_limit	100
parallel_server_limit	100

PDBがCDBに接続されており、ディレクティブが定義されていない場合、PDBではPDB用のデフォルト・ディレクティブが使用されます。

新しいPDBにディレクティブを新しく作成できます。また、DBMS_RESOURCE_MANAGERパッケージのUPDATE_CDB_DEFAULT_DIRECTIVEプロシージャを使用して、PDB用のデフォルト・ディレクティブ属性値を変更することもできます。

PDBがCDBから切断された場合、PDBのディレクティブは保持されます。同じPDBがCDBに再接続される場合、そのPDB用に定義されたディレクティブが手動で削除されていないと、PDBではこのディレクティブが使用されます。

図22-3に、CDBリソース・プランにおけるデフォルト・ディレクティブの例を示します。

図22-3 CDBリソース・プランにおけるデフォルト・ディレクティブ

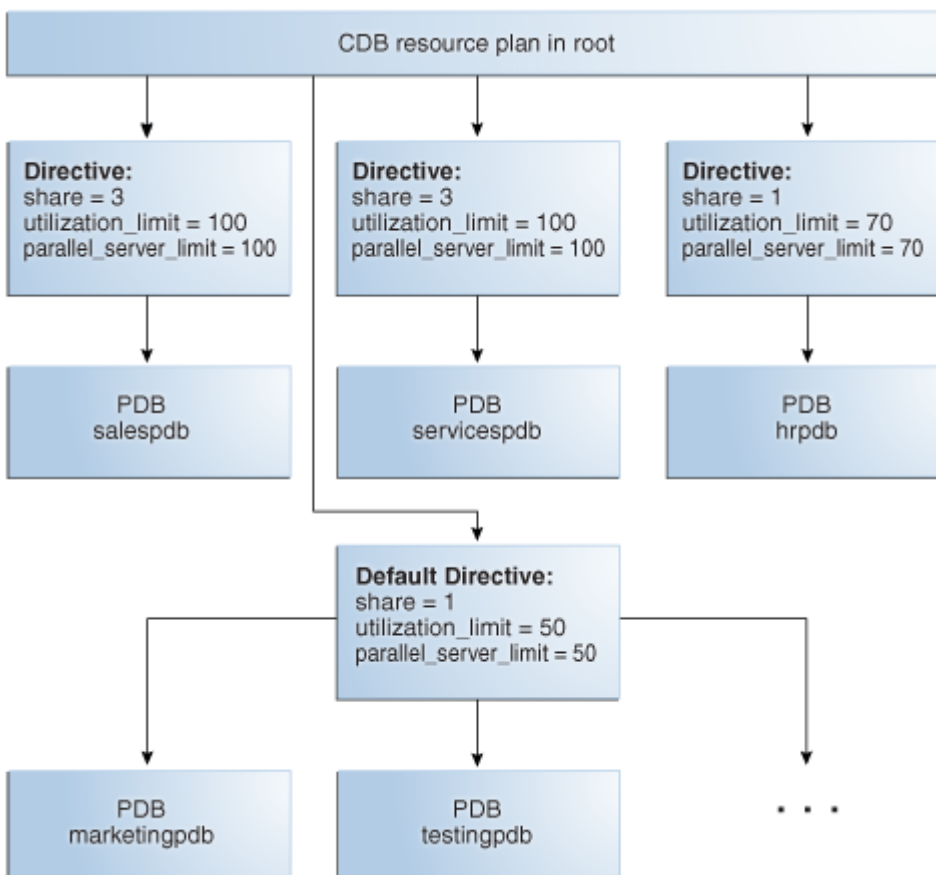


図22-3は、デフォルトPDBディレクティブで、shareが1、utilization_limitが50%およびparallel_server_limitが

50%となるよう指定されていることを示しています。CDBの一部であり、かつディレクティブが定義されていないすべてのPDBでは、デフォルトPDBディレクティブが使用されます。[図22-3](#)は、デフォルトPDBディレクティブを使用するPDB marketingpdbおよびtestingpdbを示しています。したがって、marketingpdbおよびtestingpdbは、それぞれ共有が1および使用率制限が50となります。

関連項目:

- [「PDBの新規CDBリソース・プラン・ディレクティブの作成」](#)
- [「CDBリソース・プランにおけるPDBのデフォルト・ディレクティブの更新」](#)
- [「PDBおよびアプリケーション・コンテナの作成および削除」](#)
- [「CDBからのPDBの切断」](#)
- 平行ル・サーバーの制限の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBリソース・プランについて](#)

PDBを管理するためのCDBリソース・プランの作成

個々のPDBにCDBリソース・プランを作成し、プランに対してディレクティブを定義するには、DBMS_RESOURCE_MANAGERパッケージを使用します。

個々のPDBのCDBリソース・プランの一般的な作成ステップは、次のとおりです。

1. CREATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを作成します。
2. CREATE_CDB_PLANプロシージャを使用して、CDBリソース・プランを作成します。
3. CREATE_CDB_PLAN_DIRECTIVEプロシージャを使用して、PDBにディレクティブを作成します。
4. (オプション) UPDATE_CDB_DEFAULT_DIRECTIVEプロシージャを使用して、デフォルトPDBディレクティブを更新します。
5. VALIDATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを検証します。
6. SUBMIT_PENDING_AREAプロシージャを使用して、ペンディング・エリアを発行します。

親トピック: [CDBリソース・プランの管理](#)

PDBを管理するためのCDBリソース・プランの作成: 使用例

この使用例では、個々のPDBのCDBリソース・プランの作成に必要な各ステップを示します。

この使用例では、newcdbという名前のCDBにCDBリソース・プランを作成する場合を想定しています。プランには、PDBごとにディレクティブを含めます。また、この使用例では、デフォルト・ディレクティブおよびAutoTaskディレクティブも更新します。

ディレクティブは、DBMS_RESOURCE_MANAGERパッケージ内の各種プロシージャを使用して定義されます。各ディレクティブの属性は、これらのプロシージャのパラメータを使用して定義されます。[表22-6](#)で、プランにおけるディレクティブのタイプについて説明します。

表22-6 CDBリソース・プランにおけるPDBディレクティブの属性

ディレクティブ属性	説明	関連項目
shares	CPU リソースおよび平行ル実行サーバー・リソースのリソース割当ての共有。	「PDB へのリソース割当ての共有」

ディレクティブ属性	説明	関連項目
utilization_limit	CPUのリソース使用率制限。	「PDBの使用率制限」
parallel_server_limit	<p>パラレル文をキューイングするまでに、PDBで使用できるパラレル実行サーバーの最大割合。</p> <p>PDBに対して parallel_server_limit ディレクティブが指定されている場合、制限は、CDB ルートの PARALLEL_SERVERS_TARGET 値を CREATE_CDB_PLAN_DIRECTIVE プロシージャの parallel_server_limit パラメータの値で乗算した値になります。</p> <p>ノート: CDB プランで parallel_server_limit ディレクティブを使用するかわりに、PARALLEL_SERVERS_TARGET 初期化パラメータを使用することをお勧めします。</p>	「PDBの使用率制限」

[表22-7](#)で、CDBリソース・プランにより、[表22-6](#)で説明されているディレクティブ属性を使用してリソースがそのPDBにどのように割り当てられるかについて説明します。

表22-7 CDBリソース・プランにおけるPDBのサンプル・ディレクティブ

PDB	sharesディレクティブ	utilization_limitディレクティブ	parallel_server_limitディレクティブ
salespdb	3	無制限	無制限
servicespdb	3	無制限	無制限
hrpdb	1	70	70
デフォルト	1	50	50
AutoTask	1	75	75

PDB salespdbおよびservicespdbは、CDB内の他のPDBよりも重要です。したがって、これらに対しては、高い共有(3)、無制限のCPU使用率リソースおよび無制限のパラレル実行サーバー・リソースが確保されます。

デフォルト・ディレクティブは、特定のディレクティブが定義されていないPDBに適用されます。この使用例では、CDBに、デフォルト・ディレクティブを使用する複数のPDBが含まれていると想定しています。この使用例では、デフォルト・ディレクティブを更新します。

さらに、この使用例では、自動タスク・ディレクティブも更新します。自動タスク・ディレクティブは、ルート・メンテナンス・ウィンドウで実行される自動メンテナンス・タスクに適用されます。

CDBリソース・プランを作成するには:

1. 次のCREATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

2. 次のCREATE_CDB_PLANプロシージャを使用して、newcdb_planという名前のCDBリソース・プランを作成します。

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN(
    plan    => 'newcdb_plan',
    comment => 'CDB resource plan for newcdb');
END;
/
```

3. CREATE_CDB_PLAN_DIRECTIVEプロシージャを使用して、PDBにCDBリソース・プラン・ディレクティブを作成します。各ディレクティブにより、特定のPDBへのリソースの割当て方法が指定されます。

[表22-7](#)で、この使用例のPDB salespdb、servicespdbおよびhrpdbのディレクティブについて説明します。次のプロシージャを実行して、これらのディレクティブを作成します。

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
    plan                => 'newcdb_plan',
    pluggable_database  => 'salespdb',
    shares              => 3,
    utilization_limit   => 100,
    parallel_server_limit => 100);
END;
/
```

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
    plan                => 'newcdb_plan',
    pluggable_database  => 'servicespdb',
    shares              => 3,
    utilization_limit   => 100,
    parallel_server_limit => 100);
END;
/
```

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
    plan                => 'newcdb_plan',
    pluggable_database  => 'hrpdb',
    shares              => 1,
    utilization_limit   => 70,
    parallel_server_limit => 70);
END;
/
```

このCDB内の他のすべてのPDBでは、デフォルトPDBディレクティブが使用されます。

4. PDBの現在のデフォルトCDBリソース・プラン・ディレクティブが要件を満たしていない場合、UPDATE_CDB_DEFAULT_DIRECTIVEプロシージャを使用してディレクティブを更新します。

デフォルト・ディレクティブは、特定のディレクティブが定義されていないPDBに適用されます。詳細は、[「PDBのデフォルト・ディレクティブ」](#)を参照してください。

[表22-7](#)で、この使用例のPDBで使用されているデフォルト・ディレクティブについて説明します。次のプロシージャを実

行して、デフォルト・ディレクティブを更新します。

```
BEGIN
  DBMS_RESOURCE_MANAGER.UPDATE_CDB_DEFAULT_DIRECTIVE (
    plan           => 'newcdb_plan',
    new_shares     => 1,
    new_utilization_limit => 50,
    new_parallel_server_limit => 50);
END;
/
```

5. 次のVALIDATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA ();
```

6. 次のSUBMIT_PENDING_AREAプロシージャを使用して、ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA ();
```

親トピック: [CDBリソース・プランの管理](#)

PDBパフォーマンス・プロファイルを使用するCDBリソース・プランの作成

DBMS_RESOURCE_MANAGERパッケージを使用してPDBパフォーマンス・プロファイルのCDBリソース・プランを作成し、プランに対してディレクティブを定義します。プロファイルを使用する各PDBには、CDBリソース・プラン・ディレクティブが採用されます。

PDBパフォーマンス・プロファイルのあるCDBリソース・プランの一般的な作成ステップは、次のとおりです。

1. CREATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを作成します。
2. CREATE_CDB_PLANプロシージャを使用して、CDBリソース・プランを作成します。
3. CREATE_CDB_PROFILE_DIRECTIVEプロシージャを使用して、PDBパフォーマンス・プロファイルのディレクティブを作成します。
4. (オプション) UPDATE_CDB_DEFAULT_DIRECTIVEプロシージャを使用して、デフォルトPDBディレクティブを更新します。
5. VALIDATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを検証します。
6. SUBMIT_PENDING_AREAプロシージャを使用して、ペンディング・エリアを発行します。
7. プロファイルを使用する各PDBについて、DB_PERFORMANCE_PROFILE初期化パラメータを設定し、プロファイル名を指定します。

親トピック: [CDBリソース・プランの管理](#)

PDBパフォーマンス・プロファイルに対するCDBリソース・プランの作成: 使用例

この使用例では、PDBパフォーマンス・プロファイルに対するCDBリソース・プランの作成に必要な各ステップを示します。

この使用例では、newcdbという名前のCDBにCDBリソース・プランを作成する場合を想定しています。プランには、PDBパフォーマンス・プロファイルごとにディレクティブを含めます。また、この使用例では、デフォルト・ディレクティブおよびAutoTaskディレクティブも更新します。

CDBリソース・プランで、各プロファイルに名前を付けます。各PDBで、DB_PERFORMANCE_PROFILE初期化パラメータを設定して、PDBが使用するPDBパフォーマンス・プロファイルを指定します。

ディレクティブは、DBMS_RESOURCE_MANAGERパッケージ内の各種プロシージャを使用して定義されます。各ディレクティブの属性は、これらのプロシージャのパラメータを使用して定義されます。次の表で、プランにおけるディレクティブのタイプについて説明します。

表22-8 CDBリソース・プランにおけるPDBパフォーマンス・プロファイル・ディレクティブの属性

ディレクティブ属性	説明	関連項目
shares	CPU リソースおよびパラレル実行サーバー・リソースのリソース割当ての共有。	「PDB へのリソース割当ての共有」
utilization_limit	CPU のリソース使用率制限。	「PDB の使用率制限」
parallel_server_limit	<p>PDB で使用できるパラレル実行サーバーの最大割合。</p> <p>PDB パフォーマンス・プロファイルに対して parallel_server_limit ディレクティブが指定されている場合、制限は、CDB ルートの PARALLEL_SERVERS_TARGET 初期化パラメータの設定値を CREATE_CDB_PROFILE_DIRECTIVE プロシージャの parallel_server_limit パラメータの値で乗算した値になります。</p>	「PDB の使用率制限」

次の表で、CDBリソース・プランにより、[表22-8](#)で説明されているディレクティブ属性を使用してリソースがそのPDBパフォーマンス・プロファイルにどのように割り当てられるかについて説明します。

表22-9 CDBリソース・プランにおけるPDBパフォーマンス・プロファイルのサンプル・ディレクティブ

PDB	sharesディレクティブ	utilization_limitディレクティブ	parallel_server_limitディレクティブ
gold	3	無制限	無制限
silver	2	40	40
bronze	1	20	20
デフォルト	1	10	10
AutoTask	2	60	60

デフォルト・ディレクティブは、特定のディレクティブが定義されていないPDBに適用されます。この使用例では、CDBに、デフォルト・ディレクティブを使用する複数のPDBが含まれていると想定しています。この使用例では、デフォルト・ディレクティブを更新します。

さらに、この使用例では、自動タスク・ディレクティブも更新します。自動タスク・ディレクティブは、ルート・メンテナンス・ウィンドウで実行される自動メンテナンス・タスクに適用されます。

PDBパフォーマンス・プロファイルに対してCDBリソース・プランを作成するには:

1. プロファイルを使用する各PDBについて、DB_PERFORMANCE_PROFILE初期化パラメータをそのPDBが使用するプロファ

イルの名前に設定します。

- a. ALTER SYSTEM文を実行してパラメータを設定します。

たとえば、PDBを現在のコンテナとして、次のSQL文を実行します。

```
ALTER SYSTEM SET DB_PERFORMANCE_PROFILE=gold SCOPE=spfile;
```

- b. PDBをクローズします。

```
ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;
```

- c. PDBをオープンします。

```
ALTER PLUGGABLE DATABASE OPEN;
```

2. 次のCREATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. 次のCREATE_CDB_PLANプロシージャを使用して、newcdb_planという名前のCDBリソース・プランを作成します。

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN(
    plan    => 'newcdb_plan',
    comment => 'CDB resource plan for newcdb');
END;
/
```

4. CREATE_CDB_PLAN_DIRECTIVEプロシージャを使用して、PDBにCDBリソース・プラン・ディレクティブを作成します。各ディレクティブにより、特定のPDBへのリソースの割当て方法が指定されます。

[表22-7](#)は、このシナリオのgold、silverおよびbronzeプロファイルのディレクティブを示しています。次のプロシージャを実行して、これらのディレクティブを作成します。

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE(
    plan          => 'newcdb_plan',
    profile       => 'gold',
    shares        => 3,
    utilization_limit => 100,
    parallel_server_limit => 100);
END;
/

BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE(
    plan          => 'newcdb_plan',
    profile       => 'silver',
    shares        => 2,
    utilization_limit => 40,
    parallel_server_limit => 40);
END;
/

BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE(
    plan          => 'newcdb_plan',
    profile       => 'bronze',
    shares        => 1,
    utilization_limit => 20,
```



```
parallel_server_limit => 20);
END;
/
```

このCDB内の他のすべてのPDBでは、デフォルトPDBディレクティブが使用されます。

5. PDBの現在のデフォルトCDBリソース・プラン・ディレクティブが要件を満たしていない場合、UPDATE_CDB_DEFAULT_DIRECTIVEプロシージャを使用してディレクティブを更新します。

デフォルト・ディレクティブは、特定のディレクティブが定義されていないPDBに適用されます。

[表22-7](#)で、この使用例のPDBで使用されているデフォルト・ディレクティブについて説明します。次のプロシージャを実行して、デフォルト・ディレクティブを更新します。

```
BEGIN
  DBMS_RESOURCE_MANAGER.UPDATE_CDB_DEFAULT_DIRECTIVE(
    plan          => 'newcdb_plan',
    new_shares    => 1,
    new_utilization_limit => 10,
    new_parallel_server_limit => 10);
END;
/
```

6. 次のVALIDATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

7. 次のSUBMIT_PENDING_AREAプロシージャを使用して、ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

[「PDBのデフォルト・ディレクティブ」](#)

親トピック: [CDBリソース・プランの管理](#)

CDBリソース・プランの有効化

ルートのRESOURCE_MANAGER_PLAN初期化パラメータを設定して、CDBに対してリソース・マネージャを有効化します。

このパラメータにより、CDBの現行インスタンスに使用されるトップレベルのプランが指定されます。このパラメータでプランを指定しない場合、リソース・マネージャは有効になりません。

前提条件

CDBリソース・プランを有効化する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

CDBリソース・プランを有効化するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. 次のいずれかのアクションを実行します。
 - ALTER SYSTEM文を使用して、RESOURCE_MANAGER_PLAN初期化パラメータをCDBリソース・プランに設定します。

次の例では、ALTER SYSTEM文を使用して、CDBリソース・プランをnewcdb_planに設定しています。

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'newcdb_plan';
```

- テキスト形式の初期化パラメータ・ファイルで、RESOURCE_MANAGER_PLAN初期化パラメータをCDBリソース・プランに設定し、CDBを再起動します。

次の例では、初期化パラメータ・ファイルのCDBリソース・プランをnewcdb_planに設定しています。

```
RESOURCE_MANAGER_PLAN = 'newcdb_plan'
```

関連項目:

- [「CDB内のテナントへのアクセスについて」](#)
- Oracle Schedulerを使用してCDBリソース・プランの変更をスケジュールする方法を学習するには、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBリソース・プランの管理](#)

CDBリソース・プランの変更

CDBリソース・プランの変更には、プランの更新、PDBのプラン・ディレクティブの作成、更新または削除、デフォルト・ディレクティブの更新などのタスクが含まれます。

- [CDBリソース・プランの更新](#)
UPDATE_CDB_PLANプロシージャを使用すると、CDBリソース・プランのコメントを変更して更新できます。
- [PDBのCDBリソース・プラン・ディレクティブの管理](#)
PDBのCDBリソース・プラン・ディレクティブを作成、更新および削除できます。
- [PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの管理](#)
PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを作成、更新および削除できます。
- [CDBリソース・プランにおけるPDBのデフォルト・ディレクティブの更新](#)
UPDATE_CDB_DEFAULT_DIRECTIVEプロシージャを使用して、CDBリソース・プランでPDBのデフォルト・ディレクティブを更新できます。デフォルト・ディレクティブは、特定のディレクティブが定義されていないPDBに適用されます。
- [CDBリソース・プランにおけるメンテナンス・タスクのデフォルト・ディレクティブの更新](#)
UPDATE_CDB_AUTOTASK_DIRECTIVEプロシージャを使用して、CDBリソース・プランでAutoTaskディレクティブを更新できます。自動タスク・ディレクティブは、ルート・メンテナンス・ウィンドウで実行される自動メンテナンス・タスクに適用されます。
- [CDBリソース・プランの削除](#)
DELETE_CDB_PLANプロシージャを使用して、CDBリソース・プランを削除できます。

親トピック: [CDBリソース・プランの管理](#)

CDBリソース・プランの更新

UPDATE_CDB_PLANプロシージャを使用すると、CDBリソース・プランのコメントを変更して更新できます。

前提条件

CDBリソース・プランを更新する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

CDBリソース・プランを更新するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。

2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. UPDATE_CDB_PLANプロシージャを実行して、new_commentパラメータに新しいコメントを入力します。

たとえば、次のプロシージャを実行すると、newcdb_plan CDBリソース・プランのコメントが変更されます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.UPDATE_CDB_PLAN(
    plan      => 'newcdb_plan',
    new_comment => 'CDB plan for PDBs in newcdb');
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- [「CDBリソース・プランについて」](#)
- [「CDB内のコンテナへのアクセスについて」](#)

親トピック: [CDBリソース・プランの変更](#)

PDBのCDBリソース・プラン・ディレクティブの管理

PDBのCDBリソース・プラン・ディレクティブを作成、更新および削除できます。

- [PDBの新規CDBリソース・プラン・ディレクティブの作成](#)
CDBにPDBを作成する場合、CREATE_CDB_PLAN_DIRECTIVEプロシージャを使用して、PDBにCDBリソース・プラン・ディレクティブを作成できます。このディレクティブにより、新しいPDBへのリソースの割当て方法が指定されます。
- [PDBのCDBリソース・プラン・ディレクティブの更新](#)
UPDATE_CDB_PLAN_DIRECTIVEプロシージャを使用して、PDBのCDBリソース・プラン・ディレクティブを更新できます。このディレクティブにより、PDBへのリソースの割当て方法が指定されます。
- [PDBのCDBリソース・プラン・ディレクティブの削除](#)
DELETE_CDB_PLAN_DIRECTIVEプロシージャを使用して、PDBのCDBリソース・プラン・ディレクティブを削除できます。

親トピック: [CDBリソース・プランの変更](#)

PDBの新規CDBリソース・プラン・ディレクティブの作成

CDBにPDBを作成する場合、CREATE_CDB_PLAN_DIRECTIVEプロシージャを使用して、PDBにCDBリソース・プラン・ディレクティブを作成できます。このディレクティブにより、新しいPDBへのリソースの割当て方法が指定されます。

前提条件

PDBの新しいCDBリソース・プラン・ディレクティブを作成する前に、[「マルチテナント環境でのOracle Resource Managerの](#)

[概要](#)で説明されている前提条件を満たします。

PDBに新しいCDBリソース・プラン・ディレクティブを作成するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. CREATE_CDB_PLAN_DIRECTIVEプロシージャを実行して、新しいPDBに適切な値を指定します。

たとえば、次のプロシージャを実行すると、newcdb_plan CDBリソース・プランのoperpdbという名前のPDBにリソースが割り当てられます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE (
    plan              => 'newcdb_plan',
    pluggable_database => 'operpdb',
    shares            => 1,
    utilization_limit  => 20,
    parallel_server_limit => 30);
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- [「CDBリソース・プランについて」](#)
- [「CDB内のコンテナへのアクセスについて」](#)

親トピック: [PDBのCDBリソース・プラン・ディレクティブの管理](#)

PDBのCDBリソース・プラン・ディレクティブの更新

UPDATE_CDB_PLAN_DIRECTIVEプロシージャを使用して、PDBのCDBリソース・プラン・ディレクティブを更新できます。このディレクティブにより、PDBへのリソースの割当て方法が指定されます。

前提条件

PDBのCDBリソース・プラン・ディレクティブを更新する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たしていることを確認します。

PDBのCDBリソース・プラン・ディレクティブを更新するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. UPDATE_CDB_PLAN_DIRECTIVEプロシーダを実行して、PDBに新しいリソース割当て値を指定します。

たとえば、次のプロシーダを実行すると、newcdb_plan CDBリソース・プランのoperpdbという名前のPDBに対するリソース割当てが更新されます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.UPDATE_CDB_PLAN_DIRECTIVE (
    plan                => 'newcdb_plan',
    pluggable_database  => 'operpdb',
    new_shares          => 1,
    new_utilization_limit => 10,
    new_parallel_server_limit => 20);
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- [「CDB内のテナントへのアクセスについて」](#)
- [「CDBリソース・プランについて」](#)

親トピック: [PDBのCDBリソース・プラン・ディレクティブの管理](#)

PDBのCDBリソース・プラン・ディレクティブの削除

DELETE_CDB_PLAN_DIRECTIVEプロシーダを使用して、PDBのCDBリソース・プラン・ディレクティブを削除できます。

PDBを切断または削除すると、PDBのディレクティブが削除される場合があります。ただし、ディレクティブは保持でき、将来PDBがCDBに接続された場合に、既存のディレクティブがPDBに適用されます。

前提条件

PDBのCDBリソース・プラン・ディレクティブを削除する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

PDBのCDBリソース・プラン・ディレクティブを削除するには:

1. SQL*Plusで、現在のテナントがルートであることを確認します。
2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. DELETE_CDB_PLAN_DIRECTIVEプロシーダを実行して、CDBリソース・プランおよびPDBを指定します。

たとえば、次のプロシーダを実行すると、newcdb_plan CDBリソース・プランのoperpdbという名前のPDBに対するディレクティブが削除されます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.DELETE_CDB_PLAN_DIRECTIVE (
```

```
plan          => 'newcdb_plan',
pluggable_database => 'operpdb');
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- [「CDB内のテナントへのアクセスについて」](#)
- [「CDBリソース・プランについて」](#)

親トピック: [PDBのCDBリソース・プラン・ディレクティブの管理](#)

PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの管理

PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを作成、更新および削除できます。

- [PDBパフォーマンス・プロファイルの新規CDBリソース・プラン・ディレクティブの作成](#)
新規PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブは、CREATE_CDB_PROFILE_DIRECTIVEプロシージャを使用して作成できます。ディレクティブにより、新しいプロファイルを使用するすべてのPDBのリソースの割当て方法を指定します。
- [PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの更新](#)
UPDATE_CDB_PROFILE_DIRECTIVEプロシージャを使用して、PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを更新します。このディレクティブにより、PDBパフォーマンス・プロファイルを使用するPDBへのリソースの割当て方法が指定されます。
- [PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの削除](#)
DELETE_CDB_PROFILE_DIRECTIVEプロシージャを使用して、PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを削除できます。

親トピック: [CDBリソース・プランの変更](#)

PDBパフォーマンス・プロファイルの新規CDBリソース・プラン・ディレクティブの作成

新規PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブは、CREATE_CDB_PROFILE_DIRECTIVEプロシージャを使用して作成できます。ディレクティブにより、新しいプロファイルを使用するすべてのPDBのリソースの割当て方法を指定します。

前提条件

PDBパフォーマンス・プロファイルの新しいCDBリソース・プラン・ディレクティブを作成する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

PDBパフォーマンス・プロファイルに新しいCDBリソース・プラン・ディレクティブを作成するには:

1. SQL*Plusで、現在のテナントがルートであることを確認します。

2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. CREATE_CDB_PROFILE_DIRECTIVE プロシージャを実行して、新しいPDBパフォーマンス・プロファイルに適切な値を指定します。

たとえば、次のプロシージャを実行すると、newcdb_plan CDBリソース・プランのcopperという名前のPDBパフォーマンス・プロファイルにリソースが割り当てられます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE (
    plan          => 'newcdb_plan',
    profile       => 'copper',
    shares        => 1,
    utilization_limit => 20,
    parallel_server_limit => 30);
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```



ノート:

PDB で新規プロファイルを使用するには、PDB の DB_PERFORMANCE_PROFILE 初期化パラメータがプロファイル名に設定されている必要があります。

関連項目:

- [「CDBリソース・プランについて」](#)
- [「CDB内のコンテナへのアクセスについて」](#)

親トピック: [PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの管理](#)

PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの更新

UPDATE_CDB_PROFILE_DIRECTIVE プロシージャを使用して、PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを更新します。このディレクティブにより、PDBパフォーマンス・プロファイルを使用するPDBへのリソースの割当て方法が指定されます。

PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを更新する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを更新するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。

2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. UPDATE_CDB_PROFILE_DIRECTIVEプロシージャを実行して、PDBパフォーマンス・プロファイルに新しいリソース割当て値を指定します。

たとえば、次のプロシージャを実行すると、newcdb_plan CDBリソース・プランのcopperという名前のPDBパフォーマンス・プロファイルに対するリソース割当てが更新されます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.UPDATE_CDB_PROFILE_DIRECTIVE (
    plan           => 'newcdb_plan',
    profile        => 'copper',
    new_shares     => 1,
    new_utilization_limit => 10,
    new_parallel_server_limit => 20);
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- [「CDBリソース・プランについて」](#)
- [「CDB内のコンテナへのアクセスについて」](#)

親トピック: [PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの管理](#)

PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの削除

DELETE_CDB_PROFILE_DIRECTIVEプロシージャを使用して、PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを削除できます。

PDBでパフォーマンス・プロファイルを使用しない場合は、プロファイルのディレクティブを削除できます。

前提条件

PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを削除する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブを削除するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. DELETE_CDB_PROFILE_DIRECTIVEプロシージャを実行して、CDBリソース・プランおよびPDBパフォーマンス・プロファイ

ルを指定します。

たとえば、次のプロシーダを実行すると、newcdb_plan CDBリソース・プランのoperpdbという名前のPDBに対するディレクティブが削除されます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.DELETE_CDB_PLAN_DIRECTIVE (
    plan    => 'newcdb_plan',
    profile => 'operpdb');
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- [「CDB内のコンテナへのアクセスについて」](#)
- [「CDBリソース・プランについて」](#)

親トピック: [PDBパフォーマンス・プロファイルのCDBリソース・プラン・ディレクティブの管理](#)

CDBリソース・プランにおけるPDBのデフォルト・ディレクティブの更新

UPDATE_CDB_DEFAULT_DIRECTIVEプロシーダを使用して、CDBリソース・プランでPDBのデフォルト・ディレクティブを更新できます。デフォルト・ディレクティブは、特定のディレクティブが定義されていないPDBに適用されます。

前提条件

CDBリソース・プランでPDBのデフォルト・ディレクティブを更新する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

CDBリソース・プランでPDBのデフォルトディレクティブを更新するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. UPDATE_CDB_DEFAULT_DIRECTIVEプロシーダを実行して、適切なデフォルトのリソース割当て値を指定します。

たとえば、次のプロシーダを実行すると、newcdb_plan CDBリソース・プランにおけるPDBのデフォルト・ディレクティブが更新されます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.UPDATE_CDB_DEFAULT_DIRECTIVE (
    plan                => 'newcdb_plan',
    new_shares          => 2,
    new_utilization_limit => 40);
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- 詳細は、[「PDBのデフォルト・ディレクティブ」](#)を参照してください。
- [「CDBリソース・プランについて」](#)
- [「CDB内のコンテナへのアクセスについて」](#)を参照してください。

親トピック: [CDBリソース・プランの変更](#)

CDBリソース・プランにおけるメンテナンス・タスクのデフォルト・ディレクティブの更新

UPDATE_CDB_AUTOTASK_DIRECTIVEプロシージャを使用して、CDBリソース・プランでAutoTaskディレクティブを更新できます。自動タスク・ディレクティブは、ルート・メンテナンス・ウィンドウで実行される自動メンテナンス・タスクに適用されます。

前提条件

CDBリソース・プランでメンテナンス・タスクのデフォルト・ディレクティブを更新する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

CDBリソース・プランでメンテナンス・タスクのAutoTaskディレクティブを更新するには:

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. UPDATE_CDB_AUTOTASK_DIRECTIVEプロシージャを実行して、適切なAutoTaskのリソース割当て値を指定します。

たとえば、次のプロシージャを実行すると、newcdb_plan CDBリソース・プランにおけるメンテナンス・タスクのAutoTaskディレクティブが更新されます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.UPDATE_CDB_AUTOTASK_DIRECTIVE (
    plan          => 'newcdb_plan',
    new_shares    => 2,
    new_utilization_limit => 60);
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- [「CDB内のテナントへのアクセスについて」](#)
- [「CDBリソース・プランについて」](#)

親トピック: [CDBリソース・プランの変更](#)

CDBリソース・プランの削除

DELETE_CDB_PLANプロシージャを使用して、CDBリソース・プランを削除できます。

リソース・プランを無効にする必要があります。CDBリソース・プランが必要なくなった場合は、このプランを削除できます。異なるCDBリソース・プランを有効化したり、CDBに対してリソース・マネージャを無効化できます。

前提条件

CDBリソース・プランを削除する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

CDBリソース・プランを削除するには:

1. SQL*Plusで、現在のテナントがルートであることを確認します。
2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. DELETE_CDB_PLANプロシージャを実行して、CDBリソース・プランを指定します。

たとえば、次のプロシージャを実行すると、newcdb_plan CDBリソース・プランが削除されます。

```
BEGIN
  DBMS_RESOURCE_MANAGER.DELETE_CDB_PLAN(
    plan => 'newcdb_plan');
END;
/
```

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- [「CDBリソース・プランについて」](#)
- [「CDB内のテナントへのアクセスについて」](#)を参照してください。
- [「CDBリソース・プランの有効化」](#)
- [「CDBリソース・プランの無効化」](#)

親トピック: [CDBリソース・プランの変更](#)

CDBリソース・プランの無効化

RESOURCE_MANAGER_PLAN初期化パラメータの設定を解除して、CDBに対してリソース・マネージャを無効化します。

PDB間およびPDB内のいずれの場合もCPUの管理を有効化するには、PDBの共有または使用率制限を指定するCDBリソース・プランが必要です。PDBの共有または使用率制限が指定されたリソース・プランが有効で、CDBリソース・プランが指定されていない場合、CDBリソース・プランはDEFAULT_CDB_PLANに設定されます。この設定では、すべてのPDBに等しい共有が提供され、使用率制限は指定されません。CDB全体のCPUリソース管理を無効化するには、RESOURCE_MANAGER_PLANをORA\$INTERNAL_CDB_PLANに設定します。

前提条件

CDBリソース・プランを無効化する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

CDBリソース・プランを無効化するには：

1. SQL*Plusで、現在のコンテナがルートであることを確認します。
2. 次のいずれかのアクションを実行します。
 - ALTER SYSTEM文を使用して、CDBに対してRESOURCE_MANAGER_PLAN初期化パラメータの設定を解除します。

次の例では、ALTER SYSTEM文を使用して、RESOURCE_MANAGER_PLAN初期化パラメータの設定を解除しています。

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = '';
```

- 初期化パラメータ・ファイルで、RESOURCE_MANAGER_PLAN初期化パラメータの設定を解除し、CDBを再起動します。

次の例では、初期化パラメータ・ファイルのRESOURCE_MANAGER_PLAN初期化パラメータの設定を解除しています。

```
RESOURCE_MANAGER_PLAN =
```

関連項目：

- [「CDB内のコンテナへのアクセスについて」](#)
- [「CDBインスタンスの停止」](#)
- データベースの起動の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBリソース・プランの管理](#)

CDBのプランおよびディレクティブに関する情報の表示

CDBリソース・プラン、CDBリソース・プラン・ディレクティブおよびCDBにおける事前定義のリソース・プランに関する情報を表示できます。

- [CDBリソース・プランの表示](#)
DBA_CDB_RSRC_PLANSビューを使用して、CDBに定義されているすべてのCDBリソース・プランを表示する例を示しま

- す。
- [CDBリソース・プラン・ディレクティブの表示](#)
DBA_CDB_RSRC_PLAN_DIRECTIVESビューを使用して、CDB内のすべてのCDBリソース・プランで定義されたすべてのディレクティブを表示する例を示します。

関連項目:

Oracle Database Resource Managerの監視の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください

親トピック: [CDBリソース・プランの管理](#)

CDBリソース・プランの表示

DBA_CDB_RSRC_PLANSビューを使用して、CDBに定義されているすべてのCDBリソース・プランを表示する例を示します。DEFAULT_CDB_PLANは、Oracle Databaseに付属しています。使用要件に応じて、このデフォルト・プランを使用できます。

CDBリソース・プランを表示するには:

1. SQL*PlusまたはSQL Developerを起動し、CDBルートにログインします。
2. 次の問合せを実行します。

```
COLUMN PLAN FORMAT A30
COLUMN STATUS FORMAT A10
COLUMN COMMENTS FORMAT A35

SELECT PLAN, STATUS, COMMENTS
FROM   DBA_CDB_RSRC_PLANS
ORDER BY PLAN;
```

出力は次のようになります。

PLAN	STATUS	COMMENTS
DEFAULT_CDB_PLAN		Default CDB plan
DEFAULT_MAINTENANCE_PLAN		Default CDB maintenance plan
NEWCDB_PLAN		CDB plan for PDBs in newcdb
ORA\$INTERNAL_CDB_PLAN		Internal CDB plan

ノート:



ペンディング・エリア内のプランのステータスは PENDING です。ペンディング・エリア内のプランは編集中です。ペンディング・エリアに存在しないすべてのプランのステータスは、NULL です。

関連項目:

[「CDBリソース・プランについて」](#)

CDBリソース・プラン・ディレクティブの表示

DBA_CDB_RSRC_PLAN_DIRECTIVESを使用して、CDB内のすべてのCDBリソース・プランで定義されたすべてのディレクティブを表示する例を示します。

DEFAULT_CDB_PLANは、Oracle Databaseに付属のデフォルトCDBプランです。DEFAULT_CDB_PLANでは、各PDBに共有が1および使用率制限が100と指定されています。CDBリソース・プランにCPUディレクティブが構成されていない、つまりsharesおよびutilization_limitsディレクティブが設定解除されている場合は、CPUリソース・マネージャによって、PDBレベルのCPU_MIN_COUNTおよびCPU_COUNTパラメータを使用してCPUが管理されます。ORA\$DEFAULT_PDB_DIRECTIVEは、PDBのデフォルト・ディレクティブです。

CDBリソース・プラン・ディレクティブを表示するには:

1. SQL*PlusまたはSQL Developerを起動し、CDBルートにログインします。
2. 次の問合せを実行します。

```
COLUMN PLAN HEADING 'Plan' FORMAT A24
COLUMN PLUGGABLE_DATABASE HEADING 'Pluggable Database' FORMAT A25
COLUMN SHARES HEADING 'Shares' FORMAT 999
COLUMN UTILIZATION_LIMIT HEADING 'Utilization|Limit' FORMAT 999
COLUMN PARALLEL_SERVER_LIMIT HEADING 'Parallel|Server|Limit' FORMAT 999

SELECT PLAN,
       PLUGGABLE_DATABASE,
       SHARES,
       UTILIZATION_LIMIT,
       PARALLEL_SERVER_LIMIT
FROM DBA_CDB_RSRC_PLAN_DIRECTIVES
ORDER BY PLAN;
```

出力は次のようになります。

Plan	Pluggable Database	Shares	Utilization Limit	Parallel Server Limit
DEFAULT_CDB_PLAN	ORA\$DEFAULT_PDB_DIRECTIVE	1	100	100
DEFAULT_CDB_PLAN	ORA\$AUTOTASK		90	100
DEFAULT_MAINTENANCE_PLAN	ORA\$AUTOTASK		90	100
DEFAULT_MAINTENANCE_PLAN	ORA\$DEFAULT_PDB_DIRECTIVE	1	100	100
NEWCDB_PLAN	HRPDB	1	70	70
NEWCDB_PLAN	SALESPDB	3	100	100
NEWCDB_PLAN	ORA\$DEFAULT_PDB_DIRECTIVE	1	50	50
NEWCDB_PLAN	ORA\$AUTOTASK	1	75	75
NEWCDB_PLAN	SERVICESPDB	3	100	100

前述の出力は、[「PDBを管理するためのCDBリソース・プランの作成: 使用例」](#)で作成されて[「CDBリソース・プランの変更」](#)で変更された、newcdb_planのディレクティブを示しています。

関連項目:

- [「CDBリソース・プランについて」](#)
- [「PDBのデフォルト・ディレクティブ」](#)

親トピック: [CDBのプランおよびディレクティブに関する情報の表示](#)

PDBリソース・プランの管理

個々のPDBのリソース・プランを作成し、有効化および変更できます。

- [PDBリソース・プランについて](#)
PDBリソース・プランは、このPDB内で特定のPDBのリソースがコンシューマ・グループにどのように割り当てられるかを決定します。
- [PDBリソース・プランの作成](#)
PDBリソース・プランは、非CDBのリソース・プランを作成する場合と同じ方法で作成します。プランを作成するには、DBMS_RESOURCE_MANAGER PL/SQLパッケージ内のプロシージャを使用します。
- [PDBリソース・プランの有効化](#)
現在のコンテナがPDBの場合、ALTER SYSTEM文を使用してRESOURCE_MANAGER_PLAN初期化パラメータをプランに設定し、PDBリソース・プランを有効化します。
- [PDBリソース・プランの変更](#)
DBMS_RESOURCE_MANAGERパッケージを使用すると、非CDBのリソース・プランを変更する場合と同じ方法でPDBリソース・プランを変更できます。
- [PDBリソース・プランの無効化](#)
PDBのRESOURCE_MANAGER_PLAN初期化パラメータの設定を解除して、PDBリソース・プランを無効化します。

親トピック: [PDBに対するOracle Resource Managerの使用](#)

PDBリソース・プランについて

PDBリソース・プランは、このPDB内で特定のPDBのリソースがコンシューマ・グループにどのように割り当てられるかを決定します。

PDBリソース・プランは、非CDBのリソース・プランと似ています。PDBリソース・プランは、PDBごとに割り当てられるリソース量を決定するCDBリソース・プランとは異なります。

PDBリソース・プランには、次の制限が適用されます。

- PDBリソース・プランは、サブプランを含むことができません。
- PDBリソース・プランで、複数レベルのスケジューリング・ポリシーを使用することはできません。

非CDBを使用するPDBを作成するときに、非CDBにリソース・プランが含まれていない場合は、これらのリソース・プランが前述の制限事項に従っていないことがあります。この場合、これらのリソース・プランは、Oracle Databaseによってこれらの要件を満たす同等のPDBリソース・プランに自動的に変換されます。元のリソース・プランおよびディレクティブは、DBA_RSRC_PLANSおよびDBA_RSRC_PLAN_DIRECTIVESビューにLEGACYステータスで記録されます。

- [PDBリソース・プランを作成する場合のCDBリソース・プランの要件](#)
PDBリソース・プランを作成する場合は、CDBリソース・プランが特定の要件を満たしている必要があります。
- [PDBリソース・プラン: 例](#)
CDBリソース・プランとPDBリソース・プランには、1対多の関係があります。

関連項目:

- [「CDBリソース・プランについて」](#)

- [「非CDBからPDBを作成するためのオプション」](#)
- リソース・プランについてさらに学習するには、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [PDBリソース・プランの管理](#)

PDBリソース・プランを作成する場合のCDBリソース・プランの要件

PDBリソース・プランを作成する場合は、CDBリソース・プランが特定の要件を満たしている必要があります。

CDBリソース・プランのディレクティブを作成するには、DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVEプロシージャを使用します。PDBリソース・プランにディレクティブを作成する場合は、同じパッケージ内のCREATE_PLAN_DIRECTIVEプロシージャを使用します。1つ以上のPDBリソース・プランを作成するときにCDBリソース・プランがない場合、CDBでは、Oracle Databaseに付属のDEFAULT_CDB_PLANが使用されます。

次の表で、CDBリソース・プランの要件および要件が満たされない場合の結果について説明します。「CDBリソース・プラン要件」列で説明されているパラメータ値は、CREATE_CDB_PLAN_DIRECTIVEプロシージャ用です。「要件が満たされない場合の結果」列で説明されているパラメータ値は、CREATE_PLAN_DIRECTIVEプロシージャ用です。

表22-10 PDBリソース・プランのためのCDBリソース・プラン要件

リソース	CDBリソース・プラン要件	要件が満たされない場合の結果
CPU	<p>次の要件のいずれかが満たされている必要があります。</p> <ul style="list-style-type: none"> ● 共有値が、shares パラメータを使用して PDB に指定されている必要があります。 ● 100 を下回る CPU 使用率制限が、utilization_limit パラメータを使用して PDB に指定されている必要があります。 <p>これらの値は、特定の PDB のディレクティブまたはデフォルト・ディレクティブに設定できます。</p>	<p>PDB リソース・プランの CPU 割当てポリシーは規定されません。</p> <p>PDB リソース・プランの utilization_limit パラメータで指定された CPU 制限は、規定されません。</p>
パラレル実行サーバー	<p>次の要件のいずれかが満たされている必要があります。</p> <ul style="list-style-type: none"> ● 共有値が、shares パラメータを使用して PDB に指定されている必要があります。 ● parallel_server_limit パラメータを使用して、PDB に 100 未満のパラレル・サーバー制限を指定する必要があります。 <p>これらの値は、特定の PDB のディレクティブまたはデフォルト・ディレクティブに設定できます。</p>	<p>PDB リソース・プランのパラレル実行サーバー割当てポリシーは、規定されません。</p> <p>PDB リソース・プランで parallel_server_limit によって指定されたパラレル・サーバー制限は規定されません。ただし、PDB の PARALLEL_SERVERS_TARGET 初期化パラメータを設定して、パラレル制限を規定できます。</p>

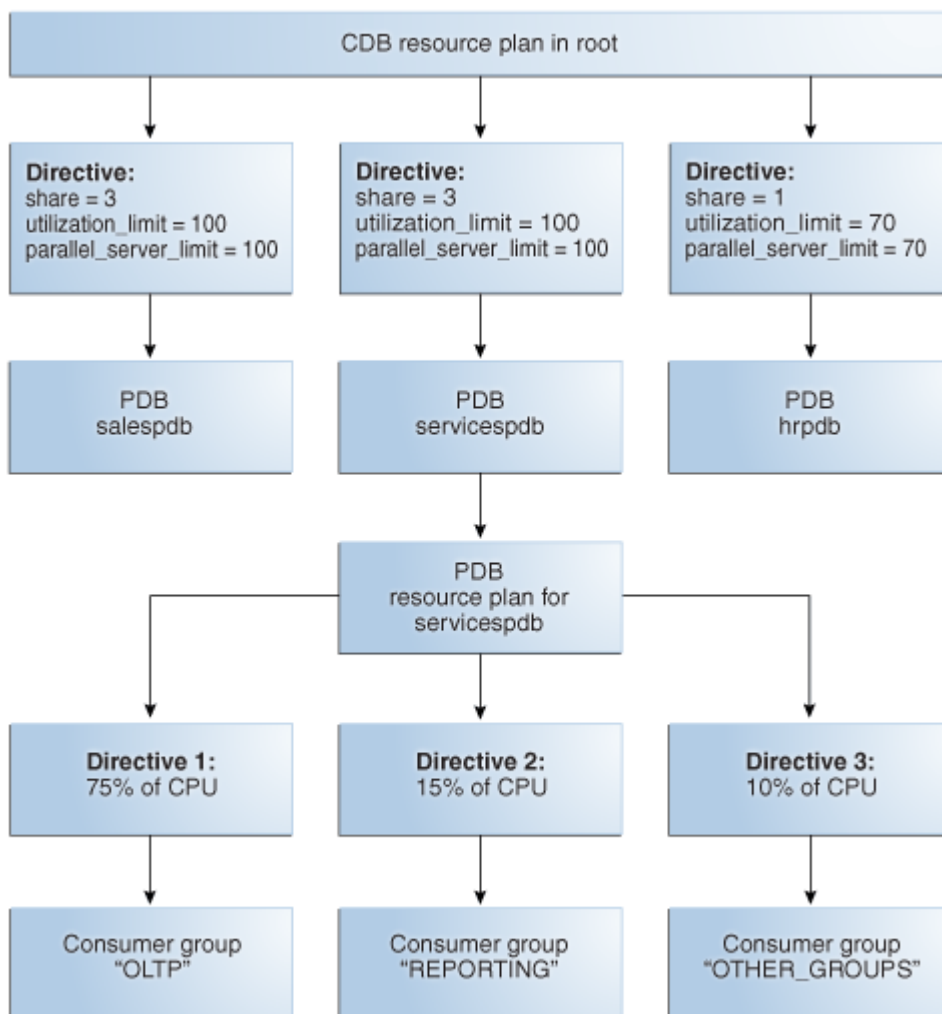
親トピック: [PDBリソース・プランについて](#)

PDBリソース・プラン: 例

CDBリソース・プランとPDBリソース・プランには、1対多の関係があります。

次の図に、CDBリソース・プランおよびPDBリソース・プランの例を示します。

図22-4 CDBリソース・プランおよびPDBリソース・プラン



前述の図は、PDB servicespdbのPDBリソース・プランにおける一部のディレクティブを示しています。CDB内の他のPDBにもPDBリソース・プランを作成できます。

親トピック: [PDBリソース・プランについて](#)

PDBリソース・プランの作成

PDBリソース・プランは、非CDBのリソース・プランを作成する場合と同じ方法で作成します。プランを作成するには、DBMS_RESOURCE_MANAGER PL/SQLパッケージ内のプロシージャを使用します。

CDBリソース・プランにより、システム・リソースの一部がPDBに割り当てられます。PDBリソース・プランにより、PDB内でのこの部分の割当て方法が決定します。

PDBリソース・プランの作成に必要なステップの概要は、次のとおりです。

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。
2. CREATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを作成します。
3. CREATE_CONSUMER_GROUPプロシージャを使用して、コンシューマ・グループを作成、変更または削除します。
4. SET_CONSUMER_GROUP_MAPPINGプロシージャを使用して、セッションをコンシューマ・グループにマップします。

5. CREATE_PLANプロシージャを使用して、PDBリソース・プランを作成します。
6. CREATE_PLAN_DIRECTIVEプロシージャを使用して、PDBリソース・プラン・ディレクティブを作成します。
7. VALIDATE_PENDING_AREAプロシージャを使用して、ペンディング・エリアを検証します。
8. SUBMIT_PENDING_AREAプロシージャを使用して、ペンディング・エリアを発行します。

現在のコンテナがPDBであり、これらのステップを完了する場合に必要な権限がユーザーに付与されていることを確認してください。これらのステップの完了の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。

CREATE_SIMPLE_PLANプロシージャを使用して、多くの状況に対応できる単純なリソース・プランを作成することもできます。単純なリソース・プランの作成の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。



ノート:

PDB リソース・プランに適用されるいくつかの制限事項があります。詳細は、[『PDB リソース・プランについて』](#)を参照してください。

親トピック: [PDBリソース・プランの管理](#)

PDBリソース・プランの有効化

現在のコンテナがPDBの場合、ALTER SYSTEM文を使用してRESOURCE_MANAGER_PLAN初期化パラメータをプランに設定し、PDBリソース・プランを有効化します。

このパラメータがプランに指定されていない場合、PDBリソース・プランはPDBに対して有効化されません。

前提条件

PDBリソース・プランを有効化する前に、[『マルチテナント環境でのOracle Resource Managerの概要』](#)で説明されている前提条件を満たします。

PDBリソース・プランを有効化するには:

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。
2. ALTER SYSTEM文を使用して、RESOURCE_MANAGER_PLAN初期化パラメータをPDBリソース・プランに設定します。

PDBリソース・プランの変更は、Oracle Schedulerを使用してスケジュールすることもできます。

例22-5 PDBリソース・プランの有効化

次の例では、PDBリソース・プランをsalespdb_planに設定しています。

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'salespdb_plan';
```

関連項目:

- [『CDB内のコンテナへのアクセスについて』](#)
- [『システム・レベルでのPDBの変更』](#)
- Oracle Schedulerを使用してPDBリソース・プランの変更をスケジュールする方法を学習するには、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [PDBリソース・プランの管理](#)

PDBリソース・プランの変更

DBMS_RESOURCE_MANAGERパッケージを使用すると、非CDBのリソース・プランを変更する場合と同じ方法でPDBリソース・プランを変更できます。

前提条件

PDBリソース・プランを変更する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

PDBリソース・プランを変更するには:

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。
2. ペンディング・エリアを作成します。

```
exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

3. 次のタスクのうち1つ以上を完了して、PDBリソース・プランを変更します。
 - UPDATE_CONSUMER_GROUPプロシージャを使用して、コンシューマ・グループを更新します。
 - DELETE_CONSUMER_GROUPプロシージャを使用して、コンシューマ・グループを削除します。
 - UPDATE_PLANプロシージャを使用して、リソース・プランを更新します。
 - DELETE_PLANプロシージャを使用して、リソース・プランを削除します。
 - UPDATE_PLAN_DIRECTIVEプロシージャを使用して、リソース・プラン・ディレクティブを更新します。
 - DELETE_PLAN_DIRECTIVEプロシージャを使用して、リソース・プラン・ディレクティブを削除します。

4. ペンディング・エリアを検証します。

```
exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5. ペンディング・エリアを発行します。

```
exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

関連項目:

- [「PDBリソース・プランについて」](#)
- [「CDB内のコンテナへのアクセスについて」](#)
- コンシューマ・グループのタスクを完了する手順は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [PDBリソース・プランの管理](#)

PDBリソース・プランの無効化

PDBのRESOURCE_MANAGER_PLAN初期化パラメータの設定を解除して、PDBリソース・プランを無効化します。

前提条件

PDBリソース・プランを無効化する前に、[「マルチテナント環境でのOracle Resource Managerの概要」](#)で説明されている前提条件を満たします。

PDBリソース・プランを無効するには:

1. SQL*Plusで、現在のコンテナがPDBであることを確認します。
2. ALTER SYSTEM文を使用して、PDBのRESOURCE_MANAGER_PLAN初期化パラメータの設定を解除します。

例22-6 PDBリソース・プランの無効化

次の例では、PDBリソース・プランを無効化しています。

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = '' ;
```

関連項目:

- [「システム・レベルでのPDBの変更」](#)
- [「CDB内のコンテナへのアクセスについて」](#)を参照してください。

親トピック: [PDBリソース・プランの管理](#)

Oracle Database Resource Managerで管理されるPDBの監視

一連の動的パフォーマンス・ビューにより、Oracle Database Resource ManagerによるPDBの設定の結果を監視できます。

- [PDBに対するリソース・マネージャのビューについて](#)
ビューを使用して、PDBに対するOracle Database Resource Managerの設定の結果を監視できます。
- [PDBのCPU使用率の監視](#)
V\$RSRCPDBMETRICビューを使用すると、セッション数の観点または過去1分間の使用状況の観点から、CPUメトリックをミリ秒単位で追跡できます。
- [PDBパラレル実行の監視](#)
V\$RSRCPDBMETRICビューを使用すると、PDBに対するパラレル文とパラレル・サーバーの使用を追跡できます。
- [PDBによって生成されるI/Oの監視](#)
V\$RSRCPDBMETRICビューを使用すると、PDBで生成されるI/Oの量を追跡できます。
- [PDBのメモリー使用率の監視](#)
V\$RSRCPDBMETRICビューを使用すると、PDBで使用されるメモリー量を追跡できます。

親トピック: [PDBに対するOracle Resource Managerの使用](#)

PDBに対するリソース・マネージャのビューについて

ビューを使用して、PDBに対するOracle Database Resource Managerの設定の結果を監視できます。

次のビューがあります。

- V\$RSRCPDBMETRIC
V\$RSRCPDBMETRICビューは、CPU使用率、パラレル実行、生成されるI/O、メモリー使用率など、PDBのリソース消費に関する現在の統計を提供します。

- V\$RSRCPDBMETRIC_HISTORY

V\$RSRCPDBMETRIC_HISTORYビューの列は、V\$RSRCPDBMETRICビューの列と同じです。これらのビューの唯一の違いは、V\$RSRCPDBMETRICビューには過去1分間のみのメトリックが含まれているのに対して、V\$RSRCPDBMETRIC_HISTORYビューには過去60分間のメトリックが含まれていることです。

- V\$RSRC_PDB

V\$RSRC_PDBビューには累積統計が表示されます。統計は、CDBリソース・プランによって設定された時間から蓄積されます。

- DBA_HIST_RSRC_PDB_METRIC

このビューには、自動ワークロード・リポジトリ(AWR)のスナップショットを使用して取得された、V\$RSRCPDBMETRIC_HISTORYの履歴統計が含まれています。

ノート:



STATISTICS_LEVEL 初期化パラメータが ALL または TYPICAL に設定されている場合、V\$RSRCPDBMETRIC および V\$RSRCPDBMETRIC_HISTORY ビューでは、リソース・マネージャによって現在管理されていないリソースの統計が記録されます。

関連項目:

- リアルタイムSQL監視の詳細は、[『Oracle Database SQLチューニング・ガイド』](#)を参照してください
- V\$RSRCPDBMETRIC、V\$RSRCPDBMETRIC_HISTORY、V\$RSRC_PDBおよびDBA_HIST_RSRC_PDB_METRICについて学習するには、[『Oracle Databaseリファレンス』](#)を参照してください

親トピック: [Oracle Database Resource Managerで管理されるPDBの監視](#)

PDBのCPU使用率の監視

V\$RSRCPDBMETRICビューを使用すると、セッション数の観点または過去1分間の使用状況の観点から、CPUメトリックをミリ秒単位で追跡できます。

ビューには各PDBのリアルタイムなメトリックが提供されるため、ワークロードの実行中に、CPUリソースの使用状況を継続的に監視する場合に非常に便利です。

アクティブなCDBリソース・プランでPDBのCPU使用状況を管理します。このビューを使用して、次のような他のPDB設定によって、PDBの最大および平均CPU使用状況を比較します。

- 使用されたCPU時間
- CPUを待機している時間
- CPUを消費しているセッションの平均数
- CPU割当てを待機しているセッションの数

たとえば、PDBが使用したCPUリソースの量およびPDBがリソース割当てを待機した期間を表示できます。あるいは、各PDBから実行されているセッションの数とアクティブなセッションの総数とを対比できます。

PDBのCPU使用率の観点からのCPU消費量の追跡

CPU使用率でCPUの消費量を追跡するには、CPU_UTILIZATION_LIMIT列およびAVG_CPU_UTILIZATION列を問い合わせます。AVG_CPU_UTILIZATIONには、PDBによるサーバーのCPUの平均使用率が表示されます。CPU_UTILIZATION_LIMITは、PDBが使用できるサーバーのCPUの最大割合を表します。この制限は、UTILIZATION_LIMITディレクティブ属性を使用して設定されます。

次の問合せでは、コンテナID (CON_ID)および各PDBの名前を表示することでこの情報が表示されます。

```
COLUMN PDB_NAME FORMAT A10
```

```
SELECT r. CON_ID,  
       p. PDB_NAME,  
       r. CPU_UTILIZATION_LIMIT,  
       r. AVG_CPU_UTILIZATION  
FROM   V$RSRCPDBMETRIC r,  
       CDB_PDBS p  
WHERE  r. CON_ID = p. CON_ID;
```

PDBのCPU消費量および制限量の追跡

各PDBのCPUの消費量および制限量をミリ秒単位で追跡するには、CPU_CONSUMED_TIME列およびCPU_TIME_WAIT列を使用します。NUM_CPUS列は、リソース・マネージャが管理しているCPUの数を表します。

次の問合せでは、コンテナID (CON_ID)および各PDBの名前を表示することでこの情報が表示されます。

```
COLUMN PDB_NAME FORMAT A10
```

```
SELECT r. CON_ID,  
       p. PDB_NAME,  
       r. CPU_CONSUMED_TIME,  
       r. CPU_WAIT_TIME,  
       r. NUM_CPUS  
FROM   V$RSRCPDBMETRIC r,  
       CDB_PDBS p  
WHERE  r. CON_ID = p. CON_ID;
```

PDBのセッション数の観点からのCPU消費量および制限量の追跡

CPUの消費量および制限量をセッションの数で追跡するには、RUNNING_SESSIONS_LIMIT、AVG_RUNNING_SESSIONS、AVG_WAITING_SESSIONSの各列を使用します。RUNNING_SESSIONS_LIMITには、特定のPDBからいつでも実行できるセッションの最大数が表示されます。この制限は、PDBに設定したUTILIZATION_LIMITディレクティブ属性によって定義されます。AVG_RUNNING_SESSIONS列にはCPUを消費しているセッションの平均数が表示され、AVG_WAITING_SESSIONS列にはCPUを待機しているセッションの平均数が表示されます。

次の問合せでは、コンテナID (CON_ID)および各PDBの名前を表示することでこの情報が表示されます。

```
COLUMN PDB_NAME FORMAT A10
```

```
SELECT r. CON_ID,  
       p. PDB_NAME,  
       r. RUNNING_SESSIONS_LIMIT,  
       r. AVG_RUNNING_SESSIONS,  
       r. AVG_WAITING_SESSIONS  
FROM   V$RSRCPDBMETRIC r,  
       CDB_PDBS p  
WHERE  r. CON_ID = p. CON_ID;
```

親トピック: [Oracle Database Resource Managerで管理されるPDBの監視](#)

PDBの平行実行の監視

V\$RSRCPDBMETRICビューを使用すると、PDBに対する平行・ステートメントと平行・サーバーの使用を追跡できます。

PDBの平行実行サーバーは、PDBのCDBのアクティブなCDBリソース・プランで管理されます。平行・ステートメントおよびPDBの平行・サーバーの使用を追跡するには、AVG_ACTIVE_PARALLEL_STMTS、AVG_QUEUED_PARALLEL_STMTS、AVG_ACTIVE_PARALLEL_SERVERS、AVG_QUEUED_PARALLEL_SERVERSおよびPARALLEL_SERVERS_LIMIT列を使用します。

AVG_ACTIVE_PARALLEL_STMTSとAVG_ACTIVE_PARALLEL_SERVERSには、実行されている平行・ステートメントの平均数、および平行・ステートメントによって使用される平行・サーバーの平均数が表示されます。AVG_QUEUED_PARALLEL_STMTSとAVG_QUEUED_PARALLEL_SERVERSには、キュー内の平行・ステートメントの平均数、およびキュー内の平行・ステートメントによって要求された平行・サーバーの平均数が表示されます。PARALLEL_SERVERS_LIMITには、PDBで使用できる平行・サーバーの数が表示されます。

次の問合せでは、コンテナID (CON_ID)および各PDBの名前を表示することでこの情報が表示されます。

```
COLUMN PDB_NAME FORMAT A10

SELECT r.CON_ID, p.PDB_NAME, r.AVG_ACTIVE_PARALLEL_STMTS, r.AVG_QUEUED_PARALLEL_STMTS,
       r.AVG_ACTIVE_PARALLEL_SERVERS, r.AVG_QUEUED_PARALLEL_SERVERS, r.PARALLEL_SERVERS_LIMIT
FROM V$RSRCPDBMETRIC r, CDB_PDBS p
WHERE r.CON_ID = p.CON_ID;
```

親トピック: [Oracle Database Resource Managerで管理されるPDBの監視](#)

PDBによって生成されるI/Oの監視

V\$RSRCPDBMETRICビューを使用すると、PDBで生成されるI/Oの量を追跡できます。

I/Oは、PDBでMAX_IOPS初期化パラメータまたはMAX_MBPS初期化パラメータを設定することで、PDBに対して制限されます。このビューを使用して、毎秒の操作数および毎秒のMB数に関してPDBで生成されるI/Oを比較します。

PDBで毎秒生成されるI/O操作数の追跡

前の1分間にPDBで毎秒生成されるI/O操作数を追跡するには、IOPS列を使用します。

次の問合せでは、コンテナID (CON_ID)および各PDBの名前を表示することでこの情報が表示されます。

```
COLUMN PDB_NAME FORMAT A10

SELECT r.CON_ID, p.PDB_NAME, r.IOPS
FROM V$RSRCPDBMETRIC r, CDB_PDBS p
WHERE r.CON_ID = p.CON_ID;
```

PDBでのI/O操作で毎秒生成されるデータ量(MB単位)の追跡

前の1分間にPDBで毎秒生成されるI/OのMB数を追跡するには、IOMBPS列を使用します。

次の問合せでは、コンテナID (CON_ID)および各PDBの名前を表示することでこの情報が表示されます。

```
COLUMN PDB_NAME FORMAT A10

SELECT r.CON_ID, p.PDB_NAME, r.IOMBPS
FROM V$RSRCPDBMETRIC r, CDB_PDBS p
WHERE r.CON_ID = p.CON_ID;
```

親トピック: [Oracle Database Resource Managerで管理されるPDBの監視](#)

PDBのメモリー使用状況の監視

V\$RSRCPDBMETRICビューを使用すると、PDBで使用されるメモリー量を追跡できます。

このビューを使用して、SGA、PGA、バッファ・キャッシュおよびPDBが現在使用している共有プール・メモリーの量を追跡します。

特定のPDBの現在のメモリー使用量(バイト単位)を追跡するには、SGA_BYTES、PGA_BYTES、BUFFER_CACHE_BYTESおよびSHARED_POOL_BYTES列を使用します。

次の問合せでは、コンテナID (CON_ID)および各PDBの名前を表示することでこの情報が表示されます。

```
COLUMN PDB_NAME FORMAT A10
```

```
SELECT r.CON_ID, p.PDB_NAME, r.SGA_BYTES, r.PGA_BYTES, r.BUFFER_CACHE_BYTES, r.SHARED_POOL_BYTES  
FROM V$RSRCPDBMETRIC r, CDB_PDBS p  
WHERE r.CON_ID = p.CON_ID;
```

親トピック: [Oracle Database Resource Managerで管理されるPDBの監視](#)

23 CDBでのOracle Schedulerの使用

Oracle Schedulerを使用して、マルチテナント・コンテナ・データベース(CDB)でのジョブをスケジュールできます。

CDBでOracle Schedulerを使用する前に、次の要件を満たします。

- CDBを構成および管理する方法について理解しています。
- Oracle Schedulerを使用して非CDBでのジョブをスケジュールする方法を理解しています。
- [CDBにおけるDBMS_SCHEDULERの起動](#)
ほとんどのスケジューラは、2つのスケジューラ・グローバル属性を除き、非CDBで行うのと同じ方法で作業を呼び出します。
- [CDBにおけるジョブ・コーディネータおよびスレーブ・プロセス](#)
CDB関連の主な変更は、ジョブ・コーディネータ・プロセスに対して行います。
- [DBMS_JOBおよびDBMS_SCHEDULER](#)
DBMS_JOBジョブを発行するスキーマにCREATE_JOB権限を付与する場合にのみ、PDBでDBMS_JOBを使用できます。
- [PDBをクローズするプロセス](#)
IMMEDIATEオプションを指定してPDBがクローズされた場合、コーディネータはPDBで実行されているジョブを停止し、そのジョブを再度実行するにはリカバリする必要があります。
- [新規および変更されたCDBビュー](#)
一部のCDBビューがCDBに固有であるのに対し、その他のCDBビューにはCDB固有の列があります。

関連項目:

- [「マルチテナント環境の管理」](#)
- [Oracle Database管理者ガイド](#)

親トピック: [マルチテナント環境でのOracle機能の使用](#)

CDBにおけるDBMS_SCHEDULERの起動

ほとんどのスケジューラは、2つのスケジューラ・グローバル属性を除き、非CDBで行うのと同じ方法で作業を呼び出します。

JOB_QUEUE_PROCESSES初期化パラメータには、DBMS_JOBジョブおよびOracle Scheduler(DBMS_SCHEDULER)ジョブの実行用に作成可能な、インスタンスごとのジョブ・スレーブの最大数を指定します。値の範囲は0から4000です(デフォルト)。

CDB環境でジョブ・スレーブを制限するには、次の場所でJOB_QUEUE_PROCESSESを設定できます。

- CDBルート
JOB_QUEUE_PROCESSESを、スケジューラがデータベース・インスタンス全体で同時に使用できるスレーブ・プロセスの最大数に設定します。

CDBルートでJOB_QUEUE_PROCESSESが0の場合、PDBレベルのJOB_QUEUE_PROCESSES設定に関係なく、DBMS_JOBおよびOracle SchedulerジョブはルートまたはPDBで実行できません。
- PDB
JOB_QUEUE_PROCESSESを、このPDBの同時ジョブの最大数に設定します。実際の数は、リソース・マネージャによって

割り当てられたリソースと他のコンテナでの需要によって異なります。複数のPDBがジョブをリクエストすると、Oracle Schedulerは、すべてのPDBにかなりの数のプロセスを付与しようとします。

PDBでJOB_QUEUE_PROCESSESが0の場合、CDBルートのJOB_QUEUE_PROCESSES設定に関係なく、DBMS_JOBとOracle SchedulerジョブはこのPDBで実行できません。

すべてのグローバルOracle Scheduler属性をPDBレベルに設定する必要があります。たとえば、DBMS_SCHEDULER.SET_ATTRIBUTEを使用してEMAIL_SENDER属性をルート・データベースで設定する場合、その属性は、特定のPDBで実行されるジョブではなく、ルートで実行されるジョブに適用されます。新しいEMAIL_SENDERをPDBに選択する場合は、このPDBでグローバル属性を設定する必要があります。

関連項目:

JOB_QUEUE_PROCESSESについてさらに学習するには、[Oracle Databaseリファレンス](#)を参照してください

親トピック: [CDBでのOracle Schedulerの使用](#)

CDBにおけるジョブ・コーディネータおよびスレーブ・プロセス

CDB関連の主な変更は、ジョブ・コーディネータ・プロセスに対して行います。

非CDBでは、コーディネータは、実行準備が完了しているすべてのジョブを参照し、実行するそれらのジョブのサブセットを選択して、ジョブ・スレーブに割り当てます。また、ウィンドウをオープンおよびクローズし、これにより、データベースに対して有効なリソース・プランが変更されます。

このことは、次のことを除き、実質的に、CDB内で発生します。

- ジョブがすべてのPDBから選択されます。

コーディネータは、ルート・データベースおよびすべての子PDBを参照し、ジョブの優先度、スケジュールされたジョブの開始時間、およびジョブを実行するためのリソースの可用性に基づいてジョブを選択します。後の基準は、ジョブのコンシューマ・グループ、および現在有効なリソース・プランによって異なります。コーディネータでは、すべてのPDBに対して公平になるようには処理しません。PDBのジョブでスタベーションが発生しないようにする唯一の方法は、十分なリソースをPDBに割り当てることです。

- ウィンドウがPDBおよびルート・データベース・レベルでオープンします。

非CDBでは、特定の時点でオープンできるウィンドウは1つのみです。CDBには、2つのレベルのウィンドウがあります。PDBレベルでは、ウィンドウを使用して、そのPDBに属するコンシューマ・グループ間でリソースを割り当てるリソース・プランを設定できます。ルート・データベース・レベルでは、ウィンドウで異なるPDBにリソースを割り当てることができます。したがって、どの時点においても、ルート・データベースで1つのウィンドウがオープンし、各PDBで1つのウィンドウがオープンしている可能性があります。

- ジョブ・スレーブは、そのジョブ・スレーブが属する特定のPDBに切り替えます。

ジョブ・スレーブは、実質的には非CDBの場合と同じですが、スレーブがジョブを実行するとき、ジョブが属するPDBに切り替えた後で実行することが異なります。残りのコードは実質的に変更されていません。

親トピック: [CDBでのOracle Schedulerの使用](#)

DBMS_JOBおよびDBMS_SCHEDULER

DBMS_JOBジョブを発行するスキーマにCREATE JOB権限を付与する場合にのみ、PDBでDBMS_JOBを使用できます。

DBMS_JOBインターフェイスは、DBMS_SCHEDULERを使用して実装されます。そのため、DBMS_JOBからDBMS_SCHEDULERに切り替えることをお勧めします。

スケジューラの場合、コーディネータは各PDBから実行するジョブを選択します。また、スケジューラの場合、スレーブ・プロセスはPDBに切り替えてからジョブを実行しますが、それ以外については、実質的にコードは変更されません。

関連項目:

DBMS_JOBのサポートの詳細は、[『Oracle Database管理者ガイド』](#)を参照してください

親トピック: [CDBでのOracle Schedulerの使用](#)

PDBをクローズするプロセス

IMMEDIATEオプションを指定してPDBがクローズされた場合、コーディネータはPDBで実行されているジョブを停止し、そのジョブを再度実行するにはリカバリする必要があります。

Oracle RACデータベースでは、コーディネータは、ほとんどの場合、そのPDBがオープンしている別のインスタンスでジョブをリカバリできます。このため、最初のインスタンスのコーディネータは、PDBがまだオープンしている別のインスタンスを見つけることができた場合は、ジョブをそこに移動します。特定のケースでは、ジョブを別のインスタンスに移動できない場合があります。たとえば、対象のPDBがいずれの場所でもオープンしていない場合、ジョブは移動できません。また、別のインスタンスへのジョブの移動は、そのジョブにINSTANCE_ID属性セットが指定されている場合は実行できません。この場合、ジョブはそのインスタンスでPDBが再びオープンされるまで実行できません。

Oracle RAC以外のケースでは、ジョブの移動が問題となることはありません。終了したジョブは、PDBが再度オープンされた後にのみリカバリできます。

親トピック: [CDBでのOracle Schedulerの使用](#)

新規および変更されたCDBビュー

一部のCDBビューがCDBに固有であるのに対し、その他のCDBビューにはCDB固有の列があります。

- V\$およびGV\$ビューには、CDB_*行が表すデータを含むコンテナを識別するCON_ID列があります。非CDBでは、CON_ID列はNULLです。
- CDB_*ビューは、すべてのスケジューラのDBA_*ビューに対応します。

PDBでは、これらのビューには対応するDBA_*ビューを介して表示可能なオブジェクトのみが表示されますが、ルートにはすべてのオブジェクトが表示されます。CDB_*ビューには、特定のDBA_*ビューで見つかったすべての列および列(CON_ID)が含まれています。非CDBでは、この列はNULLです。

親トピック: [CDBでのOracle Schedulerの使用](#)

24 CDBでのOracle Database Vaultの使用

Oracle Database Vaultは、マルチテナント・コンテナ・データベース(CDB)で使用できます。

- [Oracle Database Vaultについて](#)
Oracle Database Vaultのセキュリティ統制により、アプリケーション・データを不正アクセスから守るとともに、プライバシーおよび規制の要件に順守できます。
- [マルチテナント環境におけるOracle Database Vaultの動作について](#)
統合のセキュリティをさらに強化するために、Oracle Database VaultをOracle Multitenantとともに使用できます。
- [Database Vaultが構成および有効化されていることの確認](#)
DBA_DV_STATUS、CDB_DV_STATUS、DBA_OLS_STATUSおよびCDB_OLS_STATUSデータ・ディクショナリ・ビューでは、Oracle Databaseが構成され有効化されているかどうかを確認します。
- [マルチテナント環境におけるOracle DatabaseへのOracle Database Vaultの登録](#)
Oracle Database Vaultは、いくつかのシナリオに基づいてマルチテナント環境で登録できます。
- [レルムの構成](#)
データベース・オブジェクトを保護するために、これらのオブジェクトの周辺にレルムを作成し、このデータへのユーザー・アクセスを制御するための認可を設定します。
- [マルチテナント環境におけるルール・セットとルール](#)
マルチテナント環境では、アプリケーション・ルートでルール・セットおよびその関連付けられたルールを作成できます。
- [マルチテナント環境でのコマンド・ルール](#)
マルチテナント環境では、CDBルートまたはアプリケーション・ルートのどちらかで、共通およびローカルのコマンド・ルールを作成できます。
- [マルチテナント環境におけるOracle Database Vaultポリシー](#)
Oracle Database Vaultポリシーは、それらが作成されたプラガブル・データベース(PDB)に対してのみローカルとなります。
- [Database Vault操作制御を使用したローカルPDBデータへのマルチテナント共通ユーザー・アクセスの制限](#)
CDBルート共通ユーザー(インフラストラクチャ・データベース管理者など)によるPDBアクセスを制御できます。
- [スタンドアロンのOracle DatabaseをPDBに変換してCDBにプラグイン](#)
リリース12c以降のスタンドアロンのOracle DatabaseはPDBに変換可能で、さらにそのPDBはCDBにプラグインできます。

親トピック: [マルチテナント環境でのOracle機能の使用](#)

Oracle Database Vaultについて

Oracle Database Vaultのセキュリティ統制により、アプリケーション・データを不正アクセスから守るとともに、プライバシーおよび規制の要件に順守できます。

統制によって特権アカウントによるアプリケーション・データへのアクセスをブロックすることや、信頼できるパスの認可を使用してデータベース内の要注意操作を統制することができます。権限およびロールの分析により、最小権限のベスト・プラクティスを使用することで、既存のアプリケーションのセキュリティを強化できます。Oracle Database Vaultは、既存のデータベース環境のセキュリティを透過的に強化するので、コストと時間をかけてアプリケーションを変更する必要はありません。

関連トピック

- [Oracle Database Vaultの管理者ガイド](#)

親トピック: [CDBでのOracle Database Vaultの使用](#)

マルチテナント環境におけるOracle Database Vaultの動作について

統合のセキュリティをさらに強化するために、Oracle Database VaultをOracle Multitenantとともに使用できます。

Oracle Database Vaultでは、プラグブル・データベース(PDB)内や、PDBとコンテナ・データベースでの共通特権ユーザーとの間での、特権ユーザー・アクセスを禁止できます。各PDBには、レルム、ルール・セット、コマンド・ルール、デフォルト・ポリシー(デフォルト・レルムなど)など独自のDatabase Vaultメタデータがあります。また、任意の子のPDBでDVSYSスキーマやDVFスキーマ内のオブジェクトを自動的に利用できます。どちらのスキーマも共通のユーザー・スキーマです。

共通レルムはアプリケーション・ルートのみで構成できますが、共通ルール・セットおよびコマンド・ルールは、アプリケーション・ルートまたはCDBルートのどちらでも作成できます。アプリケーション・ルート内の共通コマンド・ルールは、その関連付けられたPDBに適用され、CDBルート内の共通コマンド・ルールは、CDB環境内のすべてのPDBに適用されます。共通レルムおよびコマンド・ルールを作成できることにより、CDB環境全体で共有の一連のレルム、ルール・セットまたはコマンド・ルールを使用するポリシーを作成できます。マルチテナント環境ですべてのPDBに対してこれらの同じコンポーネントを作成する必要はありません。共通保護は、Oracle Database Vaultが有効になっているアプリケーション・ルートに関連付けられたすべてのPDBに適用されます。

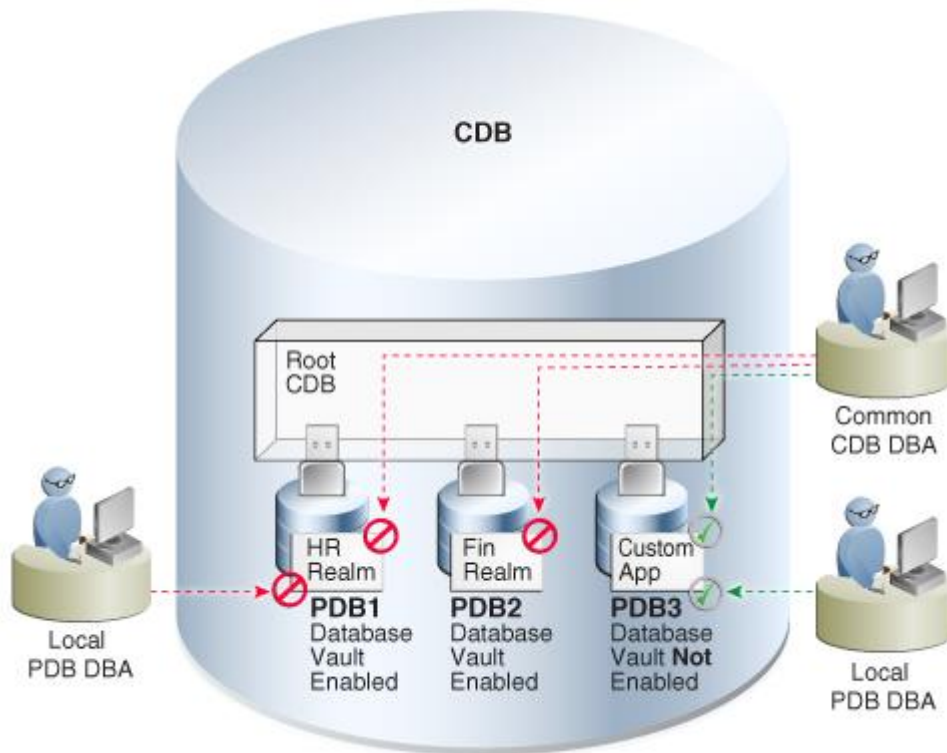
PDBごとに個別のローカル・ポリシーを作成できます。Database Vaultを使用してオブジェクトを保護する場合、Database Vaultは、共通のオブジェクトの共通の権限を、ローカル・システム権限と同じ強制ルール下に置きます。

Database Vaultを有効にしたPDBを構成すると、DVSYSスキーマが共通ユーザー・スキーマとなり、ルートに格納されます。つまり、DVSYSスキーマ内のすべてのオブジェクト(表、データ・ディクショナリ・ビュー、ユーザー・アカウント、PL/SQLパッケージ、デフォルト・ポリシーなど)が、このスキーマに利用可能な共通の権限の影響下にあるということになります。すなわち、レルム、ファクタなどをルートに作成してルートスキーマを保護できます。Database Vaultは、関連付けられたPDBで構成する前に、まずルートで構成するようにしてください。

CDBルートでOracle Database Vaultを有効にするときは、通常モードまたは厳密モードのどちらかを選択できます。設定は、選択した設定に基づいてCDB全体に伝播されます。たとえば、CDBに、Database Vaultが有効になっているPDBとDatabase Vaultが有効になっていないPDBが含まれているとします。通常モードを使用してDatabase Vaultを有効にした場合は、両方のタイプのPDBが通常どおり機能し続けます。厳密モードを使用してDatabase Vaultを有効にした場合、Database Vaultが無効になっているPDBは、制限されたモードで動作します。

[図24-1](#)では、Database Vaultが有効になっているかどうかによって、標準モードのデータベースで共通およびローカル・データベース管理者に可能になるアクセスがどのように異なるかを示します。このシナリオでは、共通ユーザーおよびローカル・ユーザーのいずれも、PDB1とPDB2のレルムにアクセスできません。共通ユーザーとPDB3のローカル・ユーザーはどちらも、Database Vaultが有効化されていないPDB3内のCustom Appアプリケーションにアクセスできます。

図24-1 標準モードでのマルチテナント環境におけるOracle Database Vault



関連トピック

- [マルチテナント環境におけるレルム](#)
- [マルチテナント環境におけるルール・セットとルール](#)
- [マルチテナント環境におけるコマンド・ルール](#)
- [スタンドアロンのOracle DatabaseをPDBに変換してCDBにプラグイン](#)

親トピック: [CDBでのOracle Database Vaultの使用](#)

Database Vaultが構成および有効化されていることの確認

DBA_DV_STATUS、CDB_DV_STATUS、DBA_OLS_STATUSおよびCDB_OLS_STATUSデータ・ディクショナリ・ビューは、Oracle Databaseが構成され有効になっているかどうかを確認します。

Oracle Database Vault管理者の他に、Oracle Database SYSユーザー、およびDBAロールを付与されているユーザーが、これらのビューを問合せできます。

- Database Vaultの場合:
 - 非マルチテナント・データベースのために、またはマルチテナント環境でルートのみか個々のPDBのために Database Vaultステータスを確認する場合は、DBA_DV_STATUSを問い合わせます。次に例を示します。

```
SELECT * FROM DBA_DV_STATUS;
```

次のような出力が表示されます。

NAME	STATUS
DV_CONFIGURE_STATUS	TRUE
DV_ENABLE_STATUS	TRUE

- 管理権限がある共通ユーザーとして、マルチテナント環境ですべてのPDBのDatabase Vaultステータスを確認する場合は、コンテナID (CON_ID)フィールドの追加を提供する、CDB_DV_STATUSを問い合わせます。
- Oracle Label Securityの場合は、次のデータ・ディクショナリ・ビューを問い合わせます。これらは、Database Vault

のそれらに相当するビューに似ています。

- DBA_OLS_STATUS
- CDB_OLS_STATUS

親トピック: [CDBでのOracle Database Vaultの使用](#)

マルチテナント環境におけるOracle DatabaseへのOracle Database Vaultの登録

Oracle Database Vaultは、いくつかのシナリオに基づいてマルチテナント環境で登録できます。

- [CDBルートでのDatabase Vaultの登録](#)
マルチテナント環境では、CDBルートでDatabase Vault適用対象のロールを使用する共通ユーザーにOracle Database Vaultを登録します。
- [個別PDBを管理するためのDatabase Vault共通ユーザーの登録](#)
マルチテナント環境では、最初にOracle Database Vaultをルートで登録し、次にPDBで登録する必要があります。
- [個別PDBを管理するためのDatabase Vaultローカル・ユーザーの登録](#)
最初にOracle Database Vaultをルートに登録し、次にPDBに登録する必要があります。
- [Database Vault対応PDBへの接続](#)
マルチテナント環境では、SQL*Plusから、すでにDatabase Vaultが有効になっているデータベースに接続できます。
- [マルチテナント環境でのOracle Database Vaultの手動インストール](#)
特定の条件のマルチテナント環境に対しては、Oracle Database Vaultを手動でインストールする必要があります。

親トピック: [CDBでのOracle Database Vaultの使用](#)

CDBルートでのDatabase Vaultの登録

マルチテナント環境では、CDBルートでDatabase Vault適用対象のロールを使用する共通ユーザーにOracle Database Vaultを登録します。

1. マルチテナント環境において、ユーザー作成権限を持ち、CREATE SESSIONおよびSET CONTAINER権限を付与する権限を持つユーザーとして、データベース・インスタンスのルートにログインします。

次に例を示します。

```
sqlplus c##dba_debra
Enter password: password
```

2. Database Vault所有者(DV_OWNERロール)およびDatabase Vaultアカウント・マネージャ(DV_ACCTMGRロール)のアカウント用に使用するユーザー・アカウントを選択します(または新しいユーザーを作成します)。

ロールごとに2つのアカウントを維持することをお勧めします。一方のアカウントはその名前付きユーザーのプライマリ・アカウントであり、日常的に使用されます。他方のアカウントは、プライマリ・アカウントのパスワードを忘れてしまいリセットする必要がある場合に備えたバックアップ・アカウントとして使用されます。

これらのアカウント名の先頭にc##または0##を付加します。次に例を示します。

```
GRANT CREATE SESSION, SET CONTAINER TO c##sec_admin_owen
  IDENTIFIED BY password CONTAINER = ALL;
GRANT CREATE SESSION, SET CONTAINER TO c##dbv_owner_root_backup
  IDENTIFIED BY password CONTAINER = ALL;
GRANT CREATE SESSION, SET CONTAINER TO c##accts_admin_ace
  IDENTIFIED BY password CONTAINER = ALL;
```

```
GRANT CREATE SESSION, SET CONTAINER TO c##dbv_acctmgr_root_backup
IDENTIFIED BY password CONTAINER = ALL;
```

詳細は、次のとおりです。

- プライマリ・アカウント(c##sec_admin_owenおよび c##accts_admin_ace)は、新しいロールのDV_ADMINおよびDV_ACCTMGRに対してまだ存在していない場合に作成します。
- passwordを安全なパスワードに置き換えます。

3. SYSDBA管理権限を持つユーザーSYSとしてルートに接続します。

```
CONNECT SYS AS SYSDBA
Enter password: password
```

4. 2つのバックアップDatabase Vaultユーザー・アカウントを構成します。

次に例を示します。

```
BEGIN
CONFIGURE_DV (
  dvowner_username      => 'c##dbv_owner_root_backup',
  dvacctmgr_username    => 'c##dbv_acctmgr_root_backup',
  force_local_dvowner   => FALSE);
END;
/
```

この例では、force_local_dvownerをFALSEに設定しているため、共通ユーザーは、このCDBルートに関連付けられているPDBのDV_OWNER権限を持つことができます。TRUEに設定すると、共通のDV_OWNERユーザーはCDBルートに対してのみDV_OWNERロール権限を持つように制限されます。CDBルート共通ユーザーにローカルにDV_OWNERを付与すると、そのユーザーはDV_OWNERロールを別のユーザーに共通に付与できなくなります。

5. utlrp.sqlスクリプトを実行して、ルートで無効化されたオブジェクトを再コンパイルします。

```
@?/rdbms/admin/utlrp.sql
```

スクリプトから指示がある場合はそれに従い、再びスクリプトを実行します。指示なしにスクリプトが異常終了した場合は、スクリプトを再実行します。

6. 先ほど構成したプライマリDatabase Vault所有者ユーザーとして、ルートに接続します。

次に例を示します。

```
CONNECT c##dbv_owner_root_backup
Enter password: password
```

7. 次のいずれかのコマンドを使用して、Oracle Database Vaultを有効にします。

- Oracle Database Vaultで標準モードを使用できるようにするには:

```
EXEC DBMS_MACADM.ENABLE_DV;
```

- 関連するすべてのPDBがこのデータベースでDatabase Vaultを有効にする必要がある場合、次のコマンドを使用します。(この手順の完了後、これらのPDBをそれぞれ有効にする必要があります。)Database Vaultが有効になっていないPDBは、データベースの再起動後からPDBでDatabase Vaultが有効になるまで、制限モードになります。

```
EXEC DBMS_MACADM.ENABLE_DV (strict_mode => 'y');
```

8. SYSDBA管理権限を使用して接続します。

```
CONNECT / AS SYSDBA
```

9. データベースを再起動します。

```
SHUTDOWN IMMEDIATE
STARTUP
```

10. Oracle Database VaultとOracle Label Securityがインストールされていて、有効になっていることを確認します。

```
SELECT * FROM DBA_DV_STATUS;
SELECT * FROM DBA_OLS_STATUS;
```

11. バックアップDV_OWNERユーザーとして接続し、前に作成したプライマリDV_OWNERユーザーにDV_OWNERロールを付与します。

次に例を示します。

```
CONNECT c##dbv_owner_root_backup
Enter password: password

GRANT DV_OWNER TO c##sec_admin_owen WITH ADMIN OPTION;
```

12. バックアップDV_ACCTMGRユーザーとして接続してから、DV_ACCTMGRロールをバックアップDV_ACCTMGRユーザーに付与します。

次に例を示します。

```
CONNECT c##dbv_acctmgr_root_backup
Enter password: password

GRANT DV_ACCTMGR TO c##accts_admin_ace WITH ADMIN OPTION;
```

13. 2つのバックアップ・アカウント・パスワードを、将来必要になる場合に備えて、特権アカウント管理(PAM)システムなどの安全な場所に格納します。

関連トピック

- [Database Vaultが構成および有効化されていることの確認](#)
- [Oracle Database Vaultの管理者ガイド](#)
- [Oracle Database Vaultの管理者ガイド](#)

親トピック: [マルチテナント環境におけるOracle DatabaseへのOracle Database Vaultの登録](#)

個別PDBを管理するためのDatabase Vault共通ユーザーの登録

マルチテナント環境では、Oracle Database Vaultをまずルートに登録して、後からPDBに登録する必要があります。

先にPDBに登録しようとすると、ORA-47503「Database VaultはCDB\$ROOTで有効化されていません。」エラーが表示されます。

1. まだ実行していない場合は、関連するバックアップ・アカウントとともにDatabase Vaultアカウントとして使用する名前付きの共通ユーザー・アカウントを特定するか作成します。
2. CDBルートにOracle Database Vaultを登録していることと、DV_OWNERロールが共通ユーザーに共通に付与されていることを確認します。
3. PDBに対してローカルである管理者として、PDBに接続します。
次に例を示します。

```
CONNECT dba_debra@pdb_name
Enter password: password
```

利用可能なPDBを検索するには、DBA_PDBSデータ・ディクショナリ・ビューを問い合わせます。現在のPDBを確認するには、show con_nameコマンドを実行します。

4. このPDBのためにユーザーにCREATE SESSION権限およびSET CONTAINER権限を付与します。

次に例を示します。

```
GRANT CREATE SESSION, SET CONTAINER TO c##sec_admin_owen CONTAINER = CURRENT;  
GRANT CREATE SESSION, SET CONTAINER TO c##accts_admin_ace CONTAINER = CURRENT;
```

5. SYSDBA管理権限を持つユーザーSYSとして接続します

```
CONNECT SYS@pdb_name AS SYSDBA  
Enter password: password
```

6. PDBにいる間に、2つのバックアップDatabase Vaultユーザー・アカウントを構成します。

```
BEGIN  
CONFIGURE_DV (  
  dvowner_uname      => 'c##dbv_owner_root_backup',  
  dvacctmgr_uname    => 'c##dbv_acctmgr_root_backup');  
END;  
/
```

この例では、force_local_dvownerパラメータは不要なために省略されています。PDB内で構成されているすべての共通ユーザーは、そのPDBのスコープに制限されます。

7. utlrlp.sqlスクリプトを実行して、このPDBで無効化されたオブジェクトを再コンパイルします。

```
@?/rdbms/admin/utlrlp.sql
```

スクリプトから指示がある場合はそれに従い、再びスクリプトを実行します。指示なしにスクリプトが異常終了した場合は、スクリプトを再実行します。

8. 前の手順で構成したバックアップDatabase Vault所有者ユーザーとして、PDBに接続します。

次に例を示します。

```
CONNECT c##dbv_owner_root_backup@pdb_name  
Enter password: password
```

9. このPDBでOracle Database Vaultを有効にします。

```
EXEC DBMS_MACADM.ENABLE_DV;
```

10. SYSDBA管理権限でCDBに接続します。

```
CONNECT / AS SYSDBA
```

11. PDBを閉じてから、再度開きます。

次に例を示します。

```
ALTER PLUGGABLE DATABASE pdb_name CLOSE IMMEDIATE;  
ALTER PLUGGABLE DATABASE pdb_name OPEN;
```

12. PDBが構成されていてDatabase Vaultに対して有効になっていることを確認します。

```
CONNECT SYS@pdb_name AS SYSDBA  
Enter password: password  
  
SELECT * FROM DBA_DV_STATUS;
```


- バックアップDV_OWNERユーザーとして接続し、前に作成したプライマリDV_OWNERユーザーにDV_OWNERロールを付与します。

次に例を示します。

```
CONNECT c##dbv_owner_root_backup@pdb_name
Enter password: password

GRANT DV_OWNER TO c##sec_admin_owen WITH ADMIN OPTION;
```

- バックアップDV_ACCTMGRユーザーとして接続してから、DV_ACCTMGRロールをプライマリDV_ACCTMGRユーザーに付与します。

次に例を示します。

```
CONNECT c##dbv_acctmgr_root_backup@pdb_name
Enter password: password

GRANT DV_ACCTMGR TO c##accts_admin_ace WITH ADMIN OPTION;
```

- 2つのバックアップ・アカウント・パスワードを、将来必要になる場合に備えて、特権アカウント管理(PAM)システムなどの安全な場所に格納します。

関連トピック

- [Database Vaultが構成および有効化されていることの確認](#)
- [Oracle Database Vaultの管理者ガイド](#)
- [Oracle Database Vaultの管理者ガイド](#)

親トピック: [マルチテナント環境におけるOracle DatabaseへのOracle Database Vaultの登録](#)

特定のPDBを管理するDatabase Vaultローカル・ユーザーの登録

Oracle Database Vaultをまずルートで登録してから、PDBで登録する必要があります。

先にPDBに登録しようとすると、ORA-47503「Database VaultはCDB\$ROOTで有効化されていません。」エラーが表示されます。

- ユーザー作成とCREATE SESSION権限およびSET CONTAINER権限の付与の権限があるユーザーとしてPDBにログインします。

次に例を示します。

```
sqlplus sec_admin@pdb_name
Enter password: password
```

利用可能なPDBを検索するには、DBA_PDBSデータ・ディクショナリ・ビューを問い合わせます。現在のPDBを確認するには、show con_nameコマンドを実行します。

- 新しいDatabase Vaultロールに既存のローカル・ユーザーの名前付きアカウントを使用しない場合は、新しい名前付きローカル・ユーザー・アカウントを作成します。

どちらの場合も、名前付きユーザーがパスワードを紛失または忘れた場合に備えて、Database Vaultロールを保持するバックアップ・アカウントを作成しておく必要があります。

```
GRANT CREATE SESSION, SET CONTAINER TO sec_admin_owen
  IDENTIFIED BY password;
GRANT CREATE SESSION, SET CONTAINER TO dbv_owner_backup
  IDENTIFIED BY password;
GRANT CREATE SESSION, SET CONTAINER TO accts_admin_ace
```



```
IDENTIFIED BY password;  
GRANT CREATE SESSION, SET CONTAINER TO dbv_acctmgr_backup  
IDENTIFIED BY password;
```

3. CDBルートにOracle Database Vaultを登録したことを確認します。
一時的にrootに接続して、DBA_DV_STATUSビューを問合せします。

```
CONNECT SYS / AS SYSDBA  
Enter password: password  
  
SELECT * FROM DBA_DV_STATUS;
```

4. SYSDBA管理権限を持つユーザーSYSとして、PDBに接続します。

```
CONNECT SYS@pdb_name AS SYSDBA  
Enter password: password
```

5. PDBにいる間に、2つのバックアップDatabase Vaultユーザー・アカウントを構成します。

```
BEGIN  
  CONFIGURE_DV (  
    dvowner_username => 'dbv_owner_backup',  
    dvacctmgr_username => 'dbv_acctmgr_backup');  
END;  
/
```

この例では、force_local_dvownerパラメータは不要なために省略されています。Database Vaultロールは、PDBで構成されているときにローカルに付与されます。

6. utlrlp.sqlスクリプトを実行して、このPDBで無効化されたオブジェクトを再コンパイルします。

```
@?/rdbms/admin/utlrlp.sql
```

スクリプトから指示がある場合はそれに従い、再びスクリプトを実行します。指示がなくスクリプトが異常終了した場合は、再びスクリプトを実行します。

7. 前の手順で構成したバックアップDatabase Vault所有者ユーザーとして、PDBに接続します。
次に例を示します。

```
CONNECT dbv_owner_backup@pdb_name  
Enter password: password
```

8. このPDBでOracle Database Vaultを有効にします。

```
EXEC DBMS_MACADM.ENABLE_DV;
```

9. SYSDBA管理権限でCDBに接続します。

```
CONNECT / AS SYSDBA
```

10. PDBを閉じてから、再度開きます。

```
ALTER PLUGGABLE DATABASE pdb_name CLOSE IMMEDIATE;  
ALTER PLUGGABLE DATABASE pdb_name OPEN;
```

11. PDBがDatabase VaultとOracle Label Securityに対して構成され有効になっていることを確認します。

```
CONNECT SYS@pdb_name AS SYSDBA  
Enter password: password
```

```
SELECT * FROM DBA_DV_STATUS;  
SELECT * FROM DBA_OLS_STATUS;
```

- バックアップDV_OWNERユーザーとして接続し、前に作成したプライマリDV_OWNERユーザーにDV_OWNERロールを付与します。

次に例を示します。

```
CONNECT dbv_owner_backup@pdb_name  
Enter password: password
```

```
GRANT DV_OWNER TO sec_admin_owen WITH ADMIN OPTION;
```

- バックアップDV_ACCTMGRユーザーとして接続してから、DV_ACCTMGRロールをバックアップDV_ACCTMGRユーザーに付与します。

次に例を示します。

```
CONNECT dbv_acctmgr_backup@pdb_name  
Enter password: password
```

```
GRANT DV_ACCTMGR TO c##accts_admin_ace WITH ADMIN OPTION;
```

- 2つのバックアップ・アカウント・パスワードを、将来必要になる場合に備えて、特権アカウント管理(PAM)システムなどの安全な場所に格納します。

親トピック: [マルチテナント環境におけるOracle DatabaseへのOracle Database Vaultの登録](#)

Database Vault対応PDBへの接続

マルチテナント環境では、SQL*Plusから、すでにDatabase Vaultが有効になっているデータベースに接続できます。

このシナリオでは、接続されるデータベースには、独自のローカルDatabase Vaultアカウントがあります。Database Vault対応データベースをDatabase Vaultが有効になっていないCDBに接続する場合は、CDBでDatabase Vaultを有効にしてCDBを再起動するまで、PDBは制限されたモードのままになることを覚えておいてください。Database Vaultが有効になっていないPDBをDatabase Vaultが有効になっているCDBに接続する場合は、PDBでDatabase Vaultを有効にしてPDBを再起動するまで、PDBは制限されたモードのままになります。このDatabase Vaultが有効になっていないPDBの接続も引き続き使用できます。ただし、CDBで厳密オプションが設定されてDatabase Vaultが有効になっている場合は、PDBでDatabase Vaultを有効にする必要があります。

Database Vault対応PDBを接続する前に、Database Vaultロールが共通ユーザーに付与されている場合は、PDBの接続が共通ユーザーに与える影響を完全に理解しておいてください。

関連トピック

- 『[Oracle Databaseセキュリティ・ガイド](#)』

親トピック: [マルチテナント環境におけるOracle DatabaseへのOracle Database Vaultの登録](#)

マルチテナント環境でのOracle Database Vaultの手動インストール

特定の条件のマルチテナント環境に対しては、Oracle Database Vaultを手動でインストールする必要があります。

たとえば、Database Vaultのないリリース11g Oracle Databaseをリリース12cにアップグレードしてから、12c Database Vault対応データベースに接続するPDBに変換する場合は、Oracle Database Vaultを手動でインストールする必要があります。また、Database VaultおよびLabel SecurityがインストールされているCDBにPDBが接続されており、このPDBにこれ

らの製品がない場合は、Oracle Database Vault (およびOracle Label Security)をPDBに手動でインストールする必要があります。

1. SYSDBA管理権限を付与されているユーザーとして、Oracle Database VaultをインストールするPDBにログインします。

たとえば、hr_pdbというPDBにログインするには、次のようにします。

```
sqlplus sec_admin@hr_pdb as sysdba
Enter password: password
```

使用可能なPDBを検索するには、show pdbsコマンドを実行します。現在のPDBを確認するには、show con_nameコマンドを実行します。

2. 必要な場合は、Oracle Database VaultおよびOracle Label SecurityがこのPDBにすでにインストールされているかどうかを確認します。

DVSYSAアカウント(Database Vault用)とLBACSYSアカウント(Label Security用)が存在している場合は、PDBにDatabase VaultとLabel Securityが存在しています。

```
SELECT USERNAME FROM DBA_USERS WHERE USERNAME IN (' DVSYS', ' LBACSYS');
```

3. Database VaultとLabel Securityのどちらもインストールされていない場合は、catols.sqlスクリプトを実行してOracle Label Securityをインストールします。

```
@$ORACLE_HOME/rdbms/admin/catols.sql
```

Oracle Label Securityは、Oracle Database Vaultをインストールする前にインストールしておく必要があります。

4. catmac.sqlスクリプトを実行することで、Oracle Database Vaultをインストールします。

```
@$ORACLE_HOME/rdbms/admin/catmac.sql
```

5. 「1に値を入力してください」プロンプトで、DVSYSAをインストールする表領域としてSYSTEMを入力します。
6. 「2に値を入力してください」プロンプトで、PDBの一時表領域を入力します。

インストールの完了後、PDBにOracle Database Vaultを登録できます。Database VaultがまだCDBに登録されていない場合は、PDBをクローズしてからCDBルートでDatabase Vaultを登録する必要があります。Database VaultはPDBで登録する前に、CDBルートで登録しておく必要があります。CDBルートでDatabase Vaultを登録してからデータベースを再起動すると、PDBをオープンしてDatabase Vaultを登録できます。

関連トピック

- [Oracle Database Vaultの管理者ガイド](#)

親トピック: [マルチテナント環境におけるOracle DatabaseへのOracle Database Vaultの登録](#)

レルの構成

データベース・オブジェクトを保護するために、これらのオブジェクトの周辺にレルを作成し、このデータへのユーザー・アクセスを制御するための認可を設定します。

- [レルの概要](#)
レルを使用すると、特定のオブジェクト・タイプなど、データベース・オブジェクトを保護できます。
- [マルチテナント環境におけるレル認可](#)
マルチテナント環境では、共通レル認可のルールおよび動作は、他の共通オブジェクトの認可と同様です。

レلمの概要

レلمを使用すると、特定のオブジェクト・タイプなど、データベース・オブジェクトを保護できます。

- [レلمについて](#)
レلمは、特定のアプリケーションのために保護する必要のあるデータベース・スキーマ、データベース・オブジェクトおよびデータベース・ロールのグループです。
- [マルチテナント環境におけるレلم](#)
マルチテナント環境では、アプリケーション・ルート内の共通オブジェクトを保護するためにレلمを作成できます。

親トピック: [レلمの構成](#)

レلمについて

レلمは、特定のアプリケーションのために保護する必要のあるデータベース・スキーマ、データベース・オブジェクトおよびデータベース・ロールのグループです。

レلمは、データベース・オブジェクトの保護ゾーンとみなすことができます。スキーマは、表、ビューおよびパッケージなどのデータベース・オブジェクトの論理的な集合で、ロールは権限の集合です。スキーマおよびロールを機能グループに分類することにより、システム権限を使用するユーザーがこれらのグループに対して行える操作を制御し、データベース管理者またはシステム権限を持つその他の強力なユーザーによる不正なデータ・アクセスを防ぐことができます。Oracle Database Vaultは、既存のOracleデータベースの任意アクセス制御モデルを置き換えません。レلمおよびコマンド・ルール両方で、このモデルの上位の層として機能します。

Oracle Database Vaultには、通常と必須という2つのタイプのレلمがあります。どちらのタイプのレلمでも、スキーマ全体の保護と、スキーマ内の重要オブジェクト(表や索引)の選択的な保護が可能です。通常レلمの場合、オブジェクト権限を付与されているオブジェクトの所有者またはユーザーはレلم認可なしで問合せやDML操作を実行できますが、DDL操作の実行にはレلم認可が必要です。必須レلمの場合、レلم内のオブジェクトにさらに厳格な保護が適用されます。必須レلمの場合、オブジェクト権限およびシステム権限に基づくアクセスの両方がブロックされ、オブジェクト権限を持つユーザーはレلم認可なしで問合せやDML操作またはDDL操作を実行することができません。つまり、オブジェクト所有者であっても、自身のオブジェクトが必須レلمで保護されている場合、適切なレلم認可を受けないとそのオブジェクトにアクセスできません。

Oracle Flashback Technologyを使用するデータベースの場合、標準レلمと必須レلمのどちらも、フラッシュバック表に対する動作が強制的に同じになります。ユーザーにレلمに対する権限がある場合、ユーザーは、レلمで保護された表に対してFLASHBACK TABLE SQL文を実行できます。

情報ライフサイクル管理(ILM)を使用するデータベースの場合、Database Vault管理者は、DBMS_MACADM.AUTHORIZE_MAINTENANCE_USERおよびDBMS_MACADM.UNAUTHORIZE_MAINTENANCE_USERプロシージャを使用して、レلمで保護されたオブジェクトに誰がILM操作を実行可能かを制御できます。

スキーマ、スキーマ内の特定のタイプのすべてのオブジェクト、またはスキーマ内の個々のオブジェクトをレلمに登録できます。レلمを作成すると、レلمで保護する一連のスキーマ・オブジェクトまたはロール(セキュア・オブジェクト)に登録し、セキュア・オブジェクトにアクセスする一連のユーザーやロールを認可できます。通常レلمで保護されるオブジェクトは、直接オブジェクト権限を持つユーザーからのDMLアクセスを許可します。

たとえば、経理部で使用される既存のすべてのデータベース・スキーマを保護するレلمを作成できます。レلمに対して認可されていないユーザーは、保護された経理データにシステム権限を使用してアクセスすることを許可されません。スキーマ全体が保護される場合は、表、索引、プロシージャおよびその他のオブジェクトなど、スキーマ内のすべてのオブジェクトが保護されます。

Oracle Database Vaultに作成するレلم上でレポートを実行できます。開発フェーズやテスト・フェーズの間、また、本番

フェーズの間でも、シミュレーション・モードを使用して、アクセスをブロックするかわりにレلم違反のみを記録できます。これにより、Database Vaultレلمを使用してアプリケーションを迅速にテストできます。

Oracle Enterprise Manager Cloud ControlでOracle Database Vault Administratorページを使用することで、レلمを構成できます。別の方法としては、Oracle Database Vaultで提供されるPL/SQLインタフェースおよびパッケージを使用することで、レلمを構成できます。

親トピック: [レلمの概要](#)

マルチテナント環境におけるレلم

マルチテナント環境では、アプリケーション・ルート内の共通オブジェクトを保護するためにレلمを作成できます。

個々のプラグブル・データベース(PDB)内でこれらのオブジェクトの周りに多数のオブジェクトおよびレلمを作成するのではなく、アプリケーション・ルートでレلمを作成する利点は、それらをアプリケーション・ルートという1つの場所で作成できるということです。このような方法で、それらを一元的に管理できます。

CDBルートでは、共通レلمは作成できません。

Database Vaultの共通レلمは、通常レلمか必須レلمのどちらかにできます。このレلمは、PDB内のローカル・オブジェクトではなく、アプリケーション・ルート内のオブジェクトのみを保護します。CDBルート、アプリケーション・ルート、および影響を受けるPDBはすべて、Database Vaultに対応している必要があります。

共通レلمを構成するには、一般に、DV_OWNERまたはDV_ADMINロールが付与されている必要があります。共通レلمの共通認可を付与するには、アプリケーション・ルートに在る必要があります。アプリケーション・ルートに関連付けられているPDBにレلمを伝播するには、アプリケーション・ルートを同期させる必要があります。たとえば、saas_sales_appというアプリケーションを同期させるには、次のようにします。

```
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;
```

関連トピック

- [Oracle Database Vaultの管理者ガイド](#)

親トピック: [レلمの概要](#)

マルチテナント環境におけるレلم認可

マルチテナント環境では、共通レلم認可のルールおよび動作は、他の共通オブジェクトの認可と同様です。

共通レلمのローカル認可

共通レلمのローカル認可とは、ユーザーがアクセスしているPDBのためにこのユーザーが保持している認可のことを指します。

共通レلمのローカル認可のルールは、次のとおりです。

- DV_OWNERまたはDV_ADMINロールを共通で付与されているユーザーは、共通ユーザー、共通ロール、ローカル・ユーザーおよびローカル・ロールにローカル認可を付与できます。共通DV_OWNERまたはDV_ADMINユーザーは、PDB内の共通レلمからローカル認可を削除することもできます。
- ローカルDatabase Vault管理者は、PDB内のローカルで認可できます(つまり、ローカル認可をローカル・ユーザーと共通ユーザーの両方に付与する)。また、共通Database Vault管理者は、各PDBで認可を付与できます。共通レلم認可は、アプリケーション・ルートで共通Database Vault管理者のみが付与できます。
- 共通Database Vault管理者は、ローカル認可を、PDB内から共通レلمに追加することも、それから削除することもできます。

- 共通ユーザーに共通レلمのローカル認可しかない場合、このユーザーは、このローカル認可以外のPDB内の共通レلمにはアクセスできません。
- 共通ユーザーまたは共通ロールは、共通レلمへのローカル認可と共通認可の両方を同時に保持できます。共通レلمから共通ユーザーのローカル認可を削除しても、共通ユーザーの共通認可には影響しません。共通レلمから共通ユーザーの共通認可を削除しても、共通ユーザーのローカル認可には影響しません。

共通レلمの共通認可

共通レلمの共通認可とは、Database Vaultに対応しているすべてのコンテナで認可が有効になっていると同時に、共通ユーザーまたは共通ロールがアプリケーション・ルートで保持している認可のことを指します。

共通レلمのローカル認可のルールは、次のとおりです。

- DV_OWNERまたはDV_ADMINロールを共通で付与されているユーザーは、アプリケーション・ルート内の共通ユーザーまたはロールに共通レلم認可を付与できます。この共通Database Vault管理者は、アプリケーション・ルート内にいながら、共通認可の削除を実行できます。
- この共通認可は、CDB内の、Database Vaultに対応しているコンテナに適用されます。
- 共通ユーザーにアプリケーション・ルート内の共通レلمに対する権限がある場合、このユーザーは、アプリケーション・ルート内およびアプリケーションPDB内の共通レلمによって保護されているオブジェクトにアクセスできます。
- 共通レلمに関連付けられているルール・セットは、共通ルール・セットである必要があります。共通認可に関連付けられている共通ルール・セットに追加されるルールに、ローカル・オブジェクトを含めることはできません。

アプリケーション・ルート内と個々のPDB内でのレلمの認可の動作

コンテナでのDatabase Vault強制の間に、共通レلمは、それがPDBでローカルで使用される場合の同じレلمと同じ強制動作を実行します。

親トピック: [レلمの構成](#)

マルチテナント環境におけるルール・セットとルール

マルチテナント環境では、アプリケーション・ルートでルール・セットおよびその関連付けられたルールを作成できます。

共通レلمでは、関連付けられたレلمまたはコマンド・ルールをDatabase Vaultで評価するときに、共通ルール・セットを使用する必要があります。共通ルール・セットとそのルールは、アプリケーション・ルートでのみ作成できます。共通ルール・セットは、作成後、共通ルール・セットを作成したルートに関連付けられているすべてのコンテナ内に存在します。共通ルール・セットは、共通ルールのみを含むことができます。

共通ルール・セットとそのルールを構成するには、DV_OWNERまたはDV_ADMINロールが共通で付与されている必要があります。

関連トピック

- [マルチテナント環境におけるコマンド・ルール](#)

親トピック: [CDBでのOracle Database Vaultの使用](#)

マルチテナント環境におけるコマンド・ルール

マルチテナント環境では、CDBルートまたはアプリケーション・ルートのどちらかで、共通およびローカルのコマンド・ルールを作成できます。

共通コマンド・ルールは、共通のレلم、ルール・セットおよびルールのみに関連付けることができます。ローカル・コマンド・ルールは、ローカルのレلم、ルール・セットおよびルールのみに関連付けることができます。

これらのコマンド・ルールをマルチテナント環境全体に適用するには、DVADMまたはDVOWNERロールを付与された共通ユーザーとして、CDBルートまたはアプリケーション・ルートからコマンド・ルール・プロシージャを実行する必要があります。CDBルートで作成される共通コマンド・ルールは、そのCDB環境内のすべてのPDBに適用されます。アプリケーション・ルートで作成される共通コマンド・ルールは、このアプリケーション・ルートに関連付けられているPDBのみに適用されます。CDBルートまたはアプリケーション・ルートに関連付けられているPDBにコマンド・ルールを伝播するには、PDBを同期させる必要があります。たとえば、saas_sales_appというアプリケーション・ルートをアプリケーションPDBと同期するには、次のようにします。

```
ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;
```

CDBルートの共通コマンド・ルールをPDBと同期するには:

```
ALTER PLUGGABLE DATABASE APPLICATION APP$CDB$SYSTEM SYNC;
```

USER_ROLE_PRIVSデータ・ディクショナリ・ビューを問い合わせることで、ユーザーのロールを確認できます。コマンド・ルールに関する情報を確認するには、DBA_DV_COMMAND_RULEデータ・ディクショナリ・ビューを問い合わせます。

親トピック: [CDBでのOracle Database Vaultの使用](#)

マルチテナント環境におけるOracle Database Vaultポリシー

Oracle Database Vaultポリシーは、それらが作成されたプラグブル・データベース(PDB)に対してのみローカルとなります。

つまり、PDBでポリシーを作成した場合は、ローカルのレلمおよびコマンド・ルールのみをそれに追加できます。共通レلمまたは共通コマンド・ルールを保持できるDatabase Vaultポリシーは作成できません。

親トピック: [CDBでのOracle Database Vaultの使用](#)

Database Vault操作制御を使用したローカルPDBデータへのマルチテナント共通ユーザー・アクセスの制限

インフラストラクチャ・データベース管理者など、CDBルートの共通ユーザーでPDBアクセスを制御できます。

- [Database Vault操作制御の使用について](#)
自律環境、通常のクラウド環境またはオンプレミス環境で、共通ユーザーによるプラグブル・データベース(PDB)のローカル・データへのアクセスを自動的に制限できます。
- [Database Vault操作制御の有効化](#)
Database Vault操作制御を有効にするには、DBMS_MACADM. ENABLE_APP_PROTECTION PL/SQLプロシージャを使用します。
- [共通ユーザーおよびパッケージの例外リストへの追加](#)
PDBのローカル・データにアクセスする必要がある共通ユーザーとアプリケーションは、例外リストに追加できます。
- [共通ユーザーおよびパッケージの例外リストからの削除](#)
PDBのローカル・データにアクセスする必要がなくなったユーザーおよびアプリケーションは、例外リストから削除できます。
- [Database Vault操作制御の無効化](#)
Database Vault操作制御を無効にするには、DBMS_MACADM. DISABLE_APP_PROTECTION PL/SQLプロシージャを使用します。

関連トピック

- [Oracle Database Vault環境におけるDBA操作は、『Oracle Database Vault管理者ガイド』を参照](#)

親トピック: [CDBでのOracle Database Vaultの使用](#)

Database Vault操作制御の使用について

自律環境、通常のクラウド環境またはオンプレミス環境で、共通ユーザーによるプラグブル・データベース(PDB)のローカル・データへのアクセスを自動的に制限できます。

これを実現するために、共通ユーザー(インフラストラクチャ・データベース管理者やアプリケーションなど)に適用されるOracle Database Vault操作制御を使用できます。

Database Vault操作制御は、データベース管理者が高い権限を持つユーザーとしてCDBルートにログインする必要があるときに、PDBの顧客データにはアクセスできないようにするような状況で役立ちます。データベース操作制御では、PDBデータベース管理者をブロックしません。こうしたユーザーをブロックするには、PDBでOracle Database Vaultを有効にして、該当するユーザーをブロックするレلم制御などのDatabase Vault機能を使用します。

共通ユーザーやアプリケーションが、PDBのローカル・データへのアクセスが必要なタスクを実行する必要がある場合は、共通ユーザーとパッケージについてのDatabase Vault操作制御の例外リストを作成できます。例外リストに指定する共通ユーザーのタイプの例として、Oracle Textを担当するCTXSYSアプリケーション・アカウントが挙げられます。例外リストでパッケージを指定すると、例外リスト内のユーザーにフル・アクセスを提供することなく詳細な制御を適用できます。

Database Vault操作制御する一般的なプロセスは、次のとおりです。

1. Database Vault操作制御を有効にして、本番環境でも有効にしたままにします。
2. この段階では、Database Vault操作制御が環境内のすべてのPDBに適用されます。PDBでDatabase Vaultを有効にしているかどうかは関係ありません。
3. 特定のユーザーとパッケージがPDBのローカル・スキーマにアクセスできるようにするには、それらを例外リストに追加します。ユーザーまたはパッケージがアクセスを必要としなくなったら、それらを例外リストから削除します。たとえば、データベースでOracle Textを使用している場合は、例外リストにCTXSYS管理ユーザー・アカウントとパッケージを追加できます。

親トピック: [Database Vault操作制御を使用したローカルPDBデータへのマルチテナント共通ユーザー・アクセスの制限](#)

Database Vault操作制御の有効化

Database Vault操作制御を有効にするには、DBMS_MACADM. ENABLE_APP_PROTECTION PL/SQLプロシージャを使用します。

マルチテナント本番サーバーにDatabase Vault操作制御を使用する場合は、Database Vault操作制御を常時有効にしておくことをお勧めします。

ほとんどの場合、特定のPDBのみでなく、CDB全体に対してデータベース操作制御を有効にします。特定のPDBに対して無効にする必要がある場合(トラブルシューティング目的など)、そのPDBでDBMS_MACADM. DISABLE_APP_PROTECTIONプロシージャを実行します。PDBのトラブルシューティングが終了したら、このトピックの例で示すように、Database Vault操作制御を再度有効化します。

Database Vault操作制御を有効にする前に、CDBルートでDatabase Vaultを有効化して構成しておく必要があります。ただし、PDBでDatabase Vaultを有効化する必要はありません。

1. DV_OWNERロールが付与されている共通ユーザーとしてCDBルートにログインします。
次に例を示します。

```
sqlplus c##sec_admin_owen_root
Enter password: password
```

2. DBMS_MACADM. ENABLE_APP_PROTECTIONプロシージャを実行します。
 - CDB環境内のすべてのPDBに対してDatabase Vault操作制御を有効化するには:

```
EXEC DBMS_MACADM. ENABLE_APP_PROTECTION;
```

- 特定のPDBの操作の制御は、トラブルシューティングの理由で無効になっている可能性があります。特定のPDB (HRPDBなど)に対するDatabase Vault操作の制御を再度有効にするには、次のようにします。

```
EXEC DBMS_MACADM. ENABLE_APP_PROTECTION (' HRPDB');
```

この段階で、1つまたはすべてのPDBのDatabase Vault操作制御が有効化されています。SYSDBA管理権限を持つユーザーSYSとして接続してからSELECT * FROM DBA_DV_STATUS;問合せを実行すると、確認できます。特定の信頼できる共通ユーザーまたはパッケージが、これらのPDBのローカル・スキーマにアクセスして特別な操作を実行する必要がある場合は、DBMS_MACADM.ADD_APP_EXCEPTIONプロシージャを使用して、そのユーザーまたはパッケージをDatabase Vault操作制御の例外リストに追加できます。

関連トピック

- [共通ユーザーおよびパッケージの例外リストへの追加方法は、『Oracle Database Vault管理者ガイド』を参照](#)

親トピック: [Database Vault操作制御を使用したローカルPDBデータへのマルチテナント共通ユーザー・アクセスの制限](#)

共通ユーザーおよびパッケージの例外リストへの追加

PDBのローカル・データにアクセスする必要がある共通ユーザーとアプリケーションは、例外リストに追加できます。

PDBのローカル・データにアクセスする必要があるユーザー・アカウントの唯一のオブジェクトがユーザー・パッケージである場合は、そのパッケージを例外リストに追加します。これにより、例外リストに登録するものを詳細に制御できます。例外リストに追加する共通ユーザーとパッケージの種類は、PDBの機能に必要なものにします。たとえば、Oracle Spatialを使用している場合は、MDSYSアカウントを例外リストに追加する必要があります。MDSYSは、Oracle Spatial関数ために顧客PDBデータにアクセスする必要があります。共通ユーザーおよびパッケージをDatabase Vault操作制御例外リストに追加するには、DBMS_MACADM.ADD_APP_EXCEPTION PL/SQLプロシージャを使用できます。既存の例外を調べるには、DBA_DV_APP_EXCEPTIONデータ・ディクショナリ・ビューを問合せします。

1. DV_OWNERロールが付与されている共通ユーザーとしてCDBルートにログインします。
次に例を示します。

```
sqlplus c##sec_admin_owen_root  
Enter password: password
```

2. 共通ユーザーに指定するパッケージが、次の要件を満たしていることを確認します。
 - パッケージは、共通ユーザーが所有している必要があります。
 - ユーザー作成のパッケージは、定義者権限のプロシージャで作成する必要があります。

ユーザー作成パッケージの詳細は、DBA_OBJECTSデータ・ディクショナリ・ビューを問い合わせることで確認できます。

3. DBMS_MACADM.ADD_APP_EXCEPTIONプロシージャを実行します。
次に例を示します。

```
DBMS_MACADM.ADD_APP_EXCEPTION (' MDSYS', ' PATCH_APP');
```

親トピック: [Database Vault操作制御を使用したローカルPDBデータへのマルチテナント共通ユーザー・アクセスの制限](#)

共通ユーザーおよびパッケージの例外リストからの削除

PDBのローカル・データにアクセスする必要がなくなったユーザーおよびアプリケーションは、例外リストから削除できます。

共通ユーザーおよびパッケージをDatabase Vault操作制御例外リストから削除するには、

DBMS_MACADM.DELETE_APP_PROTECTION PL/SQLプロシージャを使用できます。既存の例外を調べるには、DBA_DV_APP_EXCEPTIONデータ・ディクショナリ・ビューを問合せします。

1. DV_OWNERロールが付与されている共通ユーザーとしてCDBルートにログインします。

次に例を示します。

```
sqlplus c##sec_admin_owen_root
Enter password: password
```

2. DBMS_MACADM.DELETE_APP_EXCEPTIONプロシージャを実行します。

次に例を示します。

```
DBMS_MACADM.DELETE_APP_EXCEPTION ('MDSYS', 'PATCH_APP');
```

親トピック: [Database Vault操作制御を使用したローカルPDBデータへのマルチテナント共通ユーザー・アクセスの制限](#)

Database Vault操作制御の無効化

Database Vault操作制御を無効にするには、DBMS_MACADM.DISABLE_APP_PROTECTION PL/SQLプロシージャを使用します。

ほとんどの場合、Database Vault操作制御は有効のままにしておく必要があります。トラブルシューティングの際にPDBのDatabase Vault操作制御を無効にする必要がある場合は、そのPDBのDatabase Vault操作制御を一時的に無効にすることをお勧めします(その他のPDBに対する操作制御は有効のままにしておきます)。トラブルシューティングが完了したら、Database Vault操作制御を再度有効化します。

1. DV_OWNERロールが付与されている共通ユーザーとしてCDBルートにログインします。

次に例を示します。

```
sqlplus c##sec_admin_owen_root
Enter password: password
```

2. DBMS_MACADM.DISABLE_APP_PROTECTIONプロシージャを実行します。

- CDB環境内のすべてのPDBに対するDatabase Vault操作制御を無効化するには:

```
EXEC DBMS_MACADM.DISABLE_APP_PROTECTION;
```

- 特定のPDB (例: HRPDB)に対してDatabase Vault操作制御を無効化するには、次のようにします。

```
EXEC DBMS_MACADM.DISABLE_APP_PROTECTION ('HRPDB');
```

親トピック: [Database Vault操作制御を使用したローカルPDBデータへのマルチテナント共通ユーザー・アクセスの制限](#)

スタンドアロンのOracle DatabaseをPDBに変換してCDBにプラグイン

リリース12c以降のスタンドアロンのOracle DatabaseはPDBに変換可能で、さらにそのPDBはCDBにプラグインできます。

1. DV_OWNERロールを付与されているユーザーとしてルートに接続します。

次に例を示します。

```
sqlplus c##sec_admin
Enter password: password
```

2. CONTAINER = CURRENTを指定して、ユーザーSYSにDV_PATCH_ADMINロールを付与します。

```
GRANT DV_PATCH_ADMIN TO SYS CONTAINER = CURRENT;
```

3. ルートで、SYSOPERシステム権限を持つユーザーSYSとして接続します。

次に例を示します。

```
CONNECT SYS AS SYSOPER
Enter password: password
```

4. データベースを読み取り専用モードで再起動します。

次に例を示します。

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
ALTER DATABASE OPEN READ ONLY
```

5. Database Vault対応のデータベースに、DV_OWNERロールを持つユーザーとして接続します。

次に例を示します。

```
CONNECT sec_admin@pdb_name
```

6. このデータベースで、ユーザーSYSにDV_PATCH_ADMINロールを付与します。

```
GRANT DV_PATCH_ADMIN TO SYS;
```

7. オプションで、DBMS_PDB.CHECK_PLUG_COMPATIBILITYファンクションを実行して、切断されたPDBがCDBと互換性があるかどうかを確認します。

このファンクションを実行する場合は、次のパラメータを設定します。

- `pdb_descr_file`: PDBの記述を含むXMLファイルへのフルパスを設定します。
- `store_report`: PDBにCDBと互換性がない場合にレポートを生成するかどうかを指定します。レポートを生成する場合はTRUEに、レポートを生成しない場合はFALSEに設定します。生成されたレポートは、`PDB_PLUG_IN_VIOLATIONS`一時表に格納され、PDBにCDBとの互換性がない場合にのみ生成されます。

たとえば、`/disk1/usr/dv_db_pdb.xml`ファイルで記述されているPDBに現在のCDBと互換性があるかどうかを判断するには、次のPL/SQLブロックを実行します。

```
SET SERVEROUTPUT ON
DECLARE
  compatible CONSTANT VARCHAR2(3) :=
    CASE DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
      pdb_descr_file => '/disk1/usr/dv_db_pdb.xml',
      store_report => TRUE)
    WHEN TRUE THEN 'YES'
    ELSE 'NO'
END;
BEGIN
  DBMS_OUTPUT.PUT_LINE(compatible);
END;
/
```

出力がYESの場合はPDBに互換性があり、次のステップに進むことができます。

出力がNOの場合は、PDBに互換性がありません。`PDB_PLUG_IN_VIOLATIONS`一時表を調べると、互換性がない理由を確認できます。

8. PDBを記述するXMLファイルを作成します。

次に例を示します。

```
BEGIN
  DBMS_PDB.DESCRIBE (
    pdb_descr_file => '/disk1/oracle/dv_db.xml');
END;
/
```

9. CREATE PLUGGABLE DATABASE文を実行し、USING句でXMLファイルを指定します。要求された場合には、他の句を指定します。

次に例を示します。

```
CREATE PLUGGABLE DATABASE pdb_name AS CLONE USING 'dv_db.xml' NOCOPY;
```

10. 作成したPDBに、SYSDBA管理権限を持つユーザーSYSとして接続します。

```
CONNECT SYS@pdb_name AS SYSDBA
```

11. noncdb_to_pdb.sqlスクリプトを実行します。

```
@$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

12. このPDBを読み取り/書き込み制限モードでオープンします。

```
ALTER PLUGGABLE DATABASE pdb_name OPEN READ WRITE RESTRICTED;
```

13. 次のプロシージャを実行してPDBを同期します。

```
EXECUTE DBMS_PDB.SYNC_PDB;
```

14. DV_OWNERロールを付与されているユーザーとしてルートに接続します。

```
sqlplus c##sec_admin
Enter password: password
```

15. CONTAINER = CURRENTを指定して、ユーザーSYSからDV_PATCH_ADMINロールを取り消します。

```
REVOKE DV_PATCH_ADMIN FROM SYS CONTAINER = CURRENT;
```

16. Database Vault対応のレガシー・データベースに、SYSOPERシステム権限を持つユーザーSYSとして接続します。

```
CONNECT SYS@pdb_name AS SYSOPER
```

17. このデータベースを再起動します。

次に例を示します。

```
SHUTDOWN IMMEDIATE
STARUP
```

18. ユーザーSYSからDV_PATCH_ADMINロールを取り消します。

```
REVOKE DV_PATCH_ADMIN FROM SYS;
```

関連トピック

- [PDBおよびアプリケーション・コンテナの作成および削除](#)

親トピック: [CDBでのOracle Database Vaultの使用](#)

25 CDBでのXStreamの使用

マルチテナント・コンテナ・データベース(CDB)では、Oracle Database XStreamを使用できます。

- [XStreamについて](#)
XStreamは、Oracle Databaseコンポーネントおよびアプリケーション・プログラミング・インタフェース(API)で構成され、Oracleデータベースからのデータ変更をクライアント・アプリケーションで受信し、クライアント・アプリケーションからOracleデータベースにデータの変更を送信できるようにします。
- [システム生成ルールとマルチテナント環境](#)
マルチテナント環境では、移植可能な一連のスキーマ、オブジェクト、および論理的に個別のデータベースとしてアプリケーションに表示される関連する構造をOracleデータベースに含めることができます。この自己完結型コレクションは、プラグブル・データベース(PDB)と呼ばれます。CDBにはPDBが含まれています。
- [XStream Outとマルチテナント環境](#)
マルチテナント環境では、移植可能な一連のスキーマ、オブジェクト、および論理的に個別のデータベースとしてアプリケーションに表示される関連する構造をデータベースに含めることができます。
- [CDBでのXStream Outの構成](#)
CDBでXStream Outを構成する場合は、どのデータベース変更をXStream Outで取得してクライアント・アプリケーションに送信するかを決定する必要があります。
- [XStream Inとマルチテナント環境](#)
マルチテナント環境では、移植可能な一連のスキーマ、オブジェクト、および論理的に個別のデータベースとしてアプリケーションに表示される関連する構造をOracleデータベースに含めることができます。

親トピック: [マルチテナント環境でのOracle機能の使用](#)

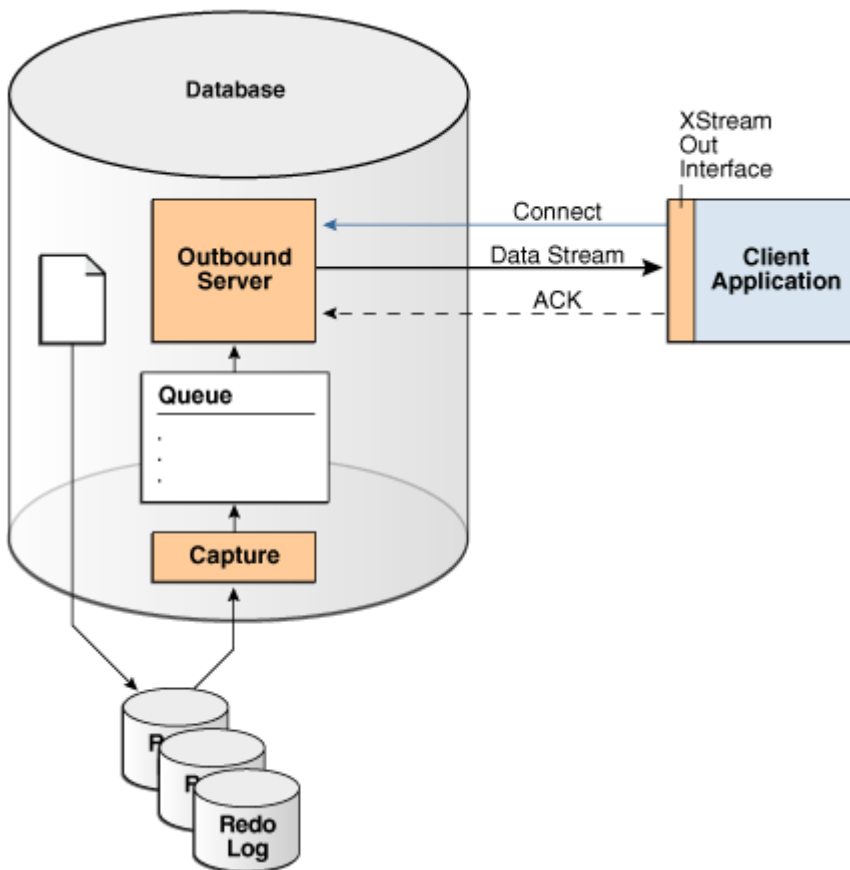
XStreamについて

XStreamは、Oracle Databaseコンポーネントおよびアプリケーション・プログラミング・インタフェース(API)で構成され、Oracleデータベースからのデータ変更をクライアント・アプリケーションで受信し、クライアント・アプリケーションからOracleデータベースにデータの変更を送信できるようにします。

これらのデータ変更は、Oracleデータベースと他のシステムで共有できます。他のシステムとは、Oracle以外のデータベース、非RDBMS Oracle製品、ファイル・システム、サード・パーティのソフトウェア・アプリケーションなどです。クライアント・アプリケーションは、ユーザーにより固有の目的およびユースケース用に設計されます。

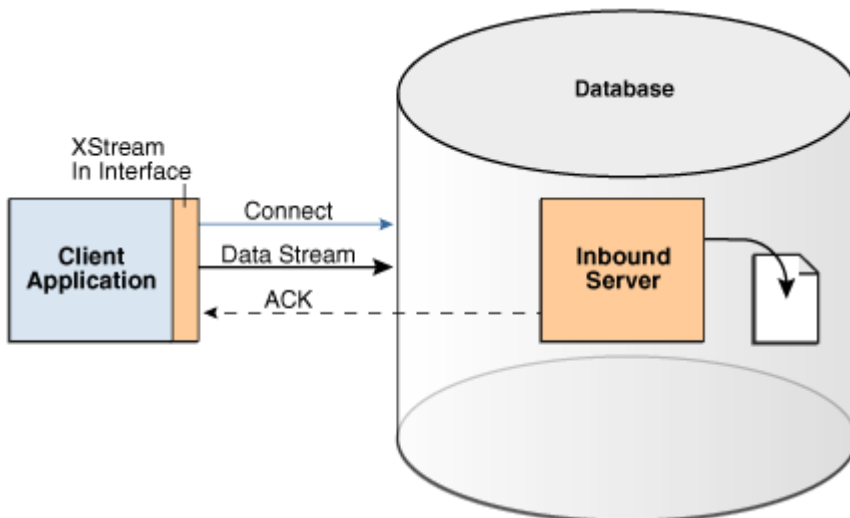
XStreamは、XStream OutおよびXStream Inの2つの主要な機能で構成されています。XStream Outには、Oracleデータベースでのデータ変更を他のシステムと共有できるようにする、Oracle DatabaseのコンポーネントおよびAPIがあります。XStream Outは、データ操作言語(DML)およびデータ定義言語(DDL)の変更をREDOログから取得し、APIを使用するクライアント・アプリケーションにこれらの変更を送信できます(次の図を参照)。

図25-1 XStream Out



XStream Inには、他のシステムでのデータ変更をOracleデータベースと共有できるようにする、Oracle DatabaseのコンポーネントおよびAPIがあります。XStream Inは、これらの変更をOracleデータベース内のデータベース・オブジェクトに適用できます(次の図を参照)。

図25-2 XStream In



XStreamは、Oracleデータベースの取得機能および適用機能を使用します。これらの機能によって、XStreamの次の機能が使用可能になります。

- データベース変更をストリーミングするための論理変更レコード(LCR)形式

LCRは、データベース変更を説明する特定の形式のメッセージです。変更がデータ操作言語(DML)操作だった場合は、行LCRにより、DML操作の結果として発生した行変更がそれぞれカプセル化されます。1つのDML操作によって複数の行変更が発生する場合があるため、1つのDML操作によって複数の行LCRが発生することがあります。変更がデータ定義言語(DDL)操作だった場合、1つのDDL LCRがDDL変更をカプセル化します。

- 包含ルールや除外ルールなど、動作を制御するルールおよびルール・セット

ルールによって、データベース・レベル、スキーマ・レベル、表レベルおよび行/列レベルでデータベースの変更がフィルタリ

が可能になります。

- 取得したデータ変更を変更するルールベースの変換
- LOB、LONG、LONG RAW、XMLTypeなど、データベースのほとんどのデータ型のサポート
- 単一のデータベース・インスタンスへの複数のインバウンド・ストリーム、単一のデータベース・インスタンスからの複数のアウトバウンド・ストリーム、単一の取得プロセスからの複数のアウトバウンド・ストリームなど、カスタマイズされた構成
- 最適なパフォーマンスのための並列性の適用、SQLの生成、競合の検出と解決、エラー処理、適用ハンドラによるカスタマイズされた適用など、XStream Inのフル装備の適用

ノート:



XStream Out 構成および XStream In 構成のいずれにおいても、クライアント・アプリケーションは専用サーバー接続を使用する必要があります。

関連トピック

- [Oracle Database XStreamガイド](#)

親トピック: [CDBでのXStreamの使用](#)

システム生成ルールとマルチテナント環境

マルチテナント環境では、移植可能な一連のスキーマ、オブジェクトおよび関連構造をOracleデータベースに含めることができ、アプリケーションには論理的に別のデータベースのように見えます。この自己完結型コレクションは、プラグابل・データベース (PDB)と呼ばれます。CDBにはPDBが含まれています。

アプリケーション・コンテナも含めることができます。アプリケーション・コンテナは、アプリケーション・ルートとそれに関連付けられたアプリケーションPDBから構成されるCDBのオプション・コンポーネントです。アプリケーション・コンテナには、1つ以上のアプリケーションのデータが格納されています。アプリケーション・コンテナは、アプリケーションのメタデータおよび共通データを共有します。CDBでは、CDBルート、各PDB、各アプリケーション・ルートおよび各アプリケーションPDBのそれぞれがコンテナです。

CDBでは、変更が発生したコンテナのグローバル名をsource_database_name属性で、CDBルートのグローバル名をroot_name属性でLCRIに含めることができます。XStreamコンポーネントのルールで、これらの属性を考慮できます。

- [CDBのシステム生成ルールとXStream Out](#)

CDBでは、CDBルートでXStream Outを構成する必要があります。そのため、システム生成ルールを作成するDBMS_XSTREAM_ADMパッケージのPL/SQLプロシージャは、CDBルートで実行する必要があり、さらに共通ユーザーとして接続されている必要があります。

- [CDBのシステム生成ルールとXStream In](#)

XStream Inは、CDBのルートまたは任意のコンテナで構成できます。

関連トピック

- [マルチテナント・アーキテクチャ](#)

親トピック: [CDBでのXStreamの使用](#)

CDBのシステム生成ルールとXStream Out

CDBでは、XStream OutをCDBルートで構成する必要があります。そのため、システム生成ルールを作成するDBMS_XSTREAM_ADMパッケージのPL/SQLプロシージャは、CDBルートで実行する必要があり、さらに共通ユーザーとして接続されている必要があります。

伝播のルールを作成するプロシージャを除く、ADD_GLOBAL_RULESプロシージャなどのXStream Outのシステム生成ルールを作成するプロシージャには、次の表に示す主要なパラメータが含まれています。

表25-1 CDBのシステム生成ルールの主要なプロシージャ・パラメータ

パラメータ	説明
source_database	ソース・データベースのグローバル名。CDB では、ルールに関連するコンテナのグローバル名を指定してください。コンテナは、CDB ルート、PDB、アプリケーション・ルートまたはアプリケーション PDB にできます。例: mycdb. example. com または hrpdb. example. com。
source_root_name	ソース CDB の CDB ルートのグローバル名。例: mycdb. example. com。
source_container_name	ソース・コンテナの短縮名。コンテナは、CDB ルート、PDB、アプリケーション・ルートまたはアプリケーション PDB にできます。例: CDB\$ROOT または hrpdb。

source_databaseまたはsource_root_nameを指定する際にドメイン名を指定しなかった場合は、名前に自動的に追加されます。たとえば、ドメイン名が EXAMPLE.COMの場合にDBS1を指定すると、自動的にDBS1.EXAMPLE.COMが指定されます。

これらの主要なパラメータの組合せによって、プロシージャで生成されたルールに基づいて、どのコンテナの変更をXStream Outが取得し、クライアント・アプリケーションにストリームするのが決まります。これらのパラメータの設定に関係なく、システム生成ルールでは、特定のスキーマおよび表に取得され、ストリームされる変更を引き続き制限できます。

ローカル取得とは、ソースCDB上で取得プロセスが実行されることを意味します。ローカル取得構成では、source_root_nameパラメータで、ローカルCDBのCDBルートのグローバル名を指定します。このパラメータがNULLの場合は、ローカルCDBのCDBルートのグローバル名が自動的に指定されます。生成されるルールには、現在のCDBのCDBルートのグローバル名に対する条件が含まれます。

ダウンストリーム取得とは、ソースCDB以外のCDB上で取得プロセスが実行されることを意味します。ダウンストリーム取得構成では、source_root_nameパラメータにNULL以外の値が必要で、リモート・ソースCDBのCDBルートのグローバル名を指定する必要があります。生成されるルールには、リモートCDBのCDBルートのグローバル名に対する条件が含まれます。このパラメータがNULLの場合は、ローカル取得とみなされます。

次の表は、ローカル取得構成でのsource_databaseおよびsource_container_nameの様々なパラメータ設定に対するルール条件を示しています。

表25-2 ローカル取得およびXStream Outのコンテナのルール条件

source_databaseパラメータの設定	source_container_nameパラメータの設定	説明
NULL	NULL	XStream Out は、CDB ルート、すべての PDB、すべてのアプリケーション・ルート、すべてのアプリケーション PDB など、ローカル CDB のコンテナに対する変更を取得してストリームしま

source_databaseパラ メータの設定	source_container_na meパラメータの設定	説明
		す。
NULL 以外	NULL	XStream Out は、ローカル CDB の指定されたソース・コンテナで行われた変更を取得してストリームします。ソース・コンテナは、CDB ルート、PDB、アプリケーション・ルートまたはアプリケーション PDB にできます。DBMS_XSTREAM_ADM プロシージャが、CDB_PDBS ビューおよび CDB_PROPERTIES ビューを問い合わせ、source_container_name 値を決定します。
NULL	NULL 以外	XStream Out は、ローカル CDB の指定されたソース・コンテナで行われた変更を取得してストリームします。ソース・コンテナは、CDB ルート、PDB、アプリケーション・ルートまたはアプリケーション PDB にできます。DBMS_XSTREAM_ADM プロシージャが、CDB_PDBS ビューおよび CDB_PROPERTIES ビューを問い合わせ、source_database 値を決定します。
NULL 以外	NULL 以外	XStream Out は、ローカル CDB の指定されたソース・コンテナで行われた変更を取得してストリームします。ソース・コンテナは、CDB ルート、PDB、アプリケーション・ルートまたはアプリケーション PDB にできます。 source_database 値の接頭辞が source_container_name 値と異なる場合、生成されるルールには source_database 値に対する条件が含まれ、内部表により source_database 値が source_container_name 値にマップされます。

次の表は、ダウンストリーム取得構成でのsource_databaseおよびsource_container_nameの様々なパラメータ設定に対するルール条件を示しています。

表25-3 ダウンストリーム取得およびXStream Outコンテナのルール条件

source_databaseパラ メータの設定	source_container_na meパラメータの設定	説明
NULL	NULL	XStream Out は、CDB ルート、すべての PDB、すべてのアプリケーション・ルート、すべてのアプリケーション PDB など、リモート・ソース CDB のコンテナに対する変更を取得してストリームします。
NULL 以外	NULL	XStream Out は、リモート・ソース CDB の指定されたソース・コンテナで行われた変更を取得してストリームします。ソース・コンテナは、CDB ルート、PDB、アプリケーション・ルートまた

source_databaseパラ メータの設定	source_container_na meパラメータの設定	説明
		はアプリケーション PDB にできます。DBMS_XSTREAM_ADM プロシージャは、source_database 値の接頭辞から source_container_name 値を導出します。
NULL	NULL 以外	DBMS_XSTREAM_ADM プロシージャでエラーが発生します。
NULL 以外	NULL 以外	XStream Out は、リモート・ソース CDB の指定されたソース・コンテナで行われた変更を取得してストリームします。ソース・コンテナは、CDB ルート、PDB、アプリケーション・ルートまたはアプリケーション PDB にできます。 source_database 値の接頭辞が source_container_name 値と異なる場合、生成されるルールには source_database 値に対する条件が含まれ、内部表により source_database 値が source_container_name 値にマップされます。

関連トピック

- [Oracle Database XStreamガイド](#)
- [Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)

親トピック: [システム作成ルールとマルチテナント環境](#)

CDBのシステム生成ルールとXStream In

XStream Inは、CDBのルートまたは任意のコンテナで構成できます。

一般的にインバウンド・サーバーはルールまたはルール・セットを使用しません。かわりに、通常はクライアント・アプリケーションから受信したすべてのLCRを処理します。インバウンド・サーバーは、現在のコンテナにのみ変更を適用できます。したがって、インバウンド・サーバーがCDBルートで設定されている場合は、CDBルートにのみ変更を適用できます。インバウンド・サーバーが1つのPDBで構成されている場合は、そのPDBにのみ変更を適用できます。インバウンド・サーバーがアプリケーション・ルートで構成されている場合は、そのアプリケーション・ルートにのみ変更を適用でき、インバウンド・サーバーが1つのアプリケーションPDBで構成されている場合は、そのアプリケーションPDBにのみ変更を適用できます。

関連トピック

- [マルチテナント環境の管理](#)

親トピック: [システム作成ルールとマルチテナント環境](#)

XStream Outとマルチテナント環境

マルチテナント環境では、移植可能な一連のスキーマ、オブジェクトおよび関連構造をデータベースに含めることができ、アプリケーションには論理的に別のデータベースのように見えます。

この自己完結型コレクションは、プラグブル・データベース(PDB)と呼ばれます。1つのマルチテナント・コンテナ・データベース

(CDB)には複数のPDBが格納されます。CDBでは、XStream OutはCDB以外での場合とほぼ同様に機能します。

CDBには、アプリケーション・コンテナを含めることもできます。アプリケーション・コンテナは、アプリケーション・ルートとそれに関連付けられたすべてのアプリケーションPDBから構成されるCDBのオプション・コンポーネントです。アプリケーション・コンテナには、1つ以上のアプリケーションのデータが格納されています。アプリケーション・コンテナは、アプリケーションのメタデータおよび共通データを共有します。CDBでは、CDBルート、各PDB、各アプリケーション・ルートおよび各アプリケーションPDBのそれぞれがコンテナです。

CDBと非CDBでのXStream Out機能の主な違いは、次のとおりです。

- XStream OutはCDBルートでのみ構成する必要があります。
- XStream OutはCDB内の任意のコンテナに適用された変更を表示できます。
- XStream Out取得ルールは、LCRをクライアント・アプリケーションに必要なLCRに制限できます。システム生成の取得ルールは、DBMS_XSTREAM_ADMパッケージのADD_OUTBOUNDおよびCREATE_OUTBOUNDプロシージャに渡されたパラメータに基づいて適切なLCRを選択します。XStream Outコンポーネントで使用されるルールよりも詳細に制御するために、同じパッケージのADD_*_RULESプロシージャを使用できます。
- XStream Outタスクを実行するユーザーは、共通ユーザーである必要があります。

XStream環境での操作の切断および接続

XStream Outに関連するPDB、アプリケーション・ルートまたはアプリケーションPDBがそのCDBから切断され、別のCDBに接続された場合、取得プロセスまたはアウトバウンド・サーバーはコンテナの一部とはみなされません。他のCDBで、取得プロセスおよびアウトバウンド・サーバーを再度構成する必要があります。

アウトバウンド・サーバーが取得プロセスとは異なるデータベースに構成されている場合は、切断および接続の操作に対して追加の考慮事項があります。

この例では、次のことを想定しています。

- CDB1という名前のCDBには、PDB PDB1が含まれています。
- 取得プロセスはCDB1で構成され、PDB1からCDB2というCDBのアウトバウンド・サーバーにLCRを送信します。
- PDB1をCDB1から切断して、CDB3というCDBに接続します。

PDB1からCDB2のアウトバウンド・サーバーにLCRを引き続き配信するには、CDB3で新しい取得プロセスを構成し、LCRを取得してCDB2に送信する必要があります。

CDB1のルートへの参照をCDB3のルートに変更するように、データベースBのアウトバウンド・サーバーで使用されるルールを変更する必要があります。また、CDB3でPDB1に別の名前が指定された場合は、新しいPDBの名前を反映するようにルールを変更する必要があります。

XStream環境のアプリケーション・コンテナ

CDBに1つ以上のアプリケーション・コンテナが含まれている場合は、XStream OutをCDBルートで構成する必要があり、それによってXStream Outは、アプリケーション・ルートやアプリケーションPDBなどのCDBのコンテナに追加された変更を取得できます。アプリケーション・コンテナに取得された変更は、PDB、アプリケーション・ルート、アプリケーションPDBなど、任意のタイプのコンテナに送信できます。

あるアプリケーション・ルートから別のアプリケーション・ルートに変更をレプリケートする場合は、XStreamでALTER PLUGGABLE DATABASE APPLICATION文をレプリケートできます。エラーを回避するには、文を適用するターゲット・アプリケーション・ルートに、ソース・アプリケーション・ルートと同じアプリケーションがインストールされ、両方のアプリケーション・ルートでアプリケーション名が同じである必要があります。

アプリケーション・ルートからアプリケーション・ルートでないコンテナに変更をレプリケートする際にエラーの発生を回避するには、ALTER PLUGGABLE DATABASE APPLICATION文がレプリケートされないようにする必要があります。

XStream OCI APIでは、OCIStreamOutAttach関数およびOCILCRHeaderGet関数を使用して、ALTER PLUGGABLE DATABASE APPLICATION文が複製されるかどうかを制御できます。XStream Java APIでは、XStreamOut.attachメソッドのmodeパラメータを使用して、この動作を制御できます。

関連トピック

- [システム生成ルールとマルチテナント環境](#)
- [CDBでのXStream Outの構成](#)
- [マルチテナント・アーキテクチャ](#)

親トピック: [CDBでのXStreamの使用](#)

CDBでのXStream Outの構成

CDBでXStream Outを構成する場合は、どのデータベース変更をXStream Outで取得してクライアント・アプリケーションに送信するかを決定する必要があります。

XStream Outでは、CDBルートとすべてのPDB、アプリケーション・ルート、アプリケーションPDBなど、すべてのコンテナに対するすべてのデータベース変更をストリームするか、特定のコンテナからの変更をストリームできます。さらに、ローカル取得を使用するようにXStream Outを構成するか、ダウンストリーム取得を使用するようにXStream Outを構成して、ソース・データベースからの変更を取得するために必要な作業を軽減できます。

CDBでXStream Outを構成する場合は、次の制限が適用されます。

- 取得プロセスおよびアウトバウンド・サーバーは、CDBルートに存在する必要があります。
- 取得プロセスおよびアウトバウンド・サーバーは、同じCDBに存在する必要があります。
- CDBの各コンテナは、XStream Out構成時にオープンされている必要があります。
- アプリケーション・ルートに対する変更を取得しているときは、ALTER PLUGGABLE DATABASE APPLICATION文が他のアプリケーション・ルートにのみレプリケートされるようにする必要があります。

また、CDBのXStream管理者を正しく作成していることを確認します。

ノート:



非 CDB を使用してコンテナが作成されている場合、コンテナでは非 CDB からの XStream Out コンポーネントを使用できません。CDB ルートで、取得プロセスやアウトバウンド・サーバーなどの XStream Out コンポーネントを削除し、再作成する必要があります。

- [ローカル取得を使用したCDBでのXStream Outの構成](#)
例では、ローカル取得を使用したCDBでのXStream Outの構成を示しています。
- [ダウンストリーム取得を使用したCDBでのXStream Outの構成](#)
ダウンストリーム取得を使用して、XStream Outコンポーネントをソース・データベース以外のデータベースに配置できます。

関連トピック

- [XStream Outとマルチテナント環境](#)
- [CDBのシステム生成ルールとXStream Out](#)
- [Oracle Database XStreamガイド](#)
- [非CDBからPDBを作成するためのオプション](#)

親トピック: [CDBでのXStreamの使用](#)

ローカル取得を使用したCDBでのXStream Outの構成

例では、ローカル取得を使用したCDBでのXStream Outの構成を示しています。

前提条件

XStream Outを構成する前に、CDBのすべてのコンテナが、XStream Out構成時に読取り/書込みモードでオープンしているようにします。

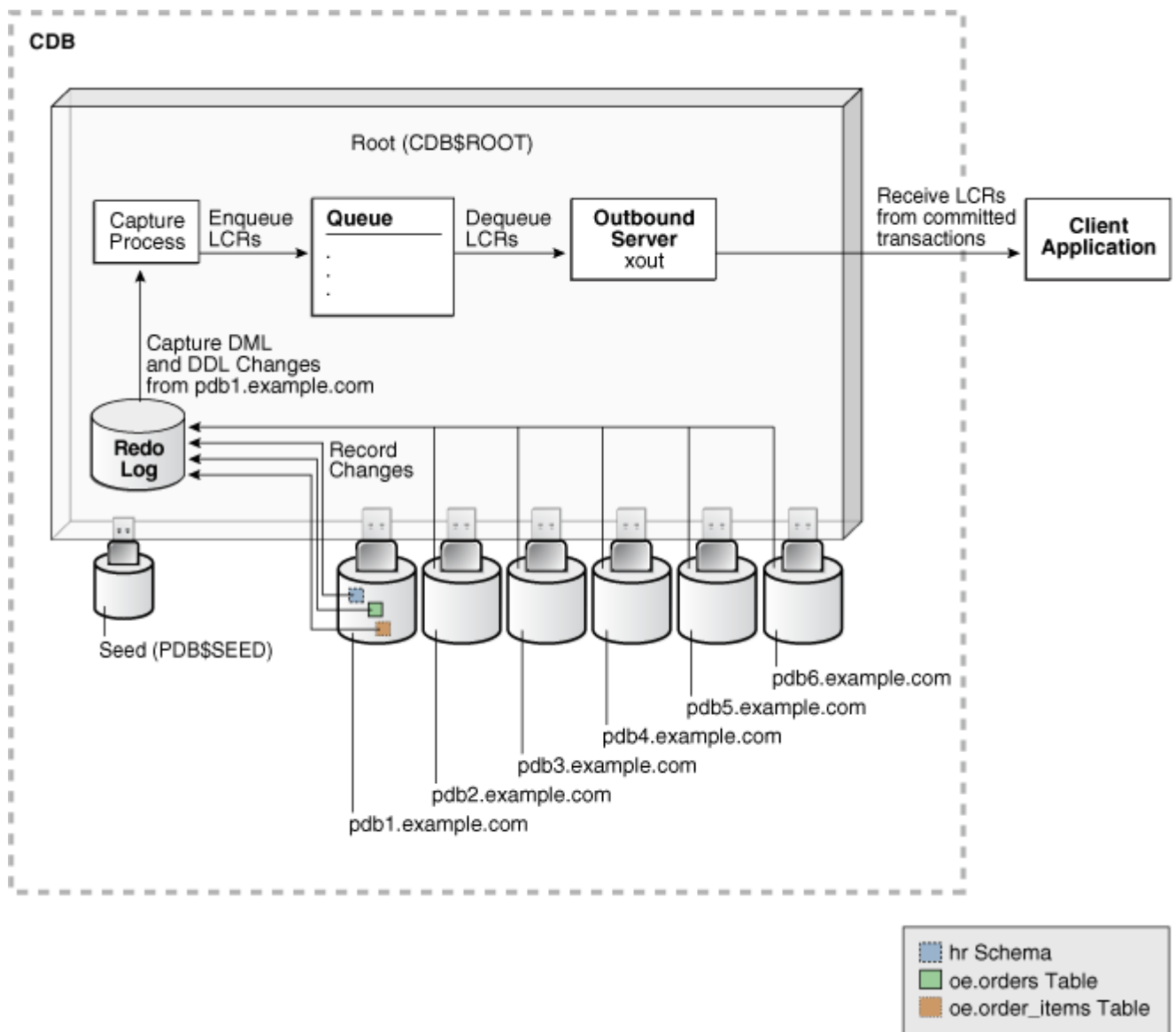
想定

この項では、次のことを想定しています。

- 取得プロセスはローカル取得プロセスであり、アウトバウンド・サーバーと同じデータベースで実行されます。
- アウトバウンド・サーバーの名前はxoutです。
- PDB `pdb1.example.com`の`oe.orders`表および`oe.order_items`表に対するデータ操作言語(DML)およびデータ定義言語(DDL)の変更は、アウトバウンド・サーバーに送信されます。
- PDB `pdb1.example.com`の`hr`スキーマに対するDML変更およびDDL変更は、アウトバウンド・サーバーに送信されません。

[図25-3](#)は、このXStream Out構成の概要を示しています。

図25-3 PDBのCREATE_OUTBOUNDを使用して作成されたサンプルXStream Out構成



CREATE_OUTBOUNDプロシージャを使用して、アウトバウンド・サーバーを作成するには:

1. SQL*Plusで、CDB (PDB pdb1. example. comではなく)のルートにXStream管理者として接続します。
2. アウトバウンド・サーバーおよび他のXStreamコンポーネントを作成します。
 - a. ソースCDB内のすべてのコンテナが読み取り/書き込みモードでオープンしていることを確認します。
 - b. CREATE_OUTBOUNDプロシージャを実行します。

この例の想定に従って、次のCREATE_OUTBOUNDプロシージャを実行します。

```

DECLARE
  tables DBMS_UTILITY.UNCL_ARRAY;
  schemas DBMS_UTILITY.UNCL_ARRAY;
BEGIN
  tables(1) := 'oe.orders';
  tables(2) := 'oe.order_items';
  schemas(1) := 'hr';
  DBMS_XSTREAM_ADM.CREATE_OUTBOUND(
    server_name => 'xout',
    source_database => 'pdb1.example.com',
    table_names => tables,
    schema_names => schemas);
END;
/

```

ノート:



CDB ルート、すべての PDB、すべてのアプリケーション・ルート、すべてのアプリケーション PDB など、CDB 内のすべてのコンテナでの変更を取得して、これらの変更内容を XStream クライアント・アプリケーションに送信するには、CREATE_OUTBOUND プロシージャを実行する際に source_database パラメータを省略します。

- c. CREATE_OUTBOUND プロシージャが正常に完了した後、必要に応じて、1つ以上のコンテナのオープン・モードを変更します。

ステップ **b** でプロシージャを実行すると、次の処理が実行されます。

- d. oe.orders 表、oe.order_items 表、および pdb1.example.com PDB の hr スキーマ内のすべての表のサブメンタル・ロギングを構成します。
- e. 取得プロセスおよびアウトバウンド・サーバーで使用されるシステム生成の名前でキューを作成します。
- f. oe.orders 表、oe.order_items 表、および pdb1.example.com PDB の hr スキーマに対する DML および DDL 変更を取得するように指示するルール・セットを備えた取得プロセスをシステム生成の名前で作成して起動します。
- g. oe.orders 表、oe.order_items 表および hr スキーマに対する DML および DDL 変更をクライアント・アプリケーションに送信することを指示するルール・セットを備えた xout という名前のアウトバウンド・サーバーを作成して起動します。
- h. 現在のユーザーをアウトバウンド・サーバーの接続ユーザーとして設定します。この例では、現在のユーザーは XStream 管理者です。クライアント・アプリケーションは、アウトバウンド・サーバーと相互作用するために接続ユーザーとしてデータベースに接続する必要があります。

ノート:



server_name 値は 30 バイトを超過できません。

ヒント:



pdb1.example.com データベースからアウトバウンド・サーバーに、データベースのすべての変更を取得して送信するには、table_names および schema_names パラメータに NULL (デフォルト) を指定します。

3. CDB のルートでアウトバウンド・サーバーに接続して LCR を受信するクライアント・アプリケーションを作成して実行します。
クライアント・アプリケーションを実行すると、アウトバウンド・サーバーが自動的に起動します。

関連トピック

- [Oracle Database XStream ガイド](#)
- [Oracle Database XStream ガイド](#)

ダウンストリーム取得を使用したCDBでのXStream Outの構成

ダウンストリーム取得を使用して、XStream Outコンポーネントをソース・データベース以外のデータベースに配置できます。

複数のCDBがある場合は、ソース・データベースを1つのCDBに配置し、ダウンストリーム取得を使用して別のCDBでの変更を取得できます。

前提条件

XStream Outを構成する前に、次の前提条件を満たす必要があります。

- CDBのすべてのコンテナが、XStream Out構成時に読取り/書込みモードでオープンしているようにします。
- この例では、ダウンストリーム取得を使用します。そのため、ソース・データベースからダウンストリーム・データベースへのログ・ファイル転送を構成する必要があります。
- リアルタイムのダウンストリーム取得を使用する場合は、必要なスタンバイREDOログも追加する必要があります。

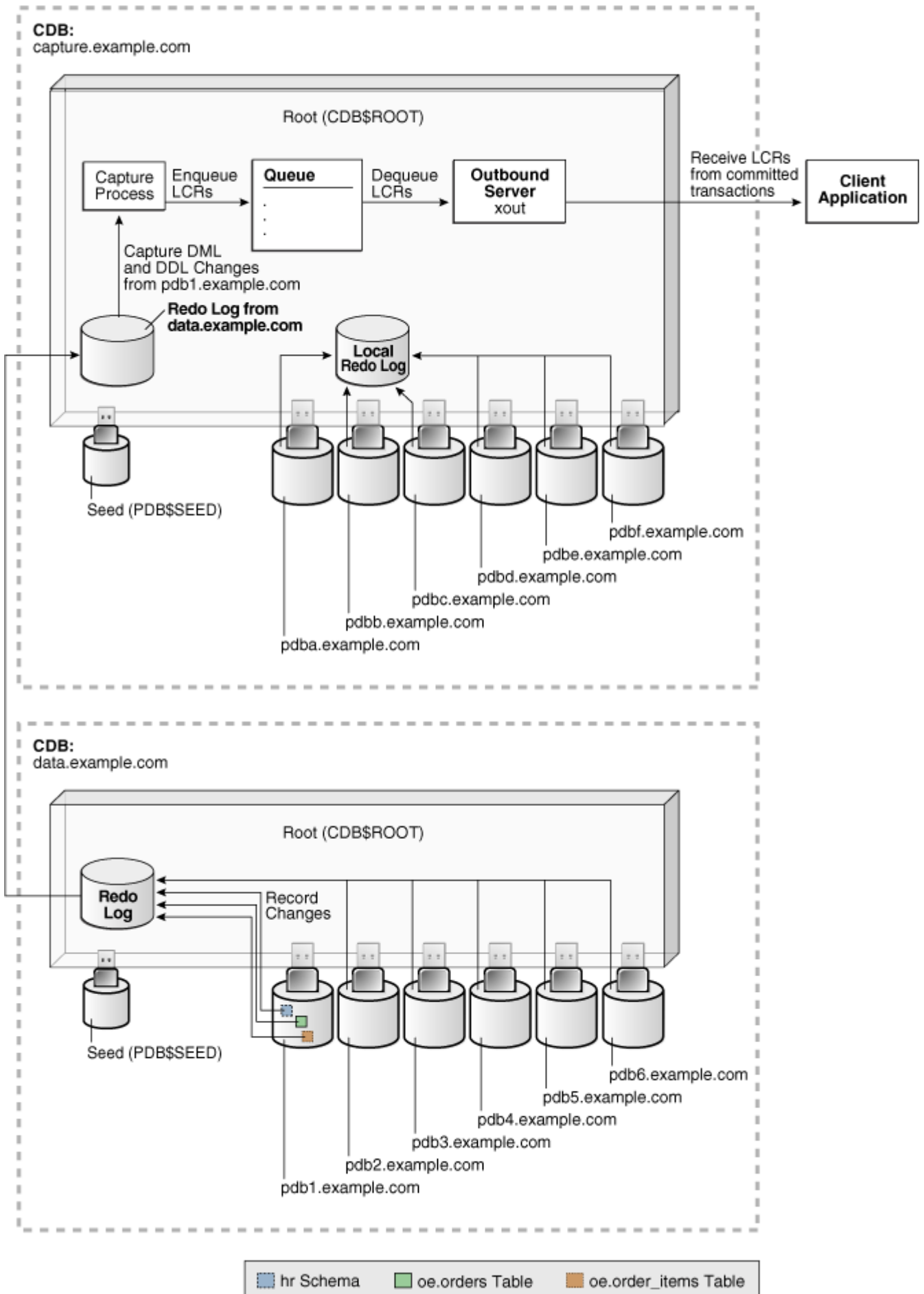
想定

この項では、次のことを想定しています。

- アウトバウンド・サーバーの名前はxoutです。
- アウトバウンド・サーバーで使用されるキューはc##xstrmadmin.xstream_queueです。
- ソース・データベースは、CDB data.example.comのPDB pdb1.example.comです。
- 取得プロセスは、CDB capture.example.comで実行されます。
- アウトバウンド・サーバーは、CDB capture.example.comで実行されます。
- PDB pdb1.example.comからのoe.orders表およびoe.order_items表に対するDML変更およびDDL変更は、アウトバウンド・サーバーに送信されます。
- PDB pdb1.example.comからのhrスキーマに対するDML変更およびDDL変更は、アウトバウンド・サーバーに送信されます。

次の図は、このXStream Out構成の概要を示しています。

図25-4 複数のCDBおよびダウンストリーム取得を使用したXStream Out構成の例



ダウンストリーム取得を使用してCDBでXStream Outを構成するには:

1. SQL*Plusで、ダウンストリーム取得CDBのルートにXStream管理者として接続します。

この例では、ダウンストリーム取得CDBはcapture.example.comです。

2. 取得プロセスで使用されるキューを作成します。

たとえば、次のプロシーダを実行します。

```
BEGIN
  DBMS_XSTREAM_ADM.SET_UP_QUEUE (
    queue_table => 'c##xstradmin.xstream_queue_table',
    queue_name => 'c##xstradmin.xstream_queue');
END;
/
```

3. 必要に応じて、ダウストリーム取得CDBのルートからソースCDBのルートに、データベース・リンクを作成します。

この例では、capture.example.comのルートからdata.example.comのルートに、データベース・リンクを作成します。たとえば、ユーザーc##xstradminが両方のデータベースでXStream管理者の場合は、次のようにデータベース・リンクを作成します。

```
CREATE DATABASE LINK data.example.com CONNECT TO c##xstradmin
  IDENTIFIED BY password USING 'data.example.com';
```

4. ソースCDB内のすべてのコンテナが読み取り/書き込みモードでオープンしていることを確認します。

5. ステップ3でデータベースを作成しなかった場合は、ソースCDBのルートで追加のステップを完了する必要があります。

ステップ3でデータベース・リンクを作成した場合、これらのステップは必要ありません。

BUILDプロシーダを実行し、ソースCDBのデータベース・オブジェクトに対して必要なサプリメンタル・ロギングが指定されていることを確認します。

a. ソースCDBのルートにXStream管理者として接続します。

b. DBMS_CAPTURE_ADM.BUILDプロシーダを実行します。次に例を示します。

```
SET SERVEROUTPUT ON
DECLARE
  scn NUMBER;
BEGIN
  DBMS_CAPTURE_ADM.BUILD (
    first_scn => scn);
  DBMS_OUTPUT.PUT_LINE('First SCN Value = ' || scn);
END;
/
First SCN Value = 409391
```

このプロシーダでは、capture.example.com CDBのルートで作成される取得プロセスの有効な先頭SCN値が表示されます。ステップ6で取得プロセスを作成するときに使用できるように、返されたSCN値をノートにとります。

c. ソースCDBのデータベース・オブジェクトに対して必要なサプリメンタル・ロギングが指定されていることを確認します。

この例では、hrスキーマ、oe.orders表、およびpdb1.example.com PDBのoe.order_items表に対してサプリメンタル・ロギングが構成されていることを確認してください。

6. ダウストリーム取得CDBのルートに接続されているときに、取得プロセスを作成します。

たとえば、capture.example.comにXStream管理者として接続されているときに、次のプロシーダを実行して取得プロセスを作成します。

```
BEGIN
```

```

DBMS_CAPTURE_ADM.CREATE_CAPTURE(
  queue_name      => 'c##xstrmadmin.xstream_queue',
  capture_name    => 'real_time_capture',
  rule_set_name   => NULL,
  start_scn       => NULL,
  source_database => NULL,
  use_database_link => TRUE,
  first_scn       => NULL,
  logfile_assignment => 'implicit',
  source_root_name => 'data.example.com',
  capture_class   => 'xstream');
END;
/

```

ステップ3でデータベースを作成しなかった場合は、DBMS_CAPTURE_ADM.BUILDプロシージャによってfirst_scnパラメータに対して返されるSCN値を指定します。

取得プロセスは起動しないでください。

7. 取得プロセスが作成された後、必要に応じて1つ以上のPDBのオープン・モードを変更します。
8. ADD_OUTBOUNDプロシージャを実行します。

この項の想定に従って、次のADD_OUTBOUNDプロシージャを実行します。

```

DECLARE
  tables DBMS_UTILITY.UNCL_ARRAY;
  schemas DBMS_UTILITY.UNCL_ARRAY;
BEGIN
  tables(1) := 'oe.orders';
  tables(2) := 'oe.order_items';
  schemas(1) := 'hr';
  DBMS_XSTREAM_ADM.ADD_OUTBOUND(
    server_name      => 'xout',
    queue_name       => 'c##xstrmadmin.xstream_queue',
    source_database  => 'pdb1.example.com',
    table_names      => tables,
    schema_names     => schemas,
    source_root_name => 'data.example.com',
    source_container_name => 'pdb1');
END;
/

```

このプロシージャを実行すると、次のアクションが実行されます。

- a. xoutという名前のアウトバウンド・サーバーを作成します。アウトバウンド・サーバーには、oe.orders表、oe.order_items表およびhrスキーマに対するDMLおよびDDL変更をクライアント・アプリケーションに送信することを指示するルール・セットがあります。ルールでは、これらの変更がCDB data.example.comのPDB pdb1.example.comで発生している必要があることを指定します。アウトバウンド・サーバーは、このキュー c##xstrmadmin.xstream_queueからLCRをデキューします。
- b. 現在のユーザーをアウトバウンド・サーバーのconnect_userとして設定します。この例では、current_userはXStream管理者です。クライアント・アプリケーションは、アウトバウンド・サーバーと相互作用するためにconnect_userとしてデータベースに接続する必要があります。



ノート:

server_name 値は 30 バイトを超過できません。

9. アウトバウンド・サーバーに接続してLCRを受信するクライアント・アプリケーションを作成して実行します。

クライアント・アプリケーションを実行すると、ダウンストリーム取得CDBでアウトバウンド・サーバーが自動的に起動します。

関連トピック

- [Oracle Database XStreamガイド](#)

親トピック: [CDBでのXStream Outの構成](#)

XStream Inとマルチテナント環境

マルチテナント環境では、移植可能な一連のスキーマ、オブジェクトおよび関連構造をOracleデータベースに含めることができ、アプリケーションには論理的に別のデータベースのように見えます。

この自己完結型コレクションは、プラグブル・データベース(PDB)と呼ばれます。1つのマルチテナント・コンテナ・データベース(CDB)には複数のPDBが格納されます。アプリケーション・コンテナも含めることができます。アプリケーション・コンテナは、アプリケーション・ルートとそれに関連付けられたすべてのアプリケーションPDBから構成されるCDBのオプション・コンポーネントです。アプリケーション・コンテナには、1つ以上のアプリケーションのデータが格納されています。アプリケーション・コンテナは、アプリケーションのメタデータおよび共通データを共有します。CDBでは、CDBルート、各PDB、各アプリケーション・ルートおよび各アプリケーションPDBのそれぞれがコンテナです。

CDBでは、インバウンド・サーバーは、単一のソース・データベースからLCRを受信して、現在のコンテナ(PDB、アプリケーション・ルート、アプリケーションPDBまたはCDBルート)でのみ変更を実行するように制限されています。単一のインバウンド・サーバーは、CDBの複数のコンテナに変更を適用できません。

インバウンド・サーバーがCDBルートにある場合、適用ユーザーは共通ユーザーである必要があります。インバウンド・サーバーがアプリケーション・ルートにある場合、適用ユーザーは共通ユーザーまたはアプリケーション共通ユーザーである必要があります。インバウンド・サーバーがPDBまたはアプリケーションPDBにある場合、適用ユーザーは共通ユーザーまたはローカル・ユーザーにできます。

ノート:



XStream では、アプリケーション・ルート・コンテナで行われた変更を同期しません。アプリケーション・ルート・コンテナで行われた操作のレプリケートに XStream In レプリケーションを使用しないでください。ターゲットのアプリケーション・ルート・コンテナでのこれらの変更は手動で適用できます。PDB で行われる操作はレプリケートできることに注意してください。

関連トピック

- [システム生成ルールとマルチテナント環境](#)
- [マルチテナント・アーキテクチャ](#)

親トピック: [CDBでのXStreamの使用](#)

用語集

アプリケーション

アプリケーション・ルート内で、アプリケーションは共通ユーザーによって作成されるデータおよびメタデータの名前付けおよびバージョンングされたセットです。アプリケーションには、アプリケーション共通ユーザー、アプリケーション共通オブジェクト、あるいはそれらの複数の組合せが含まれる場合があります。

親トピック: [用語集](#)

アプリケーション共通オブジェクト

[アプリケーション・ルート](#)への接続中に作成される共有データベース・オブジェクト。メタデータ(メタデータリンク・オブジェクト対象)またはデータ([データリンク共通オブジェクト](#)対象)が、[アプリケーション・コンテナ](#)でアプリケーションPDBによって共有されます。

親トピック: [用語集](#)

アプリケーション共通ユーザー

[アプリケーション・ルート](#)への接続中に作成される[共通ユーザー](#)。メタデータ([メタデータリンク共通オブジェクト](#)対象)またはデータ([データリンク共通オブジェクト](#)対象)が、アプリケーション・コンテナでアプリケーションPDBによって共有されます。

親トピック: [用語集](#)

アプリケーション・コンテナ

アプリケーション・ルートに接続されたアプリケーションPDBの名付けられたセット。アプリケーション・コンテナにはアプリケーション・シードが含まれる場合があります。

親トピック: [用語集](#)

アプリケーション・パッチ

[アプリケーション・コンテナ](#)内の、[アプリケーション](#)に対する小規模な変更。パッチ適用の一般的な例にはバグ修正およびセキュリティ・パッチが含まれます。アプリケーションのアップグレードはALTER PLUGGABLE DATABASE APPLICATION文で開始および終了します。

親トピック: [用語集](#)

アプリケーションPDB

[アプリケーション・コンテナ](#)に接続されたPDB。

親トピック: [用語集](#)

アプリケーション・ルート

アプリケーション・コンテナ内のルート・コンテナ。各[アプリケーション・コンテナ](#)にはアプリケーション・ルートが1つずつあります。アプリケーション・ルートは、共通オブジェクトが含まれる可能性があるため[CDBルート](#)と一部の特性を共有し、CREATE PLUGGABLE DATABASE文で作成されるためPDBと一部の特性を共有します。

親トピック: [用語集](#)

アプリケーション・シード

[アプリケーション・コンテナ](#)内で他のPDBを作成するためのテンプレートとして機能する、オプションの[アプリケーションPDB](#)。アプリケーション・コンテナには0または1のアプリケーション・シードが含まれます。

親トピック: [用語集](#)

アプリケーション・アップグレード

[アプリケーション・コンテナ](#)内の、[アプリケーション](#)の物理アーキテクチャに対する大規模な変更。アプリケーションのアップグレードはALTER PLUGGABLE DATABASE APPLICATION文で開始および終了します。

親トピック: [用語集](#)

CDB

最低1つの[PDB](#)を含むOracle Databaseインストール。PDBは、Oracle Netクライアントに従来のOracleデータベースとして表示されます。すべてのOracleデータベースは、CDBか[非CDB](#)のいずれかです。

親トピック: [用語集](#)

CDB管理者

CDBを管理するデータベース管理者。[PDB管理者](#)は、CDB内の個々のPDBを管理します。

親トピック: [用語集](#)

CDBフリート

1つの論理CDBとして管理できる様々なCDBの集合。

親トピック: [用語集](#)

CDBリストア・ポイント

CDBで、ルートへの接続時、およびFOR PLUGGABLE DATABASE句が指定されていないときに作成されるリストア・ポイント。PDBリストア・ポイントと異なり、CDBリストア・ポイントはすべてのPDBによって使用可能です。

親トピック: [用語集](#)

CDBルート

[マルチテナント・コンテナ・データベース\(CDB\)](#)において、すべてのPDBが属するスキーマ、スキーマ・オブジェクトおよび非スキーマ・オブジェクトの集合。各CDBには、PDBの管理に必要なシステム・メタデータが格納されるルート・[コンテナ](#)が1つずつあります。すべてのPDBはCDBルートに属します。

親トピック: [用語集](#)

クリーン・リストア・ポイント

PDBが閉じられるときに作成されるPDBリストア・ポイント。クリーン・リストア・ポイントに対するフラッシュバックPDBでは、バックアップのリストアまたは一時インスタンスの作成が必要ありません。

親トピック: [用語集](#)

共通オブジェクト

データ([データリンク共通オブジェクト](#))またはメタデータ([メタデータリンク共通オブジェクト](#))のいずれかを共有する、[CDBルート](#)または[アプリケーション・ルート](#)のいずれかに存在するオブジェクト。CDBルート内の共通オブジェクトはすべてOracle提供です。アプリケーション・ルート内の共通オブジェクトは[アプリケーション共通オブジェクト](#)と呼ばれています。

親トピック: [用語集](#)

共通ユーザー

[マルチテナント・コンテナ・データベース\(CDB\)](#)内で、複数のコンテナに同じIDで存在するデータベース・ユーザー。[CDBルート](#)で作成される共通ユーザーは、既存と将来のすべてのPDBにおいて同じIDを持ちます。[アプリケーション・コンテナ](#)で作成される共通ユーザーは、この[アプリケーション・コンテナ](#)の既存と将来のすべての[アプリケーションPDB](#)において同じIDを持ちます。

親トピック: [用語集](#)

コンテナ

[マルチテナント・コンテナ・データベース\(CDB\)](#)で、ルートまたはPDBのいずれか。

親トピック: [用語集](#)

コンテナ・データ・オブジェクト

CDBにおいて、複数のコンテナまたはCDB全体に関連するデータを含み、このようなオブジェクトを介して特定の共通ユーザーに表示されるデータを1つ以上のコンテナに制限するためのメカニズムを備えた表またはビュー。名前がV\$およびCDB_で始まるOracle提供ビューは、コンテナ・データベース・オブジェクトの例です。

親トピック: [用語集](#)

コンテナ間操作

CDBにおいて、CDB自体、複数のコンテナ、複数の共通ユーザーまたはロール、またはユーザーが接続されているコンテナ以外のコンテナに影響を与えるDDL文。ルートに接続されている共通ユーザーのみが、コンテナ間操作を実行できます。

親トピック: [用語集](#)

データ・リンク

[PDB](#)において、ルート内のデータ(メタデータではない)を指し示す内部メカニズム。たとえば、AWRデータはルートに常駐します。各PDBがオブジェクト・リンクを使用してルート内のAWRデータを指し示すことにより、DBA_HIST_ACTIVE_SESS_HISTORYやDBA_HIST_BASELINEなどのビューが個別のコンテナそれぞれでアクセス可能になります。

親トピック: [用語集](#)

データベース統合

1つ以上の非CDBから[マルチテナント・コンテナ・データベース\(CDB\)](#)にデータを移動する一般プロセス。

親トピック: [用語集](#)

データリンク共通オブジェクト

[CDBルート](#)または[アプリケーション・ルート](#)のいずれかに存在する[共通オブジェクト](#)。メタデータではなくデータは、共通オブジェクトを示す[データ・リンク](#)が含まれるPDBによって共有されます。

親トピック: [用語集](#)

拡張データリンク共通オブジェクト

[データリンク共通オブジェクト](#)と[メタデータリンク共通オブジェクト](#)を合成したもの。拡張データリンク・オブジェクトの場合、各[アプリケーションPDB](#)は、[アプリケーション・ルート](#)の共通データを共有しながら独自のPDB固有データを作成できます。

親トピック: [用語集](#)

高速アプリケーション通知(FAN)

アプリケーションでは、FANを使用して、迅速な障害の検出、障害発生後の接続プールの分散の均等化、および障害が発生したコンポーネントの修復時の接続プールの分散の再均等化を行うことができます。FAN通知プロセスでは、クラスタ・サーバーが使用不可になるか、またはネットワーク・インタフェースに障害が発生した場合にOracle Databaseが発行するシステム・イベントが使用されます。

親トピック: [用語集](#)

ホット・クローニング

ソースPDBが読み取り/書き込みモードで開かれている状態でのPDBのクローニング。

親トピック: [用語集](#)

リードCDB

[CDBフリート](#)で、いくつかのCDBを監視および管理するための中心的な場所。

親トピック: [用語集](#)

ローカルUNDOモード

[CDB](#)内の各[PDB](#)に対するUNDOデータ・ファイルの個別セットの使用。

親トピック: [用語集](#)

ローカル・ユーザー

[マルチテナント・コンテナ・データベース\(CDB\)](#)で、[共通ユーザー](#)でないすべてのユーザー。

親トピック: [用語集](#)

メタデータ・リンク

[PDB](#)において、ルートに格納されているディクショナリ・オブジェクト定義を指し示す内部メカニズム。たとえば、各PDB内のOBJ\$表はメタデータ・リンクを使用して、ルートに格納されているOBJ\$の定義を指し示します。

親トピック: [用語集](#)

メタデータリンク共通オブジェクト

[CDBルート](#)または[アプリケーション・ルート](#)のいずれかに存在する[共通オブジェクト](#)。データではなくメタデータは、共通オブジェクトを示す[メタデータ・リンク](#)が含まれるPDBによって共有されます。

親トピック: [用語集](#)

マルチテナント・アーキテクチャ

Oracle Databaseをマルチテナント・コンテナ・データベース([CDB](#))として動作させることを可能にする(つまり複数のPDBを含めることができる)アーキテクチャ。[PDB](#)は、Oracle Netクライアントに従来のOracleデータベース([非CDB](#))として表示されるスキーマ、スキーマ・オブジェクトおよび非スキーマ・オブジェクトのポータブル・コレクションです。

親トピック: [用語集](#)

マルチテナント・コンテナ・データベース(CDB)

[\[CDB\]](#)を参照してください。

親トピック: [用語集](#)

非CDB

[マルチテナント・コンテナ・データベース\(CDB\)](#)ではないOracle Database。Oracle Database 12cまでのすべてのデータベースは非CDBでした。Oracle Database 12cから、すべてのデータベースはCDBまたは非CDBである必要があります。

親トピック: [用語集](#)

Oracle Multitenant

1つのCDB内に複数のPDBを作成できるようにするデータベース・オプション。

親トピック: [用語集](#)

PDB

[マルチテナント・コンテナ・データベース\(CDB\)](#)内で、Oracle Netクライアントに従来のOracleデータベース([非CDB](#))として表示されるスキーマ、スキーマ・オブジェクトおよび非スキーマ・オブジェクトのポータブル・コレクション。

親トピック: [用語集](#)

PDB管理者

1つ以上のPDBを管理するデータベース管理者。[CDB管理者](#)は、CDB全体を管理します。

親トピック: [用語集](#)

PDBアーカイブ・ファイル

PDBデータ・ファイルおよびXMLメタデータ・ファイルの両方が含まれる、圧縮されたファイル。アーカイブ・ファイルを指定することでPDBを作成し、その結果、XMLファイルおよびデータ・ファイルを別々にコピーすることを回避できます。

親トピック: [用語集](#)

PDBロックダウン・プロファイル

指定されたPDBに接続中のローカル・ユーザーが使用可能な操作を制限するためのセキュリティ・メカニズム。一般的な使用方は、権限付与の影響を制限することです。たとえば、ALTER SYSTEMの権限付与をPLSQLで始まる名前のオプションのみに制限します。

親トピック: [用語集](#)

PDBパフォーマンス・プロファイル

PDBまたはPDBのセットのために指定された共有のシステム・リソース、CPU、パラレル実行サーバーおよびメモリー。

親トピック: [用語集](#)

PDBリストア・ポイント

CDB内で、特定のPDBについてのみ使用可能なリストア・ポイント。対照的に、CDBリストア・ポイントはすべてのPDBによって使用可能です。

親トピック: [用語集](#)

PDBスナップショット

ALTER PLUGGABLE DATABASE SNAPSHOTコマンドを使用して作成されたPDBの、名前付きのPoint-in-Timeコピー。ファイル・レベルでは、PDBスナップショットは、PDBコピーの内容を含むアーカイブ・ファイルです。

基礎となるファイル・システムでスパース・ファイルがサポートされている場合、最初のスナップショットはフルであり、以降のすべてのスナップショットはスパースです。

親トピック: [用語集](#)

PDBの同期化

ユーザーが開始し、[アプリケーションPDB](#)内の[アプリケーション](#)を[アプリケーション・ルート](#)の最新バージョンおよびパッチに更新する操作。

親トピック: [用語集](#)

PDB/非CDB互換性保障

マルチテナント・アーキテクチャにおいて、Oracle Netで接続しているクライアントから、PDBと非CDBの動作が同じように見えるようにすること。

親トピック: [用語集](#)

プラグブル・データベース(PDB)

[「PDB」](#)を参照してください。

親トピック: [用語集](#)

プロキシPDB

データベース・リンクを使用してリモートCDB内のPDBを参照するPDB。リモートPDBは[参照先PDB](#)と呼ばれます。

親トピック: [用語集](#)

参照先PDB

[プロキシPDB](#)によって参照されるPDB。ローカルPDBはその参照先PDBと同じCDB内にありますが、リモートPDBは異なるCDB内にあります。

親トピック: [用語集](#)

リフレッシュ可能なクローンPDB

ソースPDBと定期的に同期できる読取り専用クローン。REFRESH MODE句の値に応じて、同期は自動的に、または手動で行われます。

親トピック: [用語集](#)

リソース・プラン

リソース・コンシューマ・グループへのリソースの割当て方法を指定するリソース・プラン・ディレクティブのコンテナ。

親トピック: [用語集](#)

リソース・プラン・ディレクティブ

コンシューマ・グループ内のセッションのCPU、物理I/Oまたは論理I/O使用量に対する制限と制御のセット。

親トピック: [用語集](#)

シードPDB

[マルチテナント・コンテナ・データベース\(CDB\)](#)内の、デフォルトの[プラグブル・データベース\(PDB\)](#)で、システムがユーザー作成のPDB用のテンプレートとして使用します。PDBシードは、システム提供のPDB\$SEEDまたは[アプリケーション・シード](#)のいずれかです。

親トピック: [用語集](#)

共有UNDOモード

単一インスタンスCDBでは、アクティブなUNDO表領域が1つのみ存在します。Oracle RAC CDBの場合、それぞれのインスタンスについてアクティブなUNDO表領域が1つ存在します。

親トピック: [用語集](#)

スナップショット・コピーPDB

CREATE PLUGGABLE DATABASE ... FROM ... SNAPSHOT COPYコマンドを実行することで作成されるPDB。記憶域管理スナップショットは、特定のファイル・システムでのみサポートされている、基礎となる記憶域のコピーです。

ノート:



スナップショット・コピーPDBの作成に使用される記憶域管理スナップショットは、CREATE PLUGGABLE DATABASE ... USING SNAPSHOT コマンドで指定できるPDB管理スナップショットとは異なります。記憶域管理スナップショットは、PDBスナップショットからのクローンには含まれません。

親トピック: [用語集](#)

分割ミラーによるクローンPDB

Oracle ASMでミラーを分割することによって作成されたPDB。

親トピック: [用語集](#)

ここに用語を入力してください。

ここに説明を入力してください。

親トピック: [用語集](#)

システム・コンテナ

CDBルートおよびCDB内のすべてのPDBが含まれるコンテナ。

親トピック: [用語集](#)

切断されたPDB

自己完結型のPDBデータ・ファイルのセットと、PDBファイルの場所を指定するXMLメタデータ・ファイル。

親トピック: [用語集](#)

索引

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [H](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [U](#) [V](#) [X](#)

A

- ADD_OUTBOUNDプロシージャ [25.4.2](#)
- 管理ユーザー
 - パスワード・ファイル、マルチテナント環境 [13.2.1.4](#)
- ALTER DATABASE文 [13.1.4](#)
 - アプリケーション・ルート [2.3.1.2](#), [17.2](#)
 - CDB [13.4.1](#)
 - RECOVER句 [2.1.5](#)
- ALTER PLUGGABLE DATABASE文 [13.4.1](#), [15.4.2](#), [17.2](#)
 - DROP SNAPSHOT句 [16.5](#)
 - MATERIALIZE句 [7.7.3](#)
 - SET MAX_PDB_SNAPSHOTS句 [16.1.2.2](#), [16.2](#)
 - SNAPSHOT句 [16.3](#), [16.4](#)
 - SNAPSHOT COPY句 [7.7.3](#)
 - UNPLUG INTO句 [11.1](#), [12.1.2](#), [12.2.2](#)
- ALTER SESSION文
 - SET CONTAINER句 [13.2.2.2](#)
- ALTER SYSTEM文
 - CDB [13.3.2](#)
 - CONTAINER句 [13.3.2](#)
 - PDB [15.3](#)
- アプリケーション共通オブジェクト [2.2.5](#), [2.3.2](#), [17.4](#), [17.4.1](#)
 - CONTAINERS句 [17.5.1](#)
 - 作成 [2.3.2.1](#), [17.4.1.1](#), [17.4.3](#)
 - データリンク [2.1.2.5](#), [2.3.2.3](#)
 - DDL文 [17.4.5](#)
 - DML文 [17.4.4](#), [17.5.1](#)
 - 拡張データリンク・オブジェクト [2.3.2.4](#)
 - メタデータリンク共通オブジェクト [2.3.2.2](#)
 - メタデータ・リンク [2.3.2.2.1](#)
 - ネーミング・ルール [2.2.1.2](#)
 - 制限 [17.4.2](#)
- アプリケーション・コンテナ
 - 概要 [2.3.1](#)
 - 管理 [17](#), [17.1](#)
 - アプリケーション共通オブジェクト [2.2.1.2](#), [2.2.5](#), [2.3.2](#), [2.3.2.1](#), [17.4](#), [17.4.3](#), [17.5.1](#)
 - アプリケーション・コンテキスト [18.5.2](#)
 - アプリケーションPDB [2.3.1.3](#)
 - アプリケーション・ルート [2.3.1.2](#), [17.3.5.2](#)

- アプリケーション [2.3.3.2](#), [2.3.3.3](#)
- 暗黙的に作成されるアプリケーション [2.3.5](#)
- アプリケーション・シード [2.3.1.4](#)
 - 作成 [12.2.1](#), [12.2.1.3](#)
 - 準備 [12.2.1.2](#)
- アプリケーションの同期化 [2.3.6](#)
- アプリケーションのバージョン [2.3.3.3.2](#)
- 一括挿入 [17.3.9](#)
- 互換バージョン [17.3.8](#)
- コンテナ・マップ [2.3.7](#), [17.6.1](#)
 - 作成 [17.6.2](#)
- 作成 [12.1.1](#), [12.1.1.3](#)
- DML文 [17.5](#), [17.5.1](#)
- 削除 [12.1.3](#)
- アプリケーションのアップグレードの動作 [2.3.3.3.1](#)
- アプリケーションのインストール [2.3.3](#), [2.3.3.1](#), [2.3.3.2](#), [17.3.2.1](#)
- アプリケーションの管理 [17.3.1](#)
- アプリケーションの移行 [2.3.4](#)
- アプリケーションの移行 [17.3.5](#)
- ネーミング・ルール [2.1.2.4](#)
- アプリケーションのパッチ適用 [2.3.3.4](#), [17.3.4.1](#), [17.3.4.2](#)
- 準備 [12.1.1.2](#)
- 目的 [2.3.1.1](#), [2.3.1.1.1](#), [2.3.1.1.2](#), [2.3.1.1.3](#)
- SQL*Loader [17.3.9](#)
- アプリケーションの同期 [17.3.6](#)
- プロキシPDBとの同期 [17.3.7](#)
- Transport Layer Security [18.7](#)
- アプリケーションのアンインストール [17.3.10.1](#), [17.3.10.2](#)
- 切断 [12.1.2](#), [12.1.2.1](#)
- アプリケーションのアップグレード [2.3.3](#), [2.3.3.3](#), [17.3.3.1](#)
- 拡張データ・リンク・オブジェクトの表示 [19.11.8](#)
- 情報の表示 [19.11.1](#)
- パッチの表示 [19.11.4](#), [19.11.5](#)
- 共有オブジェクトの表示 [19.11.7](#)
- SQL文の表示 [19.11.3](#)
- ステータスの表示 [19.11.2](#)
- 同期エラーの表示 [19.11.6](#)
- ビュー [19.11](#)
 - 仮想プライベート・データベース・ポリシー [18.6.2](#)
- アプリケーション・コンテキスト
 - 概要 [18.5.1](#)
 - アプリケーション・コンテナ [18.5.2](#)
 - CDB [18.5](#)
- アプリケーションPDB [2.3.1.3](#)
 - アプリケーションの同期化 [2.3.6](#)

- クローニング [7.1.1](#)
- 作成 [12.3](#), [17.3.5.3](#)
- ネーミング・ルール [2.1.2.4](#)
- 同期 [2.3.7](#)
- アプリケーション・ルート [2.3.1.2](#)
 - ALTER DATABASE文 [17.2](#)
 - ALTER PLUGGABLE DATABASE文 [17.2](#)
 - アプリケーションPDB
 - 変更 [17.2](#)
 - 作成 [17.3.5.2](#)
 - 変更 [17.2](#)
- アプリケーション
 - CDB内
 - メタデータリンク共通オブジェクト [2.3.2.2](#)
 - アプリケーション・コンテナ内 [2.3.3](#), [2.3.3.2](#), [2.3.3.3](#), [2.3.3.3.1](#)
 - 異なるバージョン [2.3.3.3.2](#)
 - 暗黙的に作成 [2.3.5](#)
 - アプリケーションの移行 [2.3.4](#)
 - パッチ適用 [2.3.3.4](#), [17.3.4.2](#)
 - 同期化 [2.3.6](#)
 - アンインストール [17.3.10.2](#)
- アプリケーション・シード [1.3.2.1.1](#), [2.1.2.1](#), [2.3.1.4](#)
 - 作成 [12.2.1](#), [12.2.1.3](#)
 - 削除 [12.2.3](#)
 - 準備 [12.2.1.2](#)
 - 切断 [12.2.2](#)
- 監査
 - 監査構成 [2.2.6](#)
 - 監査ポリシー [2.2.6](#)
 - CDB [18.9](#), [18.9.1](#)
 - 共通オブジェクト [2.2.6](#)
 - 従来 [18.9.3.2](#)
- 認証
 - オペレーティング・システム認証 [18.4.1](#)
 - PDBのオペレーティング・システム・ユーザー [18.4.1](#)
 - PDB [18.4.1](#)
- AVAILABILITY MAX句 [8.3.2.2](#)
- AVAILABILITY NORMAL句 [8.3.2.1](#)

B

- バックアップおよびリカバリ
 - CDBおよびPDB [2.6.1](#), [20](#)
- break-glassプロトコル [24.9.1](#)

C

- [catcon.pl](#) [13.5.3](#)
- [CDB_PDB_HISTORY](#)ビュー [19.10](#)
- [CDB_PDBS](#)ビュー [19.4](#)
- [CDBリソース・プラン](#) [22.2.1](#)
 - [PDBパフォーマンス・プロファイル](#) [22.2.4](#), [22.2.7.3](#)
- [CDB](#) [3.1](#), [IV](#), [13.1.8](#), [15.6](#)
 - [概要](#) [1.1](#)
 - [管理](#) [13](#)
 - [ALTER DATABASE](#)文 [13.4.1](#)
 - [ALTER PLUGGABLE DATABASE](#)文 [13.4.1](#)
 - [ALTER SYSTEM](#)文 [13.3.2](#)
 - [アプリケーション共通オブジェクト](#) [2.2.1.2](#), [2.3.2](#), [2.3.2.1](#), [17.4](#), [17.4.1.1](#), [17.5.1](#)
 - [問合せ](#) [19.7.3](#)
 - [アプリケーション・コンテナ](#) [2.3.1](#), [2.3.1.1](#), [2.3.1.1.1](#), [2.3.1.1.2](#), [2.3.1.1.3](#)
 - [アプリケーション共通オブジェクト](#) [2.1.2.5](#), [17.4.1.1](#)
 - [アプリケーション・アップグレード](#) [2.3.3.3.1](#)
 - [一括挿入](#) [17.3.9](#)
 - [互換バージョン](#) [17.3.8](#)
 - [作成](#) [12.1.1](#), [12.1.1.3](#)
 - [DML](#)文 [17.5](#)
 - [削除](#) [12.1.3](#)
 - [アプリケーションのインストール](#) [2.3.3.1](#), [2.3.3.2](#), [17.3.2.1](#)
 - [アプリケーションの移行](#) [17.3.5](#)
 - [アプリケーションのパッチ適用](#) [17.3.4.1](#)
 - [準備](#) [12.1.1.2](#)
 - [アプリケーションの同期](#) [17.3.6](#)
 - [アプリケーションのアンインストール](#) [17.3.10.1](#), [17.3.10.2](#)
 - [切断](#) [12.1.2](#)
 - [アプリケーションのアップグレード](#) [2.3.3.3](#), [17.3.3.1](#)
 - [アプリケーション・コンテキスト](#) [18.5](#)
 - [アプリケーションPDB](#) [2.3.1.3](#)
 - [クローニング](#) [7.1.1](#)
 - [作成](#) [12.3](#), [17.3.5.3](#)
 - [アプリケーション・シード](#) [2.3.1.4](#)
 - [作成](#) [12.2.1](#)
 - [削除](#) [12.2.3](#)
 - [準備](#) [12.2.1.2](#)
 - [切断](#) [12.2.2](#)
 - [監査](#) [18.9](#)
 - [影響](#) [18.9.1](#)
 - [従来](#) [18.9.3.2](#)
 - [バックアップおよびリカバリ](#) [2.6.1](#), [20](#)

- CDBフリート [14](#), [14.2](#)
 - CDBメンバー [14.4](#)
 - リードCDB [14.1](#), [14.3](#)
- CDBリソース・プラン
 - 情報の表示 [22.2.9](#)
- 文字セット [2.1.1](#)
- 共通オブジェクト [2.1.2.5](#), [2.2.5](#)
- 共通権限付与 [2.2.4.3](#), [2.2.4.5](#), [18.1.2](#)
- 共通ロール [2.2.3](#), [2.2.3.1](#)
- 共通ユーザー [2.2](#), [2.2.2](#), [2.2.2.1](#), [2.2.4.3](#), [2.2.4.5](#)
 - 定義 [3.1.1](#)
 - ネーミング・ルール [2.2.1.2](#)
- 互換性違反 [15.4.5.1](#)
- 接続 [13.2.1](#)
 - ALTER SESSION文 [13.2.2.2](#)
 - CONNECTコマンド [13.2.2.1](#)
- コンテナ・データ・オブジェクト [2.1.3.3](#), [19.1.2](#)
 - 問合せ [19.6](#)
- コンテナ・マップ [2.3.7](#), [17.6](#)
- コンテナ [1.1.1](#), [13.5.1.1](#), [19.3](#)
- CONTAINERS句 [19.7.2](#)
- 作成 [4](#), [4.3.1](#)
- 作成 [1.3.1](#)
- コンテナ間操作 [2.1.5](#)
- 現在のコンテナ [2.1.4](#), [13.1.1](#), [19.8](#)
- データベース・リソース・マネージャ [22](#)
- Database Vault操作の制御 [24.9.1](#)
- データ定義言語(DDL) [13.5.2](#)
- データ・ディクショナリ [2.1.3](#)
- データ・リンク [2.1.3.2](#)
- データ・リダクションのマスキング・ポリシー [18.8](#)
- DBMS_SQLパッケージ [13.5.4](#)
- DML文 [13.5.1](#), [17.5.1](#)
- EM Express [4.4](#)
- ENABLE PLUGGABLE DATABASE句 [4.3.2.1](#)
- PL/SQLコードの実行 [13.5.4](#)
- ファイル [2.5](#)
- フラッシュバック [20.5](#)
- PDBのフラッシュバック [2.6.2](#)
- Oracle Database Vaultでの機能 [24.2](#)
- 共通のロールおよび権限の付与 [2.2.4.3.1](#)
- 権限とロールの付与 [2.2.4.2.1](#), [18.1.5](#)
- 初期化パラメータ [19.9](#)
- ローカル権限付与 [18.1.2](#)
- ローカル・ロール [2.2.3](#), [2.2.3.2](#), [2.2.4.2](#)

- ローカル・ユーザー [2.2.2](#), [2.2.2.2](#)
 - 定義 [3.1.1](#)
- メタデータ・リンク [2.1.3.2](#)
- 変更 [13.3.2](#), [13.4.1](#), [13.4.2](#)
- 監視 [19](#)
- オブジェクト権限 [18.1.4](#)
- Oracle Database Vault [13.1.2](#), [24](#)
- Oracle Data Pump [1.3.2.2.2](#)
- Oracle Managed Files [4.3.2.2.2](#)
- PDB_FILE_NAME_CONVERT初期化パラメータ [4.3.2.2.3](#)
- PDBロックダウン・プロファイル [2.2.7](#), [13.1.7.1](#), [18.3.1](#)
- PDB
 - 変更 [13.4.1](#)
 - リフレッシュ [15.4.4.1](#)
- PDBスナップショット [16](#)
 - 自動作成の構成 [16.3](#)
 - 手動作成 [7.6.1](#), [16.4](#)
 - 削除 [16.5](#)
 - 最大数の設定 [16.1.2.2](#), [16.2](#)
- 作成計画 [4.2](#)
- PDBの接続
 - 方法 [5.1](#)
 - 準備 [5.5](#)
- Point-in-Timeリカバリ [20.4](#)
- 前提条件 [3.2](#)
- 付与の原則 [2.2.4.1](#)
- 権限管理 [18.1](#)
- レルム [24.5.1.2](#)
- リカバリ [20](#)
- リソース管理 [2.7](#)
- 権限の取消し [18.1.5](#)
- ロール
 - 共通の作成 [18.2.6](#)
 - ローカルの作成 [18.2.8](#)
 - 共通の付与 [2.2.4.3](#), [2.2.4.5](#), [18.2.9](#)
 - 共通ロールの使用方法 [18.2.2](#)
 - 管理 [18.2](#)
 - 管理に必要な権限 [18.2.4](#)
 - 共通の作成規則 [18.2.5](#)
- ルート・コンテナ [2.1.1](#)
 - 変更 [13.4.4](#)
- ルール・セット [24.6](#)
- セキュリティ [18](#)
- SEED FILE_NAME_CONVERT句 [4.3.2.2.1](#)
- シードPDB [1.3.2.1.1](#), [2.1.2.1](#)

- サービス [2.4](#)
- 停止 [13.6](#)
- スナップショット・コピーPDB [7.7.1.1](#)
 - マテリアライズ [7.7.3](#)
- SQLスクリプト [13.5.3](#)
- スタンバイ・データベース [13.1.2](#)
- システム・コンテナ [2.1.1](#)
- システム権限 [18.1.3](#)
- タスク [3.3.1](#)
- 一時ファイル [2.5](#)
- ツール [3.3.2](#)
- 透過的データ暗号化 [13.1.2](#)
- UNDOモード [2.5](#), [4.3.2.4](#), [13.4.3](#)
- PDBの切断 [11.1](#)
- ユーザー権限、影響 [18.1.1](#)
- データ・ポンプを使用したデータの移動 [21.1.2](#)
- 情報の表示 [18.1.7.1](#), [19](#)
- ビュー [19.1.3](#)
- 仮想プライベート・データベース [18.6](#)
 - ポリシー [18.6.2](#)
- XStream [25](#)
- XStream In [25.5](#)
- XStream Out [25.3](#)
 - 構成 [25.4.1](#)
 - 複数の構成 [25.4.2](#)
- CDBS
 - インフラストラクチャDBAによるPDBアクセス [24.9.1](#)
- CLONEDBパラメータ [7.1](#)
- アプリケーションPDBのクローニング [7.1.1](#)
- 非CDBのクローニング [7.4.1](#)
- PDBのクローニング [7.1](#)
 - ローカル [7.2.1](#), [7.2.2](#)
 - リフレッシュ可能なクローンPDB [7.5](#), [15.4.4.2](#)
 - リモート [7.3.1](#), [7.3.2](#)
 - 分割ミラーの使用 [7.8](#)
- コマンド・ルール
 - PDB [24.7](#)
- COMMON_USER_PREFIXパラメータ [13.5.2.1](#)
- 共通性、原則 [2.2.1.1](#)
- 共通オブジェクト [2.2.5](#)
- 共通権限付与 [2.2.4.3](#), [2.2.4.5](#)
 - 概要 [18.1.2](#)
 - 付与 [18.1.5](#)
 - 取消し [18.1.5](#)
 - オブジェクト権限の使用 [18.1.4](#)

- システム権限の使用 [18.1.3](#)
- 共通ロール [2.2.3](#), [2.2.3.1](#)
 - 概要 [18.2.1](#)
 - 作成 [18.2.6](#)
 - 付与 [2.2.4.3](#), [2.2.4.5](#), [18.2.9](#)
 - 動作の概要 [18.2.2](#)
 - 管理に必要な権限 [18.2.4](#)
 - 作成のルール [18.2.5](#)
- 共通ユーザー・アカウント [1.1.1](#), [2.2](#), [2.2.2](#)
 - 定義 [3.1.1](#)
 - 他のPDBへのアクセスの有効化 [18.1.7](#)
 - 権限の付与先 [2.2.4.3](#), [2.2.4.5](#), [18.1](#)
 - ネーミング・ルール [2.2.1.2](#)
- 共通ユーザー [2.2.2.1](#)
 - PDBのデータへのアクセス [18.1.7.2](#)
 - 接頭辞 [13.5.2.1](#)
- CONFIGURE_DVプロシージャ
 - Database Vaultの登録 [24.4.2](#), [24.4.3](#)
- CONNECTコマンド, SQL*Plus
 - CDB [13.2.2.1](#)
- CONTAINER_DATAオブジェクト
 - 情報の表示 [18.1.7](#)
- コンテナ・データベース
 - 「CDB」を参照
- コンテナ・データ・オブジェクト [2.1.3.3](#), [19.1.2](#)
 - 概要 [18.1.7.1](#)
 - 定義 [19.1.2](#)
 - 問合せ [19.6](#)
- コンテナ・マップ [2.3.7](#), [17.6](#), [17.6.1](#)
 - 作成 [17.6.2](#)
- CONTAINERS_DEFAULT_TARGETプロパティ [2.1.5](#)
- CONTAINERS_DEFAULT属性 [17.4.1](#)
- CONTAINERS_PARALLEL_DEGREEパラメータ [19.7.2](#), [19.7.3](#)
- コンテナ、CDB [1.1.1](#), [2.1](#), [2.6.2](#)
 - ルート [2.1.1](#)
- CONTAINERS句 [17.4.1](#), [17.5.1](#), [19.7.2](#), [19.7.3](#)
- CPU_COUNT初期化パラメータ [22.1.4.1](#)
- CREATE_FILE_DEST句 [5.2.4.2](#), [7.2.4.3](#), [13.1.7.3](#)
- CREATE_SIMPLE_PLANプロシージャ
 - データベース・リソース・マネージャ [22.3.2](#)
- CREATE DATABASE文
 - CDB [4.3](#)
 - ENABLE PLUGGABLE DATABASE句 [4.3.2.1](#)
 - SEED FILE_NAME_CONVERT句 [4.3.2.2.1](#)
 - undo_mode_clause [4.3.2.4](#)

- CREATE LOCKDOWN PROFILE文 [2.2.7](#), [18.3.3](#)
- CREATE PLUGGABLE DATABASE文 [1.3.2.1.1](#), [1.3.2.1.2](#), [2.1.2.1](#)
 - アプリケーション・コンテナ [2.3.1](#)
 - AS PROXY句 [1.3.2.4](#), [2.1.2.3](#), [10.1](#)
 - 句 [5.4](#)
 - DEFAULT TABLESPACE句 [5.2.2](#)
 - ファイルの場所 [5.2.4](#)
 - HOST句 [10.1.4.1](#)
 - logging_clause [15.4.1](#), [15.4.1.2](#)
 - MAX_AUDIT_SIZE句 [5.2.1](#)
 - MAX_DIAG_SIZE句 [5.2.1](#)
 - NO DATA句 [7.2.2](#)
 - PATH_PREFIX句 [5.2.4.3](#)
 - pdb_force_logging_clause [15.4.1](#)
 - PDBリスナー・ホスト名 [10.1.4](#)
 - PDBリスナー・ポート番号 [10.1.4](#)
 - PORT句 [10.1.4.2](#)
 - REFRESH MODE句 [7.5](#)
 - RELOCATE句 [1.3.2.3](#), [8.1](#), [8.4](#), [8.5](#)
 - SERVICE_NAME_CONVERT句 [5.3](#)
 - SNAPSHOT COPY句 [1.3.2.1.2.2](#), [7.1](#), [7.7.1.1](#)
 - SNAPSHOT MODE句 [16.1.3](#)
 - SOURCE_FILE_DIRECTORY句 [9.1.2.2](#)
 - SOURCE_FILE_NAME_CONVERT句 [9.1.2.1](#)
 - ソース・ファイルの場所 [9.1.2](#)
 - STORAGE句 [5.2.1](#)
 - USER_TABLESPACES句 [5.2.3](#)
 - USING句 [1.3.2.2.1](#), [9.2](#)
 - USING SNAPSHOT句 [7.6.1](#)
- CREATE ROLE文 [2.2.3.1](#)
- アプリケーションPDBの作成 [17.3.5.3](#)
- CDBの作成 [4](#)
 - CREATE DATABASE文 [4.3](#)
 - Database Configuration Assistant [4.3.1](#)
 - ENABLE PLUGGABLE DATABASE句 [4.3.2.1](#)
 - スクリプトによる手動 [4.3](#)
 - Oracle Managed Files [4.3.2.2.2](#)
 - PDB_FILE_NAME_CONVERT初期化パラメータ [4.3.2.2.3](#)
 - 計画 [4.2](#)
 - SEED FILE_NAME_CONVERT句 [4.3.2.2.1](#)
 - undo_mode_clause [4.3.2.4](#)
- PDBの作成 [5](#)
- コンテナ間操作 [2.1.5](#)
- 現在のコンテナ [2.1.4](#), [13.1.1](#)

D

- Database Configuration Assistant
 - CDB [4.3.1](#)
- データベース統合 [1.2.2](#)
- データベース・リンク、PDB [2.1.2.5](#)
- データベース・リソース・マネージャ [1.2.2](#)
 - CDBリソース・プラン [22.2.1](#)
 - CDB [2.7](#), [22](#)
 - CREATE_SIMPLE_PLANプロシージャ [22.3.2](#)
 - PDBの監視 [22.4](#)
 - PDBリソース・プラン [22.3.1](#)
 - PDB [22](#)
- データベース
 - グループ化されたスキーマ
 - 「レルム」を参照 [24.5.1.1](#)
- データベース・サービス [2.4.1](#), [2.4.1.2](#)
 - CDB内 [2.4.1.1](#)
 - PDB [2.4](#)
- Database Upgrade Assistant(DBUA)
 - マルチテナント・アーキテクチャ・サポート [1.1.2](#)
- Database Vault
 - 「Oracle Database Vault」を参照
- Database Vault操作の制御
 - 例外リストへのユーザーとパッケージの追加 [24.9.3](#)
 - 例外リストからのユーザーとパッケージの削除 [24.9.4](#)
 - 無効化 [24.9.5](#)
 - 有効化 [24.9.2](#)
- Database Vaultレルム保護 [24.5.1.1](#)
- Database Vaultレルム保護 [24.5.1.1](#)
- データ定義言語(DDL)
 - CDB [13.5.2](#)
- データ・ディクショナリ
 - CDB [2.1.3](#)
 - PDB [1.2.3](#)
 - CDBのストレージ [2.1.3.4](#)
- データリンク・アプリケーション共通オブジェクト [2.3.2.1](#), [12.1.1.1](#)
- データリンク共通オブジェクト [2.1.2.5](#), [2.3.2](#), [2.3.2.3](#), [17.4.1](#)
- データ・リンク [2.1.3.2](#), [2.3.2.3](#)
- データ操作言語
 - CDB [13.5.1](#)
- DB_CACHE_SIZEパラメータ [22.1.4.2](#)
- DB_CREATE_FILE_DEST初期化パラメータ [5.2.4.2](#)
- DB_PERFORMANCE_PROFILEパラメータ [22.2.5](#)
- DBA_APP_ERRORSビュー [19.11.6](#)

- DBA_APP_PATCHESビュー [19.11.5](#)
- DBA_APP_PDB_STATUSビュー [19.11.2](#)
- DBA_APP_STATEMENTSビュー [19.11.3](#)
- DBA_APP_VERSIONSビュー [19.11.4](#)
- DBA_APPLICATIONSビュー [19.11.1](#)
- DBA_CONTAINER_DATAデータ・ディクショナリ・ビュー [18.1.7.1](#)
- DBA_OBJECTSビュー [17.3.5.2](#)
 - 共有オブジェクト [19.11.7](#)
- DBA_PDB_SAVED_STATESビュー [15.4.5.4](#)
- DBA_PDB_SNAPSHOTFILEビュー [16.6](#)
- DBA_PDB_SNAPSHOTSビュー [16.6](#)
- DBA_PROFILESビュー [17.3.5.2](#)
- DBA_ROLESビュー [17.3.5.2](#)
- DBA_TABLESビュー
 - 拡張データリンク・オブジェクト [19.11.8](#)
- DBA_USERSビュー [17.3.5.2](#)
- DBMS_CREDENTIALパッケージ [13.1.7.2](#)
- DBMS_JOB
 - PDBでの使用 [23.3](#)
- DBMS_MACADM.ENABLE_DVプロシージャ
 - Database Vaultの登録 [24.4.1](#)、[24.4.2](#)、[24.4.3](#)
- DBMS_PDBパッケージ [1.3.2.2.2](#)、[5.1](#)、[5.1.2](#)、[7.3.2](#)、[9.1.3](#)、[17.3.5.2](#)
- DBMS_SCHEDULERパッケージ [23.3](#)
- DBMS_SQLパッケージ
 - CDB [13.5.4](#)
- DEFAULT TABLESPACE句 [5.2.2](#)
- デフォルト一時表領域
 - rootのための指定 [4.3.4.1](#)、[4.3.4.2](#)
- DESCRIBEプロシージャ [9.1.3](#)
- ダイレクト・パス・ロード [1.3.2.2.2](#)
- DROP PLUGGABLE DATABASE文 [11.2](#)、[12.1.3](#)、[12.2.3](#)
- DVSYSスキーマ
 - CDB [24.2](#)

E

- EM Express [1.1.2](#)
 - CDB [4.4](#)
- ENABLE_PLUGGABLE_DATABASE初期化パラメータ [4.3.3](#)
- ENABLE PLUGGABLE DATABASE句 [4.3.2.1](#)
- Enterprise Manager
 - マルチテナント・アーキテクチャ・サポート [1.1.2](#)
- 拡張データ・リンク・アプリケーション共通オブジェクト [12.1.1.1](#)
- 拡張データリンク・オブジェクト [2.3.2.4](#)、[17.4.1](#)

F

- FILE_NAME_CONVERT句 [7.2.4.3](#)
 - フラッシュバックPDB [2.6.2](#), [13.1.6](#)
 - FLASHBACK TABLE SQL文 [24.5.1.1](#)
 - フリート、CDB [14](#), [14.1](#), [14.2](#)
-

H

- HOST句 [10.1.4.1](#)
-

I

- 情報ライフサイクル管理 [24.5.1.1](#)
 - インストール
 - マルチテナント環境でのDatabase VaultおよびLabel Security [24.4.5](#)
-

L

- LEAD_CDB_URIデータベース・プロパティ [14.4](#)
 - LEAD_CDBデータベース・プロパティ [14.3](#)
 - リードCDB [14.1](#)
 - ローカル権限付与
 - 概要 [18.1.2](#)
 - 付与 [18.1.5](#)
 - 取消し [18.1.5](#)
 - ローカル権限
 - 付与 [2.2.4.2.1](#)
 - ローカル・ロール [2.2.3](#), [2.2.3.2](#), [2.2.4.2](#)
 - 概要 [18.2.1](#)
 - 作成 [18.2.8](#)
 - 付与 [2.2.4.2.1](#)
 - 作成のルール [18.2.7](#)
 - ローカル・ユーザー [2.2.2](#), [2.2.2.2](#)
 - 定義 [3.1.1](#)
 - ロックダウン・プロファイル、PDB [2.2.7](#), [18.3.1](#)
 - logging_clause [15.4.1](#), [15.4.1.2](#)
 - 論理変更レコード(LCR) [25.1](#)
 - LogMinerユーティリティ
 - CDBでの使用 [21.2](#)
-

M

- MAX_AUDIT_SIZE句 [5.2.1](#)
 - MAX_DIAG_SIZE句 [5.2.1](#)
 - MAX_IOPSパラメータ [22.1.4.4](#)
 - MAX_MBPSパラメータ [22.1.4.4](#)
 - MAX_PDB_SNAPSHOTSデータベース・プロパティ [16.1](#)
 - メタデータリンク・アプリケーション共通オブジェクト [2.3.2.1](#), [12.1.1.1](#)
 - メタデータリンク共通オブジェクト [2.3.2](#), [2.3.2.2](#), [17.4.1](#)
 - メタデータ・リンク [2.3.2.2.1](#)
 - メタデータ・リンク [2.1.3.2](#), [2.3.2.2.1](#)
 - マルチテナント・アーキテクチャ [1.1](#), [3.1](#), [IV](#), [21.2.2](#)
 - 利点 [1.2.1](#), [1.2.2](#), [1.2.3](#)
 - 概要 [2](#)
 - ユーザー・インタフェース [1.1.2](#)
 - XStream In [25.5](#)
 - XStream Out [25.3](#)
 - マルチテナント・コンテナ・データベース
 - 「CDB」を参照
 - マルチテナント・コンテナ・データベース [21.2](#)
 - 「CDB」を参照
 - マルチテナント環境 [3.1](#)
-

N

- NO DATA句 [7.2.2](#)
 - noncdb_to_pdb.sqlスクリプト [1.3.2.2.2](#), [7.3.2](#), [9.3](#)
 - 非CDB [1.1](#), [1.2.1](#)
 - CDBとしてクローニング [1.3.2.2.2](#)
 - PDBとしてクローニング [1.3.2.1.2](#), [7](#), [7.3.2](#), [7.4.1](#)
 - PDBへの移動 [5.1.2](#)
-

O

- オブジェクト権限
 - 共通権限付与 [18.1.4](#)
- オープン・モード
 - PDB [15.4.5](#)
- オペレーティング・システム [18.4.1](#)
 - 認証
 - PDBのオペレーティング・システム・ユーザー [18.4.1](#)
- オペレーティング・システム・ユーザー
 - PDBの構成 [18.4.2](#)
- ORA-39357: 警告: コンテナ・データベースのルートまたはシードに接続する場合、Oracle Data Pumpの操作は

- 通常必要ありません。 [21.1.1](#)
- ORA-47503エラー [24.4.2](#), [24.4.3](#)
 - Oracle ASM [5.2.4.2](#), [7.8](#)
 - Oracle Database Configuration Assistant(DBCA) [1.1.2](#)
 - Oracle Database Vault
 - 概要 [24.1](#)
 - CDB [13.1.2](#), [24](#)
 - Oracle Database Vault操作の制御
 - 概要 [24.9.1](#)
 - Oracle Database Vaultポリシー
 - マルチテナント環境 [24.8](#)
 - Oracle Database Vault登録
 - 特定のPDBを管理する共通ユーザー [24.4.2](#)
 - CDBルートを管理する共通ユーザー [24.4.1](#)
 - 特定のPDBを管理するローカル・ユーザー [24.4.3](#)
 - Database Vault対応データベースへの接続 [24.4.4](#)
 - 構成と有効化の確認 [24.3](#)
 - Oracle Data Guard
 - CDB [13.1.2](#)
 - Oracle Data Pump [1.3.2.2.2](#), [5.1.2](#)
 - Oracle Data Redaction
 - CDB [18.8](#)
 - Oracle Enterprise Manager Database Express。
 - 「EM Express」を参照
 - Oracle Flashback Technology [24.5.1.1](#)
 - Oracle GoldenGate [1.3.2.2.2](#), [5.1.2](#)
 - Oracle Managed Files [4.3.2.2.2](#), [5.2.4.2](#)
 - Oracle Multitenant [13.1.8](#)
 - Oracle Multitenantオプション [1](#), [IV](#)
 - Oracle Universal Installer [4.1](#)
 - Oracle Virtual Private Database
 - CDB [18.6](#)
 - Oracle Virtual Private Database(VPD)
 - 概要 [18.6.1](#)
 - アプリケーション・コンテナ [18.6.2](#)
 - CDB [18.6.2](#)
-

P

- PATH_PREFIX句 [5.2.4.3](#), [7.2.4.3](#), [13.1.7.3](#)
- PDB_FILE_NAME_CONVERT初期化パラメータ [4.3.2.2.3](#), [7.3.2](#)
- pdb_force_logging_句 [15.4.1](#), [15.4.1.2](#)
- PDB_OS_CREDENTIAL初期化パラメータ [13.1.7.1](#), [13.1.7.2](#)
- PDB_PLUG_IN_VIOLATIONSビュー [15.4.5.1](#)

- pdb_save_or_discard_state句 [15.4.5.4](#)
- pdb_to_apppdb.sqlスクリプト [17.3.5.3](#)
- PDB\$SEED [2.1.2.1](#)
- PDBロックダウン・プロファイル
 - 概要 [18.3.1](#)
 - 作成 [18.3.3](#)
 - デフォルト [18.3.2](#)
 - 無効化 [18.3.4](#)
 - 削除 [18.3.5](#)
 - 有効化 [18.3.4](#)
- PDBパフォーマンス・プロファイル [22.2.4](#)
 - 管理 [22.2.7.3](#)
- PDBの再配置
 - 基本的なステップ [8.5](#)
 - 仕組み [8.3](#)
 - ユーザー・インタフェース [8.4](#)
- PDBリソース・プラン [22.3.1](#)
- PDB [3.1](#), [IV](#), [13.1.8](#), [15.6](#)
 - 概要 [1.1](#)
 - 管理 [15](#)
 - ALTER SYSTEM文 [15.3](#)
 - アーカイブ・ファイル [1.3.2.2.1](#)
 - 監査
 - 許可された監査設定のタイプ [18.9.1](#)
 - バックアップおよびリカバリ [2.6.1](#)
 - 利点 [1.2.1](#)
 - バッファ・プール・サイズ [22.1.4.2](#)
 - CDBリソース・プラン [22.2.1](#)
 - 作成 [22.2.2](#), [22.2.4](#)
 - ディレクティブ [22.2.1.3](#)
 - 無効化 [22.2.8](#)
 - 有効化 [22.2.6](#)
 - 管理 [22.2.7](#)
 - 共有 [22.2.1.1](#)
 - 使用率制限 [22.2.1.2](#)
 - 情報の表示 [22.2.9](#)
 - 文字セット [2.1.1](#)
 - クローニング [1.3.2.1.2](#), [2.1.2.2](#), [7](#), [7.1](#)
 - アプリケーションのクローニング [7.1.1](#)
 - ローカルのクローニング [7.2.1](#), [7.2.2](#), [7.3.1](#)
 - コマンド・ルール [24.7](#)
 - 共通ロール
 - 概要 [18.2.1](#)
 - 作成 [18.2.6](#)
 - 付与 [18.2.9](#)

- 動作の概要 [18.2.2](#)
- 管理に必要な権限 [18.2.4](#)
- 取消し [18.2.9](#)
- 作成のルール [18.2.5](#)
- 共通ユーザー [2.2.1.2](#)
 - PDBのデータへのアクセス [18.1.7.2](#)
 - 権限情報の表示 [18.1.7.1](#)
- 互換性違反 [15.4.5.1](#)
- 接続 [2.4.2](#), [13.2.1](#), [15.2.1](#)
 - ALTER SESSION文 [13.2.2.2](#)
 - CONNECTコマンド [13.2.2.1](#)
- データの統合 [1.3](#)
- コンテナ [2.1](#)
- CPU制限 [22.1.4.1](#)
- プロキシとして作成 [10](#)
- 接続による作成 [1.3.2.2.2](#), [9.2](#)
- シードからの作成 [1.3.2.1.1](#), [6](#)
- 作成 [1.3.2](#), [5](#)
- 現在のコンテナ [2.1.4](#), [13.1.1](#)
- データベース・リンク [2.1.2.5](#)
- データベース・リソース・マネージャ [22](#)
- データ・ディクショナリ [1.2.3](#)
- データ・リダクション・ポリシー [18.8](#)
- DBMS_SQLパッケージ [13.5.4](#)
- 定義 [2.1.2](#), [3.1](#)
- 削除 [11.2](#)
- EM Express [4.4](#)
- 暗号化 [7.1](#)
- PL/SQLコードの実行 [13.5.4](#)
- ファイングレイン監査ポリシー [18.9.4](#)
- フラッシュバック [2.6.2](#), [13.1.6](#), [20.5](#)
- 権限とロールの付与 [2.2.4](#)
- ホット・クローニング [7.1](#)
- I/O制限 [22.1.4.4](#)
- instances_clause [15.4.5.2](#)
- キーストア [7.1](#)
- ローカル・ロール
 - 概要 [18.2.1](#)
 - 作成 [18.2.8](#)
 - 作成のルール [18.2.7](#)
- ロックダウン・プロファイル [2.2.7](#), [13.1.7.1](#)
- 管理 [15.6](#)
- 変更 [15.4](#)
- 移動 [8](#), [8.3.2.1](#), [8.3.2.2](#)
 - 仕組み [8.3](#)

- 目的 [8.2](#)
- 非CDBの移動 [5.1.2](#), [9.1.3](#)
- ネーミング・ルール [2.1.2.4](#)
- オープン・モード [15.4.5](#), [19.5](#)
 - 再起動時の保持 [15.4.5.4](#)
- オペレーティング・システム・ユーザーの構成 [18.4.2](#)
- オペレーティング・システム・ユーザー、設定 [18.4.1](#)
- PDBリソース・プラン [22.3.1](#)
 - 作成 [22.3.2](#)
 - 無効化 [22.3.5](#)
 - 有効化 [22.3.3](#)
 - 変更 [22.3.4](#)
- PGAサイズ [22.1.4.2](#)
- Database Vaultが有効になっているPDBをCDBに接続 [24.10](#)
- 接続 [9](#)
 - 方法 [5.1](#)
 - 準備 [5.5](#)
- Point-in-Timeリカバリ [20.4](#)
- 前提条件 [3.2](#)
- 権限
 - 共通 [18.1.3](#)
 - 付与 [18.1.5](#)
 - 影響 [18.1.1](#)
 - オブジェクト [18.1.4](#)
 - 取消し [18.1.5](#)
 - 情報の表示 [18.1.7.1](#)
- プロキシ [1.3.2](#), [1.3.2.4](#), [2.1.2.1](#), [2.1.2.3](#), [10.1.1](#), [17.3.7.1](#)
- PUBLICロール [18.2.3](#)
- 目的 [2.1.2.2](#)
- リフレッシュ可能なクローン [1.3.2.1.2.3](#)
- リフレッシュ [15.4.4.1](#)
- relocate_clause [15.4.5.2](#)
- 再配置 [1.3.2.3](#), [8](#), [8.1](#), [8.3.2.1](#), [8.3.2.2](#), [8.5](#)
 - 仕組み [8.3](#)
 - 目的 [8.2](#)
 - ユーザー・インタフェース [8.4](#)
- 名前の変更 [15.4.3](#)
- リソース管理 [2.7](#)
- リストア・ポイント [2.6.2](#)
- サービス [2.4](#), [2.4.1](#), [2.4.1.1](#), [2.4.1.2](#), [15.2.2](#)
- services_clause [15.4.5.2](#)
- SGAサイズ [22.1.4.2](#)
- 共有プール・サイズ [22.1.4.2](#)
- SHUTDOWNコマンド [15.5](#), [15.5.4](#)
- 停止 [13.6](#)

- スナップショット・コピー [1.3.2.1.2.2](#), [7.7.1.1](#)
- スナップショット [7.7.3](#), [16.2](#), [16](#), [16.1](#), [16.1.2.1](#), [16.1.2.2](#), [16.1.3](#), [16.3](#), [16.4](#), [16.5](#), [16.6](#)
- STARTUPコマンド [15.5](#), [15.5.2](#), [15.5.3](#)
- タスク [3.3.1](#)
- 一時ファイル [2.5](#)
- ツール [3.3.2](#)
- タイプ [2.1.2.1](#)
- 切断 [1.3.2.2.1](#)
- 切断 [11.1](#)
- 情報の表示 [18.1.7.1](#)
- ビュー [19.1.3](#)
- 仮想プライベート・データベース・ポリシー [18.6.2](#)
- XStream In [25.5](#)
- XStream Out [25.3](#)
 - 構成 [25.4.1](#)
- PDBスナップショット・カルーセル
 - 概要 [16.1](#)
 - 管理 [16](#)
 - 内容 [16.1.2.2](#)
 - 仕組み [16.1.2](#)
 - 目的 [16.1.1](#)
 - スナップショットの最大数の設定 [16.2](#)
 - スナップショットの表示 [16.6](#)
- PDBスナップショット
 - 表示 [16.6](#)
- PGA_AGGREGATE_LIMITパラメータ[22.1.4.2](#)
- PGA_AGGREGATE_TARGETパラメータ[22.1.4.2](#)
- プラガブル・データベース
 - 「PDB」を参照
- プラガブル・データベース
 - 「PDB」を参照
- 切断されたPDBの接続 [9.2](#)
- PORT句 [10.1.4.2](#)
- 権限
 - 共通の付与 [2.2.4.3](#), [2.2.4.3.1](#), [2.2.4.5](#)
 - CDBでの付与 [2.2.4](#), [2.2.4.1](#)
 - ローカル [2.2.4.2.1](#)
- プロキシPDB [1.3.2](#), [1.3.2.4](#), [2.1.2.1](#), [2.1.2.3](#), [2.1.2.5](#), [10](#)
 - 作成 [17.3.7.2](#)
 - 参照先PDB
 - リスナー・ホスト名の変更 [15.2.3.1](#)
 - リスナー・ポート番号の変更 [15.2.3.2](#)
 - アプリケーション・ルート・レプリカの同期 [17.3.7.1](#)
- PUBLICロール, CDB [18.2.3](#)

R

- レルム
 - 概要 [24.5.1.1](#)
 - マルチテナント環境での認可 [24.5.2](#)
 - マルチテナント環境
 - 概要 [24.5.1.2](#)
 - CDBのリカバリ [20](#)
 - リフレッシュ可能なクローンPDB [1.3.2.1.2.3](#), [7.5](#)
 - スイッチオーバー [15.4.4.2](#)
 - REFRESH MODE句 [7.5](#)
 - PDBの再配置 [8](#)
 - 共通のリスナー・ネットワーク [8.3.2.1](#)
 - 独立したリスナー・ネットワーク [8.3.2.2](#)
 - ユーザー・インタフェース [8.4](#)
 - REMOTE_RECOVERY_FILE_DESTパラメータ [7.5](#)
 - リソース・プラン
 - CDB [22.2.1](#)
 - PDB [22.3.1](#)
 - ロール
 - 共通 [2.2.4.3.1](#)
 - 共通、付与 [18.2.9](#)
 - CDBでの付与 [2.2.4](#), [2.2.4.1](#)
 - CDB内 [2.2.3](#)
 - ローカル [2.2.3.2](#), [2.2.4.2](#), [2.2.4.2.1](#)
 - ルート・コンテナ [1.1.1](#), [1.3.1](#), [2.1.1](#)
 - 変更 [13.4.4](#)
 - 情報の表示 [18.1.7.1](#)
 - 行レベルのセキュリティ
 - 「ファイングレイン・アクセス・コントロール」、「Oracle Virtual Private Database (VPD)」を参照
 - rsmgr:I/Oレート制限 [22.1.4.4](#)
 - ルールベースの変換 [25.1](#)
 - ルール [25.1](#)
 - ルール・セット
 - マルチテナント環境
 - 概要 [24.6](#)
-

S

- スケジューラ
 - CDB [23.1](#)
 - PDBのクローズ [23.4](#)
 - CDBの起動 [23.1](#)
 - CDBにおけるジョブ・コーディネータの使用 [23.2](#)
 - CDBにおけるスレーブ・プロセスの使用 [23.2](#)

- ビュー [23.5](#)
- セキュリティ
 - CDB [18](#)
 - PDB [13.1.7](#)
- SEED FILE_NAME_CONVERT句 [4.3.2.2.1](#)
- シードPDB [1.1.1](#), [1.3.1](#)
- SERVICE_NAME_CONVERT句 [5.3](#), [7.2.4.3](#)
- サービス
 - PDB [15.2.2](#)
- SGA_MIN_SIZEパラメータ [22.1.4.2](#)
- SGA_TARGETパラメータ [22.1.4.2](#)
- SHARED_POOL_SIZEパラメータ [22.1.4.2](#)
- SHUTDOWNコマンド
 - PDBのクローズ [15.5](#)
 - PDB [15.5.4](#)
- インスタンスの停止
 - CDB [13.6](#)
- SNAPSHOT COPY句 [1.3.2.1.2.2](#), [7.1](#), [7.7.1.1](#)
- スナップショット・コピーPDB [1.3.2.1.2.2](#)
- SNAPSHOT MODE句 [16.1.3](#)
- スナップショット、PDB [16.1](#)
 - 内容 [16.1.2.1](#)
 - 表示 [16.6](#)
- SOURCE_FILE_DIRECTORY句 [9.1.2.2](#)
- SOURCE_FILE_NAME_CONVERT句 [9.1.2.1](#)
- 分割ミラーによるクローンPDB [7.8](#)
- SQL*Loader
 - アプリケーション・コンテナ [17.3.9](#)
- SQL*Plus
 - マルチテナント・アーキテクチャ・サポート [1.1.2](#)
- SQLスクリプト
 - CDB [13.5.3](#)
- STANDBY_PDB_SOURCE_FILE_DBLINK初期化パラメータ [7.3.2](#)
- STANDBY_PDB_SOURCE_FILE_DIRECTORY初期化パラメータ [9.2](#)
- スタンバイ・データベース
 - CDB [13.1.2](#)
- STARTUPコマンド
 - PDB [15.5.2](#), [15.5.3](#)
 - PDBの起動 [15.5](#)
- STORAGE句 [5.2.1](#)
- リフレッシュ可能なクローンPDBのスイッチオーバー [15.4.4.2](#)
- アプリケーションの同期 [17.3.6](#)
- システム・コンテナ [2.1.1](#)
- システム権限
 - CDB [18.1.3](#)

- 共通権限付与 [18.1.3](#)
-

T

- 透過的データ暗号化
 - CDB [13.1.2](#)
 - Transport Layer Security(TLS)
 - アプリケーション・コンテナ [18.7](#)
-

U

- undo_mode_clause [4.3.2.4](#)
 - UNDOモード
 - CDB [2.5](#), [13.4.3](#)
 - UNDO表領域 [13.1.4](#)
 - CDBに対する指定 [4.3.4.1](#), [4.3.4.2](#)
 - 統合監査ポリシー、アプリケーション・コンテナ
 - 例 [18.9.3.6](#)
 - 統合監査ポリシー、CDB
 - 概要 [18.9.3.1](#)
 - 監査証跡での表示方法 [18.9.3.7](#)
 - 構成 [18.9.3.3](#)
 - 例 [18.9.3.4](#), [18.9.3.5](#)
 - 統合監査ポリシー、ロール
 - 例 [18.9.2](#)
 - 切断 [11.1](#), [12.1.2](#), [12.2.2](#)
 - アップグレード
 - データベース [1.2.2](#), [2.1.3.1](#)
 - USER_TABLESPACES句 [5.2.3](#)
 - ユーザー権限
 - CDB [18.1.1](#)
 - ユーザー
 - 共通 [1.1.1](#), [2.2.1.2](#), [2.2.2.1](#)
 - ユーザー名、CREATE USER文での指定 [13.2.1.3](#)
 - USING SNAPSHOT句 [7.6.1](#)
 - PDBの使用率制限 [22.2.1.2](#)
-

V

- V\$CON_SYS_TIME_MODELビュー [19.1.2](#)
- V\$CON_SYSSTATビュー [19.1.2](#)
- V\$CON_SYSTEM_EVENTビュー [19.1.2](#)
- V\$CON_SYSTEM_WAIT_CLASSビュー [19.1.2](#)

- V\$CONTAINERSビュー [19.3](#)
 - V\$PDBSビュー [15.4.1.2](#), [19.4](#), [19.5](#)
 - V\$RSRCPDBMETRIC_HISTORYビュー [22.4](#)
 - V\$RSRCPDBMETRICビュー [22.4](#)
 - CPU使用率 [22.4.2](#)
 - 生成I/O [22.4.4](#)
 - メモリ使用量 [22.4.5](#)
 - パラレル実行 [22.4.3](#)
 - 仮想プライベート・データベース
 - 「Oracle Virtual Private Database」を参照
 - VPD
 - 「Oracle Virtual Private Database」を参照
-

X

- XStream
 - CDB [25](#)