Oracle® Database Vault

管理者ガイド

19c

F16132-11(原本部品番号:E96302-21)

2023年7月

タイトルおよび著作権情報

Oracle Database Vault管理者ガイド, 19c

F16132-11

原本著者: Patricia Huey

原本協力者: Taousif Ansari、Tom Best、Sanjay Bharadwaj、Ji-won Byun、Martin Cheng、Chi Ching Chui、Scott Gaetjen、Viksit Gaur、Rishabh Gupta、Lijie Heng、Dominique Jeunot、Peter Knaggs、Suman Kumar、Chon Lee、Rudregowda Mallegowda、Yi Ouyang、Hozefa Palitanawala、Gayathri Sairamkrishnan、Vipin Samar、James Spiller、Srividya Tata、Kamal Tbeileh、Saravana Soundararajan、Sudheesh Varma、Peter Wahl、Alan Williams

目次

図一覧

表一覧

はじめに

『Oracle Database Vault管理者ガイド』では、Oracle Database Vaultを使用したOracleデータベース環境におけるアクセス制御ベースのセキュリティの構成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
- 関連ドキュメント
- 表記規則

対象読者

このマニュアルは、セキュリティ管理者、監査管理者、ラベル管理者、およびOracle Database Vaultの構成を担当するOracleデータベース管理者(DBA)を対象としています。

親トピック: はじめに

ドキュメントのアクセシビリティ

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サポートへのアクセス

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、次のマニュアルを参照してください。

- Oracle Databaseセキュリティ・ガイド
- Oracle Label Security管理者ガイド
- Oracle Database管理者ガイド
- Oracle Database SQL言語リファレンス
- Oracle Multitenant管理者ガイド

Oracleテクニカル・サービス

製品データ・シート、よくある質問、最新の製品ドキュメントへのリンク、製品のダウンロードおよびその他の関連ドキュメントは、Oracle Technical Resources (旧称: Oracle Technology Network)からダウンロードできます。Oracleテクニカル・サービスを使用するには、オンライン登録が必要です。登録は、次の場所から無償で行えます。

```
https://www.oracle.com/technical-resources/
```

My Oracle Support

セキュリティ・パッチ、動作要件およびサポート・ナレッジ・ベースに関する情報を確認するには、My Oracle Support(旧OracleMetaLink)に接続してください。場所は次のとおりです。

https://support.oracle.com

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 Database Vault管理者ガイドのこのリリースの変更点

この章の内容は次のとおりです。

Oracle Database Vault 19cでの変更点

Oracle Database 19cのOracle Database Vault管理者ガイドの変更点は次のとおりです。

- 統合監査ポリシーのコマンド・ルールのサポート
  統合監査ポリシー用のOracle Database Vaultコマンド・ルールを作成できるようになりました。
- インフラストラクチャ・データベース管理者のDatabase Vault操作の制御
  マルチテナント・データベースでは、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)による自律型で通常のクラウドまたはオンプレミス環境のプラガブル・データベース(PDB)のローカル・データへのアクセスをブロックできるようになりました。
- 権限分析ドキュメントのOracle Databaseセキュリティ・ガイドへの移動
  権限分析のドキュメントは、『Oracle Database Vault管理者ガイド』から『*Oracle Database*セキュリティ・ガイド』に移動しました。

親トピック: Oracle Database Vault管理者ガイドのこのリリースの変更点

統合監査ポリシーのコマンド・ルールのサポート

統合監査ポリシー用のOracle Database Vaultコマンド・ルールを作成できるようになりました。

コマンド・ルールを使用して、個々の統合監査ポリシーを有効および無効にできるようになりました。この拡張によって、1つのコマンド・ルールを介してすべての統合監査ポリシーを同じように管理するのではなく、各ポリシーの管理方法をきめ細かく制御できます。たとえば、HR監査者は、CRM統合監査ポリシーではなく、自分自身のHR統合監査ポリシーを制御できます。この新機能は、コマンド・ルールに対して`AUDIT`および`NOAUDIT`の使用を拡張しますが、コマンド・ルールに統合監査ポリシーを指定する場合、`AUDIT POLICY`または`NOAUDIT POLICY`を指定する必要があります。

関連項目

- コマンド・ルールで保護できるSQL文
- コマンド・ルールの作成
- CREATE_COMMAND_RULEプロシージャ

親トピック: Oracle Database Vault 19cでの変更点

インフラストラクチャ・データベース管理者のDatabase Vault操作の制御

マルチテナント・データベースでは、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)による自律型で通常のクラウドまたはオンプレミス環境のプラガブル・データベース(PDB)のローカル・データへのアクセスをブロックできるようになりました。

この拡張により、共通ユーザーはPDBに存在するローカル・データにアクセスできなくなります。これにより、ビジネス・アプリケーショ

ンの機密データを格納できるようになり、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する操作が許可されます。

関連項目

- 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

親トピック: Oracle Database Vault 19cでの変更点

権限分析ドキュメントのOracle Databaseセキュリティ・ガイドへの移動

権限分析のドキュメントは、『Oracle Database Vault管理者ガイド』から『*Oracle Database*セキュリティ・ガイド』に移動しました。

権限分析のライセンス情報については、『*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。

関連トピック

- 『Oracle Databaseセキュリティ・ガイド』
- Oracle Databaseライセンス情報ユーザー・マニュアル

親トピック: Oracle Database Vault 19cでの変更点

Oracle Database Vault 18cでの変更点

Oracle Database 18cのOracle Database Vault管理者ガイドの変更点は次のとおりです。

- Oracle Database Vaultシミュレーション・モードの拡張機能
  このリリースのOracle Database Vaultでは、シミュレーション・モードにいくつかの変更が加えられています。
- 新規ファクタ・ファンクション
  このリリースから、4つの新しいファクタ・ファンクションが導入されました。
- ロールへのData PumpおよびDatabase Vault認可の付与機能
  このリリースから、Oracle Database Vault環境でOracle Data Pump操作を実行するための認可をロールに付与できるようになりました。
- Oracle Database VaultでのOracle Database Replayのサポート
  このリリースから、Oracle Database Vault環境でOracle Database Replay操作を実行できるようになりました。

親トピック: Oracle Database Vault管理者ガイドのこのリリースの変更点

Oracle Database Vaultシミュレーション・モードの拡張機能

このリリースのOracle Database Vaultでは、シミュレーション・モードにいくつかの変更が加えられています。

- シミュレーション・モードで、SQL文からすべての必須レルムの違反が取得されるようになりました。
- シミュレーション・モードで、完全なコール・スタック情報を取得できるようになりました。
- デフォルトの信頼できるパス・コンテキスト・ファクタを、連結された列ではなく、個別の列として使用できるようになりました。

SQL文から必須レルムの違反をすべて取得することで、加える必要がある可能性があるすべての変更を確認できます。そのようにしない場合、最初の必須レルムの違反によって他の違反が隠され、元の修正が完了し、新たなリグレッション・テストが実行されるまで、その違反が認識されない可能性があります。この拡張機能によって、リグレッション・テストおよびアプリケーション認証にかかる時間を短縮できます。

完全なコール・スタックを表示すると、違反がある元のSQL文を特定できます。多くの場合、アプリケーションの様々な部分で同様のSQL文が呼び出されています。この機能によりアプリケーション開発者は、違反をトリガーしたアプリケーション・コードを正確かつ迅速に特定できます。

コンテキスト・ファクタは、レルムおよびコマンド・ルールの信頼できるパスを作成するために使用されます。複数の信頼できるパスに共通で使用されるファクタがあり、前のリリースでは、これらのファクタが単一の文字列表現から個別の列に抽出されていました。この拡張機能によって、信頼できるパス・ルール・セットで使用するファクタをより簡単に識別できるようになりました。

関連トピック

- シミュレーション・モードについて

親トピック: Oracle Database Vault 18cでの変更点

新規ファクタ・ファンクション

このリリースから、4つの新しいファクタ・ファンクションを使用できるようになりました。

このファクタ・ファンクションは次のとおりです。

- `F$DV$_CLIENT_IDENTIFIER`
- `F$DV$_DBLINK_INFO`
- `F$DV$_MODULE`
- `F$PROXY_USER`

関連トピック

- Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

親トピック: Oracle Database Vault 18cでの変更点

ロールへのData PumpおよびDatabase Vault認可の付与機能

このリリースから、Oracle Database Vault環境でOracle Data Pump操作を実行するための認可をロールに付与できるようになりました。

以前のリリースでは、この認可を付与できるのは個別のユーザーに対してのみでした。この拡張機能によって管理者は、ユーザーに対するこのタイプの認可をロールを介して簡単に管理できます。

関連トピック

- Oracle Database VaultでのOracle Data Pumpの使用

親トピック: Oracle Database Vault 18cでの変更点

Oracle Database VaultでのOracle Database Replayのサポート

このリリースから、Oracle Database Vault環境でOracle Database Replay操作を実行できるようになりました。

次の機能でサポートされている機能を示します。

- `DBMS_MACADM` PL/SQLプロシージャ:
  - `DBMS_MACADM.AUTHORIZE_DBCAPTURE`
  - `DBMS_MACADM.AUTHORIZE_DBREPLAY`
  - `DBMS_MACADM.UNAUTHORIZE_DBCAPTURE`

        - DBMS_MACADM.UNAUTHORIZE_DBREPLAY
    - データ・ディクショナリ・ビュー:
        - DBA_DV_DBCAPTURE_AUTH
        - DBA_DV_DBREPLAY_AUTH

関連トピック

- Oracle Database VaultにおけるOracle Database Replayの使用

親トピック: Oracle Database Vault 18cでの変更点

1 Oracle Database Vaultの概要

Oracle Database Vaultを使用すると、データに対する管理アクセスを制御できます。

- Oracle Database Vaultの概要
  Oracle Database Vaultは、認可されていない特権ユーザーによる機密データへのアクセスを防ぐためと認可されていないデータベース変更を防ぐための制御を提供します。
- Oracle Database Vaultの使用に必要な権限
  Oracle Database Vaultは、様々なユーザーが職務の分離ガイドラインに基づいて特定のタスクを実行するためのデータベース・ロールを提供します。
- Oracle Database Vaultのコンポーネント
  Oracle Database Vaultには、PL/SQLパッケージおよび他の特定のツールを含む一連のコンポーネントがあります。
- Oracle Database Vaultのコンプライアンスへの対応
  法令を順守することで得られる最も大きな副産物の1つは、セキュリティに対する意識の向上です。
- Oracle Database Vaultによるユーザー・アカウントの保護
  多くのセキュリティ侵害は、外部と内部の両方とも、特権データベース・ユーザー・アカウントを標的にし、データベースからデータを盗み出します。
- Oracle Database Vaultによる柔軟なセキュリティ・ポリシーの実現
  Oracle Database Vaultは、データベースの柔軟なセキュリティ・ポリシーの設計を支援します。
- Oracle Database Vaultのデータベース統合に関する問題への対応
  統合とクラウド環境によってコストは削減されますが、機密アプリケーション・データが、本来アクセスする必要のない人にも公開されるおそれがあります。
- 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。

1.1 Oracle Database Vaultの概要

Oracle Database Vaultは、認可されていない特権ユーザーによる機密データへのアクセスを防ぐためと認可されていないデータベース変更を防ぐための制御を提供します。

- Oracle Database Vaultについて
  Oracle Database Vaultのセキュリティ統制は、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、プライバシおよび規制の要件の順守に役立ちます。
- 特権アカウントに対する統制
  特権データベース・アカウントは、データベース内の機密性の高いアプリケーション・データへのアクセスを獲得する手段として最もよく使われています。
- データベース構成に対する統制
  監査で明らかになる事項として最も多いのが、データベース権限の無認可での変更およびDBAロールのユーザーへの付与が多すぎることです。
- エンタープライズ・アプリケーションの保護ポリシー
  アプリケーション固有のOracle Database Vault保護ポリシーおよびガイドラインを、主要なエンタープライズ・アプリケーションで利用可能です。

親トピック: Oracle Database Vaultの概要

1.1.1 Oracle Database Vaultについて

Oracle Database Vaultのセキュリティ統制は、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、プライバシおよび規制の要件の順守に役立ちます。

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での要注意操作を統制することができます。Oracle Database Vaultでは、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。

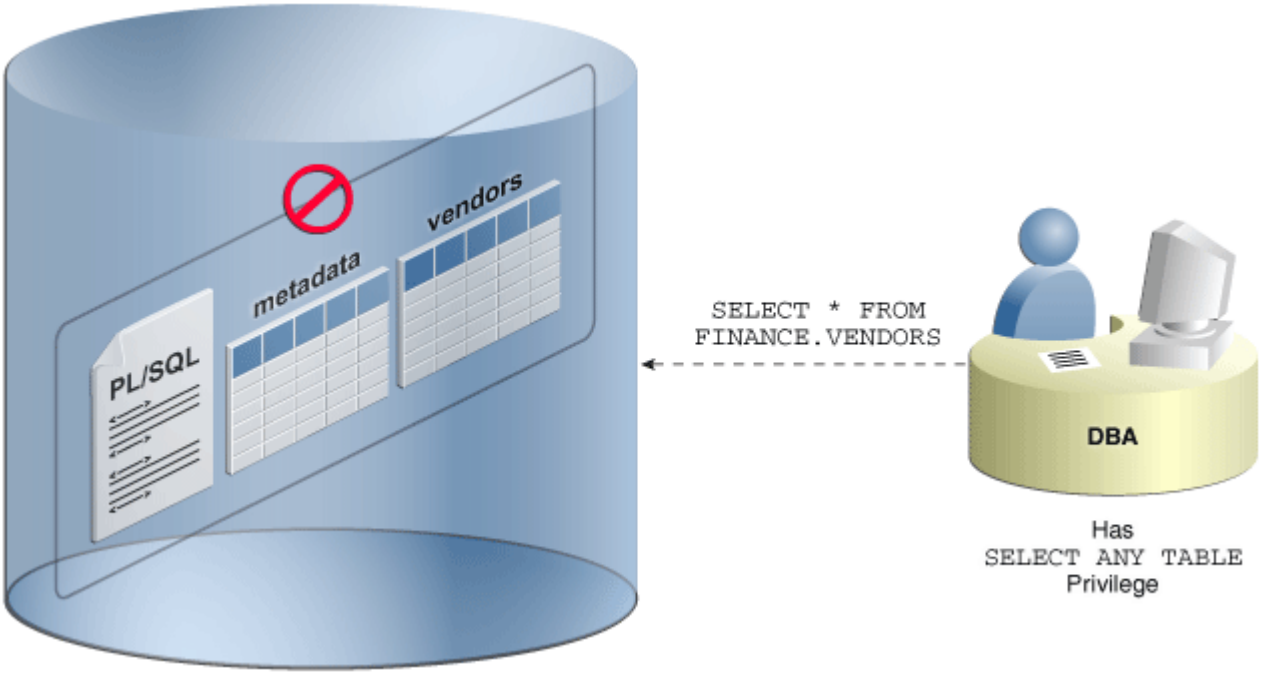
親トピック: Oracle Database Vaultの概要

1.1.2 特権アカウントに対する統制

特権データベース・アカウントは、データベース内の機密性の高いアプリケーション・データへのアクセスを獲得する手段として最もよく使われています。

広範囲にわたる無制限のアクセス権が付与されているので、データベースの保守がしやすくなりますが、同じアクセス権が攻撃ポイントとなって大量のデータへのアクセスを許してしまうおそれがあります。Oracle Database Vaultのレルムにアプリケーション・スキーマ、機密性の高い表、およびストアド・プロシージャを入れておくと、特権アカウントを悪用した侵入者や内部犯行者による機密アプリケーション・データへのアクセスを阻止するための統制を実施できるようになります。

図1-1 DBAによるデータへのアクセスのOracle Database Vaultレルムによるブロッキング



親トピック: Oracle Database Vaultの概要

1.1.3 データベース構成に対する統制

監査で明らかになる事項として最も多いのが、データベース権限の無認可での変更およびDBAロールのユーザーへの付与が多すぎることです。

本番環境に対する無認可での変更を阻止することが重要であるのは、セキュリティのためだけではなく、コンプライアンスのためでもあります。そのような変更によってセキュリティが弱まるおそれがあり、侵入口を開くことになって、プライバシーおよびコンプライアンスの規制に対する違反となるからです。Oracle Database VaultのSQLコマンド・ルールを使用すると、データベース内部で

の操作(たとえば、CREATE TABLE、TRUNCATE TABLE、DROP TABLEなどのコマンド)を統制できます。IPアドレス、認証方法、プログラム名などの多様なファクタがあらかじめ定義されているので、盗んだパスワードを悪用する攻撃を阻止するための信頼できるパスの認可の実装に役立ちます。このような統制は、意図しない構成変更を防ぐだけでなく、ハッカーや内部犯行者によるアプリケーションの改悪を阻止します。

Oracle Database Vaultの必須モードのレルムを利用すると、アプリケーション・オブジェクトへのアクセスを禁止でき、オブジェクト所有者など、オブジェクトに対する権限が直接付与されている場合もアクセスできなくなります。必須レルムでは、アクセスできるユーザーを分析する必要はありません。これは、認可ユーザーのリストからそれが明らかであるためです。

親トピック: Oracle Database Vaultの概要

1.1.4 エンタープライズ・アプリケーションの保護ポリシー

アプリケーション固有のOracle Database Vault保護ポリシーおよびガイドラインを、主要なエンタープライズ・アプリケーションで利用可能です。

これらのエンタープライズ・アプリケーションには、Oracle Fusion Applications、Oracle E-Business Suit、Oracle PeopleSoft、Oracle Siebel、Oracle Financial Services (i-Flex)、Oracle Primavera、SAPおよびInfosysのFinacleが含まれています。Oracle Database Vaultは、アプリケーションを変更したりクライアントを変更する必要がないため、ほとんどの既製アプリケーションやカスタム・アプリケーションで使用できます。

親トピック: Oracle Database Vaultの概要

1.2 Oracle Database Vaultの使用に必要な権限

Oracle Database Vaultは、様々なユーザーが職務の分離ガイドラインに基づいて特定のタスクを実行するためのデータベース・ロールを提供します。

よく使用されるロールは次のとおりです。

- DV_OWNERおよびDV_ADMINでは、Database Vaultポリシーを作成および管理できます。
- DV_ACCTMGRでは、ユーザー・アカウントを管理できます。

Oracle Database Vaultを構成して有効化すると、DV_OWNERロールがユーザーに付与されます。このユーザーは、構成プロセスの開始前に存在する必要があります。また、DV_ACCTMGRロールが、2人目となるオプションのユーザーに付与されます。このユーザーも構成前に存在する必要があります。Database Vaultロールは他のユーザーに付与できますが、それらのユーザーが信頼できることを確認してください。

登録プロセスの間に、DV_OWNERユーザーとDV_ACCTMGRユーザーのバックアップ・アカウントを作成する必要があります。ベスト・プラクティスとして、これらのバックアップ・アカウントを保持し続けることをお薦めします。

関連トピック

- Oracle Database Vaultロール
- バックアップOracle Database Vaultアカウント

親トピック: Oracle Database Vaultの概要

1.3 Oracle Database Vaultのコンポーネント

Oracle Database Vaultには、PL/SQLパッケージおよび他の特定のツールを含む一連のコンポーネントがあります。

- Oracle Database Vaultアクセス制御コンポーネント
  Oracle Database Vaultを使用して一連のコンポーネントを作成し、データベース・インスタンスのセキュリティを管理できます。
- Oracle Enterprise Manager Cloud ControlのDatabase Vault Administratorページ
  Oracle Database Vaultはデフォルトで事前インストールされており、簡単に有効化できます。
- Oracle Database Vault DVSYSおよびDVFスキーマ
  Oracle Database Vaultのデータベース・オブジェクトおよびパブリック・ファンクションは、それぞれDVSYSとDVFスキーマに格納されます。
- Oracle Database Vault PL/SQLインタフェースおよびパッケージ
  Oracle Database Vaultには、セキュリティ管理者またはアプリケーション開発者がアクセス制御ポリシーを構成するためのPL/SQLインタフェースおよびパッケージが用意されています。
- Oracle Database Vaultレポートおよびモニタリング・ツール
  Oracle Enterprise Managerは、Oracle Database Vaultレポートを生成および保守します。

親トピック: Oracle Database Vaultの概要

1.3.1 Oracle Database Vaultアクセス制御コンポーネント

Oracle Database Vaultを使用して一連のコンポーネントを作成し、データベース・インスタンスのセキュリティを管理できます。

これらのコンポーネントは次のとおりです。

- レルム。レルムとは、データベース内で、スキーマ、オブジェクトおよびロールを保護できる保護ゾーンです。たとえば、会計、販売、人事に関する一連のスキーマ、オブジェクトおよびロールを保護できます。これらをレルムに保護すると、レルムを使用してシステム権限とオブジェクト権限の利用を特定のアカウントまたはロールに限定することができます。これにより、これらのスキーマ、オブジェクト、ロールを使用するユーザーに対してきめ細かいアクセス制御を提供できるようになります。レルムの構成では、レルムについて詳しく説明します。「Oracle Database VaultレルムのAPI」も参照してください。
- コマンド・ルール。コマンド・ルールとは、ユーザーによるSELECT文、ALTER SYSTEM文、データベース定義言語(DDL)文およびデータ操作言語(DML)文などのほぼすべてのSQL文の実行方法を制御するために作成する特別なセキュリティ・ポリシーです。コマンド・ルールでは、ルール・セットを使用して、文が許可されるかどうかを判断します。コマンド・ルールの詳細は、「コマンド・ルールの構成」で説明しています。「Oracle Database Vaultコマンド・ルールのAPI」も参照してください。
- ルール・セット。ルール・セットとは、レルム認可、コマンド・ルール、ファクタ割当て、保護アプリケーション・ロールに関連付けることのできる1つ以上のルールの集まりです。ルール・セットは、それに含まれる各ルールと評価タイプ(「すべてのTrue」または「いずれかTrue」)に基づいて、trueまたはfalseに評価されます。ルール・セットは、ゼロ、1つまたは複数のレルム認可、コマンド・ルールまたはセキュア・アプリケーション・ロールに関連付けることができます。ルール・セットの詳細は、「ルール・セットの構成」で説明しています。「Oracle Database Vaultルール・セットのAPI」も参照してください。
- ルール。ルールは、TrueまたはFalseに評価されるPL/SQL式です。複数のルール・セットで同じルールを使用できます。詳細は、ルール・セットの機能を参照してください。
- ファクタ。ファクタとは、ユーザー・ロケーション、データベースIPアドレス、セッション・ユーザーなどOracle Database Vaultで信頼できるパスとして認識および使用できる名前付きの変数または属性です。ルールでファクタを使用すると、データベースに接続するデータベース・アカウントを認可したり、データの可視性および管理性を制限する特定のデータベース・コマンドを実行したりするアクティビティを制御できます。各ファクタは1つ以上のアイデンティティを持ちます。アイデンティティは、ファクタの実際の値です。ファクタの取得メソッド、またはそのアイデンティティ・マッピング・ロジックによって、

ファクタは複数のアイデンティティを持つ場合があります。ファクタの詳細は、「ファクタの構成」で説明しています。「Oracle Database Vaultファクタ API」も参照してください。

- セキュア・アプリケーション・ロール。セキュア・アプリケーション・ロールは、Oracle Database Vaultルール・セットの評価に基づいて有効化できる特別なOracle Databaseロールです。セキュア・アプリケーション・ロールの詳細は、「Oracle Database Vaultのセキュア・アプリケーション・ロールの構成」で説明しています。「Oracle Database Vaultセキュア・アプリケーション・ロールのAPI」も参照してください。

これらのコンポーネントを補強するため、Oracle Database Vaultには一連のPL/SQLインタフェースおよびパッケージが用意されています。「Oracle Database Vault PL/SQLインタフェースおよびパッケージ」で概要を説明しています。

一般的に、最初のステップは、保護するデータベース・スキーマまたはデータベース・オブジェクトを含むレルムを作成することです。ルール、コマンド・ルール、ファクタ、アイデンティティ、ルール・セットおよびセキュア・アプリケーション・ロールを作成すると、さらに強固にレルムを保護できます。また、これらのコンポーネントが監視および保護するアクティビティでレポートを実行できます。「Oracle Database Vaultの開始」には、Oracle Database Vaultの基本的な機能を紹介する簡単なチュートリアルが用意されています。以降の章では、さらに高度なチュートリアルを扱います。「Oracle Database Vaultレポート」では、構成、およびOracle Database Vaultで実行されるその他のアクティビティを確認するためのレポート実行方法の詳細が提供されます。

親トピック: Oracle Database Vaultのコンポーネント

1.3.2 Oracle Enterprise Manager Cloud ControlのDatabase Vault Administratorページ

Oracle Database Vaultはデフォルトで事前インストールされており、簡単に有効化できます。

Oracle Database Vaultの管理はOracle Enterprise Manager Cloud Controlに完全統合されているので、セキュリティ管理者は効率的な中央集中型インタフェースでOracle Database Vaultを管理できます。

Oracle Enterprise Manager Cloud Controlには、Oracle Database Vaultのポリシーの表示と構成、およびOracle Database Vaultのアラートとレポートの表示に使用できるグラフィカル・ユーザー・インタフェースが用意されています。Oracle Database Vault Administratorには、ベースライン・セキュリティ構成の理解をサポートするセキュリティ関連のレポートが多数用意されています。これらのレポートは、このベースラインからの偏差の特定にも役立ちます。

「Oracle Database Vaultの開始」から「Oracle Database Vault環境でのDBA操作」では、レルム、コマンド・ルール、ファクタ、ルール・セット、セキュア・アプリケーション・ロールで定義されるアクセス制御ポリシーを構成するためのOracle Database Vault Administratorページの使用方法と、他のOracle製品へのOracle Database Vaultの統合方法について説明します。「Oracle Database Vaultの監視」では、これらのページを使用してDatabase Vaultアクティビティを監視する方法について、「Oracle Database Vaultレポート」では、Oracle Database Vaultのレポート作成について説明します。

親トピック: Oracle Database Vaultのコンポーネント

1.3.3 Oracle Database Vault DVSYSおよびDVFスキーマ

Oracle Database Vaultのデータベース・オブジェクトおよびパブリック・ファンクションは、それぞれDVSYSとDVFスキーマに格納されます。

Oracle Database Vaultには、OracleデータをOracle Database Vault用に処理する際に必要なデータベース・オブジェクトを保存するDVSYSスキーマが用意されています。このスキーマには、Oracle Database Vaultが使用するロール、ビュー、アカウント、ファンクションおよびその他のデータベース・オブジェクトが含まれます。DVFスキーマには、Oracle Database Vault

アクセス制御構成内に設定されたファクタ値を(実行時に)取得するパブリック・ファンクションが含まれます。これらのスキーマは両方ともスキーマ専用アカウントとして認証されます。これらのアカウントはデフォルトでロックされ、Oracle Supportからの指示がないかぎりロックされたままにしておきます。

関連トピック

- Oracle Database Vaultのスキーマ、ロールおよびアカウント

親トピック: Oracle Database Vaultのコンポーネント

1.3.4 Oracle Database Vault PL/SQLインタフェースおよびパッケージ

Oracle Database Vaultには、セキュリティ管理者またはアプリケーション開発者がアクセス制御ポリシーを構成するためのPL/SQLインタフェースおよびパッケージが用意されています。

PL/SQLプロシージャおよびファンクションを使用すると、指定されたデータベース・セッションのコンテキスト内にあるアクセス制御ポリシーの範囲内において、通常のデータベース・アカウントでの操作が可能になります。

詳細は、「Oracle Database VaultレルムのAPI」から「Oracle Database Vault APIリファレンス」までを参照してください。

親トピック: Oracle Database Vaultのコンポーネント

1.3.5 Oracle Database Vaultレポートおよびモニタリング・ツール

Oracle Enterprise Managerは、Oracle Database Vaultレポートを生成および保守します。

Oracle Database Vaultには、Oracle Database Vaultの構成設定に関する情報(ステータスやコンポーネント情報など)を取得できるデータベース・ビューが用意されています。

また、Oracle Database統合監査証跡またはOracle Enterprise Managerを使用して、ポリシーの変更、セキュリティ違反の試行、Oracle Database Vaultの構成および構造の変更を監視できます。

関連トピック

- Oracle Database Vaultレポート
- Oracle Database Vaultの監視

親トピック: Oracle Database Vaultのコンポーネント

1.4 Oracle Database Vaultのコンプライアンスへの対応

法令を順守することで得られる最も大きな副産物の1つは、セキュリティに対する意識の向上です。

これまで、情報技術(IT)部門は高可用性とパフォーマンスを重視してきました。法令順守に重点を置くことで、ITインフラストラクチャ、データベースおよびアプリケーションをセキュリティという観点から客観的に見ることが必要になりました。一般的な疑問には次のものがあります。

- どこに機密情報が保存されているのか。
- 誰がこの情報へのアクセス権を持っているのか。

サーベンス・オクスリー法(Sarbanes-Oxley Act)、医療保険の相互運用性と説明責任に関する法律(Health Insurance Portability and Accountability Act、HIPAA)、国際業務を行う銀行の自己資本比率に関する国際統一基準(改定版)(バーゼルII、International Convergence of Capital Measurement and Capital Standards: a Revised Framework)、日本の個人情報保護法、PCIデータ・セキュリティ基準(Payment Card Industry Data

Security Standard、PCI DSS)、欧州連合のプライバシと電子通信に関する指令(European Union Directive on Privacy and Electronic Communications)などの規制には、内部規制、職務分離およびアクセス制御を含む共通のテーマがあります。

サーベンス・オクスリーやHIPAAなどの規制による変更の多くは性質上手続き的なものであるのに対し、その他は技術的な投資を必要とする場合があります。規制に共通に見られるセキュリティ要件は厳密な内部規制です。Oracle Database Vaultを使用することで企業が達成できるコンプライアンスのレベルは規制によって異なります。一般的に、Oracle Database Vaultレルム、コマンド・ルール、ファクタおよび職務の分離機能は、規制により世界的に対応が求められている、全体的なセキュリティ・リスクを低減します。

表1-1に、潜在的なセキュリティの脅威に対応している規制を示します。

表1-1 潜在的なセキュリティの脅威に対応している規制

規制	潜在的なセキュリティの脅威
サーベンス・オクスリー法(Sarbanes-Oxley)302 条	データの不正な変更
サーベンス・オクスリー法(Sarbanes-Oxley)404 条	データの変更、不正なアクセス
サーベンス・オクスリー法(Sarbanes-Oxley)409 条	サービス妨害、不正なアクセス
グラム・リーチ・ブライリー(Gramm-Leach-Bliley)	不正なアクセス、変更または公開
医療保険の相互運用性と説明責任に関する法律(Health Insurance Portability and Accountability Act、HIPAA)164.306	データへの不正なアクセス
HIPAA 164.312	データへの不正なアクセス
バーゼル II(Basel II) - 内部リスク管理	データへの不正なアクセス
CFR Part 11	データへの不正なアクセス
日本の個人情報保護法	データへの不正なアクセス
欧州連合のプライバシと電子通信に関する指令(EU Directive on Privacy and Electronic Communications)	データへの不正なアクセス
PCI データ・セキュリティ基準(Payment Card Industry Data Security Standard、PCI DSS)	データの不正な変更

親トピック: Oracle Database Vaultの概要

1.5 Oracle Database Vaultによるユーザー・アカウントの保護

多くのセキュリティ侵害は、外部と内部の両方とも、特権データベース・ユーザー・アカウントを標的にし、データベースからデータを盗み出します。

Oracle Database Vaultは、レルム、ファクタ、コマンド・ルールを使用して権限ユーザー・アカウントへの攻撃を防ぐのに役立ちます。さらに、これらのコンポーネントはデータベース、アプリケーションおよび機密情報への安全なアクセスを支援する強力なセキュリティ・ツールを提供します。ルールとファクタを結合して、データベースのコマンドの実行が可能な条件の制御や、レルムによって保護されているデータへのアクセスの制御ができます。たとえば、ルールとファクタを作成し、IPアドレス、日時および特定のプログラム(JDBC、SQL Developer、SQL*Plusなど)に基づいてデータへのアクセスを制御できます。これにより、指定の条件を満たす接続のみにアクセスを制限できます。また、認可されていないアプリケーションによるデータベースへのアクセスとともに、アプリケーション・データへの不正なアクセスも防ぐことができます。たとえば、ルールを定義してDROP TABLE文の実行を特定のIPアドレスやホスト名に制限できます。

親トピック: Oracle Database Vaultの概要

1.6 Oracle Database Vaultによる柔軟なセキュリティ・ポリシーの実現

Oracle Database Vaultは、データベースの柔軟なセキュリティ・ポリシーの設計を支援します。

たとえば、DBAロールを持つデータベース・ユーザーは、そのロールに付与されたDROP ANY TABLEシステム権限を使用できます。経験のない管理者が、DROP TABLEコマンドを実行したときに非本番データベース上にいると思い込み、実際には本番システム上にいて重要なアプリケーション表を削除するとします。これにより、アプリケーションの停止、データの損失およびリカバリ時間が発生する可能性があります。Oracle Database Vaultを使用すると、このユーザーによるDROP TABLE文の使用を制限することでそのような変更を防ぐように、コマンド・ルールを作成できます。また、文の実行を次のようにして制限するなど、アクティビティをさらに制限するルール・セットをコマンド・ルールに追加できます。

- 時間別(たとえば、営業時間の午前8時から午後6時まで、月曜日から金曜日まで)
- リモートではなく、ローカル・アクセスのみに制限
- アクションを認可するために、1人のユーザーではなく2人のデータベース・ユーザーが必要
- ユーザーがOracle Database Vaultセキュア・アプリケーション・ロールを有効にしている場合
- ホスト名またはIPアドレス別(たとえば、ホスト名を%appserver%にするか、192.0.2.150のIPアドレスと一致させることができます)

Oracle Database Vaultの職務分離は、規模を問わず企業の要件に合わせてカスタマイズできます。たとえば、専門のITスタッフや、アウトソーシング先のバックエンド事業者を抱えている大規模なユーザーは、職務分離をさらに微調整して、アウトソーシング先のデータベース管理者が実行できる処理を制御することができます。一方、一部のユーザーが複数の職責を兼ねる小規模な組織であれば、職務分離を整理し、それらのユーザーが職責ごとに個別の専用アカウントを作成することができます。実行されたすべてのアクションを追跡できるので、侵入者が権限のあるデータベース・アカウントを侵害して不正利用し、機密データを盗み出すことも防止できます。また、監査者によるコンプライアンスの検証も容易になります。

親トピック: Oracle Database Vaultの概要

1.7 Oracle Database Vaultのデータベース統合に関する問題への対応

統合とクラウド環境によってコストは削減されますが、機密アプリケーション・データが、本来アクセスする必要のない人にも公開されるおそれがあります。

ある国からのデータがまったく別の国でホスティングされることもありますが、そのデータへのアクセスはデータが属する国の規制に基づいて制限する必要があります。Oracle Database Vaultによる統制の下で、データベース管理者によるアプリケーション・データへのアクセスを禁止することによって、このような環境のセキュリティを強化できます。加えて、統制はアプリケーションのバイパスのブロックにも役立ち、アプリケーション層からアプリケーション・データへは信頼できるパスだけを強制的に使用させることができます。

Oracle Database Vaultでは、セキュリティ管理のために次の4つの個別の職務分離制御が提供されます。

- DBAロールを使用した日常的なデータベース管理者作業
- DV_OWNERロールとDV_ADMINロールを使用したセキュリティ管理者作業
- DV_ACCTMGRロールを使用したアカウント管理者作業
- 信頼できる名前付きユーザーによるロールおよび権限の付与

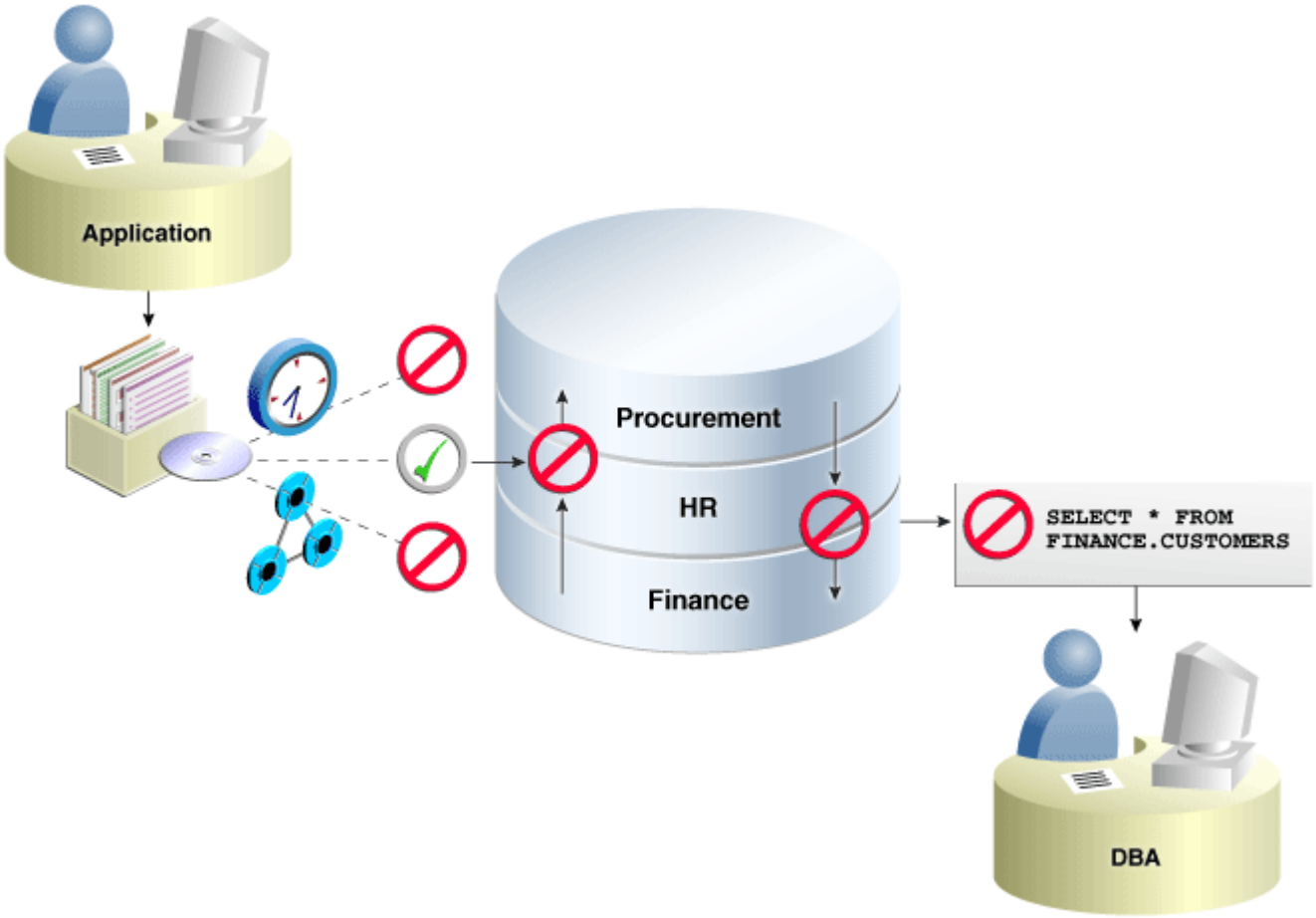
Oracle Database Vault職務分離制御はカスタマイズでき、リソースが限られている組織は、複数のOracle Database Vault責務を同じ管理者に割り当てることができますが、あるアカウントが盗用された場合にデータベースへの損害が最小限に抑えられるよう、職務分離ロールごとに別個のアカウントを使用します。

今日においても、非常に多くのOracleデータベースがエンタープライズ、さらには世界中に分散しています。ただし、今後数年間のコスト削減戦略としてデータベース統合が効果を発揮するには、専用データベース・アーキテクチャによって提供される物理的なセキュリティも、統合環境で利用できなければなりません。Oracle Database Vaultは、データベース統合に関する主要なセキュリティの問題に対応します。

図1-2に、次のデータベースのセキュリティ問題にOracle Database Vaultがどのように対応するかを示します。

- 管理権限のあるアカウントからのアプリケーション・データへのアクセス: この場合、Oracle Database Vaultが許可しないため、データベース管理者はFinanceレルムにより保護されているスキーマにアクセスできません。データベース管理者は最も権限が多く信頼されているユーザーですが、データベースに存在するアプリケーション・データへのアクセス権を付与する必要はありません。
- アプリケーション・データへのアクセスに関する職務の分離: この場合、HRレルムの所有者には、HRレルム・スキーマへのアクセス権がありますが、ProcurementやFinanceへのアクセス権はありません。

図1-2 Oracle Database Vaultのセキュリティ



データベースを統合すると、複数の強力なユーザー・アカウントが1つのデータベース内に存在することになります。つまり、データベース全体の管理者だけでなく、個々のアプリケーション・スキーマの所有者も強力な権限を持つ可能性があるということです。権限の取消しは、既存のアプリケーションに悪影響を及ぼす可能性があります。Oracle Database Vaultレルムを使用すると、信頼できるアプリケーション・パスを介したアプリケーションへのアクセスを強制し、アプリケーション・スキーマのユーザー名とパスワードがアプリケーション自体以外のユーザーによって使用されるのを防止できます。たとえば、SELECT ANY TABLEシステム権限を持つデータベース管理者が、その権限を利用して同じデータベースに存在する他のアプリケーション・データを表示することを制限できます。

親トピック: Oracle Database Vaultの概要

1.8 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、プラガブル・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。

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

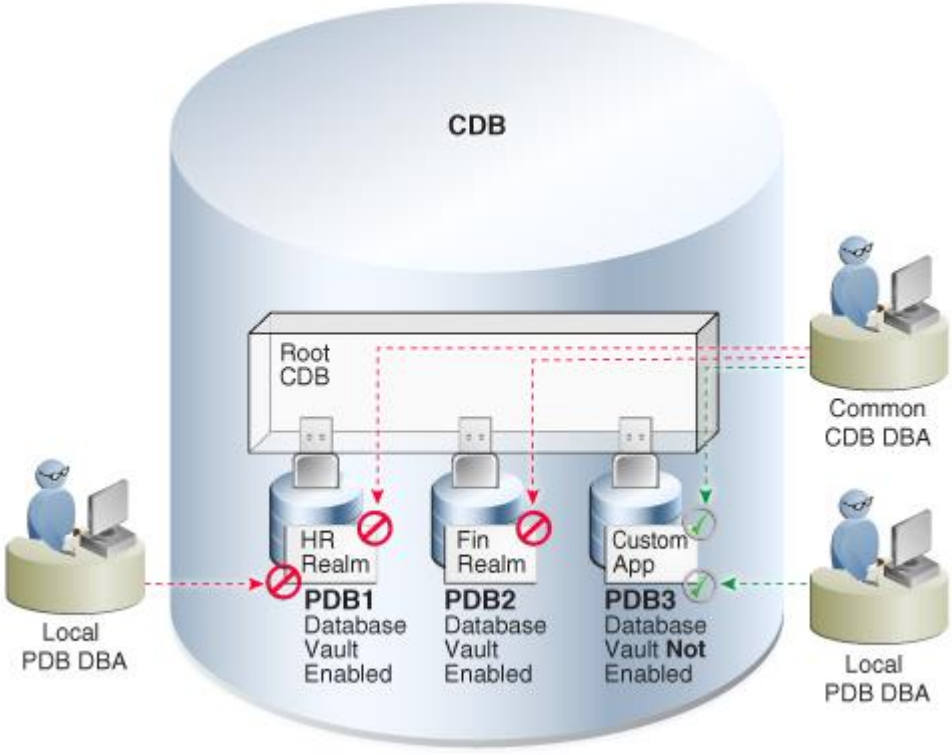
Vaultは、共通のオブジェクトの共通の権限を、ローカル・システム権限と同じ強制ルール下に置きます。

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の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1-3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1-3 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: Oracle Database Vaultの概要

2 Oracle Database Vaultの有効化後のヒント

Oracle Database Vaultを有効化すると、デフォルトのユーザー認可など、いくつかのOracle Databaseセキュリティ機能が、より強力なセキュリティ制限を提供するよう変更されます。

- 変更される初期化およびパスワード・パラメータ設定
  Oracle Database Vault構成では、データベース構成の安全を強化するために複数のデータベース初期化パラメータ設定を変更できます。
- Oracle Database Vaultによるユーザー認可の制限
  Oracle Database構成には、4つの管理データベース・アカウント名(2つのプライマリ・アカウントと2つのバックアップ・アカウント)が必要です。
- 職務分離を実施するためのOracle Database Vault固有のデータベース・ロール
  Oracle Database Vaultの構成は、セキュリティを改善し、法規、プライバシーおよびその他のコンプライアンス要件を満たすことができるように、職務分離の概念を実現します。
- 既存のユーザーおよびロールから取り消される権限
  Oracle Database Vault構成により、職務分離を強化するために、Oracle Databaseで提供されている複数のユーザーおよびロールから権限が取り消されます。
- 既存のユーザーおよびロールに対して阻止される権限
  Oracle Database Vaultの構成により、ユーザーSYSおよびSYSTEMなど、これらの権限が付与されているすべてのユーザーおよびロールに対していくつかの権限が阻止されます。
- 非統合監査環境の変更されたAUDIT文の設定
  Oracle Database Vaultを構成する際に統合監査を使用しない場合、Database Vaultは、複数のAUDIT文を構成します。

2.1 変更される初期化およびパスワード・パラメータ設定

Oracle Database Vault構成では、データベース構成の安全を強化するために複数のデータベース初期化パラメータ設定を変更できます。

これらの変更が組織のプロセスやデータベースのメンテナンス手順に影響する場合は、問題を解決するためにOracleサポートにご連絡ください。

表2-1で、Oracle Database Vaultにより変更される初期化パラメータ設定を説明します。初期化パラメータは、`init.ora`初期化パラメータ・ファイルに格納されます。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。

表2-1 変更されるデータベース初期化パラメータ設定

パラメータ	データベースのデフォルト値	Database Vaultにより設定される新しい値	変更の影響
AUDIT_SYS_OPERATIONS	FALSE	TRUE	ユーザーSYS、および SYSDBA または SYSOPER 権限で接続しているユーザーにより直接発行された上位レベルの

パラメータ	データベースのデフォルト値	Database Vaultにより設定される新しい値	変更の影響
			操作の監査を有効にします。
OS_ROLES	構成されていません	FALSE	オペレーティング・システムによる、ユーザーへのロールの付与と取消しの完全な管理を無効にします。以前に GRANT 文を使用してユーザーに付与されたロールはデータ・ディクショナリにまだリストされているため、変更されません。オペレーティング・システム・レベルでのユーザーへのロールの付与のみに適用されます。この場合も、ユーザーは権限をロールとユーザーに付与できます。
REMOTE_LOGIN_PASSWORDFILE	EXCLUSIVE	EXCLUSIVE	Oracle Database がパスワード・ファイルをチェックするかどうかを指定します。REMOTE_LOGIN_PASSWORDFILE が EXCLUSIVE に設定されていないデータベースに Oracle Database Vault をインストールした場合、EXCLUSIVE に設定するとパスワード・ファイルが使用されます。
SQL92_SECURITY	TRUE	TRUE	ユーザーに UPDATE または DELETE オブジェクト権限が付与されている場合、WHERE または SET 句を指定した UPDATE または DELETE 操作を表に対して実行するには、ユーザーに SELECT オブジェクト権限も付与されるようにする必要があります。 ユーザーに(SELECT ではなく)READ オブジェクト権限のみが付与される場合、UPDATE または DELETE 操作を実行できないことに留意してください。

親トピック: Oracle Database Vaultの有効化後のヒント

2.2 Oracle Database Vaultによるユーザー認可の制限

Oracle Database構成には、4つの管理データベース・アカウント名(2つのプライマリ・アカウントと2つのバックアップ・アカウント)

が必要です。

また、データベース・ロールが複数作成されます。これらのロールは、Oracle Database Vaultにより実現される職務分離の一部です。複数の大企業に影響を与えている共通の監査問題の1つは、データベース管理者による本番インスタンス内での新規データベース・アカウントの不正な作成です。Oracle Database Vaultのインストール時に、Oracle Database Vaultアカウント・マネージャ、またはOracle Database Vaultアカウント・マネージャ・ロールを付与されたユーザー以外はデータベース内にユーザーを作成することができなくなります。

関連トピック

- 職務分離のガイドライン

親トピック: Oracle Database Vaultの有効化後のヒント

2.3 職務分離を実施するためのOracle Database Vault固有のデータベース・ロール

Oracle Database Vaultの構成は、セキュリティを改善し、法規、プライバシーおよびその他のコンプライアンス要件を満たすことができるように、職務分離の概念を実現します。

Oracle Database Vaultでは、アカウント管理の職責、データ・セキュリティの職責およびデータベース管理の職責がデータベース内で明確に分離されます。つまり、データとシステム構成の両方について1人のユーザーがすべての権限を持たないように、多くの権限を持つロール(DBAなど)という概念が複数の新しいデータベース・ロールに分割されています。Oracle Database Vaultでは、特権ユーザー(DBAとその他の特権ロールおよびシステム権限があるユーザー)が、レルムという指定され保護されたデータベース領域にアクセスできないようになっています。Oracle Database Vault所有者(DV_OWNER)およびOracle Database Vaultアカウント・マネージャ(DV_ACCTMGR)と呼ばれる新しいデータベース・ロールも導入されています。これらの新規データベース・ロールにより、データ・セキュリティとアカウント管理が従来のDBAロールから分離されます。これらのロールは、組織内の別々のセキュリティの専門家にマップします。

関連トピック

- 職務分離のガイドライン
- Oracle Database Vaultロール

親トピック: Oracle Database Vaultの有効化後のヒント

2.4 既存のユーザーおよびロールから取り消される権限

Oracle Database Vault構成により、職務分離を強化するために、Oracle Databaseで提供されている複数のユーザーおよびロールから権限が取り消されます。

表2-2に、Oracle Database Vaultによって、Oracle Databaseで提供されている複数のユーザーおよびロールから取り消される権限の一覧を示します。Oracle Database Vaultを無効にすると、これらの権限は取り消されたままになるので注意してください。アプリケーションがこれらの権限に依存する場合は、アプリケーションの所有者に権限を直接付与します。マルチテナント環境では、これらの権限はCDBルートとそのPDBのユーザーおよびロールからと、アプリケーション・ルートとそのPDBから取り消されます。

表2-2 Oracle Database Vaultで取り消される権限

ユーザーまたはロール	取り消される権限

ユーザーまたはロール	取り消される権限
DBA ロール	● BECOME USER ● SELECT ANY TRANSACTION ● CREATE ANY JOB ● CREATE EXTERNAL JOB ● EXECUTE ANY PROGRAM ● EXECUTE ANY CLASS ● MANAGE SCHEDULER ● DEQUEUE ANY QUEUE ● ENQUEUE ANY QUEUE ● MANAGE ANY QUEUE
IMP_FULL_DATABASE ロール脚注 1	● BECOME USER ● MANAGE ANY QUEUE
EXECUTE_CATALOG_ROLE ロール	● EXECUTE ON DBMS_LOGMNR_D ● EXECUTE ON DBMS_LOGMNR_LOGREP_DICT ● EXECUTE ON DBMS_FILE_TRANSFER ● EXECUTE ON SYS.DBMS_LOGMNR
PUBLIC ユーザー	● CONFIGURE_DV プロシージャの実行中に EXECUTE ON UTL_FILE を実行しますが、この取消しが発生する前に、CONFIGURE_DV により、このプロシージャに依存するすべてのスキーマにオブジェクト権限が直接付与されます
SCHEDULER_ADMIN ロール脚注 2	● CREATE ANY JOB ● CREATE EXTERNAL JOB ● EXECUTE ANY PROGRAM ● EXECUTE ANY CLASS ● MANAGE SCHEDULER

脚注1

Oracle Data Pumpを使用してデータをエクスポートおよびインポートするようユーザーに認可を与えるには、「Oracle

Database VaultでのOracle Data Pumpの使用」を参照してください。

脚注2

データベース・ジョブをスケジュールするようユーザーに認可を与えるには、「Oracle Database VaultでのOracle Schedulerの使用」を参照してください。

ノート:

SYS および SYSTEM ユーザーは両方とも、デフォルト・パスワードを使用するユーザー・アカウントを示す DBA_USERS_WITH_DEFPWD データ・ディクショナリ・ビューに対する SELECT 権限を保持します。他のユーザーにこのビューへのアクセス権を付与するには、SELECT 権限を付与します。

関連トピック

- Oracle Database Vaultロールの権限
- DV_ACCTMGR Database Vaultアカウント・マネージャ・ロール

親トピック: Oracle Database Vaultの有効化後のヒント

2.5 既存のユーザーおよびロールに対して阻止される権限

Oracle Database Vaultの構成により、ユーザーSYSおよびSYSTEMなど、これらの権限が付与されているすべてのユーザーおよびロールに対していくつかの権限が阻止されます。

DV_ACCTMGRロールには、職務分離に対する次の権限があります。

- ALTER PROFILE
- ALTER USER
- CREATE PROFILE
- CREATE USER
- DROP PROFILE
- DROP USER

セキュリティの強化および職務分離規定の維持のために、SYSユーザーやSYSTEMユーザーにはユーザー・アカウントを作成または管理する権限を与えないでください。

任意のロールをユーザーSYSに付与できますが、SYSセッションではロールが有効化されていないため、SYSはこのロールを使用できません。

親トピック: Oracle Database Vaultの有効化後のヒント

2.6 非統合監査環境の変更されたAUDIT文の設定

Oracle Database Vaultを構成する際に統合監査を使用しない場合、Database Vaultは、複数のAUDIT文を構成します。

関連トピック

- Oracle Database Vault用に作成されるOracle Database監査設定

親トピック: Oracle Database Vaultの有効化後のヒント

3 Oracle Database Vaultの開始

Oracle Database Vaultを使用して開始する前に、これをOracle Databaseで構成および有効化する必要があります。

- Oracle DatabaseでのOracle Database Vaultの構成および有効化について
  インストール・プロセスでデフォルト・データベースを含めるように選択した場合、Oracle DatabaseにはDatabase Vaultが付属していますが、使用するには構成して有効化する必要があります。
- マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化
  いくつかのシナリオに基づいて、マルチテナント環境でOracle Database Vaultを構成して有効化できます。
- 非マルチテナント環境におけるOracle Database Vaultの登録
  ユーザーを登録した後に、これらのアカウントを保護するプロファイルを作成する必要があります。
- Oracle Real Application Clusters環境でのOracle Database Vaultの構成および有効化
  各Oracle RACノードを含むOracle Real Application Clusters (Oracle RAC)環境に対してOracle Database Vaultを構成できます。
- 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。
- Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン
  Oracle Enterprise Manager Cloud Control (Cloud Control)には、Oracle Database Vaultの管理用のページが用意されています。
- クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護
  このチュートリアルでは、HRスキーマの周辺でレルムを作成する方法を示します。

3.1 Oracle DatabaseでのOracle Database Vaultの構成および有効化について

インストール・プロセスでデフォルト・データベースを含めるように選択した場合、Oracle DatabaseにはDatabase Vaultが付属していますが、使用するには構成して有効化する必要があります。

Oracle Databaseには、インストール・プロセスでデフォルトのデータベースを含めるように選択した場合、Database Vaultが付属しますが、このDatabase Vaultを使用するには登録する必要があります。カスタム・データベースを作成する場合、DBCAを使用してDatabase Vaultをインストールし、そのデータベースに対して有効にすることができます。登録プロセスでは、Oracle Label Securityがまだ有効ではない場合、有効になります。Oracle Label SecurityはOracle Database Vaultに必要ですが、別にOracle Label Securityの使用を開始してOracle Label Securityポリシーを作成する場合を除き、別個のライセンスは必要ありません。この手順は、CDBルート、アプリケーション・ルートおよび現在のプラガブル・データベース(PDB)に適用され、単一インスタンスとOracle Real Application Clusters (Oracle RAC)の両方のインストールに適用されます。マルチテナント・データベースでは、PDBのいずれかでDatabase Vaultを構成する前に、Database VaultをCDBルートで構成する必要があります。

構成プロセスの一環として、Database Vault管理者アカウントを作成しました。これらは、Database Vaultのロールである DV_OWNERおよびDV_ACCTMGRを保持するアカウントです。最初にこれらのアカウントを使用して、管理権限を持つ名前付きユーザーにロールをプロビジョニングします。SYSではこれらのロールを持つユーザーのパスワードをリセットできないため、バックアップ・アカウントの保持により、名前付きユーザーの資格証明の紛や資格証明の配置の誤りからリカバリできます。

Oracle Databaseは、マルチテナント環境と非マルチテナント環境の両方に登録できます。マルチテナント環境の場合、登録

用に選択できるメソッドがいくつかあります。

ノート:

Oracle Database 12c より前のリリースからアップグレードしてあり、その以前のリリースで以前の Oracle Database Vault が有効になっている場合は、アップグレード・プロセスの完了後に、DBMS_MACADM.ENABLE_DV プロシージャを使用することで Oracle Database Vault を有効にする必要があります。

マルチテナント環境では、リリース 12c より前のリリースから非 Database Vault の登録済 Oracle Database を移行する場合、Database Vault の手動インストールを実行する必要があります。

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

親トピック: Oracle Database Vaultの開始

3.2 マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化

いくつかのシナリオに基づいて、マルチテナント環境でOracle Database Vaultを構成して有効化できます。

- マルチテナント環境でのDatabase Vaultの構成および有効化について
  関連するPDBで同じアクションを実行する前に、CDBルートでOracle Database Vaultを構成して有効化する必要があります。
- CDBルートでのDatabase Vaultの構成および有効化
  マルチテナント環境では、CDBルートでDatabase Vault対応ロールを使用する共通ユーザーによりOracle Database Vaultを構成して有効化します。
- 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、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。
- 特定のPDBを管理するためのDatabase Vaultローカル・ユーザーの構成および有効化
  マルチテナント環境では、最初にルートでOracle Database Vaultを構成して有効化し、次にPDBで構成して有効化する必要があります。
- DV_OWNERおよびDV_ACCTMGRユーザーを保護するプロファイルの作成
  プロファイルは、DV_OWNERおよびDV_ACCTMGRロールを付与されたユーザーに追加の保護を提供します。
- 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。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に変換します。

親トピック: Oracle Database Vaultの開始

3.2.1 マルチテナント環境でのDatabase Vaultの構成および有効化について

関連するPDBで同じアクションを実行する前に、CDBルートでOracle Database Vaultを構成して有効化する必要があります。

CDBルートでDV_OWNERロールとDV_ACCTMGRロールを割り当てられた共通ユーザーも、PDBで同じロールを持つことができます。PDBでは、同じ共通ユーザーを使用してDatabase Vaultを構成して有効化することも、別のPDBローカル・ユーザーを使用することもできます。DV_ACCTMGRロールは、CDBルートの共通ユーザーに共通に付与されます。Database Vaultを構成してCDBルートに登録するときに、DV_OWNERをローカルに、または共通にCDBルート共通ユーザーに付与できます。DV_OWNERを共通ユーザーにローカルに付与すると、共通DV_OWNERユーザーは、どのPDBでもこのロールを使用できなくなります。

親トピック: マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化

3.2.2 CDBルートでのDatabase Vaultの構成および有効化

マルチテナント環境では、CDBルートでDatabase Vault対応ロールを使用する共通ユーザーによりOracle Database Vaultを構成して有効化します。

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。

   たとえば:

   ```
   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されるユーザー・アカウントを選択(または新規ユーザーを作成)選択します。

   これらのアカウント名の先頭にc##またはC##を付加します。たとえば:

   ```
   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;
   ```

   この指定では、2つのシステム権限を付与し、アカウントが存在しない場合はアカウントを作成し、パスワードを割り当て、すべてのユーザーがCDBおよびすべてのPDBデータベースにアクセスできるようにしています。

   - プライマリ・アカウント(c##sec_admin_owenおよびc##accts_admin_ace)が新規ロールDV_ADMINおよびDV_ACCTMGRにまだ存在しない場合、これらを作成します。
   - 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。

   たとえば:

   ```
   BEGIN
   ```

```
CONFIGURE_DV (
dvowner_uname         => 'c##dbv_owner_root_backup',
dvacctmgr_uname       => 'c##dbv_acctmgr_root_backup',
  force_local_dvowner   => FALSE);
END;
/
```

この例では、force_local_dvownerをFALSEに設定すると、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DV_OWNERユーザーはCDBルートにのみDV_OWNERロール権限を持つように制限されます。DV_OWNERをCDBルート共通ユーザーにローカルに付与すると、そのユーザーはその他のユーザーに共通にDV_OWNERロールを付与できません。

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。

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。

たとえば:

```
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で通常モードを使用できるようにするには、次のようにします。

```
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を有効にしていないPDBは、データベースの再起動後、Database VaultがPDBで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. SYSOPER管理権限で接続します。

```
CONNECT / AS SYSOPER
```

9. データベースを再起動します。
単一インスタンス・データベースの場合:

```
SHUTDOWN IMMEDIATE
STARTUP
```

Oracle Real Application Clusters (Oracle RAC)環境にいる場合は、Oracle RACのローリング有効化を実行できます。

10. SYSDBA管理権限を使用して接続します。

```
CONNECT / AS SYSDBA
```

11. Oracle Database VaultおよびOracle Label Securityがインストールされ、有効になっていることを確認します。

```
SELECT * FROM CDB_DV_STATUS;
```

```
SELECT * FROM CDB_OLS_STATUS;
```

12. 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(その際はWITH ADMIN OPTION句を指定します)。

    たとえば:

```
CONNECT c##dbv_owner_root_backup
Enter password: password
GRANT DV_OWNER TO c##sec_admin_owen WITH ADMIN OPTION;
```

13. バックアップDV_ACCTMGRユーザーとして接続し、DV_ACCTMGRロールをバックアップDV_ACCTMGRユーザーに付与します(その際はWITH ADMIN OPTION句を指定します)。

    たとえば:

```
CONNECT c##dbv_acctmgr_root_backup
Enter password: password
GRANT DV_ACCTMGR TO c##accts_admin_ace WITH ADMIN OPTION
CONTAINER=ALL;
```

14. 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ロール
- Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン
- DV_PATCH_ADMIN Database Vaultデータベース・パッチ・ロール
- CONFIGURE_DVの一般システム・メンテナンス・プロシージャ
- Oracle Real Application Clusters環境でのOracle Database Vaultの構成および有効化

親トピック: マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化

3.2.3 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、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・アカウントを指定または作成します。
2. CDBルートでOracle Database Vaultを構成して有効化し、共通ユーザーにDV_OWNERロールが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。
   たとえば:

```
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。

たとえば:

```
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省略されます。PDB内で構成されているすべての共通ユーザーは、PDBの範囲に制限されます。

7. utlrp.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/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。

8. 先ほど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。

たとえば:

```
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。

たとえば:

```
ALTER PLUGGABLE DATABASE pdb_name CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE pdb_name OPEN;
```

12. PDBが構成され、Database Vaultに対して有効になっていることを確認します。

```
SELECT * FROM DBA_DV_STATUS;
```

13. 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、WITH ADMIN OPTION句を含むDV_OWNERロールを付与します。

たとえば:

```
CONNECT c##dbv_owner_root_backup@pdb_name
```

```
Enter password: password
GRANT DV_OWNER TO c##sec_admin_owen WITH ADMIN OPTION;
```

14. バックアップDV_ACCTMGRユーザーとして接続し、DV_ACCTMGRロールをプライマリDV_ACCTMGRユーザーに付与します(その際はWITH ADMIN OPTION句を指定します)。

    たとえば:

```
CONNECT c##dbv_acctmgr_root_backup@pdb_name
Enter password: password
GRANT DV_ACCTMGR TO c##accts_admin_ace WITH ADMIN OPTION;
```

15. 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ロール
- Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン
- DV_PATCH_ADMIN Database Vaultデータベース・パッチ・ロール
- CONFIGURE_DVの一般システム・メンテナンス・プロシージャ
- CDBルートでのDatabase Vaultの構成および有効化

親トピック: マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化

3.2.4 特定のPDBを管理するためのDatabase Vaultローカル・ユーザーの構成および有効化

マルチテナント環境では、最初にルートでOracle Database Vaultを構成して有効化し、次にPDBで構成して有効化する必要があります。

最初にPDBで構成して有効化しようとすると、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. 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、PDBにログインします。
   たとえば:

```
sqlplus sec_admin@pdb_name
Enter password: password
```

2. 新しいDatabase Vaultロールに対して既存のローカル・ユーザーの名前付きアカウントを使用していない場合は、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した場合またはパスワードを忘れた場合に備えて、Database Vaultロールを保持するためにバックアップ・アカウントを作成する必要があります。

```
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ルートでOracle Database Vaultを構成して有効化したことを確認します。
   ルートに一時的に接続してから、DBA_DV_STATUSビューを問い合せます。

```
SELECT * FROM SYS.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_uname     => 'dbv_owner_backup',
   dvacctmgr_uname   => 'dbv_acctmgr_backup');
 END;
/
```

   この例では、force_local_dvownerパラメータは不要なため省略されます。Database Vaultロールは、PDBで構成したときにローカルに付与されます。

6. utlrp.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/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. 先ほど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。
   たとえば:

```
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が構成され、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;
```

12. 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、WITH ADMIN OPTION句を含むDV_OWNERロールを付与します。

   たとえば:

```
CONNECT dbv_owner_backuppdb_name
Enter password: password
```

```
GRANT DV_OWNER TO sec_admin_owen WITH ADMIN OPTION;
```

13. バックアップDV_ACCTMGRユーザーとして接続し、バックアップDV_ACCTMGRユーザーに、WITH ADMIN OPTION句を含むDV_ACCTMGRロールを付与します。

    たとえば:

```
CONNECT dbv_acctmgr_backup@pdb_name
Enter password: password
GRANT DV_ACCTMGR TO c##accts_admin_ace WITH ADMIN OPTION;
```

14. 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ロール
- CDBルートでのDatabase Vaultの構成および有効化
- Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン

親トピック: マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化

3.2.5 DV_OWNERおよびDV_ACCTMGRユーザーを保護するプロファイルの作成

プロファイルは、DV_OWNERおよびDV_ACCTMGRロールを付与されたユーザーに追加の保護を提供します。

DV_OWNERまたはDV_ACCTMGRロールを付与されたデータベース・ユーザーは、クリティカルな特権アカウントとみなされます。通常、これらのアカウントはサービス・アカウントとみなしてパスワードのロックアウト要件が適用されないようにする必要があります。Oracleでは、アカウントがロックされないカスタム・プロファイルを作成することをお薦めします。また、これらのDatabase Vault関連のアカウントの失敗したログイン試行の監査も必要です。

1. CREATE PROFILEシステム権限を持つユーザーとして、データベース・インスタンスにログインします。
   - 共通DV_OWNERおよびDV_ACCTMGRユーザーの場合: データベース・インスタンスのルートにログインします。
   - ローカルのDV_OWNERおよびDV_ACCTMGRユーザーの場合: ユーザーを作成したPDBにログインします。
2. 次のようなプロファイルを作成します。
   - 共通DV_OWNERおよびDV_ACCTMGRユーザーの場合: ルートで、次のようなプロファイルを作成します。

```
CREATE PROFILE c##dv_profile limit
FAILED_LOGIN_ATTEMPTS UNLIMITED
PASSWORD_VERIFY_FUNCTION ORA12C_VERIFY_FUNCTION
PASSWORD_LOCK_TIME UNLIMITED
CONTAINER=CURRENT;
```

   - ローカルDV_OWNERおよびDV_ACCTMGRユーザーの場合: PDBで、次のようなプロファイルを作成します。

```
CREATE PROFILE dv_profile limit
FAILED_LOGIN_ATTEMPTS UNLIMITED
PASSWORD_VERIFY_FUNCTION ORA12C_VERIFY_FUNCTION
PASSWORD_LOCK_TIME UNLIMITED
CONTAINER=CURRENT;
```

3. このプロファイルを使用するために、DV_OWNERおよびDV_ACCTMGRユーザー・アカウントを更新します。
   - 共通DV_OWNERおよびDV_ACCTMGRユーザーの場合:

```
ALTER USER c##sec_admin_owen PROFILE c##dv_profile CONTAINER=ALL;
ALTER USER c##dbv_owner_root_backup PROFILE c##dv_profile CONTAINER=ALL;
```

```
ALTER USER c##accts_admin_ace PROFILE c##dv_profile CONTAINER=ALL;
ALTER USER c##dbv_acctmgr_root_backup PROFILE c##dv_profile
CONTAINER=ALL;
```

- ローカルDV_OWNERおよびDV_ACCTMGRユーザーの場合:

```
ALTER USER sec_admin_owen PROFILE dv_profile CONTAINER=CURRENT;
ALTER USER dbv_owner_backup PROFILE dv_profile CONTAINER=CURRENT;
ALTER USER accts_admin_ace PROFILE dv_profile CONTAINER=CURRENT;
ALTER USER dbv_acctmgr_backup PROFILE dv_profile CONTAINER=CURRENT;
```

4. AUDIT_ADMINロールを付与されたユーザーとして接続します。
5. 統合監査ポリシーを作成して有効化し、DV_OWNERまたはDV_ACCTMGRロールを付与されたユーザーによる失敗したログインを追跡します。
   - 共通DV_OWNERおよびDV_ACCTMGRユーザーの場合: ルートで、次のようなポリシーを作成します。

```
CREATE AUDIT POLICY c##dv_logins ACTIONS LOGON;
AUDIT POLICY c##dv_logins BY USERS WITH GRANTED ROLES DV_OWNER,
DV_ACCTMGR WHENEVER NOT SUCCESSFUL;
```

   - ローカルDV_OWNERおよびDV_ACCTMGRユーザーの場合: PDBで、次のようなポリシーを作成します。

```
CREATE AUDIT POLICY dv_logins ACTIONS LOGON;
AUDIT POLICY dv_logins BY USERS WITH GRANTED ROLES DV_OWNER, DV_ACCTMGR
WHENEVER NOT SUCCESSFUL;
```

関連トピック

- 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: マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化

3.2.6 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ということを知っておいてください。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。それでも、この接続したDatabase Vault無効のPDBは使用可能です。ただし、このCDBでDatabase Vaultが厳密オプションで有効になっている場合、このPDBでDatabase Vaultを有効にする必要があります。

Database Vault対応PDBに接続する前に、Database Vaultロールが共通ユーザーに付与されている場合、PDBへの接続が共通ユーザーに与える影響を十分に理解してください。

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でのOracle Database Vaultの構成および有効化

3.2.7 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に変換します。

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を見つけるには、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。
   DVSYSおよびDVFアカウント(Database Vault用)および LBACSYSアカウント(Label Security用)が存在する場合、Database VaultおよびLabel SecurityがPDBに存在します。

   ```
   SELECT USERNAME FROM DBA_USERS WHERE USERNAME IN ('DVSYS', 'DVF', 'LBACSYS');
   ```

3. Database VaultもLabel Security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 Database Vaultをインストールする前に、Oracle Label Securityをインストールする必要があります。

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、DVSYSをインストールするための表領域として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でOracle Database Vaultを構成して有効化できます。Database Vaultが構成されておらず、CDBで有効になっていない場合は、PDBを閉じてから、CDBルートでDatabase Vaultを構成して有効化する必要があります。Database VaultをPDBで構成して有効化するには、事前にCDBルートで構成して有効化しておく必要があります。CDBルートでDatabase Vaultを構成して有効化し、データベースを再起動した後、PDBを開いてDatabase Vaultを構成して有効化できます。

関連トピック

- マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化

親トピック: マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化

3.3 非マルチテナント環境におけるOracle Database Vaultの登録

ユーザーを登録したら、これらのアカウントを保護するプロファイルを作成する必要があります。

- Database Vaultユーザーの登録
  SQL*Plusを使用してOracle Database Vaultを登録できます。
- DV_OWNERおよびDV_ACCTMGRユーザーを保護するプロファイルの作成
  プロファイルは、DV_OWNERおよびDV_ACCTMGRロールを付与されたユーザーに追加の保護を提供します。

親トピック: Oracle Database Vaultの開始

3.3.1 Database Vaultユーザーの登録

SQL*Plusを使用してOracle Database Vaultを登録できます。

1. ユーザー・アカウントを作成する権限があるユーザーとして、データベース・インスタンスにログインし、CREATE SESSION権限を他のユーザーに付与します。

   たとえば:

   ```
   sqlplus sec_admin
   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される名前付きユーザー・アカウントを指定(または必要に応じて新規名前付きユーザーを作成)します。

   ロールごとに2つのアカウントを作成することをお薦めします。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。

   たとえば:

   ```
   GRANT CREATE SESSION TO sec_admin_owen IDENTIFIED BY password;
   GRANT CREATE SESSION TO dbv_owner_backup IDENTIFIED BY password;
   GRANT CREATE SESSION TO accts_admin_ace IDENTIFIED BY password;
   GRANT CREATE SESSION TO dbv_acctmgr_backup IDENTIFIED BY password;
   ```

   『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

3. SYSDBA管理権限を使用して接続します。

   ```
   CONNECT / AS SYSDBA
   Enter password: password
   ```

4. 2つのバックアップ・ユーザー・アカウントを使用してDatabase Vaultを構成します。

   ```
   BEGIN
    CONFIGURE_DV (
    dvowner_uname         => 'dbv_owner_backup',
    dvacctmgr_uname       => 'dbv_acctmgr_backup');
    END;
   /
   ```

   これらのユーザー・アカウントに、DV_OWNER、DV_ACCTMGRなどの、Database Vaultロールの名前を入力しないでください。

5. utlrp.sqlスクリプトを実行して、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。

6. 先ほど構成したバックアップDatabase Vault所有者ユーザーとして接続します。

たとえば:

```
CONNECT dbv_owner_backup
Enter password: password
```

7. Oracle Database Vaultを有効化します。

```
EXEC DBMS_MACADM.ENABLE_DV;
```

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. 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。

たとえば:

```
CONNECT dbv_owner_backup
Enter password: password
GRANT DV_OWNER TO sec_admin_owen WITH ADMIN OPTION;
```

11. 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。

たとえば:

```
CONNECT dbv_acctmgr_backup
Enter password: password
GRANT DV_ACCTMGR TO accts_admin_ace WITH ADMIN OPTION;
```

12. 構成が成功したことを確認します。

```
CONNECT / AS SYSDBA
SELECT * FROM SYS.DBA_DV_STATUS;
SELECT * FROM DBA_OLS_STATUS;
```

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ロール
- Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン

親トピック: 非マルチテナント環境におけるOracle Database Vaultの登録

3.3.2 DV_OWNERおよびDV_ACCTMGRユーザーを保護するプロファイルの作成

プロファイルは、DV_OWNERおよびDV_ACCTMGRロールを付与されたユーザーに追加の保護を提供します。

DV_OWNERまたはDV_ACCTMGRロールを付与されたデータベース・ユーザーは、クリティカルな特権アカウントとみなされます。通常、これらのアカウントはサービス・アカウントとみなしてパスワードのロックアウト要件が適用されないようにする必要があります。Oracleでは、アカウントがロックされないカスタム・プロファイルを作成することをお薦めします。また、これらのDatabase Vault関連のアカウントの失敗したログイン試行の監査も必要です。

1. CREATE PROFILEシステム権限を持つユーザーとして、データベース・インスタンスにログインします。
2. 次のようなプロファイルを作成します。

```
CREATE PROFILE dv_profile limit
FAILED_LOGIN_ATTEMPTS UNLIMITED
PASSWORD_VERIFY_FUNCTION ORA12C_VERIFY_FUNCTION
PASSWORD_LOCK_TIME UNLIMITED
CONTAINER=CURRENT;
```

3. このプロファイルを使用するために、DV_OWNERおよびDV_ACCTMGRユーザー・アカウントを更新します。

   たとえば:

```
ALTER USER sec_admin_owen PROFILE dv_profile;
ALTER USER dbv_owner_backup PROFILE dv_profile;
ALTER USER accts_admin_ace PROFILE dv_profile;
ALTER USER dbv_acctmgr_backup PROFILE dv_profile;
```

4. AUDIT_ADMINロールを付与されたユーザーとして接続します。
5. 統合監査ポリシーを作成して有効化し、DV_OWNERまたはDV_ACCTMGRロールを付与されたユーザーによる失敗したログインを追跡します。

   たとえば:

```
CREATE AUDIT POLICY dv_logins ACTIONS LOGON;
AUDIT POLICY dv_logins BY USERS WITH GRANTED ROLES DV_OWNER, DV_ACCTMGR
WHENEVER NOT SUCCESSFUL;
```

関連トピック

- 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: 非マルチテナント環境におけるOracle Database Vaultの登録

3.4 Oracle Real Application Clusters環境でのOracle Database Vaultの構成および有効化

各Oracle RACノードを含むOracle Real Application Clusters (Oracle RAC)環境用にOracle Database Vaultを構成できます。

Oracle RAC環境にOracle Database Vaultを構成するには、1つのノードでOracle Database Vaultを構成して有効化し、各インスタンス・ノードを再起動して、すべてのノードで有効にする必要があります。次の手順では、ノードごとに別個のOracleホームがあると仮定します。

1. CDBルートでOracle Database Vaultを構成して有効化します。
2. SYSDBA管理権限を持つユーザーSYSとして、データベース・インスタンスにログインします。

```
sqlplus sys as sysdba
Enter password: password
```

3. 各Oracle RACノードで次のALTER SYSTEM文を実行します。

```
ALTER SYSTEM SET AUDIT_SYS_OPERATIONS=TRUE SCOPE=SPFILE; -- For non-unified
auditing environments
ALTER SYSTEM SET OS_ROLES=FALSE SCOPE=SPFILE;
ALTER SYSTEM SET RECYCLEBIN='OFF' SCOPE=SPFILE;
ALTER SYSTEM SET REMOTE_LOGIN_PASSWORDFILE='EXCLUSIVE' SCOPE=SPFILE;
ALTER SYSTEM SET SQL92_SECURITY=TRUE SCOPE=SPFILE;
```

4. Oracle Databaseを再起動します。

```
CONNECT / AS SYSOPER
Enter password: password
SHUTDOWN IMMEDIATE
STARTUP
```

関連トピック

- CDBルートでのDatabase Vaultの構成および有効化

親トピック: Oracle Database Vaultの開始

3.5 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Database Vaultステータス、またはマルチテナント環境においてルートのみか個々のPDBのDatabase Vaultステータスを検索する場合は、データベースにどのユーザーとして接続しているかに応じて、DBA_DV_STATUSまたはSYS.DBA_DV_STATUSビューを問い合せます。次に例を示します。
    - DBAロールまたはSYSDBA管理権限を持つユーザーとして接続している場合:

      ```
      SELECT * FROM DBA_DV_STATUS;
      ```

    - DV_OWNERロールまたはDV_ADMINロールを持つユーザーとして接続している場合:

      ```
      SELECT * FROM SYS.DBA_DV_STATUS;
      ```

    次のような出力が表示されます。

    ```
    NAME                 STATUS
    -------------------- -----------
    DV_APP_PROTECTION    NOT CONFIGURED
    DV_CONFIGURE_STATUS  TRUE
    DV_ENABLE_STATUS     TRUE
    ```

DV_APP_PROTECTIONは、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PDBローカル・データへのアクセスを自動的に制限する操作制御を指します。

- 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

関連トピック

- 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

親トピック: Oracle Database Vaultの開始

3.6 Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン

Oracle Enterprise Manager Cloud Control (Cloud Control)には、Oracle Database Vaultの管理用のページが用意されています。

Oracle EM Expressではなく、Oracle Enterprise Manager Cloud Controlのみがサポートされています。Oracle Database Vaultページを使用して、Database Vaultで保護されたデータベースを一元的なコンソールから管理および監視できます。このコンソールでは、アラートの自動化、Database Vaultレポートの表示、およびDatabase Vaultで保護されたその他のデータベースへのDatabase Vaultポリシーの伝播が可能です。
ログインする前に、Oracle Enterprise Managerのオンライン・ヘルプに従って、Database Vaultとともに使用するCloud Controlターゲット・データベースを構成していることを確認してください。また、OracleデータベースでOracle Database Vaultを構成して有効化することも必要です。

1. Cloud Control管理者から提供された資格証明を使用してOracle Enterprise Manager Cloud Controlにログインします。
2. Cloud Controlのホーム・ページで、「ターゲット」メニューから「データベース」を選択します。
3. 「データベース」ページで、接続先のOracle Database Vaultで保護されているデータベースのリンクを選択します。

   データベースのホームページが表示されます。

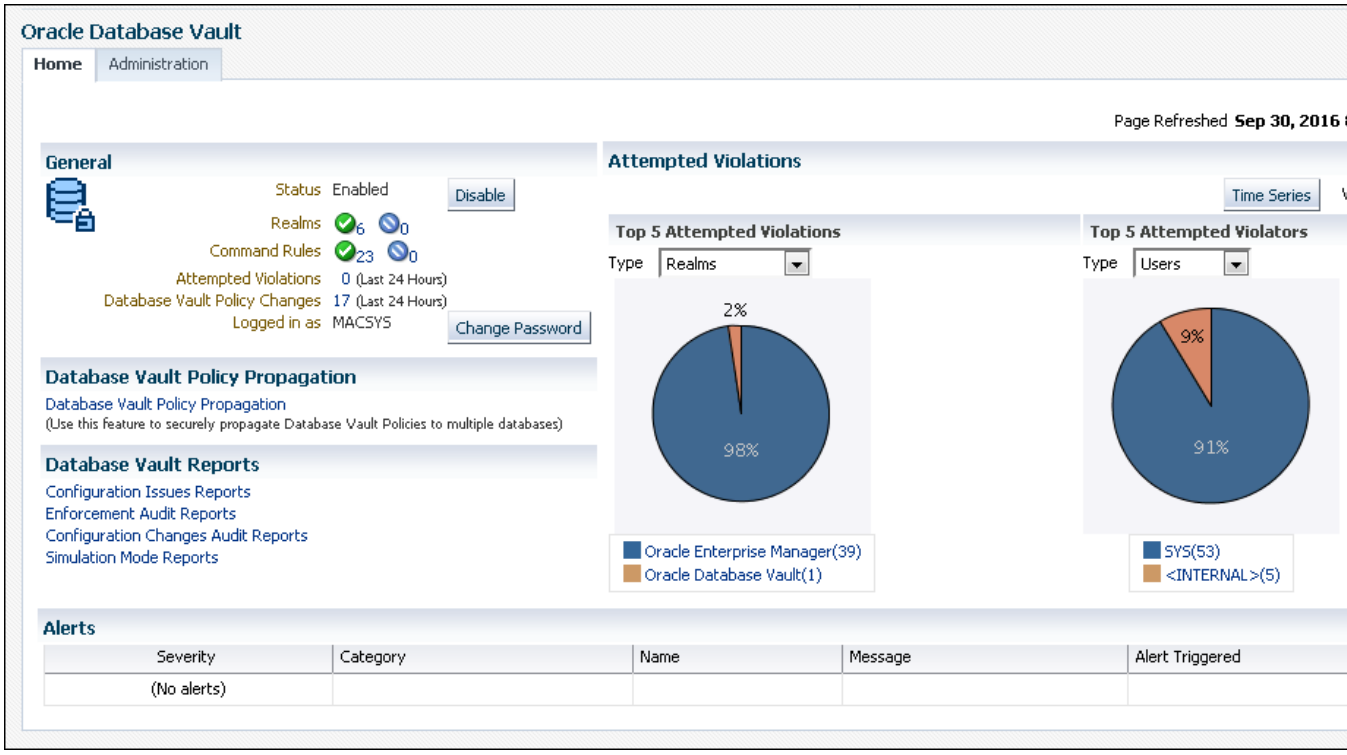
4. 「セキュリティ」メニューから、「Database Vault」を選択します。

   「データベース・ログイン」ページが表示されます。

5. 次の情報を入力します。
   - ユーザー名: 適切なOracle Database Vaultロールを付与されているユーザーの名前を入力します。
     - Database Vaultポリシーの作成および伝播: DV_OWNERまたはDV_ADMINロール、SELECT ANY DICTIONARY権限
     - Database Vaultアラートおよびレポートの表示: DV_OWNER、DV_ADMINまたはDV_SECANALYSTロール、SELECT ANY DICTIONARY権限
   - パスワード: パスワードを入力します。

- ロール：リストから「通常」を選択します。
- 別名保存：次回このページが表示されるときに、これらの資格証明が自動入力されているようにするには、このチェック・ボックスを選択します。資格証明は、Enterprise Managerに安全な方法で格納されます。これらの資格証明へのアクセスは、現在ログインしているユーザーによって異なります。

Database Vaultホームページが表示されます。



関連トピック

- Oracle Database Vaultロールについて
- Oracle Database VaultのOracle Enterprise Managerとの使用

親トピック: Oracle Database Vaultの開始

3.7 クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護

このチュートリアルでは、HRスキーマの周辺でレルムを作成する方法を示します。

- このチュートリアルについて
  このチュートリアルでは、Oracle Database Vault PL/SQLパッケージを使用することで、HRサンプル・データベース・スキーマの周辺でレルムを作成します。
- ステップ1: SYSTEMとしてログインしHRスキーマにアクセスする
  このチュートリアル用にHRスキーマを有効にする必要があります。
- ステップ2: レルムの作成
  レルムでは、1つ以上のスキーマ、個々のスキーマ・オブジェクトおよびデータベース・ロールを保護できます。
- ステップ3: SEBASTIANユーザー・アカウントの作成
  この時点では、レルムが保護するデータベース・オブジェクトにアクセスする、またはそれらを操作するデータベース・アカウントもロールもありません。
- ステップ4: ユーザーSEBASTIANによるレルムのテスト

この段階で、ユーザーSEBASTIANは、HR.EMPLOYEES表を問い合せることでレルムをテストできます。

- ステップ5: レルムの認可の作成
  次に、HR.EMPLOYEES表にアクセスできるよう、ユーザーSEBASTIANにHR Appsレルムへの認可を与える必要があります。
- ステップ6: レルムのテスト
  レルムをテストするには、HR以外のユーザーとしてEMPLOYEES表にアクセスを試みる必要があります。
- ステップ7: 統合監査が有効ではない場合のレポートの実行
  HR Appsレルムの失敗時の監査を有効にしたため、レポートを生成してセキュリティ違反を検出できます。
- ステップ8: このチュートリアルのコンポーネントの削除
  コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

親トピック: Oracle Database Vaultの開始

3.7.1 このチュートリアルについて

このチュートリアルでは、Oracle Database Vault PL/SQLパッケージを使用することで、HRサンプル・データベース・スキーマの周辺でレルムを作成します。

HRスキーマのEMPLOYEES表には、管理権限を使用したアクセスも含め、企業内のほとんどの社員に公開しない給与などの情報が含まれています。これを実現するには、HRスキーマをデータベース内の保護ゾーン(Oracle Database Vaultではレルムと呼ぶ)のセキュア・オブジェクトに追加します。そして、このレルムに制限付きの認可を付与します。その後、レルムをテストして適切に保護されていることを確認します。最後に、レポートを実行し、レルムをテストする際に試行するような疑わしいアクティビティの監査証跡をOracle Database Vaultがどのように作成するかを確認します。

親トピック: クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護

3.7.2 ステップ1: SYSTEMとしてログインしHRスキーマにアクセスする

このチュートリアル用にHRスキーマを有効にする必要があります。

このチュートリアルを開始する前に、HRサンプル・スキーマがインストールされていることを確認してください。Oracle Database サンプル・スキーマは、サンプル・スキーマのインストール方法を説明しています。

1. DBAロールを付与されたユーザーとしてデータベース・インスタンスにログインしてから、HRスキーマにアクセスします。

   たとえば:

   ```
   sqlplus system
   Enter password: password
   ```

2. マルチテナント環境で、適切なPDBに接続します。

   たとえば:

   ```
   CONNECT SYSTEM@my_pdb
   Enter password: password
   ```

   使用可能なPDBを見つけるには、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。

3. HR.EMPLOYEES表に次のように問い合せます。

   ```
   SELECT FIRST_NAME, LAST_NAME, SALARY FROM HR.EMPLOYEES WHERE ROWNUM < 10;
   ```

次のような出力が表示されます。

```
FIRST_NAME           LAST_NAME                      SALARY
-------------------- ------------------------- ----------
Steven               King                            24000
Neena                Kochhar                         17000
Lex                  De Haan                         17000
Alexander            Hunold                           9000
Bruce                Ernst                            6000
David                Austin                           4800
Valli                Pataballa                        4800
Diana                Lorentz                          4200
Nancy                Greenberg                       12008
9 rows selected.
```

4. HRスキーマがロックされて無効になっている場合は、DV_ACCTMGRユーザーとしてデータベース・インスタンスにログインし、アカウントをロック解除して有効にします。たとえば:

```
sqlplus accts_admin_ace -- For a multitenant environment, sqlplus
bea_dvacctmgr@hrpdb
Enter password: password
ALTER USER HR ACCOUNT UNLOCK IDENTIFIED BY password
```

『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

この例からわかるように、SYSTEMにはHRスキーマのEMPLOYEES表の給与情報へのアクセス権があります。SYSTEMには、SELECT ANY TABLEシステム権限を含むDBAロールが自動的に付与されるためです。

5. SQL*Plusを終了しないでください。

親トピック: クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護

3.7.3 ステップ2: レルムの作成

レルムでは、1つ以上のスキーマ、個々のスキーマ・オブジェクトおよびデータベース・ロールを保護できます。

レルムを作成したら、レルム内のスキーマやスキーマ・オブジェクトに適用するセキュリティ制限を作成します。HRスキーマにレルムを作成する必要があります。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「レルム」をクリックします。
3. Oracle Database Vault Administratorの「レルム」ページで、「作成」をクリックします。
4. 「レルムの作成」ページの「一般」で、「名前」の後にHR Appsと入力します。
5. 「説明」フィールドに、Realm to protect the HR schemaと入力します。
6. 「必須レルム」チェック・ボックスの選択を解除したままにします。
7. 「ステータス」に「有効」が選択されていて、そのレルムが使用可能になっていることを確認します。
8. 「監査オプション」で、「失敗時に監査」が選択されていて、後から監査証跡を作成できることを確認します。
9. 「次へ」をクリックして、「レルム・セキュア・オブジェクト」ページを表示します。
10. 「追加」ボタンをクリックして、「セキュア・オブジェクトの追加」ダイアログ・ボックスで次の情報を入力します。
    - 所有者: HRと入力してHRスキーマを選択します。
    - オブジェクト・タイプ: TABLEと入力します。

- オブジェクト名: EMPLOYEESと入力します。

11. 「OK」をクリックします。

    HR.EMPLOYEES表は、「レルムの作成 : レルム・セキュア・オブジェクト」ページに追加されます。

12. 「完了」、「終了」の順にクリックします。

この段階で、レルムが作成されていますが、それに対する認可は割り当てられていません。このチュートリアルで後ほど対応します。

親トピック: クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護

3.7.4 ステップ3: SEBASTIANユーザー・アカウントの作成

この時点では、レルムが保護するデータベース・オブジェクトにアクセスする、またはそれらを操作するデータベース・アカウントもロールもありません。

そのため、このステップでデータベース・アカウントまたはデータベース・ロールを認可し、レルム内のスキーマにアクセスできるようにします。SEBASTIANユーザー・アカウントを作成します。

1. DV_ACCTMGRロールを持つDatabase Vaultアカウント・マネージャとしてSQL*Plusに接続し、ローカル・ユーザーSEBASTIANを作成します。

   たとえば:

   ```
   CONNECT accts_admin_ace -- Or, CONNECT accts_admin_ace@hrpdb
   Enter password: password
   GRANT CREATE SESSION TO SEBASTIAN IDENTIFIED BY password;
   ```

   passwordを安全なパスワードに置き換えます。パスワードを作成するための最小限の要件は、『Oracle Databaseセキュリティ・ガイド』を参照してください。

2. SYSDBA権限を持つSYSとして接続し、SEBASTIANに次の追加権限を付与します。

   ```
   CONNECT SYS AS SYSDBA -- Or, CONNECT SYS@hrpdb AS SYSDBA
   Enter password: password
   GRANT READ ANY TABLE TO SEBASTIAN;
   ```

(「ステップ6: レルムのテスト」でレルムをテストする際に必要なため、SQL*Plusを終了しないでください。)

親トピック: クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護

3.7.5 ステップ4: ユーザーSEBASTIANによるレルムのテスト

この段階で、ユーザーSEBASTIANは、HR.EMPLOYEES表を問い合せることでレルムをテストできます。

1. ユーザーSEBASTIANとして接続します。

   ```
   CONNECT sebastian
   Enter password: password
   ```

2. HR.EMPLOYEES表を問い合せます。

   ```
   SELECT COUNT(*) FROM HR.EMPLOYEES;
   ```

   次の出力が表示されます。

   ```
   ERROR at line 1:
   ORA-01031: insufficient privileges
   ```

ユーザーSEBASTIANはREAD ANY TABLEシステム権限がありますが、HR.EMPLOYEES表を問合せできません。これは、HR AppsレルムがREAD ANY TABLEシステム権限よりも優先されるためです。

親トピック: クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護

3.7.6 ステップ5: レルムの認可の作成

次に、HR.EMPLOYEES表にアクセスできるよう、ユーザーSEBASTIANにHR Appsレルムへの認可を与える必要があります。

この認可により、SEBASTIANは、このレルムによって保護されているHR.EMPLOYEES表に対するREAD ANY TABLEシステム権限を使用できます。

1. Database Vault Administratorの「レルム」ページで、レルムのリストから「HR Apps」を選択し、「編集」をクリックします。
2. 「レルム認可」ページになるまで、「次へ」ボタンをクリックします。
3. 「追加」をクリックし、「認可の追加」ダイアログ・ボックスに次の情報を入力します。
   - レルム認可の権限受領者: SEBASTIANと入力します。
   - レルム認可タイプ: リストから「参加者」を選択します。
   - レルム認可ルールセット: このフィールドは空白のままにします。
4. 「OK」をクリックします。

   「参加者」認可により、HR Appsレルム内のユーザーSEBASTIANは、HR Appsレルムで保護されるオブジェクトのアクセス、操作および作成を管理できるようになります。この場合、EMPLOYEES表の表示を許可されているのは、HRユーザーおよびSEBASTIANのみです。
5. 「完了」、「終了」の順にクリックします。

親トピック: クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護

3.7.7 ステップ6: レルムのテスト

レルムをテストするには、HR以外のユーザーとしてEMPLOYEES表にアクセスを試みる必要があります。

SYSTEMアカウントには通常、SELECT ANY TABLE権限があるため、HRスキーマのすべてのオブジェクトに対するアクセス権がありますが、この場合はOracle Database Vaultを使用してEMPLOYEES表を保護しているため、アクセス権はありません。

1. SQL*Plusで、SYSTEMとして接続します。

   ```
   CONNECT SYSTEM -- Or, CONNECT SYSTEM@hrpdb
   Enter password: password
   ```

2. EMPLOYEES表の任意の行を再度問い合せてみます。

   たとえば:

   ```
   SELECT FIRST_NAME, LAST_NAME, SALARY FROM HR.EMPLOYEES WHERE ROWNUM <10;
   ```

   次の出力が表示されます。

   ```
   Error at line 1:
   ORA-01031: insufficient privileges
   ```

   SYSTEMにはEMPLOYEES表へのアクセス権はなくなります。(実際に、ユーザーSYSでもこの表にはアクセスできません。)ただし、ユーザーSEBASTIANはHR Appsレルムの認可された参加者であるため、SEBASTIANにはこの情報へ

のアクセス権があります。

3. ユーザーSEBASTIANとして接続します。

```
CONNECT sebastian -- Or, CONNECT sebastian@hrpdb
Enter password: password
```

4. 次の問合せを実行します。

```
SELECT FIRST_NAME, LAST_NAME, SALARY FROM HR.EMPLOYEES WHERE ROWNUM <10;
```

次のような出力が表示されます。

```
FIRST_NAME           LAST_NAME                      SALARY
-------------------- ------------------------- ----------
Steven               King                            24000
Neena                Kochhar                         17000
Lex                  De Haan                         17000
Alexander            Hunold                           9000
Bruce                Ernst                            6000
David                Austin                           4800
Valli                Pataballa                        4800
Diana                Lorentz                          4200
Nancy                Greenberg                       12008
9 rows selected.
```

親トピック: クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護

3.7.8 ステップ7: 統合監査が有効ではない場合のレポートの実行

HR Appsレルムの失敗時の監査を有効にしたため、レポートを生成してセキュリティ違反を検出できます。

たとえば、「ステップ6: レルムのテスト」で試行した違反のレポートを生成できます。

1. ユーザーSYSTEMとしてSQL*Plusに接続し、統合監査が有効ではないことを確認します。

```
CONNECT SYSTEM -- Or, CONNECT SYSTEM@hrpdb
Enter password: password
SQL> SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Unified Auditing';
```

VALUEがTRUEを戻す場合、この項を完了できません。「ステップ8: このチュートリアルのコンポーネントの削除」に進みます。

統合監査が有効な場合、イベントを取得する統合監査ポリシーを作成する必要があります。Oracle Database Vaultの統合監査ポリシーの作成方法については、『Oracle Databaseセキュリティ・ガイド』を参照してください。

2. 「Database Vault Administrator」ページで、「ホーム」をクリックしてホーム・ページを表示します。
3. Database Vaultの「ホーム」ページの「レポート」で、「Database Vaultレポート」を選択します。
4. 「Database Vaultレポート」ページで、「Database Vault強制監査レポート」を選択します。
5. Database Vault監査レポートリストで、「レルム監査レポート」を選択します。
6. 「検索」領域で、「コマンド」メニューから「等しい」を選択し、テキスト・フィールドにSELECTと入力します。次に「検索」をクリックします。

「検索」リージョンの後の表にレポートが表示されます。

7. 「OK」をクリックしてレポートを終了します。

Oracle Database Vaultにより、違反の種類(この場合は前の項目で入力したSELECT文)、発生した時間と場所、違反を試みたユーザーのログイン・アカウント、および違反の内容などを示すレポートが生成されます。

3.7.9 ステップ8: このチュートリアルのコンポーネントの削除

コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

1. ユーザーSEBASTIANを削除します。

   SQL*Plusで、次のようにOracle Database Vaultアカウント・マネージャ(たとえばaccts_admin_ace)としてログインし、SEBASTIANを削除します。

   ```
   sqlplus accts_admin_ace -- Or, CONNECT bea_dvacctmgr@hrpdb
   Enter password: password
   DROP USER SEBASTIAN;
   ```

2. HR Appsレルムを削除します。

   a. Cloud Controlで、DV_OWNERロールがあるユーザーとしてログインしていることを確認します。

   b. Database Vaultの「ホーム」ページで、「管理」をクリックします。

   c. 「レルム」ページで、レルムのリストからHR Appsを選択します。

   d. 「削除」をクリックして、「確認」ウィンドウで「はい」をクリックします。

3. 必要な場合は、SQL*Plusで、HRアカウントをロックし、無効にします。

   a. DV_ACCTMGRロールを持つユーザーとして接続します(ユーザーaccts_admin_aceなど)。

   b. 次のALTER USER文を実行します。

   ```
   ALTER USER HR ACCOUNT LOCK PASSWORD EXPIRE;
   ```

4 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。

- レルムの概要
  レルムを使用すると、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。
- デフォルト・レルム
  Oracle Database Vaultには、Database VaultおよびSYS関連のスキーマ、システム権限とオブジェクト権限、ロールおよび監査関連オブジェクトを保護するためのデフォルト・レルムが用意されています。
- レルムの作成
  レルムの保護を有効にするには、レルムを作成して、レルム・セキュア・オブジェクト、ロールおよび認可を含めるようにレルムを構成します。
- レルム・セキュア・オブジェクトについて
  レルム・セキュア・オブジェクトにより、レルムによって保護されるテリトリ(一連のスキーマおよびデータベースのオブジェクト、およびロール)が定義されます。
- レルム認可について
  レルム認可では、レルムで保護されているオブジェクトの管理またはアクセスを実行する一連のデータベース・アカウントおよびロールを決定します。
- 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では、共通レルム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。
- レルムの有効化ステータスの変更
  Enterprise Manager Cloud Controlから、レルムを無効または有効にすることや、シミュレーション・モードを使用するようレルムを設定することができます。
- レルムの削除
  Enterprise Manager Cloud Controlを使用して、レルムを削除できます。
- レルムの動作
  適切な権限を持つデータベース・アカウントにより、レルム内のオブジェクトに影響するSQL文が発行されると、特定のアクティビティ・セットが発生します。
- レルムでの認可の動作
  ユーザーに正しい権限がない場合、レルム認可により、そのユーザーによるアクティビティの実行が阻止されます。
- レルムで保護されたオブジェクトへのアクセス
  レルムでオブジェクトを保護できますが、このレルムで保護されているオブジェクトに含まれるオブジェクトへのアクセスは可能です。
- レルムの動作の例
  レルムにより、同じ権限のある2人のユーザーに、オブジェクトに対する別々のアクセス・レベルが必要な場合に保護を提供できます。
- その他のOracle Database Vaultコンポーネントへのレルムの影響
  レルムはファクタ、アイデンティティおよびルール・セットには影響しませんが、コマンド・ルールには影響します。
- レルム設計のガイドライン
  Oracleでは、一連のレルム設計のガイドラインを提供しています。
- レルムのパフォーマンスへの影響
  レルムは、DDLおよびDML操作などの様々な状況で、データベース・パフォーマンスに影響します。

- レルム関連のレポートおよびデータ・ディクショナリ・ビュー
  Oracle Database Vaultには、レルムの分析に役立つ、レポートとデータ・ディクショナリ・ビューが用意されています。

4.1 レルムの概要

レルムを使用すると、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。

- レルムについて
  レルムは、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。
- 必須レルムによるレルム内のオブジェクトへのユーザー・アクセスの制限
  デフォルトでは、オブジェクト権限を所有または保持するユーザーには、明示的なレルム認可を受けずにレルム保護オブジェクトへのアクセスが許可されます。
- 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ルムを作成できます。
- レルムで保護できるオブジェクト・タイプ
  特定のオブジェクト・タイプのスキーマ内のすべてのオブジェクトの周りにレルムを作成できます。

親トピック: レルムの構成

4.1.1 レルムについて

レルムは、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。

レルムは、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。レルムおよびコマンド・ルールの両方で、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つのタイプのレルムがあります。どちらのタイプのレルムでも、スキーマ全体、個々のデータベース・ロール、またはスキーマ内の重要なオブジェクト(表や索引)を選択的に保護できます。通常レルムの場合、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ルム認可なしで問合せやDML操作を実行できますが、DDL操作の実行にはレルム認可が必要です。必須レルムの場合、レルム内のオブジェクトにさらに厳格な保護が適用されます。必須レルムの場合、オブジェクト権限とシステム権限の両方へのアクセスがブロックされ、オブジェクト権限を持つユーザーが、レルム認可なしで問合せ、DMLまたはDDL操作を実行することは許可されません。つまり、オブジェクトが必須レルムで保護されている場合、オブジェクト所有者であっても、適切なレルム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、標準レルムと必須レルムのどちらも、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。ユーザーにレルムに対する権限がある場合、ユーザーは、レルム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、レルム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。

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ルムを作成できます。レルム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に作成するレルム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ルム違反のみを記録できます。これにより、Database Vaultレルム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、レルムを構成できます。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、レルムを構成できます。

親トピック: レルムの概要

4.1.2 必須レルムによるレルム内のオブジェクトへのユーザー・アクセスの制限

デフォルトでは、オブジェクト権限を所有または保持するユーザーには、明示的なレルム認可を受けずにレルム保護オブジェクトへのアクセスが許可されます。

レルムを必須レルムになるように構成することで、これらのユーザーがアクセスできないようにオプションでレルムを構成できます。必須レルムは、システム権限に基づくアクセスとオブジェクト権限に基づくアクセスの両方をブロックします。つまり、オブジェクト所有者は、レルムへのアクセスが認可されない場合、アクセスできません。ユーザーは、ユーザーまたはロールがレルムにアクセスする認可を受けている場合のみ、必須レルムのセキュア・オブジェクトにアクセスできます。

必須レルムには、その他に次の特徴があります。

- ロールが必須レルムで保護される場合、レルム所有者を除き、保護されたロールに対する権限の付与や取消しを行うことはできません。
- 以前のリリースで作成した通常のレルムを更新して必須レルムにすることができます。こうすると、所有者のアクセスをブロックし、オブジェクト権限ユーザーがレルム保護オブジェクトにアクセスしないようにできます。
- SYS所有オブジェクトは、データ・ディクショナリ保護によってすでに保護されており、Oracle Database Vaultによって個別に保護されてはいません。

必須レルムには、次の利点があります。

- 必須レルムは、オブジェクト所有者およびオブジェクト権限ユーザーをブロックできます。以前のリリースでは、複雑なコマンド・ルールを定義しないとこれらのユーザーをブロックできませんでした。
- 必須レルムを使用すると、アクセス制御を構成する柔軟性が高くなります。たとえば、ユーザーが特定の条件(日中の特定の時間範囲内など)でオブジェクトにアクセスできるようにするとします。レルムはオブジェクト権限をブロックしないため、オブジェクト権限をそのユーザーに付与できません。システム権限のみをユーザーに付与し、ルールを使用してこのユーザーをレルムに対して認可するか、コマンド上でコマンド・ルールを直接作成できます。これらのソリューションは、計算上コストの面でかなり高額になったり、システム権限などの権限をユーザーに過度に付与する必要があるため望ましい方法ではありません。必須レルムの場合、特定の条件に関するルールとともにオブジェクト権限をユーザーに付与し、このユーザーをレルム所有者または参加者として認可するだけで済みます。そのため、必須レルムを使用する場合、ユーザーに過度の権限が付与されず、Oracle Database Vaultポリシーの柔軟性が増します。
- 必須レルムにより、パッチのアップグレード時に保護レイヤーが追加されます。パッチのアップグレード時に、データベース管理者は、オブジェクトにパッチを実行するためにレルム保護オブジェクトへの直接アクセス権が必要になります。社会保障番号などの機密データが含まれる表がある場合、パッチのアップグレード時に必須レルムを使用して、管理者がこ

れらの表にアクセスしないようにすることができます。パッチ適用が完了し、データベース管理者がオブジェクトにアクセスする必要がなくなったら、必須レルムの保護を無効化し、通常のアプリケーション・レルム保護を再び有効化して、アプリケーション保護を通常の状態に戻すことができます。

- 必須レルムを使用してランタイムで表を保護できます。ランタイムで、アプリケーション・データを多くの表に格納できます。データの整合性や正確性が維持されるように、ランタイム・スキーマなどの単一ユーザーがこれらの表にアクセスする方法が適しています。アプリケーション・データが多くの異なるスキーマに分散する場合、スキーマ所有者とオブジェクト権限を持つユーザーは、データベースに直接ログインするとデータを変更できます。ユーザーがランタイム・スキーマのプロシージャを実行せずにこれらの表を更新できないように、必須レルムを使用して表を保護し、認可されたユーザーのプロシージャのみがアクセスするようにできます。通常のレルムはオブジェクト所有者とオブジェクト権限ユーザーをブロックしないため、必須レルムを使用してこれらをブロックできます。このように、認可されたユーザーのみがランタイムでこれらの表にアクセスできます。
- 構成済のロールに対する変更を禁止することで、セキュリティ設定を固定できます。

同じオブジェクトに必須レルムが複数ある場合、保護されたオブジェクトにアクセスするには、すべての必須レルムに対してユーザーまたはロールを認可する必要があります。

関連トピック

- CREATE_REALMプロシージャ
- UPDATE_REALMプロシージャ

親トピック: レルムの概要

4.1.3 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ルムを作成できます。

個々のプラガブル・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ルムを作成するのでなく、アプリケーション・ルートでレルムを作成する利点は、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、共通レルムは作成できません。

Database Vaultの共通レルムは、通常レルムか必須レルムのどちらかにできます。このレルムは、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。

共通レルムを構成するには、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。共通レルムの共通認可を付与するには、アプリケーション・ルートにいる必要があります。アプリケーション・ルートに関連付けられているPDBにレルムを伝播するには、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

- レルム認可について

親トピック: レルムの概要

4.1.4 レルムで保護できるオブジェクト・タイプ

特定のオブジェクト・タイプのスキーマ内のすべてのオブジェクトの周りにレルムを作成できます。

これらのオブジェクト・タイプは、次のとおりです。

オブジェクト・タイプC-J	オブジェクト・タイプL-P	オブジェクト・タイプR-V
CLUSTER	LIBRARY	ROLE
DIMENSION	MATERIALIZED VIEW	SEQUENCE
FUNCTION	MATERIALIZED VIEW LOG	SYNONYM
INDEX	OPERATOR	TABLE
INDEX PARTITION	PACKAGE	TRIGGER
INDEXTYPE	PROCEDURE	TYPE
JOB	PROGRAM	VIEW

親トピック: レルムの概要

4.2 デフォルトのレルム

Oracle Database Vaultには、Database VaultおよびSYS関連のスキーマ、システム権限とオブジェクト権限、ロールおよび監査関連オブジェクトを保護するためのデフォルト・レルムが用意されています。

デフォルトのレルムで保護されているタスクをユーザーが実行できるように、ユーザーをレルムに追加できます。

- Oracle Database Vaultレルム
  Oracle Database Vaultレルムは、Oracle Database VaultのDVSYS、DVFおよびLBACSYSの各スキーマの構成およびロール情報を保護します。
- Database Vaultアカウント管理レルム
  Database Vaultアカウント管理レルムは、データベース・アカウントとデータベース・プロファイルを管理および作成する管理者のレルムを定義します。
- Oracle Enterprise Managerレルム
  Oracle Database Vaultには、Oracle Enterprise Managerモニタリング・アカウント専用のレルムが用意されています。
- Oracleデフォルト・スキーマ保護レルム
  Oracleデフォルト・スキーマ保護レルムは、Oracle TextなどOracleの機能で使用されるロールおよびスキーマを保護します。
- Oracleシステム権限およびロール管理レルム
  Oracleシステム権限およびロール管理レルムは、Oracleデータベース内にあるすべてのOracle提供ロールを保護します。
- Oracleデフォルト・コンポーネント保護レルム
  Oracleデフォルト・コンポーネント保護レルムは、SYSTEMスキーマとOUTLNスキーマを保護します。

親トピック: レルムの構成

4.2.1 Oracle Database Vaultレルム

Oracle Database Vaultレルムは、Oracle Database VaultのDVSYS、DVFおよびLBACSYSの各スキーマの構成およびロール情報を保護します。

DVSYS、DVFおよびLBACSYSの3つのすべてのスキーマの所有者は、このレルムの所有者です。

このレルムで保護されているオブジェクトを検索するには、次の問合せを実行します。

```
SELECT OWNER, OBJECT_NAME, OBJECT_TYPE
FROM DBA_DV_REALM_OBJECT
WHERE REALM_NAME = 'Oracle Database Vault Realm'
ORDER BY OWNER, OBJECT_NAME;
```

レルム認可ユーザー、その(参加者または所有者としての)ロール、およびOracle Database Vaultルール・セットが認可ユーザーに適用されているかどうかを検索するには、次の問合せを実行します。

```
SELECT GRANTEE, AUTH_OPTIONS, AUTH_RULE_SET_NAME
FROM DBA_DV_REALM_AUTH
WHERE REALM_NAME = 'Oracle Database Vault Realm'
ORDER BY GRANTEE;
```

関連トピック

- Oracle Database Vaultスキーマ

親トピック: デフォルトのレルム

4.2.2 Database Vaultアカウント管理レルム

Database Vaultアカウント管理レルムは、データベース・アカウントとデータベース・プロファイルを管理および作成する管理者のレルムを定義します。

このレルムの所有者は、ユーザーに対してCREATE SESSION権限の付与と取消しを行うことができます。

このレルムが保護するオブジェクトを検索するには、次の問合せを実行します。

```
SELECT OWNER, OBJECT_NAME, OBJECT_TYPE
FROM DBA_DV_REALM_OBJECT
WHERE REALM_NAME = 'Database Vault Account Management'
ORDER BY OWNER, OBJECT_NAME;
```

レルム認可ユーザー、その参加者としてのロールまたは所有者としてのロールを検索する際に、Oracle Database Vaultルール・セットが認可ユーザーに適用されている場合は、次の問合せを実行します。

```
SELECT GRANTEE, AUTH_OPTIONS, AUTH_RULE_SET_NAME
FROM DBA_DV_REALM_AUTH
WHERE REALM_NAME = 'Database Vault Account Management'
ORDER BY GRANTEE;
```

関連トピック

- DV_ACCTMGR Database Vaultアカウント・マネージャ・ロール

親トピック: デフォルトのレルム

4.2.3 Oracle Enterprise Managerレルム

Oracle Database Vaultには、Oracle Enterprise Managerモニタリング・アカウント専用のレルムが用意されています。

Oracle Enterprise Managerレルムは、監視と管理に使用されるOracle Enterprise Managerアカウント(DBSNMPユーザーおよび OEM_MONITORロール)を保護します。

このレルムが保護するオブジェクトを検索するには、次の問合せを実行します。

```
SELECT OWNER, OBJECT_NAME, OBJECT_TYPE
FROM DBA_DV_REALM_OBJECT
WHERE REALM_NAME = 'Oracle Enterprise Manager'
ORDER BY OWNER, OBJECT_NAME;
```

レルム認可ユーザー、その参加者としてのロールまたは所有者としてのロールを検索する際に、Oracle Database Vaultルール・セットが認可ユーザーに適用されている場合は、次の問合せを実行します。

```
SELECT GRANTEE, AUTH_OPTIONS, AUTH_RULE_SET_NAME
FROM DBA_DV_REALM_AUTH
WHERE REALM_NAME = 'Oracle Enterprise Manager'
ORDER BY GRANTEE;
```

関連トピック

- Oracle Database VaultのOracle Enterprise Managerとの使用

親トピック: デフォルトのレルム

4.2.4 Oracleデフォルト・スキーマ保護レルム

Oracleデフォルト・スキーマ保護レルムは、Oracleの機能(Oracle Textなど)で使用されるロールやスキーマを保護します。

このグループ分けには、Oracle SpatialスキーマMDSYS、MDDATA)がOracle Text(CTXSYS)で拡張して使用されるメリットと、Oracle OLAPがコアOracle Databaseカーネル機能ではなくアプリケーションになるメリットがあります。

Oracleデフォルト・スキーマ保護レルムでは、いくつかのロールとスキーマが保護されます。

- このレルムが保護するオブジェクトを検索するには、次の問合せを実行します。

  ```
  SELECT OWNER, OBJECT_NAME, OBJECT_TYPE
  FROM DBA_DV_REALM_OBJECT
  WHERE REALM_NAME = 'Oracle Default Schema Protection Realm'
  ORDER BY OWNER, OBJECT_NAME;
  ```

- レルム認可ユーザー、その参加者としてのロールまたは所有者としてのロールを検索する際に、Oracle Database Vaultルール・セットが認可ユーザーに適用されている場合は、次の問合せを実行します。

  ```
  SELECT GRANTEE, AUTH_OPTIONS, AUTH_RULE_SET_NAME
  FROM DBA_DV_REALM_AUTH
  WHERE REALM_NAME = 'Oracle Default Schema Protection Realm'
  ORDER BY GRANTEE;
  ```

- デフォルトで保護されるスキーマ: CTXSYS、EXFSYS、MDDATA、MDSYS
- 保護をお薦めするロール: APEX_ADMINISTRATOR_ROLE、SPATIAL_CSW_ADMIN、WFS_USR_ROLE、CSW_USR_ROLE、SPATIAL_WFS_ADMIN、WM_ADMIN_ROLE
- 保護をお薦めするスキーマ: APEX_030200、OWBSYS、WMSYS

SYS、CTXSYSおよびEXFSYSユーザーは、Oracleデフォルト・スキーマ保護レルムのデフォルトの所有者です。これらのユー

ザーは、このレルムで保護されるロールを他のユーザーに付与して、そのスキーマに対する権限も他のユーザーに付与できます。

親トピック: デフォルトのレルム

4.2.5 Oracleシステム権限およびロール管理レルム

Oracleシステム権限およびロール管理レルムは、Oracleデータベース内にあるすべてのOracle提供ロールを保護します。

このレルムには、システム権限を付与する必要があるユーザーの認可も含まれます。

ユーザーSYSは、このレルムの唯一のデフォルトの所有者です。システム権限の管理を担当するどのユーザーも、このレルムの所有者として認可される必要があります。これらのユーザーは、このレルムで保護されるロールを他のユーザーに付与できます。

Oracleシステム権限およびロール管理レルムで保護されるロールの例としては、DBA、IMP_FULL_DATABASE、SELECT_CATALOG_ROLEおよびSCHEDULER_ADMINがあります。

このレルムが保護するオブジェクトを検索するには、次の問合せを実行します。

```
SELECT OWNER, OBJECT_NAME, OBJECT_TYPE
FROM DBA_DV_REALM_OBJECT
WHERE REALM_NAME = 'Oracle System Privilege and Role Management Realm'
ORDER BY OWNER, OBJECT_NAME;
```

レルム認可ユーザー、その参加者としてのロールまたは所有者としてのロールを検索する際に、Oracle Database Vaultルール・セットが認可ユーザーに適用されている場合は、次の問合せを実行します。

```
SELECT GRANTEE, AUTH_OPTIONS, AUTH_RULE_SET_NAME
FROM DBA_DV_REALM_AUTH
WHERE REALM_NAME = 'Oracle System Privilege and Role Management Realm'
ORDER BY GRANTEE;
```

親トピック: デフォルトのレルム

4.2.6 Oracleデフォルト・コンポーネント保護レルム

Oracleデフォルト・コンポーネント保護レルムは、SYSTEMスキーマとOUTLNスキーマを保護します。

このレルムの認可されたユーザーは、ユーザーSYSとSYSTEMです。

このレルムが保護するオブジェクトを検索するには、次の問合せを実行します。

```
SELECT OWNER, OBJECT_NAME, OBJECT_TYPE
FROM DBA_DV_REALM_OBJECT
WHERE REALM_NAME = 'Oracle Default Component Protection Realm'
ORDER BY OWNER, OBJECT_NAME;
```

レルム認可ユーザー、その参加者としてのロールまたは所有者としてのロールを検索する際に、Oracle Database Vaultルール・セットが認可ユーザーに適用されている場合は、次の問合せを実行します。

```
SELECT GRANTEE, AUTH_OPTIONS, AUTH_RULE_SET_NAME
FROM DBA_DV_REALM_AUTH
WHERE REALM_NAME = 'Oracle Default Component Protection Realm'
ORDER BY GRANTEE;
```

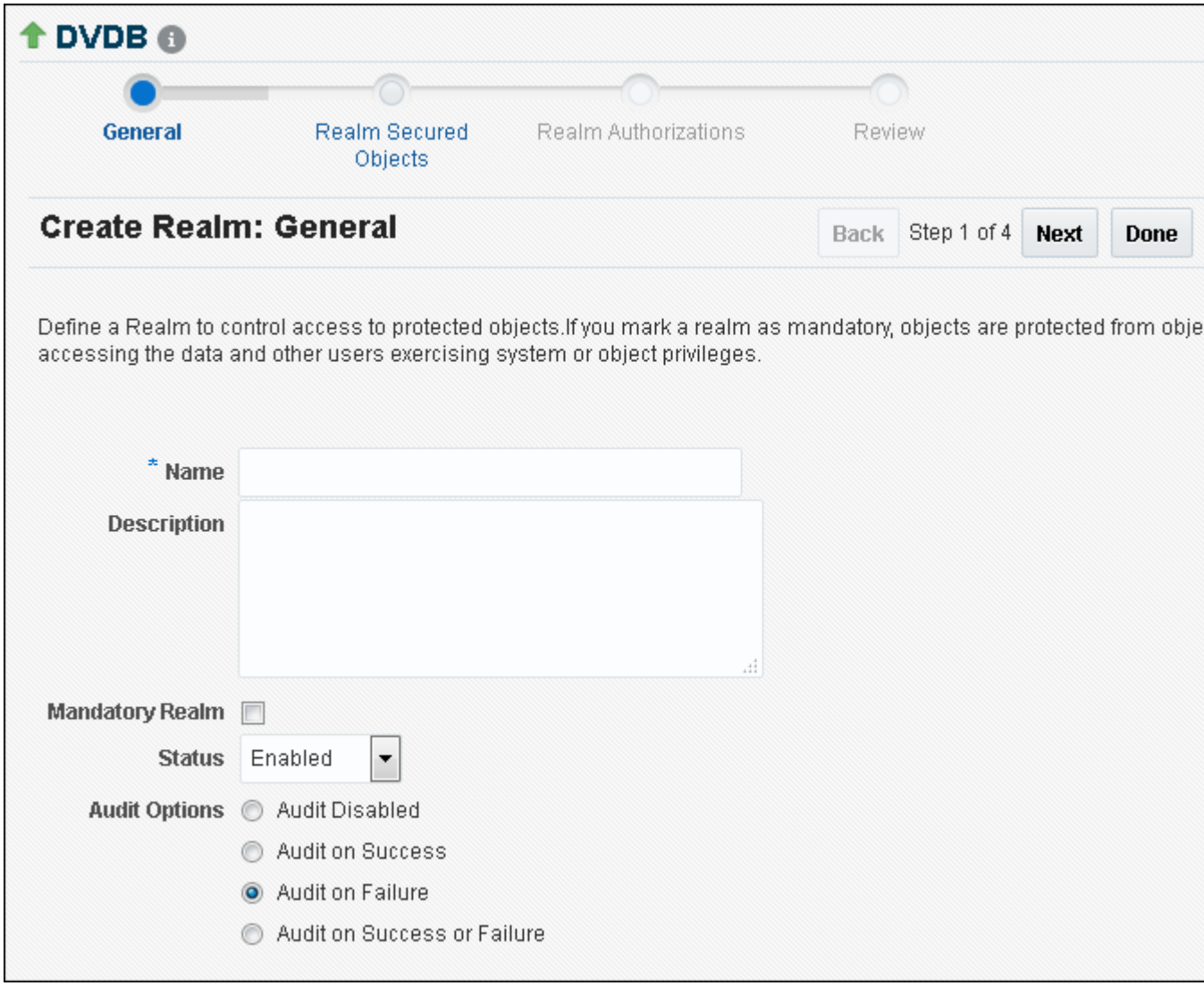
親トピック: デフォルトのレルム

4.3 レルムの作成

レルムの保護を有効にするには、レルムを作成して、レルム・セキュア・オブジェクト、ロールおよび認可を含めるようにレルムを構

成します。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「レルム」をクリックします。
3. 「レルム」ページで、「作成」をクリックして「レルムの作成」ページを表示します。



4. 「レルムの作成」ページで、次の設定を入力します。
   - 名前: レルムの名前を入力します。大/小文字の両方を使用して90文字以内で指定できます。この属性は必須です。

     保護されているアプリケーションの名前をレルム名として使用することをお薦めします(人事アプリケーションにはhr_appなど)。
   - 説明: レルムの簡単な説明を入力します。説明は大/小文字混在で最大で1024文字まで入力できます。この属性はオプションです。

     説明には、指定されたアプリケーション保護のビジネス目標や、レルムによる保護の重要性を示すその他のすべてのセキュリティ・ポリシーを含めることができます。また、レルムに対して認可されているユーザーとその目的、および緊急時の認可についても説明できます。
   - 必須レルム: レルムを必須レルムとして作成するには、このチェック・ボックスを選択します。必須レルムの詳細

は、「必須レルムによるレルム内のオブジェクトへのユーザー・アクセスの制限」を参照してください。

- ステータス: 「有効」、「無効」または「シミュレーション」のいずれかを選択します。この属性は必須です。
- 監査オプション: 次のいずれかを選択します。
  - 監査無効: 監査レコードは作成されません。
  - 成功時に監査: 認可されたアクティビティの監査レコードが作成されます。
  - 失敗時に監査: 認可されていないユーザーがレルムによって保護されているオブジェクトの変更を試行するというようなレルム違反が発生した場合に、監査レコードが作成されます。
  - 成功時または失敗時に監査: アクティビティが認可されている場合でも認可されていない場合でも、レルム内で発生したすべてのアクティビティに関する監査レコードが作成されます。

  非統合監査環境では、Oracle Database Vaultは、監査証跡をDVSYS.AUDIT_TRAIL$表に書き込みます。詳細は、「Oracle Database Vaultの監査」を参照してください。統合監査が有効な場合、この設定では監査レコードは取得されません。かわりに、『Oracle Databaseセキュリティ・ガイド』の説明に従い、この情報を取得する監査ポリシーを作成する必要があります。Oracle Databaseでは、デフォルト・ポリシーORA_DV_AUDPOLも用意されています。これは、Oracle Database VaultのDVSYSおよびDVFスキーマ・オブジェクトと、Oracle Label SecurityのLBACSYSスキーマ・オブジェクトに対して実行されるすべてのアクションを監査します。

5. 「次へ」をクリックして、「レルム・セキュア・オブジェクト」ページを表示します。

   このページの設定の概念的な詳細は、「レルム・セキュア・オブジェクトについて」を参照してください。

6. 「追加」ボタンをクリックして、「セキュア・オブジェクトの追加」ダイアログ・ボックスで次の情報を入力します。
   - オブジェクト所有者: リストから、データベース・スキーマ所有者の名前を選択します。レルムで保護するオブジェクトがロールの場合、%文字を入力できます。この属性は必須です。
   - オブジェクト・タイプ: リストから、TABLE、INDEXまたはROLEなどのデータベース・オブジェクトのタイプを選択します。この属性は必須です。

     任意のタイプのオブジェクトを必要なだけレルムに追加できます。

     デフォルトで、「オブジェクト・タイプ」ボックスには%ワイルドカード文字が入力されており、指定されたオブジェクト所有者にすべてのオブジェクト・タイプを含めることができます。ただし、データベースに特定のスキーマ所有者がいないロールは含まれず、明示的に指定する必要があります。
   - オブジェクト名: レルムで保護する必要があるデータベースのオブジェクトの名前を入力するか、%を入力して、指定したオブジェクト所有者のすべてのオブジェクト(ロール以外)を指定します。%を入力する場合は、%が「オブジェクト・タイプ」設定に対しても使用されていると、それにスキーマ内のすべてのオブジェクトを含めることができます。ただし、「オブジェクト・タイプ」が「表」に設定されている場合、「オブジェクト名」に対する%の使用は、スキーマ内のすべての表を表します。この属性は必須です。

     デフォルトで、「オブジェクト名」フィールドには%ワイルドカード文字が入力されており、オブジェクト・タイプおよびオブジェクト所有者に指定されているスキーマ全体を含めることができます。%ワイルドカード文字は、現在存在するオブジェクトだけでなく、まだ存在していないオブジェクトにも適用されることに注意してください。

     %を「オブジェクト名」フィールドに入力する場合は、%を「オブジェクト・タイプ」フィールドにも入力していると、それにスキーマ内のすべてのオブジェクトが含まれます。ただし、「オブジェクト・タイプ」を「表」などの特定のオブジェクト・タイプに設定した場合、「オブジェクト・タイプ」を%に設定すると、スキーマ内のそのタイプ(この場合は表)の

すべてのオブジェクトを表します。

7. 「次へ」をクリックして、「レルム認可」ページを表示します。

このページの設定の概念的な詳細は、「レルム認可について」を参照してください。

8. 「追加」ボタンをクリックして、「認可の追加」ダイアログ・ボックスで次の情報を入力します。

- レルム認可の権限受領者: リストから、レルム認可を付与するデータベース・アカウントまたはロールを選択します。

  このリストには、システム権限のあるアカウントのみでなく、システム内のすべてのアカウントおよびロールが表示されます。

- レルム認可タイプ: 次のいずれかの設定を選択します。この属性は必須です。

  - 参加者: これらの権限が標準のOracle Database権限付与プロセスを使用して付与されている場合、このアカウントまたはロールは、レルムで保護されているオブジェクトに対するアクセス、操作および作成を行うためのシステム権限を使用できます。1つのレルムには複数の参加者を設定できます。

  - 所有者: このアカウントまたはロールには、レルムの参加者と同じ権限に加えて、レルム・セキュア・データベース・ロールを付与または取り消す認可があります。レルム所有者は、他のユーザーに対して、レルム保護オブジェクトの権限の付与または取消しを行うことができます。1つのレルムに複数の所有者を設定できます。

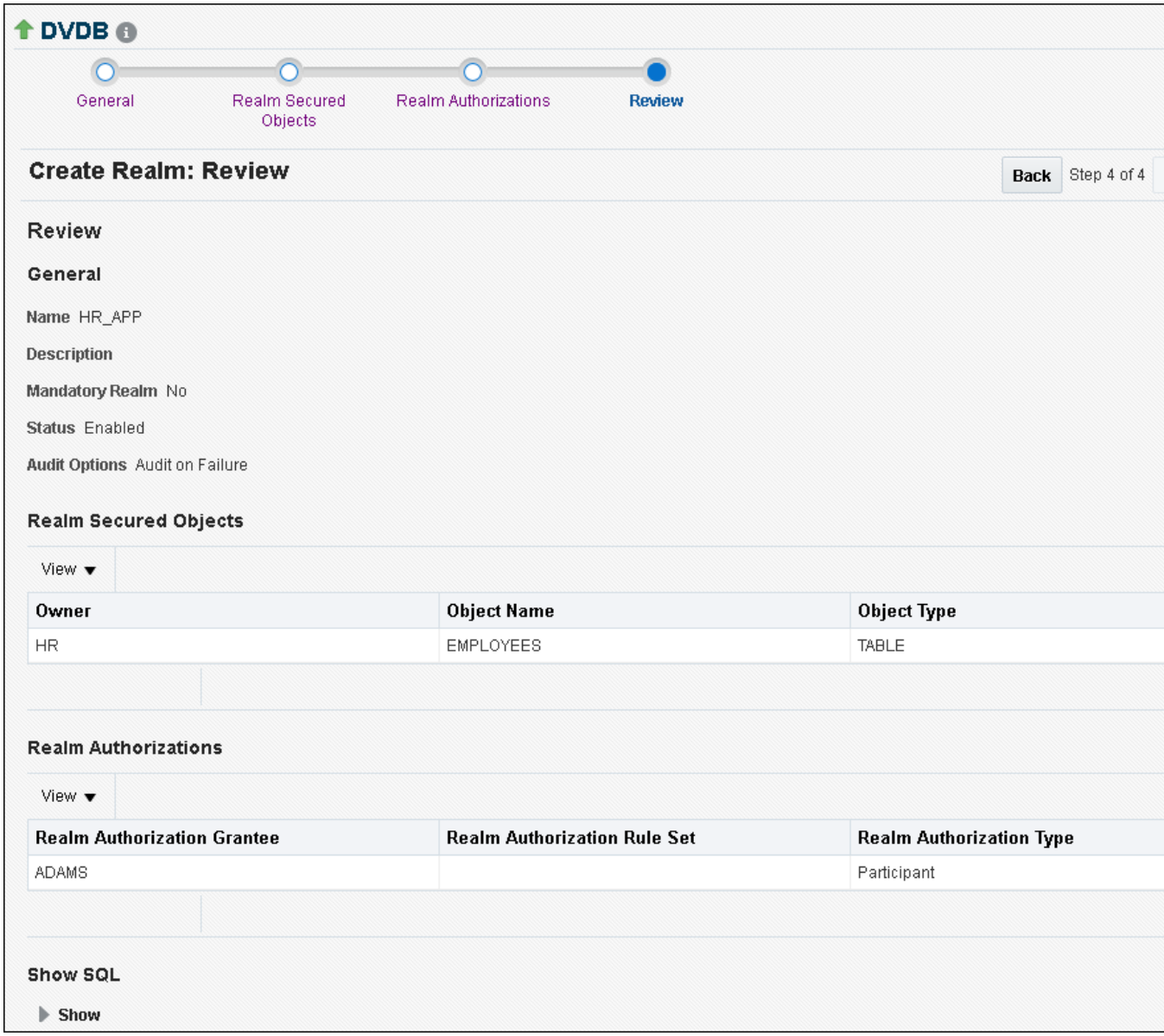
- レルム認可ルール・セット: サイトに作成された使用可能なルール・セットから選択します。選択できるのは1つのルール・セットのみですが、ルール・セットには複数のルールがあります。

  ルールを定義することによるレルム認可の制御の詳細は、「ルール・セットに追加するルールの作成」を参照してください。

  ルール・セットに関連付けられている監査およびカスタム・イベント処理は、レルム認可処理の一部として発生します。

9. 「次へ」をクリックして確認ページを表示します。
10. 「確認」ページで、作成した設定を確認します。

たとえば:



11. 「終了」をクリックして、レルムの作成を完了します。

関連トピック

- レルム・セキュア・オブジェクトについて
- レルム認可について
- 「他のデータベースへのOracle Database Vault構成の伝播」

親トピック: レルムの構成

4.4 レルム・セキュア・オブジェクトについて

レルム・セキュア・オブジェクトにより、レルムによって保護されるテリトリ(一連のスキーマおよびデータベースのオブジェクト、およびロール)が定義されます。

次のような保護方法があります。

- 複数のデータベース・アカウントまたはスキーマのオブジェクトを同じレルムに追加できます。
- 1つのオブジェクトは複数のレルムに属することができます。

  オブジェクトが複数のレルムに属する場合、Oracle Database Vaultは、適切な認可があるかどうかレルムを確認し

ます。SELECT、DDLおよびDML文の場合、ユーザーがいずれかのレルムの参加者であり、コマンド・ルールで許可されれば、そのユーザーが入力するコマンドは許可されます。複数のレルムにおけるデータベース・ロールによるGRANTおよびREVOKE操作の場合、GRANTまたはREVOKE操作を実行するユーザーはレルムの所有者である必要があります。スキーマ所有者は、複数の通常レルムに保護されたオブジェクトに対してDML操作を行うことができます。

いずれかのレルムが必須レルムの場合、オブジェクトにアクセスするユーザーは、レルム所有者か必須レルムの参加者である必要があります。認可チェック・プロセスでは、必須レルムではないレルムは無視されます。オブジェクトを保護する必須レルムが複数ある場合、オブジェクトにアクセスするユーザーは、すべての必須レルムで認可される必要があります。

- SYS所有オブジェクトは、データ・ディクショナリ保護によってすでに保護されており、Oracle Database Vaultによって個別に保護されてはいません。

親トピック: レルムの構成

4.5 レルム認可について

レルム認可では、レルムで保護されているオブジェクトの管理またはアクセスを実行する一連のデータベース・アカウントおよびロールを決定します。

次の状況でシステム権限を使用できるように、レルム認可をアカウントまたはロールに付加できます。

- レルム・セキュア・オブジェクトの作成またはレルム・セキュア・オブジェクトへのアクセスが必要な場合
- レルム・セキュア・ロールを付与または取り消す必要がある場合

レルム所有者またはレルム参加者としてレルム認可を付与されたユーザーは、システム権限を使用してレルム内のセキュア・オブジェクトにアクセスできます。

次のことに注意してください。

- レルム所有者は、自分のレルムに他のユーザーを所有者または参加者として追加できません。DV_OWNERまたはDV_ADMINロールのあるユーザーのみ、所有者または参加者としてユーザーをレルムに追加できます。
- DV_OWNERロールを付与されているユーザーは、自分自身をレルム認可に追加できます。
- レルム所有者(ただし、レルム参加者ではない)は、レルム・セキュア・ロールの付与または取消し、あるいはレルム・セキュア・オブジェクトに対するオブジェクト権限の付与または取消しを誰に対しても行うことができます。
- ユーザーはレルム所有者またはレルム参加者のいずれかとして付与されますが、両方を付与されることはありません。また一方で、既存のレルム認可の認可タイプを更新できます。

「レルムの編集: HR Realm」ページを使用して、レルム認可を管理します。レルム認可を作成、編集および削除できます。

関連トピック

- 「レルム認可構成の問題」レポート

親トピック: レルムの構成

4.6 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では、共通レルム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。

共通レルムのローカル認可

共通レルム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。

共通レルム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内の共通レルム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。共通レルム認可は、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内から共通レルムに追加することも、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ルム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内の共通レルム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、共通レルム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。共通レルム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。共通レルム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。

共通レルムの共通認可

共通レルムの共通認可とは、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。

共通レルム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ルム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・ルート内の共通レルム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内の共通レルム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。
- 共通レルム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内でのレルムの認可の動作

コンテナでのDatabase Vault強制の間に、共通レルムは、それがPDBでローカルで使用される場合の同じレルムと同じ強制動作を実行します。

親トピック: レルムの構成

4.7 レルムの有効化ステータスの変更

Enterprise Manager Cloud Controlから、レルムを無効または有効にすることや、シミュレーション・モードを使用するようレルムを設定することができます。

レルムがポリシーによって管理されており、ポリシー・ステータスが部分になっている場合は、レルムの有効化ステータスを変更できます。ポリシーが有効、無効またはシミュレーション・モードに設定されている場合、レルムの有効化ステータスは変更できません。

1. Oracle Database Vaultの「管理」ページで、「レルム」を選択します。

2. 「レルム」ページで無効または有効にするレルムを選択し、「編集」を選択します。
3. 「レルムの編集」ページの「一般」セクションの「ステータス」の下で、「無効」、「有効」または「シミュレーション」のいずれかを選択します。
4. 「完了」、「終了」の順にクリックします。

親トピック: レルムの構成

4.8 レルムの削除

Enterprise Manager Cloud Controlを使用して、レルムを削除できます。

1. レルムに関連するOracle Database Vaultデータ・ディクショナリ・ビューに問い合せて、削除するレルムへの様々な参照を特定します。
2. レルムがポリシーの一部である場合は、ポリシーからレルムを削除します。
    a. Oracle Database Vaultの「管理」ページで、「ポリシー」を選択します。
    b. レルムを含むポリシーを選択してから、「編集」をクリックします。
    c. 「レルム」領域を展開します。
    d. レルムを選択してから、「削除」をクリックします。
    e. 「次へ」、「終了」、の順にクリックします。
3. 「管理」ページの「Database Vaultコンポーネント」で、「レルム」を選択します。
4. 「レルム」ページで削除するレルムを選択し、「削除」を選択します。
5. 「確認」ウィンドウで、「はい」をクリックします。

    Oracle Database Vaultは、レルムの構成(レルム認可を含む)を削除します。レルム認可に使用されるルール・セットは削除しません。

関連トピック

- Oracle Database Vaultのデータ・ディクショナリ・ビュー

親トピック: レルムの構成

4.9 レルムの動作

適切な権限を持つデータベース・アカウントにより、レルム内のオブジェクトに影響するSQL文が発行されると、特定のアクティビティ・セットが発生します。

これらの権限には、DDL、DML、EXECUTE、GRANT、REVOKEまたはSELECT権限が含まれます。

1. ユーザーのオブジェクト権限は適切ですか。

    Oracle Database Vaultは、ユーザーの権限をチェックしてから、ユーザーの続行を許可します。ユーザーに適切な権限がない場合は、ユーザーにその権限を付与します。ユーザーの権限が適切な場合は、ステップ2に進みます。レルム認可では、追加の権限がユーザーに暗黙的に付与されることはありません。

2. SQL文はレルムによって保護されているオブジェクトに影響しますか。

    影響する場合は、ステップ3に進みます。実行しない場合、レルムはSQL文に影響しません。ステップ8に進みます。コマンドによって影響されるオブジェクトがレルムで保護されていない場合、レルムも実行されるSQL文には影響しません。

3. 必須レルムですか、通常のレルムですか。

    当てはまる場合は、ステップ5に進みます。通常レルムの場合は、ステップ4に進みます。

4. データベース・アカウントで、システム権限を使用してSQL文を実行しますか。

   当てはまる場合は、ステップ5に進みます。当てはまらない場合は、ステップ7に進みます。セッションが、対象となるオブジェクトに対して、SELECT、EXECUTEおよびDML文のオブジェクト権限しか持っていない場合、レルム保護は実施されません。レルムは、レルムで保護されているオブジェクトまたはロールに対するシステム権限の使用を保護します。通常レルムによって保護されているオブジェクトに関するオブジェクト権限を持つユーザーでも、DDL操作を行うことはできません。

5. データベース・アカウントは、レルム所有者またはレルム参加者ですか。

   当てはまる場合は、ステップ6に進みます。いずれでもない場合は、レルム違反が発生し、文は失敗します。コマンドがレルムで保護されているロールのGRANTかREVOKEである場合、またはレルムで保護されているオブジェクトに対するオブジェクト権限GRANTかREVOKEである場合、セッションはロールを介して直接的または間接的にレルム所有者として認可されている必要があります。

6. データベース・アカウントに対するレルム認可は、状況に応じてルール・セットに基づきますか。

   影響する場合は、ステップ7に進みます。当てはまらない場合は、ステップ8に進みます。

7. ルール・セットはTRUEに評価されますか。

   影響する場合は、ステップ8に進みます。そうでない場合は、レルム違反が発生し、SQL文は失敗します。

8. コマンド・ルールでコマンドの実行が阻止されますか。

   阻止される場合は、コマンド・ルール違反が発生しSQL文は失敗します。阻止されない場合は、レルム違反もコマンド・ルール違反も発生しないため、コマンドは成功します。

   たとえば、HRアカウントにDROP ANY TABLE権限があり、HRレルムの所有者であるとしても、月ごとのメンテナンス・ウィンドウでないかぎり、コマンド・ルールでHRによるHRスキーマの任意の表の削除を阻止できます。コマンド・ルールは、オブジェクト権限だけでなくANYシステム権限の使用にも適用され、レルム・チェック後に評価されます。

また、セッションはレルム内で認可されるため、アカウントはレルムによって保護されているオブジェクトを完全に制御できるわけではありません。レルム認可は、アカウントに追加権限を暗黙的に付与しません。アカウントは、従前どおり、オブジェクトにアクセスするためのシステム権限またはオブジェクト権限を持っている必要があります。たとえば、アカウントまたはロールにSELECT ANY表権限があり、HRレルムの参加者であるとします。これは、そのアカウントまたはロールが付与されているアカウントによる、HR.EMPLOYEES表への問合せが可能であることを意味します。レルムの参加者であることは、そのアカウントまたはロールでHR.EMPLOYEES表をDROPできるということではありません。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。レルムおよびコマンド・ルールの両方で、このモデルの上位の層として機能します。

次のことに注意してください。

- デフォルトでは、レルム内の表の保護では、ビューは保護されません。ビューが作成された時期が、表がレルムに追加される前か後かに関係なく、保護する必要のあるビューは、レルムに追加しておく必要があります。
- レルムで保護されたオブジェクトにアクセスする起動者権限でのプロシージャでは、プロシージャの起動者がレルムに対して認可されている必要があります。
- 表のアクセスがPUBLICに付与されている場合、レルム保護では表は保護されないので注意してください。たとえば、SELECT ON table_nameがPUBLICに付与されている場合、table_name(表が必須レルムで保護される場合を除く)がレルムで保護されていても、全ユーザーがこの表にアクセスできます。不要な権限はPUBLICから取り消すことをお薦めします。

親トピック: レルムの構成

4.10 レルムでの認可の動作

ユーザーに正しい権限がない場合、レルム認可により、そのユーザーによるアクティビティの実行が阻止されます。

- レルムの認可について
  レルムは、システム権限によるアクセスからデータを保護します。
- レルム認可の例
  たとえば、システム権限および他の強力な権限を持つユーザーからオブジェクトを保護するレルムを作成できます。

親トピック: レルムの構成

4.10.1 レルムの認可について

レルムは、システム権限によるアクセスからデータを保護します。

レルムは、データ所有者または参加者に追加の権限は付与しません。

レルム認可により、ユーザーのコマンドがコマンド内に指定されているオブジェクトへのアクセスや、そのコマンドの実行を許可または拒否される必要がある場合、論理的に確認する実行時メカニズムが提供されます。

システム権限は、CREATE ANY TABLEおよびDELETE ANY TABLEなどのデータベース権限より優先されます。通常これらの権限はスキーマ全体に適用されるため、オブジェクト権限の必要はありません。DBA_SYS_PRIVS、USER_SYS_PRIVSおよびROLE_SYS_PRIVSなどのデータ・ディクショナリ・ビューには、データベース・アカウントまたはロールのシステム権限が表示されています。データベース認可は、レルムに保護されていないオブジェクトを対象としています。ただし、レルムによって保護されているオブジェクトのシステム権限を正常に使用するには、ユーザーがレルム所有者または参加者として認可されている必要があります。レルム違反によりシステム権限の使用を阻止し、監査することができます。

必須レルムは、オブジェクト権限およびシステム権限に基づく両方のアクセスをブロックします。つまり、オブジェクト所有者は、レルムへのアクセスが認可されない場合、アクセスできません。ユーザーは、ユーザーまたはロールがレルムにアクセスする認可を受けている場合のみ、必須レルムのセキュア・オブジェクトにアクセスできます。

親トピック: レルムでの認可の動作

4.10.2 レルム認可の例

システム権限および他の強力な権限を持つユーザーからオブジェクトを保護するレルムを作成できます。

- 例: 認可されていないユーザーによる表作成の試行
  認可されていないユーザーが表を作成しようとすると、ORA-47401エラーが発生します。
- 例: 認可されていないユーザーによるDELETE ANY TABLE権限の使用の試行
  認可されていないユーザー・アクセスに対して、ORA-01031: 権限が不十分ですというエラーが表示されます。
- 例: 認可されたユーザーによるDELETE操作の実行
  認可されたユーザーには、認可されたアクティビティの実行が許可されます。

親トピック: レルムでの認可の動作

4.10.2.1 例: 認可されていないユーザーによる表作成の試行

権限のないユーザーが表を作成しようとすると、ORA-47401エラーが発生します。

例4-1に、レルムによってHRスキーマが保護されているレルムに、CREATE ANY TABLEシステム権限を持つ認可されていないユーザーが表の作成を試行すると発生する動作を示します。

例4-1 認可されていないユーザーによる表作成の試行

```
CREATE TABLE HR.demo2 (col1 NUMBER(1));
```

次のような出力結果が表示されます。

```
ORA-47401: Realm violation for CREATE TABLE on HR.DEMO2
```

この例からわかるように、認可されていないユーザーの試みは失敗します。SELECT ANY TABLE、CREATE ANY TABLE、DELETE ANY TABLE、UPDATE ANY TABLE、INSERT ANY TABLE、CREATE ANY INDEXなどのシステム権限の不正な使用は失敗します。

親トピック: レルム認可の例

4.10.2.2 例: 認可されていないユーザーによるDELETE ANY TABLE権限の使用の試行

認可されていないユーザー・アクセスに対して、ORA-01031: 権限が不十分ですというエラーが表示されます。

例4-2に、認可されていないデータベース・アカウントがDELETE ANY TABLEシステム権限を使用して既存のレコードの削除を試行すると発生する動作を示します。データベース・セッションにより次のエラーが返されます。

例4-2 認可されていないユーザーによるDELETE ANY TABLE権限の使用の試行

```
DELETE FROM HR.EMPLOYEES WHERE EMPNO = 8002;
```

次の出力が表示されます。

```
ERROR at line 1:
ORA-01031: insufficient privileges
```

レルムはオブジェクトの直接権限には影響しません。たとえば、HR.EMPLOYEES表の削除権限を付与されているユーザーは、レルム認証なしで正常にレコードを削除できます。そのため、レルムには、データベース・アカウントによる通常のビジネス・アプリケーションの使用に最低限の影響力が必要です。

親トピック: レルム認可の例

4.10.2.3 例: 認可されたユーザーによるDELETE操作の実行

認可されたユーザーには、認可されたアクティビティの実行が許可されます。

例4-3に、認可されたユーザーがレルム内で許可された標準的なタスクをどのように実行するかを示します。

例4-3 認可されたユーザーによるDELETE操作の実行

```
DELETE FROM HR.EMPLOYEES WHERE EMPNO = 8002;
1 row deleted.
```

親トピック: レルム認可の例

4.11 レルムで保護されたオブジェクトへのアクセス

レルムでオブジェクトを保護できますが、このオブジェクトに含まれるオブジェクトにはアクセスできます。

たとえば、特定の表にレルムを作成するとします。ただし、この表にユーザーが索引を作成できるようにします。これは、次のシナリオに応じて次のように実行できます。

- ユーザーにCREATE ANY INDEX権限がない場合。表のレルム所有者として、索引を作成する必要のあるユーザーにCREATE INDEX ON table権限を付与します。

- ユーザーにCREATE ANY INDEX権限がある場合。この場合、別のレルムを作成して、すべての索引タイプをセキュア・オブジェクトとし、そのユーザーにレルムの参加者認可を付与します。(レルム参加者以外がレルムで保護された表に索引を作成する場合、CREATE ANY INDEXのみでは不十分なことに注意してください。)
- すべてのデータベース管理者が索引を作成できるようにし、データベース管理者にCREATE ANY INDEX権限がある場合。データ保護レルムで、索引タイプを除いて保護するすべてのオブジェクト・タイプを指定します。これにより、保護された表の索引をすべての管理者が作成できます。

親トピック: レルムの構成

4.12 レルムの動作の例

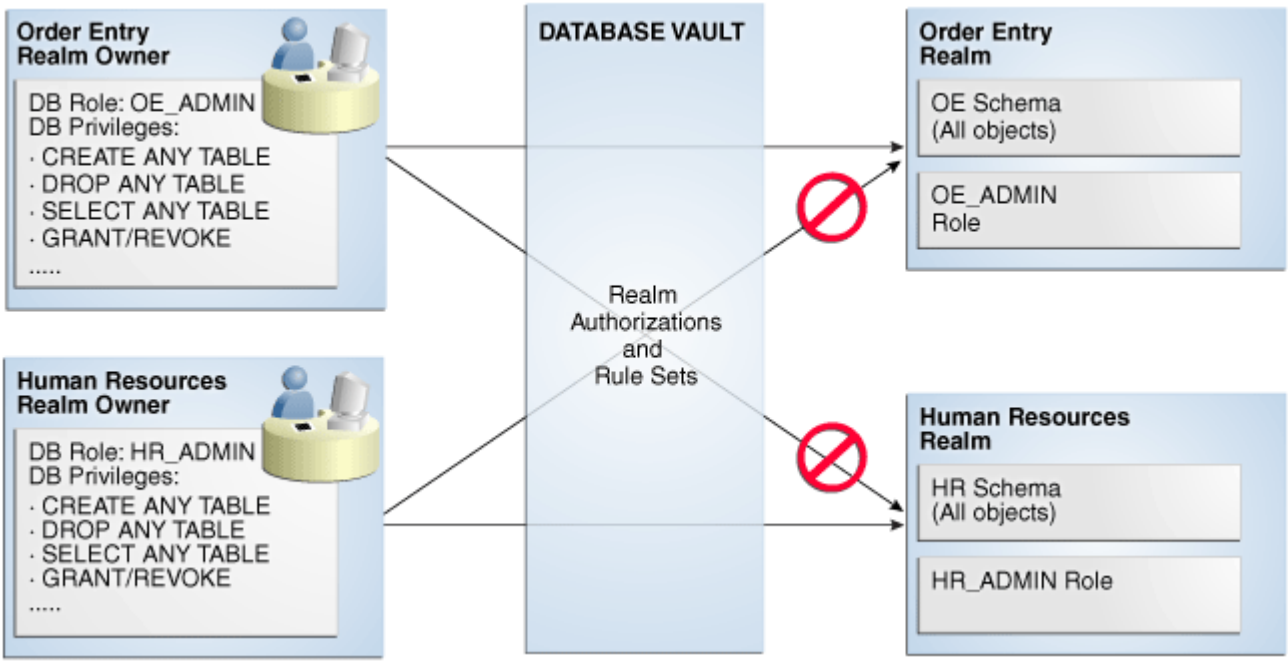
レルムにより、同じ権限のある2人のユーザーに、オブジェクトに対する別々のアクセス・レベルが必要な場合に保護を提供できます。

図4-1は、レルム内のデータがどのように保護されているかを示しています。

このシナリオでは、別々のレルムを担当する2人のユーザーに同じシステム権限があります。レルム所有者は、データベース・アカウントまたはデータベース・ロールのいずれかです。OE_ADMINおよびHR_ADMINの2つのロールのそれぞれは、セキュア・オブジェクトとしてレルムで保護され、かつレルム所有者として構成されます。

さらに、OE_ADMINなどのレルム所有者は、レルムで保護されているデータベース・ロールの付与または取消しを実行できます。レルム所有者は、他のレルムで保護されるロール(Oracleシステム権限およびロール管理レルムでSYSによって作成されるDBAロールなど)を管理できません。レルムで保護されているオブジェクトにアクセスするためにシステム権限の不正な使用を試行することで、監査可能なレルム違反が生成されます。各レルム所有者の権限はそのレルム内に制限されます。たとえば、OE_ADMINには人事レルムへのアクセス権はなく、HR_ADMINには受注レルムへのアクセス権はありません。

図4-1 レルムおよびレルム所有者に対する認可の動作



関連トピック

- クイック・スタート・チュートリアル: DBAアクセスからのスキーマの保護

親トピック: レルムの構成

4.13 その他のOracle Database Vaultコンポーネントへのレルムの影響

レルムにはファクタ、アイデンティティ、またはルール・セットに対する影響力はありませんが、コマンド・ルールに対してはあります。

コマンド・ルールでは、SQL文を処理する際にOracle Database Vaultによってまずレルム認可が評価されます。

「レルムの動作」では、レルム内のオブジェクトに影響を与えるSQL文を処理する際にOracle Database Vaultで実行されるステップが説明されています。「コマンド・ルールの動作」では、コマンド・ルールがどのように処理されるかが説明されています。

親トピック: レルムの構成

4.14 レルム設計のガイドライン

Oracleでは、一連のレルム設計のガイドラインを提供しています。

- データベース・アプリケーションを構成するスキーマおよびロールに基づいてレルムを作成します。

  最小限かつ特定のロールおよびアプリケーション・オブジェクトの維持に必要なシステム権限でデータベース・ロールを定義し、名前付きアカウントにロールを付与します。これにより、レルムの認可されたメンバーとしてロールを追加できます。レルムで保護されていてアプリケーションに必要なオブジェクトのオブジェクト・レベルの権限の場合、ロールを作成して、最低限かつ特定のオブジェクト・レベルの権限をロールに付与し、名前付きアカウントを付与します。多くの場合、ANYの付くシステム権限がすでに使用中でないかぎり、これらのタイプのロールをレルムで認可する必要はありません。権限はできるだけ少なくするという方針のモデルがデータベース・アプリケーションには理想的です。

- データベース・オブジェクトは複数のレルムに属することが可能で、アカウントまたはロールは複数のレルムで認可することが可能です。

  データベース・スキーマのサブセットに制限付きのアクセス権(HRスキーマのEMPLOYEES表のみ、またはレルムで保護されたロールなど)を付与するには、最低限必要なオブジェクトと認可で新しいレルムを作成します。

- ロールを権限受領者としてレルムに追加する場合は、ロールを保護するレルムを作成します。これにより、SYSTEMユーザー・アカウントなどGRANT ANY ROLEシステム権限が付与されているユーザーが、自分自身にロールを付与することができなくなります。

- SYSユーザー・アカウントをレルム認可に追加する場合は、ロール(DBAロールなど)を介さずに、ユーザーSYSを明示的に追加する必要があります。

- レルム認可として追加する予定で、現在ロールに許可されている権限に注意してください。

  SYSまたはSYSTEMなどのアカウントが初めてロールを作成し、Oracle Database Vault管理者がそのロールをレルム認可として追加した場合、ロールのレルム認可は意図せず付与される可能性があり、すぐには気付きません。これはロールを作成するアカウントが作成される際に、ロールが暗黙的に付与されるためです。

- 管理タスクに対しては、一時的なレルム保護の緩和が必要な場合もあります。レルムを無効にするのではなく、セキュリティ管理者(DV_ADMINまたはDV_OWNER)をログインさせて、レルムの認可されたアカウントに名前付きのアカウントを追加し、認可ルール・セットを有効に設定します。その後、有効なルール・セットで、ルール・セットの監査をすべてオンにします。管理タスクが完了するとレルム認可を削除できます。

- 新しいユーザーにANY権限を付与する場合は、必要に応じて他のユーザーにANY権限を付与できるよう、Oracleシステム権限およびロール管理レルムにデータベース管理ユーザーを追加することをお薦めします。たとえば、名前付きアカウントを使用してANY操作のGRANTを実行すると、これらの操作の監査が可能になり、説明責任を果すための監査証跡が作成されます。

- レルムによって保護されていた表、索引またはロールを削除し、同じ名前を使用して再作成した場合、レルムの保護はリストアされません。新しい表、索引またはロールに対してレルムの保護を再作成する必要があります。ただし、指定したスキーマ内の以後の表、索引およびロールすべてに対して自動的に保護を実施できます。たとえば、以後の表すべてに保護を実施するには、次のようにします。

```
BEGIN
 DBMS_MACADM.ADD_OBJECT_TO_REALM('realm_name', 'schema_name', '%', 'TABLE');
END;
/
```

- 制限を適用せずにレルムを有効にする、シミュレーション・モードを使用して、レルムの開発フェーズをテストできます。シミュレーション・モードでは違反に関する詳細情報が書き込まれ、適用されたアクティビティを表示できます。DV_OWNERまたはDV_ADMINロールを持つユーザーは、DBA_DV_SIMULATION_LOGデータ・ディクショナリ・ビューを問い合せてシミュレーション・ログを表示できます。

関連トピック

- レルムおよびコマンド・ルール・アクティビティのログ記録のためのシミュレーション・モードの使用

親トピック: レルムの構成

4.15 レルムのパフォーマンスへの影響

レルムは、DDLおよびDML操作などの様々な状況で、データベース・パフォーマンスに影響します。

- レルムで保護されているオブジェクトに対するDDLおよびDML操作によって、Oracle Databaseはそれほど影響を受けません。スキーマ全体を対象としてレルムを作成し、割り当てられたタスクに関連する特定の操作のみを実行できるよう特定のユーザーを認可することをお薦めします。ファイングレイン制御の場合は、個々の表を対象とするレルムを定義し、表に対する特定の操作を実行できるようユーザーを認可すると同時に、スキーマ全体を対象としたレルムでアプリケーション全体を保護することもできます。このタイプの構成(つまり複数のレルムによる同じオブジェクトの保護)は重大なパフォーマンス低下をもたらすことはなく、スキーマ内の一部のオブジェクトにレルム認可を付与できるようになります。
- 監査はパフォーマンスに影響します。最高のパフォーマンスを実現するためには、すべての操作を監査するのではなくファイングレイン監査を使用することをお薦めします。
- システム・パフォーマンスを定期的に確認します。確認するには、Oracle Enterprise Manager(Oracle Databaseと一緒にデフォルトでインストールされるOracle Enterprise Manager Cloud Controlを含む)、自動ワークロード・リポジトリ(AWR)およびTKPROFなどのツールを実行します。

関連項目:

- データベース・パフォーマンスの監視方法を学習するには、『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
- 個々のSQL文およびPL/SQL文の実行を監視するには、『Oracle Database SQLチューニング・ガイド』を参照してください

親トピック: レルムの構成

4.16 レルムに関連するレポートおよびデータ・ディクショナリ・ビュー

Oracle Database Vaultには、レルムの分析に役立つ、レポートとデータ・ディクショナリ・ビューが用意されています。

表4-1では、Oracle Database Vaultレポートを示します。これらのレポートの実行方法の詳細は、「Oracle Database Vaultレポート」を参照してください。

表4-1 レルムに関連するレポート

レポート	用途
「レルムの監査」レポート	レルムの保護およびレルム認可操作により生成されたレコードが監査されます。
「レルム認可構成の問題」レポート	不完全または無効なルール・セット、またはレルムに影響する権限受領者や所有者が存在しないなどの認可構成情報が表示されます。
「ルール・セット構成の問題」レポート	ルールが定義されていないか、有効ではなく、それらを使用するレルムに影響を与える可能性があるルール・セットが表示されます。
オブジェクト権限レポート	レルムが影響するオブジェクト権限が表示されます。
権限管理 - サマリー・レポート	レルムの権限受領者および所有者の情報が示されます。
機密オブジェクト・レポート	コマンド・ルールが影響するオブジェクトが表示されます。

表4-2に、既存のレルムに関する情報を提供するデータ・ディクショナリ・ビューを示します。

表4-2 レルムに使用されるデータ・ディクショナリ・ビュー

データ・ディクショナリ・ビュー	説明
DBA_DV_REALM ビュー	現行のデータベース・インスタンスで作成されたレルムが表示されます。
DBA_DV_REALM_AUTH ビュー	特定のレルムのレルム・オブジェクトにアクセスするための、名前付きデータベース・ユーザー・アカウントまたはデータベース・ロール(GRANTEE)の認可が表示されます。
DBA_DV_REALM_OBJECT ビュー	データベース・スキーマ、または特定のデータベース・オブジェクトが含まれている(つまり、レルムによって保護されている)スキーマのサブセットが表示されます。

親トピック: レルムの構成

5 ルール・セットの構成

ルール・セットにより1つ以上のルールがグループ化され、ルールによりユーザーがオブジェクトでアクションを実行できるかどうかが決定されます。

- ルール・セットの概要
  ルール・セットとは、1つ以上のルールの集合です。
- 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、PDBまた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。
- リリース12.2より前のリリースのデフォルト・ルールおよびデフォルト・ルール・セット
  以前のリリースで提供されていた多数のデフォルト・ルールおよびデフォルト・ルール・セットは、サポートされなくなりましたが、現在のOracle Databaseインストールで使用されている場合があります。
- デフォルトのルール・セット
  Oracle Database Vaultにはデフォルトで一連のルール・セットが用意されており、ニーズにあわせてカスタマイズできます。
- ルール・セットの作成
  ルール・セットを作成するには、まずルール・セットを作成し、次にそのルール・セットを編集して1つ以上のルールに関連付けることができます。
- ルール・セットに追加するルールの作成
  ルールは制御する動作を定義し、ルール・セットは名前付きのルールの集合です。
- Oracle Database Vaultコンポーネントへのルール・セット参照の削除
  ルール・セットを削除する前に、ルール・セットのOracle Database Vaultコンポーネントへの参照を削除する必要があります。
- ルール・セットの削除
  Enterprise Manager Cloud Controlを使用して、ルール・セットへの参照を見つけてからルール・セットを削除できます。
- ルール・セットの動作
  ルール・セットの動作を理解すると、より効果的なルール・セットを作成できます。
- チュートリアル: セキュリティ違反の電子メール・アラートの作成
  このチュートリアルは、UTL_MAIL PL/SQLパッケージおよびアクセス制御リストを使用して、セキュリティ違反の電子メール・アラートを作成する方法を示します。
- チュートリアル: 二人制整合性(デュアル・キー・セキュリティ)の構成
  このチュートリアルでは、Oracle Database Vaultを使用して2人のユーザーの認可を制御する方法を示します。
- ルール・セット設計のガイドライン
  Oracleでは、ルール・セット設計のガイドラインを提供しています。
- ルール・セットのパフォーマンスへの影響
  ルールの数および複雑さにより、データベースのパフォーマンスが低下する場合があります。
- ルール・セットとルールに関連するレポートおよびデータ・ディクショナリ・ビュー
  Oracle Database Vaultには、ルール・セットおよびそれに含まれるルールの分析に役立つ、レポートとデータ・ディクショナリ・ビューが用意されています。

5.1 ルール・セットの概要

ルール・セットとは、1つ以上のルールの集合のことです。

ルール・セットは、レルム認可、ファクタ割当て、コマンド・ルールまたはセキュア・アプリケーション・ロールに関連付けることができます。

ルール・セットは、それに含まれる各ルールと評価タイプ(「すべてのTrue」または「いずれかTrue」)に基づいて、trueまたはfalseに評価されます。ルール・セット内のルールは、TrueまたはFalseと評価されるPL/SQL式です。ルールを作成し、そのルールを複数のルール・セットに追加できます。

ルール・セットを使用して次のアクティビティを実行できます。

- レルム認可がアクティブになる条件の定義(レルム認可をさらに制限することが目的)
- コマンド・ルールを許可する時期の定義
- セキュア・アプリケーション・ロールの有効化
- ファクタのアイデンティティを割り当てる時期の定義

ルール・セットを作成すると、レルムの認証、コマンド・ルール、ファクタまたはセキュア・アプリケーション・ロールの構成時に選択できるようになります。

関連トピック

- ルール・セットとルールに関連するレポートおよびデータ・ディクショナリ・ビュー
- Oracle Database Vaultルール・セットのAPI

親トピック: ルール・セットの構成

5.2 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、PDBまた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。

共通レルムでは、関連付けられたレルムまたはコマンド・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

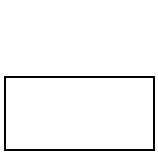
親トピック: ルール・セットの構成

5.3 リリース12.2より前のリリースのデフォルト・ルールおよびデフォルト・ルール・セット

以前のリリースで提供されていた多数のデフォルト・ルールおよびデフォルト・ルール・セットは、サポートされなくなりましたが、現在のOracle Databaseインストールで使用されている場合があります。

Oracle Databaseリリース12.2より前のリリースのデフォルト・ルールおよびデフォルト・ルール・セットを使用している場合、

Oracle Databaseでは、それらは、独自の用途のためにカスタマイズしてあった場合にはアップグレード中に削除されません。これらのルールおよびルール・セットをカスタマイズした場合、またはこれらの古いデフォルト・ルール・セットを使用する場合は、ALTER SYSTEMおよびALTER SESSIONコマンド・ルールを使用することで、カスタマイズしたそれらのルールおよびルール・セットを再実装してから、古いルールおよびルール・セットを無効化し削除することをお薦めします。これらのルールおよびルール・セットをカスタマイズしていない場合、または使用しない場合は、以降のデフォルトのコマンド・ルールで同じ機能を使用できるため、以前のルールおよびルール・セットを削除する必要があります。



ノート:

影響を受ける可能性があるルールおよびルール・セットの完全なリストは、リリース 12.2 バージョンの Oracle Database Vault 管理者ガイドを参照してください。

親トピック: ルール・セットの構成

5.4 デフォルトのルール・セット

Oracle Database Vaultにはデフォルトで一連のルール・セットが用意されており、ニーズにあわせてカスタマイズできます。

DBA_DV_RULE_SETデータ・ディクショナリ・ビューを問い合せることで、ルール・セットをすべて示すリストを確認できます。ルール・セットに関連付けられているルールを確認するには、DBA_DV_RULE_SET_RULEデータ・ディクショナリ・ビューを問い合せます。

デフォルトのルール・セットは次のとおりです。

- データファイル・ヘッダーのダンプを許可: このルール・セットは、データ・ブロックのダンプを防ぎます。
- システム変更用のファイングレイン・コントロールを許可: このルール・セットにより、ユーザーがALTER SYSTEM SQL文を使用して初期化パラメータを設定できるかどうかを制御できるようになります。
- システム・パラメータのファイングレイン・コントロールを許可: ノート: このルール・セットは非推奨になりました。

  このルール・セットにより、システム・セキュリティ、ダンプまたは宛先の場所、バックアップとリストアの設定、オプティマイザの設定、PL/SQLデバッグおよびセキュリティ・パラメータを管理する初期化パラメータを非常に柔軟にきめ細かく制御できるようになります。これは、このルール・セットに関連付けられたルールに基づいて、次の初期化パラメータに影響します。

  - 「バックアップ・リストア・パラメータが許可されているか」のルール: RECYCLEBINを設定できません(ただしリサイクルビンの無効化は妨げられません)。
  - 「データベース・ファイル・パラメータが許可されているか」のルール: CONTROL_FILESを設定できません。
  - 「オプティマイザ・パラメータが許可されているか」のルール: OPTIMIZER_SECURE_VIEW_MERGING = FALSEを設定できます(ただしTRUEは指定できません)。
  - 「PL-SQLパラメータが許可されているか」のルール: PLSQL_DEBUG = FALSEを設定できます(ただし、TRUEは指定できません)。
  - 「セキュリティ・パラメータが許可されているか」のルール: 次のパラメータを設定できません。

パラメータA-A	パラメータO-S
AUDIT_SYS_OPERATIONS = FALSE	OS_ROLES = TRUE

パラメータA-A	パラメータO-S
AUDIT_TRAIL = NONE または FALSE	REMOTE_OS_ROLES = TRUE
AUDIT_SYSLOG_LEVEL	SQL92_SECURITY = FALSE

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。

- セッションを許可: データベースにセッションを作成する権限を制御します。このルール・セットを使用すると、CONNECTコマンド・ルールを使用してデータベース・ログインを制御するためのルールを追加できます。CONNECTコマンド・ルールは、その使用が必要なプログラムへのSYSDBAアクセスの制御または制限に有用です。このルール・セットは移入されていません。
- VPD管理権限を付与可能: GRANTおよびREVOKE文を使用して、Oracle Virtual Private DatabaseのDBMS_RLSパッケージのGRANT EXECUTEまたはREVOKE EXECUTE権限を付与する権限を制御します。
- アカウント/プロファイルを保守可能: CREATE USER、DROP USER、CREATE PROFILE、ALTER PROFILEまたはDROP PROFILE文を使用して、ユーザー・アカウントおよびプロファイルを管理するロールを制御します。
- 自分のアカウントを保守可能: DV_ACCTMGRロールのあるアカウントを許可し、ALTER USER文を使用したユーザー・アカウントおよびプロファイルの管理を可能にします。また、ALTER USER文を使用した、個々のアカウントによる個人のパスワードの変更を許可します。DV_ACCTMGRロールの詳細は、「DV_ACCTMGR Database Vaultアカウント・マネージャ・ロール」を参照してください。
- 無効: レルム、コマンド・ルール、ファクタおよびセキュア・アプリケーション・ロールを迅速に無効にするための簡易ルール・セットです。
- 有効: システム機能を迅速に有効にするための簡易ルール・セットです。
- AUDIT_SYS_OPERATIONSをFalseに設定することはできません: AUDIT_SYS_OPERATIONS初期化パラメータがFALSEに設定されないようにします。統合監査が有効になっている場合は、AUDIT_SYS_OPERATIONSパラメータの効果はありません。
- OPTIMIZER_SECURE_VIEW_MERGINGをTrueに設定することはできません: OPTIMIZER_SECURE_VIEW_MERGING初期化パラメータがTRUEに設定されないようにします。
- OS_ROLESをTrueに設定することはできません: OS_ROLES初期化パラメータがTRUEに設定されないようにします。
- PLSQL_DEBUGをTrueに設定することはできません: PLSQL_DEBUG初期化パラメータがTRUEに設定されないようにします。
- REMOTE_OS_ROLESをTrueに設定することはできません: REMOTE_OS_ROLES初期化パラメータがTRUEに設定されないようにします。
- SQL92_SECURITYをFalseに設定することはできません: SQL92_SECURITYがFALSEに設定されないようにします。
- AUDIT_TRAILをオフにすることはできません: AUDIT_TRAIL初期化パラメータが無効にされないようにします。統合監査が有効になっている場合は、AUDIT_TRAILパラメータの効果はありません。

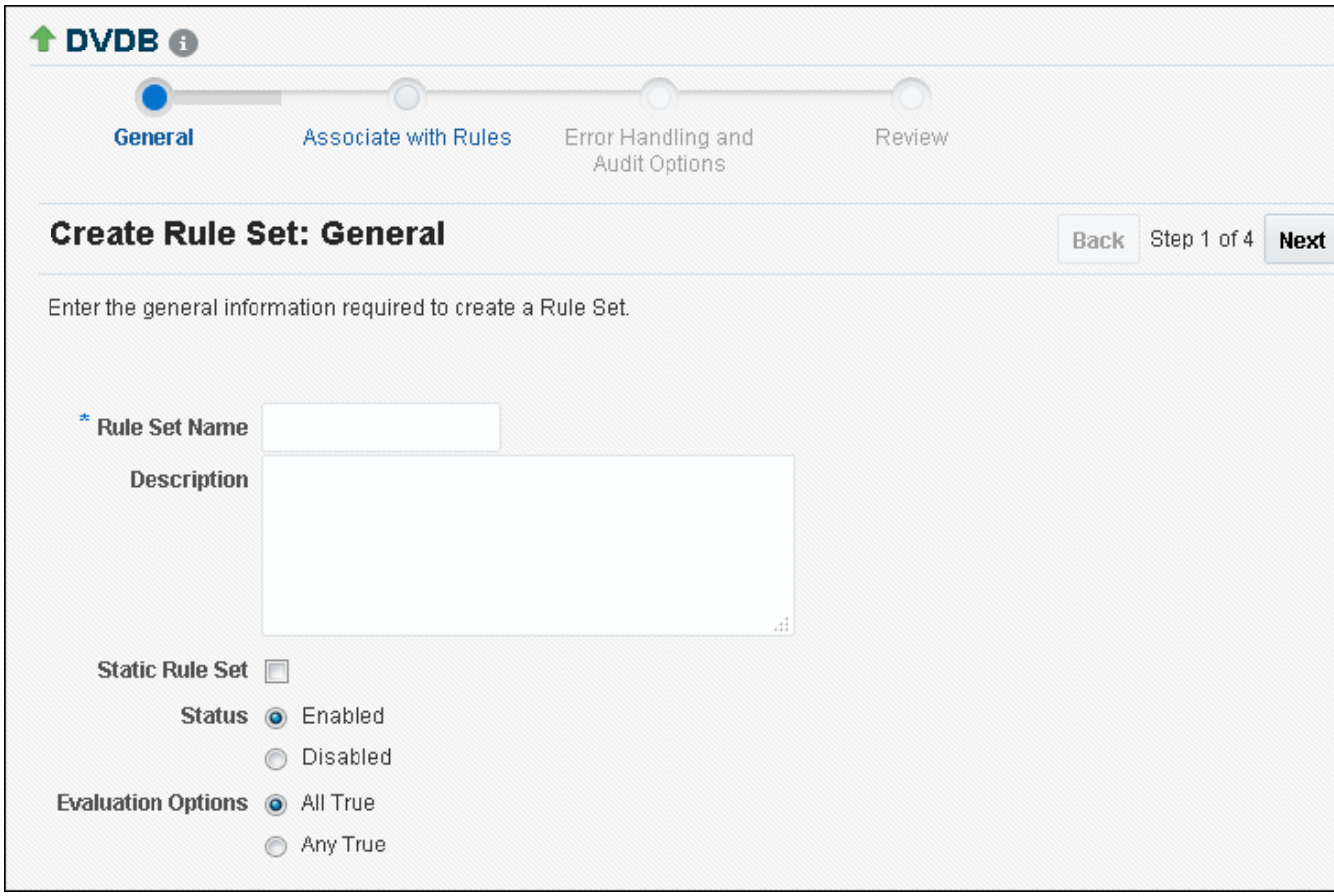
親トピック: ルール・セットの構成

5.5 ルール・セットの作成

ルール・セットを作成するには、まずルール・セットを作成し、次にそのルール・セットを編集して1つ以上のルールに関連付けることができます。

作成したルール・セットに新しいルールを関連付ける、既存のルールを追加する、またはそのルール・セットからルールの関連付けを削除することが可能です。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「ルール・セット」をクリックします。
3. 「ルール・セット」ページで、「作成」をクリックして「ルール・セットの作成」ページを表示します。



4. 「一般」ページで、次の情報を入力します。
   - 名前: ルール・セットの名前を入力します。大/小文字の両方を使用して90文字以内で指定できます。空白を使用できます。この属性は必須です。

     名前は動詞で始まり、ルール・セットが関連付けられるレルムまたはコマンド・ルールの名前で終わることをお薦めします。たとえば:

     ```
     Limit SQL*Plus access
     ```

   - 説明: ルール・セットの機能の説明を入力します。大/小文字の両方を使用して1024文字以内で指定できます。この属性はオプションです。

     ルール・セットのビジネス要件を説明できます。たとえば:

```
Rule to limit access to SQL*Plus
```

- 統計ルール・セット: ルール・セットがユーザー・セッション中にアクセスされるときの評価の頻度を制御できます。統計ルール・セットは、ユーザー・セッションで初めてアクセスされる際に一度評価されます。その後、評価された値はユーザー・セッションで再利用されます。その一方で、非統計ルール・セットは、アクセスされるたびに評価されます。
- ステータス: 「有効」または「無効」のいずれかを選択し、実行時にルール・セットを有効または無効にします。この属性は必須です。
- 評価オプション: ルール・セットに複数のルールを割り当てる場合は、次の設定のいずれかを選択します。
  - すべてTrue: ルール・セット自体がTrueと評価されるために、ルール・セットのルールはすべてTrueと評価される必要があります。
  - いずれかTrue: ルール・セット自体がTrueと評価されるために、少なくともルール・セットの1つのルールがTrueと評価される必要があります。

5. 「次へ」をクリックして、「ルールとの関連付け」ページを表示します。
6. 次のいずれかのオプションを選択します。
   - 既存のルールの追加: 「使用可能なルール」リストをダブルクリックして、ルールを「選択したルール」リストに移動し、「OK」をクリックします。
   - ルールの作成: 名前と、TrueまたはFalseに評価されるWHERE句式を入力します。「OK」をクリックします。詳細は、「ルール・セットに追加するルールの作成」を参照してください。
7. 「次へ」をクリックして、「エラー処理および監査オプション」ページを表示します。
8. 次の情報を入力します。
   - エラー処理: 「エラー・メッセージの表示」または「エラー・メッセージを表示しない」を選択します。

     「エラー・メッセージを表示しない」を選択して監査を有効にする利点は、潜在的な侵入者のアクティビティを追跡できるということです。監査レポートにより侵入者のアクティビティを把握できますが、エラー・メッセージが表示されないため、侵入者は監査が行われていることに気付きません。
   - 失敗コード: -20000から-20999または20000から20999の範囲の数値を入力します。ルール・セットがFalseと評価されるか、関連付けられたルールのいずれかに無効なPL/SQL式が含まれる場合に、失敗メッセージ(次で作成)付きのエラー・コードが表示されます。この設定を省略すると、Oracle Database Vaultにより一般的なエラー・コードが表示されます。
   - 失敗メッセージ: 大/小文字混在で80文字以内のメッセージを入力し、「失敗コード」で指定した失敗コードに関連付けます。ルール・セットがFalseと評価されるか、関連付けられたルールのいずれかに無効なPL/SQL式が含まれる場合に、エラー・メッセージが表示されます。エラー・メッセージを指定しない場合、Oracle Database Vaultにより通常のエラー・メッセージが表示されます。
   - カスタム・イベント・ハンドラ・オプション: 次のオプションのいずれかを選択し、カスタム・イベント・ハンドラ・ロジック(次で作成)を実行する時期を決定します。
     - ハンドラ無効: カスタム・イベント・メソッドを実行しません。
     - 失敗時に実行: ルール・セットがFalseと評価されるか、関連付けられたルールのいずれかに無効なPL/SQL式が含まれる場合に、カスタム・イベント・メソッドが実行されます。
     - 成功時に実行: ルール・セットがTrueと評価されるとカスタム・イベント・メソッドが実行されます。

カスタム・イベント・メソッドを作成して、標準のOracle Database Vaultルール・セットの監査機能以外の特別な処理を実行できます。たとえば、イベント・ハンドラを使用して、ワークフロー・プロセスの開始や外部システムへのイベント情報の送信を実行できます。

- カスタム・イベント・ハンドラ・ロジック: 大/小文字混在で255文字以内のPL/SQL式を入力します。式には、任意のパッケージ・プロシージャまたはスタンドアロン・プロシージャを含めることができます。独自の式を作成するか、「Oracle Database Vaultルール・セットのAPI」で説明されているPL/SQLインタフェースを使用できます。

  完全修飾プロシージャとして式を記述します(`schema.procedure_name`など)。その他の形式のSQL文を含めないでください。アプリケーション・パッケージ・プロシージャまたはスタンドアロン・プロシージャを使用している場合は、オブジェクトに対するEXECUTE権限を持つDVSYSを指定する必要があります。プロシージャ・シグネチャは、次の2つの書式のいずれかになります。

  - `PROCEDURE my_ruleset_handler(p_ruleset_name IN VARCHAR2, p_ruleset_rules IN BOOLEAN)`: ハンドラ処理にルール・セットの名前およびその戻り値が必要な場合に、この書式を使用します。
  - `PROCEDURE my_ruleset_handler`: ハンドラ処理にルール・セットの名前と戻り値が不要な場合に、この書式を使用します。

  起動者権限プロシージャをイベント・ハンドラとして使用できないことに注意してください。使用した場合、ルール・セットの評価が予想外に失敗することがあります。定義者権限プロシージャのみをイベント・ハンドラとして使用してください。

  次の構文を使用します。

  ```
  myschema.my_ruleset_handler
  ```

- 監査オプション: 次のオプションから選択すると、非統合監査環境でルール・セットの監査レコードが生成されます。Oracle Database Vaultは、監査証跡を`DVSYS.AUDIT_TRAIL$`表に書き込みます。(統合監査が有効な場合、この設定では監査レコードは取得されません。かわりに、この情報を取得する統合監査ポリシーを作成する必要があります。)
  - 監査無効: どのような場合にも監査レコードは作成されません。
  - 成功時に監査: ルール・セットがTrueに評価されると、監査レコードが作成されます。
  - 失敗時に監査: ルール・セットがFalseに評価されたとき、または関連付けられたルールのいずれかに無効なPL/SQL式が含まれているときに、監査レコードが作成されます。
  - 成功時または失敗時に監査: ルール・セットが評価されるたびに監査レコードが作成されます。

9. 「次へ」をクリックして確認ページを表示します。
10. 設定を確認して、納得のいく場合は「終了」をクリックします。

関連項目:

- `DVSYS.AUDIT_TRAIL$`表の監査レコードの詳細は、「Oracle Database Vaultの監査」を参照してください
- Database Vaultの統合監査ポリシーの作成については、『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: ルール・セットの構成

5.6 ルール・セットに追加するルールの作成

ルールは制御する動作を定義し、ルール・セットは名前付きのルールの集合です。

- ルールの作成について
  ルールは、ルール・セットの作成プロセス中またはそれに関係なく作成できます。
- デフォルト・ルール
  デフォルト・ルールとは、アクションがtrueとfalseのいずれに評価されるかのチェックなど、一般的に使用される動作が含まれるルールです。
- 新規ルールの作成
  Enterprise Manager Cloud Controlに新しいルールを作成できます。
- 既存のルールのルール・セットへの追加
  1つ以上のルールを作成すると、Enterprise Manager Cloud Controlを使用してルール・セットに追加できます。
- ルール・セットからのルールの削除
  ルール・セットからルールを削除する前に、Cloud Controlを使用してそのルールへの様々な参照を見つけることができます。

親トピック: ルール・セットの構成

5.6.1 ルールの作成について

ルールは、ルール・セットの作成プロセス中またはそれに関係なく作成できます。

ルールを作成したら、ルール・セットを1つ以上の追加ルールに関連付けられます。

ルール・セットの作成プロセス中に新しいルールを作成する場合、そのルールは、現在のルール・セットに自動で追加されます。既存のルールをルール・セットに追加することもできます。また、ルール・セットにルールを追加せずに、今後作成するルール・セットのテンプレートとして使用することもできます。

ルールをルール・セットに必要なだけ追加できますが、適切な設計とパフォーマンスの向上のために、ルール・セットを簡単にしておく必要があります。その他のアドバイスについては、「ルール・セットの設計のガイドライン」を参照してください。

ルール・セットの評価は、評価オプション(「すべてTrue」または「いずれかTrue」)を使用するルールの評価に依存します。ルール・セットが無効である場合、Oracle Database Vaultは、ルールを評価せずにルール・セットをTrueに評価します。

関連トピック

- ルール・セットの動作

親トピック: ルール・セットに追加するルールの作成

5.6.2 デフォルト・ルール

デフォルト・ルールとは、アクションがtrueとfalseのいずれに評価されるかのチェックなど、一般的に使用される動作が含まれるルールです。

DBA_DV_RULEデータ・ディクショナリ・ビューを問い合せることで、ルールをすべて示すリストを確認できます。表5-1に、現在のデフォルトOracle Databaseルールを示します。

表5-1 Oracle Database Vaultの現在のデフォルト・ルール

ルール	説明
バックアップ・リストア・パラメータが許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 現在の SQL 文が RECYCLEBIN パラメータを有効にしようとするかどうかを確認します。
データベース・ファイル・パラメータが許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 現在の SQL 文が制御ファイル関連の構成を変更しようとするかどうかを確認します。
ダンプ・パラメータが許可されているか	現在の SQL 文がダンプの宛先に関連する初期化パラメータを変更しようとするかどうかを確認します。
宛先パラメータが許可されているか	現在の SQL 文がダンプのサイズ制限に関連する初期化パラメータを変更しようとするかどうかを確認します。
ダンプまたは宛先パラメータが許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 現在の SQL 文が、ダンプのサイズ制限や宛先に関連する初期化パラメータを変更しようとするかどうかを確認します。
オプティマイザ・パラメータが許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 現在の SQL 文が OPTIMIZER_SECURE_VIEW_MERGING パラメータの設定を変更しようとするかどうかを確認します。
PL-SQL パラメータが許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 現在の SQL 文が、PLSQL_DEBUG 初期化パラメータを変更しようとするかどうかを確認します。
セキュリティ・パラメータが許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 次の初期化パラメータを無効にしようとするかどうかを確認します。 ● AUDIT_SYS_OPERATIONS ● AUDIT_TRAIL ● AUDIT_SYSLOG_LEVEL ● SQL92_SECURITY

ルール	説明
	統合監査が有効な場合、AUDIT_SYS_OPERATIONS、AUDIT_TRAIL および AUDIT_SYSLOG_LEVEL パラメータによる影響はありません。 このルールにより、次のパラメータは有効になりません。 ● OS_ROLES ● REMOTE_OS_ROLES
システム・セキュリティ・パラメータが許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 次のパラメータが変更しないようにします。 ● DYNAMIC_RLS_POLICIES ● _SYSTEM_TRIG_ENABLED
False	FALSE に評価されます
Alter DVSYS が許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 ログインしたユーザーが、他のユーザーに対する ALTER USER 文を正常に実行できるかどうかを確認します。
データベース管理者であるか	ユーザーに DBA ロールが付与されているかどうかを確認します。
Drop User が許可されているか	ログインしたユーザーがユーザーを削除できるかどうかを確認します。
ブロックのダンプが許可されているか	ブロックのダンプが許可されているかどうかを確認します。
月の最初の日であるか	指定した日が月の最初の日であるかどうかを確認します。
ラベル管理者であるか	ユーザーに LBAC_DBA ロールが付与されているかどうかを確認します。
月の最後の日であるか	指定した日が月の最後の日であるかどうかを確認します。
_dynamic_rls_init パラメータが許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 DYNAMIC_RLS_POLICIES パラメータが変更されないようにします。

ルール	説明
パラメータ値が False か	指定したパラメータ値が FALSE に設定されているかどうかを確認します。
パラメータ値が None か	指定したパラメータ値が NONE に設定されているかどうかを確認します。
パラメータ値が False でないか	指定したパラメータ値が<> FALSE に設定されているかどうかを確認します。
パラメータ値が None でないか	指定したパラメータ値が<> NONE に設定されているかどうかを確認します。
パラメータ値が Off でないか	指定したパラメータ値が<> OFF に設定されているかどうかを確認します。
パラメータ値が On でないか	指定したパラメータ値が<> ON に設定されているかどうかを確認します。
パラメータ値が True でないか	指定したパラメータ値が<> TRUE に設定されているかどうかを確認します。
パラメータ値が Off か	指定したパラメータ値が OFF に設定されているかどうかを確認します。
パラメータ値が On か	指定したパラメータ値が ON に設定されているかどうかを確認します。
パラメータ値が True か	指定したパラメータ値が TRUE に設定されているかどうかを確認します。
SYS または SYSTEM ユーザーであるか	ユーザーが SYS または SYSTEM であるかどうかを確認します。
セキュリティ管理者であるか	ユーザーに DV_ADMIN ロールが付与されているかどうかを確認します。
セキュリティ所有者であるか	ユーザーに DV_OWNER ロールが付与されているかどうかを確認します。
ユーザー・マネージャであるか	ユーザーに DV_ACCTMGR ロールが付与されているかどうかを確認します。
_system_trig_enabled パラメータが許可されているか	ノート: このデフォルト・ルールは非推奨になりました。 ユーザーが次のシステム・パラメータを変更しようとするかどうかを確認しますが、データベース・リカバリ操作では、このルールにより、これらのパラメータの変更が許可されます。 ● AUDIT_SYS_OPERATIONS: ユーザーがこのルールを FALSE に設定できないようにします。

ルール	説明
	• AUDIT_TRAIL: ユーザーがこのルールを NONE または FALSE に設定できないようにします。 • AUDIT_SYSLOG_LEVEL: このパラメータに対するすべての操作をブロックします。 • CONTROL_FILES: すべての操作をブロックします。 • OPTIMIZER_SECURE_VIEW_MERGING: ユーザーがこのルールを TRUE に設定できないようにします。 • OS_ROLES: ユーザーがこのルールを TRUE に設定できないようにします。 • PLSQL_DEBUG: ユーザーがこのルールを ON に設定できないようにします。 • RECYCLEBIN: ユーザーがこのルールを ON に設定できないようにします。 • REMOTE_OS_ROLES: ユーザーがこのルールを TRUE に設定できないようにします。 • SQL92_SECURITY: ユーザーがこのルールを FALSE に設定できないようにします。 統合監査が有効な場合、AUDIT_SYS_OPERATIONS、AUDIT_TRAIL および AUDIT_SYSLOG_LEVEL パラメータによる影響はありません。
ログイン・ユーザーがオブジェクト・ユーザーである	ログインしたユーザーが、現在の SQL 文で変更されようとしているユーザーと同じであるかどうかを確認します。
EXEMPT ACCESS POLICY ロールがない	ユーザーに EXEMPT ACCESS POLICY ロールが付与されているかどうか、またはユーザーSYS であるかどうかを確認します。
エクスポート・セッションではない	廃止
True	TRUE に評価されます

親トピック: ルール・セットに追加するルールの作成

5.6.3 新規ルールの作成

Enterprise Manager Cloud Controlに新しいルールを作成できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「ルール」をクリックします。
3. 「作成」ボタンをクリックします。
4. 「ルールの作成」ページで、次の設定を入力します。
    - 名前: ルールの名前を入力します。大/小文字混在で最大で90文字まで入力できます。

      名前は動詞で始まり、ルールの目的で終わることをお薦めします。たとえば:

      ```
      Prevent non-admin access to SQL*Plus
      ```

      ルールには「説明」フィールドがないため、明示的な名前を指定してください。ただし、90文字は超えないようにしてください。

    - ルール式: 次の要件に一致するPL/SQL式を入力します。
        - SQLのWHERE句で有効です。
        - 次に示すような、独立していて有効なPL/SQLブール式です。

          ```
          TO_CHAR(SYSDATE,'HH24') = '12'
          ```

        - ブール(TRUEまたはFALSE)値と評価される必要があります。
        - 1024文字以内である必要があります。
        - 現行のデータベース・インスタンスから既存のコンパイルされたPL/SQLファンクションを含めることができます。完全修飾ファンクションであることを確認してください(schema. function_name)。その他の形式のSQL文を含めないでください。

          起動者権限プロシージャとルール式を一緒に使用できないことに注意してください。一緒に使用すると、ルール式が予想外に失敗します。定義者権限プロシージャのみルール式と組み合せて使用できます。

          アプリケーション・パッケージ・ファンクションまたはスタンドアロン・ファンクションを使用する場合は、ファンクションのEXECUTE権限のあるDVSYSアカウントを付与する必要があります。これを行うことで、新しいルールを追加するときに発生するエラーが少なくなります。

        - ルールが機能することを確認してください。SQL*Plusで次の文を実行すると、構文をテストできます。

          ```
          SELECT rule_expression FROM DUAL;
          ```

          たとえば、次のルール式を作成したとします。

          ```
          SYS_CONTEXT('USERENV','SESSION_USER') != 'TSMITH'
          ```

          この式は、次のようにテストできます。

          ```
          SELECT SYS_CONTEXT('USERENV','SESSION_USER') FROM DUAL;
          ```

          以前にリストしたブール例の場合、次のように入力します。

```
SELECT TO_CHAR(SYSDATE,'HH24')FROM DUAL;
```

5. 「OK」をクリックします。

関連トピック

- Oracle Database VaultのPL/SQLルール・セット・ファンクション
- DBMS_MACADMルール・セットのプロシージャ
- Oracle Database VaultユーティリティのAPI

親トピック: ルール・セットに追加するルールの作成

5.6.4 既存のルールのルール・セットへの追加

1つ以上のルールを作成すると、Enterprise Manager Cloud Controlを使用してルール・セットに追加できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「ルール・セット」をクリックします。
3. 既存のルールの追加先のルール・セットを選択して、「編集」をクリックします。
4. 「ルールとの関連付け」ページになるまで、「次へ」をクリックします。
5. 「既存のルールの追加」をクリックして、「既存のルールの追加」ダイアログ・ボックスを表示します。
6. 「既存のルールの追加」ページで、対象のルールを選択して「移動」(すべて移動する場合は「すべて移動」)をクリックし、それらを「選択したルール」リストに移動します。

   [Ctrl]キーを押しながら各ルールをクリックすると、複数のルールを選択できます。

7. 「OK」をクリックします。
8. 「完了」、「終了」の順にクリックします。

親トピック: ルール・セットに追加するルールの作成

5.6.5 ルール・セットからのルールの削除

ルール・セットからルールを削除する前に、Cloud Controlを使用してそのルールへの様々な参照を見つけることができます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「ルール・セット」をクリックします。

   削除するルールを含むルール・セットが不明な場合は、「Database Vaultコンポーネント」から「ルール」を選択し、削除するルールを選択してから、「表示」オプション(ただし、「表示」メニューではない)を選択します。そのルールに関連付けられているルール・セットが「ルール・セットの使用方法」に示されます。

3. 既存のルールの追加先のルール・セットを選択して、「編集」をクリックします。
4. 「ルールとの関連付け」ページになるまで、「次へ」をクリックします。
5. 削除するルールを選択して、「削除」をクリックします。
6. 「完了」、「終了」の順にクリックします。

ルール・セットからルールを削除しても、そのルールはまだ存在します。必要に応じて、そのルールをその他のルール・セットに関連

付けることができます。ルールを削除する場合、「ルール」ページから行うことができます。

親トピック: ルール・セットに追加するルールの作成

5.7 Oracle Database Vaultコンポーネントへのルール・セット参照の削除

ルール・セットを削除する前に、ルール・セットのOracle Database Vaultコンポーネントへの参照を削除する必要があります。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 削除するルール・セットの参照を見つけます。

   「ルール・セット」ページで、ルール・セットを選択して「表示」ボタン(「表示」メニューではありません)をクリックします。
   「ルール・セットの表示」ページで、削除するルール・セットの参照の「ルールセットの使用方法」領域を確認します。
   「OK」をクリックします。

3. 「管理」ページの「Database Vaultコンポーネント」で、ルール・セットの参照を含むコンポーネントを選択します(「レルム」など)。
4. オブジェクトを選択して「編集」をクリックします。
5. 認可ページになるまで、「次へ」をクリックします。
6. ルール・セットとともに認可を選択してから、「編集」をクリックし、参照されているオブジェクトを削除します。
7. 「完了」、「終了」の順にクリックします。

親トピック: ルール・セットの構成

5.8 ルール・セットの削除

Enterprise Manager Cloud Controlを使用して、ルール・セットへの参照を見つけてからルール・セットを削除できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. ルール・セットへの参照を削除します。
3. 削除するルール・セットを選択して、「削除」をクリックします。
4. 「確認」ウィンドウで、「はい」をクリックします。

   ルール・セットが削除されます。オプションで、ルール・セットを削除する前に、ルールとの既存の関連付けを削除できます。

関連トピック

- Oracle Database Vaultコンポーネントへのルール・セット参照の削除

親トピック: ルール・セットの構成

5.9 ルール・セットの動作

ルール・セットの動作を理解すると、より効果的なルール・セットを作成できます。

- Oracle Database Vaultによるルールの評価方法
  Oracle Database Vaultは、ルール・セット内のルールを式の集合として評価します。
- ルール・セット内でのネストされたルール

ルール・セット内に1つ以上のルールをネストできます。

- 1人のユーザーを除く全員に適用するルールの作成
  1人のユーザー(たとえば特権ユーザー)を除く全員に適用するルールを作成することも可能です。

親トピック: ルール・セットの構成

5.9.1 Oracle Database Vaultによるルールの評価方法

ルール・セット内のルールは、式の集合として評価されます。

「評価オプション」が「すべてTrue」に設定されている場合にルールの評価に失敗すると、ルール・セット内の残りのルールの評価は試行されず、その時点で評価が停止します。同様に、「評価オプション」が「いずれかTrue」に設定されている場合にルールがTrueと評価されると、評価はその時点で停止します。ルール・セットが無効である場合、Oracle Database Vaultは、ルールを評価せずにルール・セットをTrueと評価します。

親トピック: ルール・セットの動作

5.9.2 ルール・セット内でのネストされたルール

ルール・セット内に1つ以上のルールをネストできます。

たとえば、Is Corporate Network During Maintenanceという、次の2つのタスクを実行するネストされたルールを作成するとします。

- データベース・セッションが企業ネットワーク内から発生した場合のみ、表の変更を制限します。
- 午後10時から午後10時59分の間にスケジュールされているシステム・メンテナンス・ウィンドウに、表の変更を制限します。

ルールの定義は次のようになります。

```
DVF.F$NETWORK = 'Corporate' AND TO_CHAR(SYSDATE,'HH24') between '22' AND '23'
```

関連トピック

- Oracle Database VaultのDVF PL/SQLファクタ・ファンクション
- ファクタの構成

親トピック: ルール・セットの動作

5.9.3 1人のユーザーを除く全員に適用するルールの作成

1人のユーザー(たとえば特権ユーザー)を除く全員に適用するルールを作成することも可能です。

- 特定のユーザーを除外するルールを作成するには、SYS_CONTEXTファンクションを使用します。

たとえば:

```
SYS_CONTEXT('USERENV','SESSION_USER') = 'SUPERADMIN_USER' OR additional_rule
```

現行ユーザーが特権ユーザーの場合、システムでは、additional_ruleは評価されず、ルールはTrueに評価されます。現行ユーザーが特権ユーザーでない場合、ルールの評価はadditional_ruleの評価によって決まります。

親トピック: ルール・セットの動作

5.10 チュートリアル: セキュリティ違反の電子メール・アラートの作成

このチュートリアルは、UTL_MAIL PL/SQLパッケージおよびアクセス制御リストを使用して、セキュリティ違反の電子メール・アラートを作成する方法を示します。

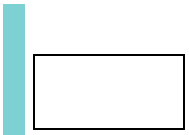
- このチュートリアルについて
  チュートリアルでは、ユーザーがメンテナンス期間外に表を変更しようとしたときに送信される、電子メール・アラートを作成します。
- ステップ1: UTL_MAIL PL/SQLパッケージのインストールおよび構成
  手動でインストールする必要があるUTL_MAIL PL/SQLパッケージには、電子メール通知を管理するためのプロシージャが含まれています。
- ステップ2: 電子メール・セキュリティ・アラートPL/SQLプロシージャの作成
  ユーザーleo_dvownerは、CREATE PROCEDURE文を使用して、電子メール・セキュリティ・アラートを作成できます。
- ステップ3: ネットワーク・サービス用のアクセス制御リストの構成
  UTL_MAILを使用するには、あらかじめ、外部ネットワーク・サービスに対してファイングレイン・アクセスを有効にするアクセス制御リスト(ACL)を構成する必要があります。
- ステップ4: 電子メール・セキュリティ・アラートを使用するためのルール・セットおよびコマンド・ルールの作成
  ルール・セットおよびコマンド・ルールを作成するには、DBMS_MACADM PL/SQLパッケージを使用します。
- ステップ5: 電子メール・セキュリティ・アラートのテスト
  アラートを作成すると、テストする準備ができます。
- ステップ6: このチュートリアルのコンポーネントの削除
  コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

親トピック: ルール・セットの構成

5.10.1 このチュートリアルについて

チュートリアルでは、ユーザーがメンテナンス期間外に表を変更しようとしたときに送信される、電子メール・アラートを作成します。

これを行うには、メンテナンスの時間帯を設定するルールを作成し、このルールをルール・セットに追加してから、ユーザーに表の変更を許可するコマンド・ルールを作成する必要があります。次に、ルール・セットをこのコマンド・ルールに関連付けます。これにより、ユーザーがメンテナンスの時間帯以外にALTER TABLE SQL文を使用しようとすると、電子メール・アラートが送信されます。



ノート:

このチュートリアルを実行するには、SMTP サーバーがあるデータベースを使用する必要があります。

親トピック: チュートリアル: セキュリティ違反の電子メール・アラートの作成

5.10.2 ステップ1: UTL_MAIL PL/SQLパッケージのインストールおよび構成

手動でインストールする必要があるUTL_MAIL PL/SQLパッケージには、電子メール通知を管理するためのプロシージャが含まれています。

1. SYSDBA管理権限を使用してSYSとしてデータベース・インスタンスにログインします。

```
sqlplus sys as sysdba
Enter password: password
```

2. マルチテナント環境で、適切なプラガブル・データベース(PDB)に接続します。

   たとえば:

   ```
   CONNECT SYS@my_pdb AS SYSDBA
   Enter password: password
   ```

   使用可能なPDBを見つけるには、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。

3. UTL_MAILパッケージをインストールします。

   ```
   @$ORACLE_HOME/rdbms/admin/utlmail.sql
   @$ORACLE_HOME/rdbms/admin/prvtmail.plb
   ```

   UTL_MAILパッケージにより、電子メールの管理が可能になります。UTL_MAIL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。ただし、現在、UTL_MAIL PL/SQLパッケージではSSLサーバーをサポートしていないことに注意してください。

4. SMTP_OUT_SERVERパラメータの現行値を調べ、このチュートリアル完了時に元に戻せるように、この値をノートにとっておきます。

   たとえば:

   ```
   SHOW PARAMETER SMTP_OUT_SERVER
   ```

   次のような出力が表示されます。

   ```
   NAME                 TYPE             VALUE
   -------------------- ---------------- --------------------------------
   SMTP_OUT_SERVER      string           some_value.example.com
   ```

5. 次のALTER SYSTEM文を発行します。

   ```
   ALTER SYSTEM SET SMTP_OUT_SERVER="imap_mail_server.example.com";
   ```

   imap_mail_server.example.comを、電子メール・ツールのアカウント設定にあるSMTPサーバーの名前に置き換えます。これらの設定を二重引用符で囲んでください。たとえば:

   ```
   ALTER SYSTEM SET SMTP_OUT_SERVER="my_imap_mail_server.example.com"
   ```

6. SYSOPER権限を使用してSYSとして接続し、データベースを再起動します。

   ```
   CONNECT SYS AS SYSOPER -- Or, CONNECT SYS@hrpdb AS SYSOPER
   Enter password: password
   SHUTDOWN IMMEDIATE
   STARTUP
   ```

7. SMTP_OUT_SERVERパラメータの設定が正しいことを確認します。

   ```
   CONNECT SYS AS SYSDBA -- Or, CONNECT SYS@hrpdb AS SYSDBA
   Enter password: password
   SHOW PARAMETER SMTP_OUT_SERVER
   ```

   次のような出力が表示されます。

   ```
   NAME                 TYPE             VALUE
   -------------------- ---------------- --------------------------------
   ```

```
SMTP_OUT_SERVER          string              my_imap_mail_server.example.com
```

親トピック: チュートリアル: セキュリティ違反の電子メール・アラートの作成

5.10.3 ステップ2: 電子メール・セキュリティ・アラートPL/SQLプロシージャの作成

ユーザーleo_dvownerは、CREATE PROCEDURE文を使用して、電子メール・セキュリティ・アラートを作成できます。

1. このステップで説明する付与を実行する権限を持つユーザーとして接続しているか確認し、DV_OWNERロールを付与されているユーザーにそのような権限を付与します。また、Oracleシステム権限およびロール管理レルムの所有者として認可されている必要があります。

   (DV_ADMINロールを付与されているユーザーを選択することもできますが、このチュートリアルでは、DV_OWNERロールを持つユーザーを選択します。)

   たとえば:

   ```
   CONNECT dba_psmith -- Or, CONNECT dba_psmith@hrpdb
   Enter password: password
   GRANT CREATE PROCEDURE, DROP ANY PROCEDURE TO leo_dvowner;
   GRANT EXECUTE ON UTL_TCP TO leo_dvowner;
   GRANT EXECUTE ON UTL_SMTP TO leo_dvowner;
   GRANT EXECUTE ON UTL_MAIL TO leo_dvowner;
   GRANT EXECUTE ON DBMS_NETWORK_ACL_ADMIN TO leo_dvowner;
   ```

   PL/SQLパッケージUTL_TCP、UTL_SMTP、UTL_MAILおよびDBMS_NETWORK_ACL_ADMINは、作成する電子メール・セキュリティ・アラートで使用されます。

2. DV_OWNERユーザーとしてSQL*Plusに接続します。

   たとえば:

   ```
   CONNECT leo_dvowner -- Or, CONNECT leo_dvowner@hrpdb
   Enter password: password
   ```

3. 次のプロシージャを作成します。

   ```
   CREATE OR REPLACE PROCEDURE email_alert AS
   msg varchar2(20000) := 'Realm violation occurred for the ALTER TABLE Command
   Security Policy rule set. The time is: ';
   BEGIN
     msg := msg||to_char(SYSDATE, 'Day DD MON, YYYY HH24:MI:SS');
   UTL_MAIL.SEND (
       sender      => 'youremail@example.com',
       recipients  => 'recipientemail@example.com',
       subject     => 'Table modification attempted outside maintenance!',
       message     => msg);
   END email_alert;
   /
   ```

   youremail@example.comを自分の電子メール・アドレスに置き換え、recipientemail@example.comを通知を受け取るユーザーの電子メール・アドレスに置き換えます。

4. このプロシージャに対するEXECUTE権限をDVSYSに付与します。

   ```
   GRANT EXECUTE ON email_alert TO DVSYS;
   ```

親トピック: チュートリアル: セキュリティ違反の電子メール・アラートの作成

5.10.4 ステップ3: ネットワーク・サービス用のアクセス制御リストの構成

UTL_MAILを使用するには、あらかじめ、外部ネットワーク・サービスに対してファイングレイン・アクセスを有効にするアクセス制御リスト(ACL)を構成する必要があります。

外部ネットワーク・サービスへのファイングレイン・アクセスの詳細は、『Oracle Databaseセキュリティ・ガイド』を参照してください。

1. SQL*Plusで、DV_OWNERユーザーとして次のアクセス制御設定およびその権限定義を構成します。

```
BEGIN
 DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
  host       => 'SMTP_OUT_SERVER_setting',
  lower_port => 25,
  ace        =>  xs$ace_type(privilege_list => xs$name_list('smtp'),
                             principal_name => 'LEO_DVOWNER,
                             principal_type => xs_acl.ptype_db));
END;
/
```

この例の説明は、次のとおりです。

- lower_port: 電子メール・ツールで送信サーバーに指定されているポート番号を入力します。通常、この設定は25です。lower_portとupper_portの両方の設定に対してこの値を入力します。(現在、UTL_MAILパッケージではSSLをサポートしていません。メール・サーバーがSSLサーバーの場合、そのメール・サーバーが別のポート番号を使用していても、ポート番号に25を入力します。)
- principal_name: LEO_DVOWNERを、DV_OWNERユーザーの名前に置き換えます。
- host: SMTP_OUT_SERVER_settingの場合、「ステップ1: UTL_MAIL PL/SQLパッケージのインストールと構成」のSMTP_OUT_SERVERパラメータに設定したSMTP_OUT_SERVER設定を入力します。この設定は、電子メール・ツールで送信サーバーに指定されている設定と完全に一致させてください。

2. 変更をデータベースにコミットします。

```
COMMIT;
```

3. これまでに作成した設定をテストします。

```
EXEC EMAIL_ALERT;
COMMIT;
```

SQL*Plusに「PL/SQL procedure successfully completed」というメッセージが表示されます。まもなく、電子メール・サーバーの速度に応じて、電子メール・アラートを受信します。

ORA-24247: 「アクセス制御リスト(ACL)によりネットワーク・アクセスが拒否されました」エラーの後にORA-06512: 「string行 string」エラーが発生した場合は、アクセス制御リスト・ファイル内の設定を確認してください。

親トピック: チュートリアル: セキュリティ違反の電子メール・アラートの作成

5.10.5 ステップ4: 電子メール・セキュリティ・アラートを使用するためのルール・セットおよびコマンド・ルールの作成

ルール・セットおよびコマンド・ルールを作成するには、DBMS_MACADM PL/SQLパッケージを使用します。

1. DV_OWNERユーザーとして、次のルール・セットを作成します。

```
BEGIN
```

```
 DBMS_MACADM.CREATE_RULE_SET(
rule_set_name    => 'ALTER TABLE Command Security Policy',
description      => 'This rule set allows ALTER TABLE only during the
                       maintenance period.',
enabled          => DBMS_MACUTL.G_YES,
eval_options     => DBMS_MACUTL.G_RULESET_EVAL_ALL,
audit_options    => DBMS_MACUTL.G_RULESET_AUDIT_FAIL,
fail_options     => DBMS_MACUTL.G_RULESET_FAIL_SILENT,
fail_message     =>'',
fail_code        => NULL,
handler_options  => DBMS_MACUTL.G_RULESET_HANDLER_FAIL,
handler          => 'leo_dvowner.email_alert');
END;
/
```

2. 次のようなルールを作成します。

   ここでは、テストする間持続するようにルールを設定します。たとえば、午後2時から午後3時までの時間帯にテストする場合、次のようにルールを作成します。

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name  => 'Restrict Access to Maintenance Period',
  rule_expr  => 'TO_CHAR(SYSDATE,''HH24'') BETWEEN ''14'' AND ''15''');
END;
/
```

   HH24、14および15には、二重引用符ではなく、必ず2つの一重引用符を使用してください。

   コンピュータのシステム時間は、次のSQL文を発行してチェックできます。

```
SELECT TO_CHAR(SYSDATE,'HH24') FROM DUAL;
```

   次のような出力が表示されます。

```
TO
--
14
```

   後から、ルールの動作に問題がなければ、次のように、サイトで通常メンテナンス作業が実行される時間(たとえば、午後7時から午後10時までの間)にルールを更新できます。

```
BEGIN
 DBMS_MACADM.UPDATE_RULE(
  rule_name  => 'Restrict Access to Maintenance Period',
  rule_expr  => 'TO_CHAR(SYSDATE,''HH24'') BETWEEN ''16'' AND ''22''');
END;
/
```

3. Restrict Access to Maintenance PeriodルールをALTER TABLE Command Security Policyルール・セットに追加します。

```
BEGIN
 DBMS_MACADM.ADD_RULE_TO_RULE_SET(
  rule_set_name => 'ALTER TABLE Command Security Policy',
  rule_name     => 'Restrict Access to Maintenance Period');
END;
/
```

4. 次のコマンド・ルールを作成します。

```
BEGIN
 DBMS_MACADM.CREATE_COMMAND_RULE(
```

```
command          => 'ALTER TABLE',
rule_set_name    => 'ALTER TABLE Command Security Policy',
object_owner     => 'SCOTT',
object_name      => '%',
enabled          => DBMS_MACUTL.G_YES);
END;
/
```

5. これらの更新をデータベースにコミットします。

```
COMMIT;
```

親トピック: チュートリアル: セキュリティ違反の電子メール・アラートの作成

5.10.6 ステップ5: 電子メール・セキュリティ・アラートのテスト

アラートを作成すると、テストする準備ができます。

1. ユーザーSCOTTとしてSQL*Plusに接続します。

   たとえば:

   ```
   CONNECT SCOTT -- Or, CONNECT SCOTT@hrpdb
   Enter password: password
   ```

   SCOTTアカウントがロックされて無効になっている場合、DV_ACCTMGRロールを持つユーザーが、このアカウントのロックを解除し、新しいパスワードを次のように作成できます。

   ```
   ALTER USER SCOTT ACCOUNT UNLOCK IDENTIFIED BY password;
   ```

   『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

2. ユーザーSCOTTとして、テスト表を作成します。

   ```
   CREATE TABLE mytest (col1 number);
   ```

3. コンピュータのシステム時間を、ALTER TABLE Command Security Policyルール・セットが実行される時間に変更します。

   たとえば、テスト時間帯を午後2時から午後3時の間に設定する場合、次のようにします。

   UNIX: rootとしてログインし、dateコマンドを使用して時間を設定します。たとえば、今日の日付が2012年8月15日だとすると、次のように入力します。

   ```
   $ su root
   Password: password
   $ date -s "08/15/2012 14:48:00"
   ```

   Windows: 通常画面の右下隅にある時計アイコンをダブルクリックします。「日付と時刻のプロパティ」ウィンドウで、時刻を午後2時に設定し、「OK」をクリックします。

4. my_test表の変更を試みます。

   ```
   ALTER TABLE mytest ADD (col2 number);
   Table altered.
   ```

   SCOTTは、この時間帯にmytest表を変更できます。

5. システム時間をRestrict Access to Maintenance Period時間外の時刻に再設定します。

6. SCOTTとしてログインし、再度my_test表の変更を試みます。

```
CONNECT SCOTT -- Or, CONNECT SCOTT@hrpdb
Enter password: password
ALTER TABLE mytest ADD (col3 number);
```

次の出力が表示されます。

```
ORA-47400: Command Rule violation for ALTER TABLE on SCOTT.MYTEST
```

SCOTTはmytest表を変更できません。まもなく、「Table modification attempted outside maintenance!」という件名で、次のようなメッセージの電子メールを受信します。

```
Realm violation occurred for the ALTER TABLE Command Security Policy rule set.
The time is: Wednesday 15 AUG, 2012 14:24:25
```

7. システム時間を正しい時刻に再設定します。

親トピック: チュートリアル: セキュリティ違反の電子メール・アラートの作成

5.10.7 ステップ6: この例で使用したコンポーネントの削除

コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

1. DV_OWNERユーザーとしてSQL*Plusに接続します。

```
CONNECT sec_admin_owen -- Or, CONNECT sec_admin_owen@hrpdb
Enter password: password
```

2. 表示された順序で、Oracle Database Vaultのルール・コンポーネントを削除します。

```
EXEC DBMS_MACADM.DELETE_RULE_FROM_RULE_SET('ALTER TABLE Command Security
Policy', 'Restrict Access to Maintenance Period');
EXEC DBMS_MACADM.DELETE_RULE('Restrict Access to Maintenance Period');
EXEC DBMS_MACADM.DELETE_COMMAND_RULE('ALTER TABLE', 'SCOTT', '%');
EXEC DBMS_MACADM.DELETE_RULE_SET('ALTER TABLE Command Security Policy');
```

3. email_alert PL/SQLプロシージャを削除します。

```
DROP PROCEDURE email_alert;
```

4. ユーザーSCOTTとして接続し、mytest表を削除します。

```
CONNECT SCOTT -- Or, CONNECT SCOTT@hrpdb
Enter password: password
DROP TABLE mytest;
```

5. 他のユーザーから権限を取り消す権限を持つユーザーとして接続します。

たとえば:

```
CONNECT accts_admin_ace -- Or, CONNECT accts_admin_ace@hrpdb
Enter password: password
```

6. DV_OWNERユーザーから、UTL_TCP、UTL_SMTPおよびUTL_MAIL PL/SQLパッケージに対するEXECUTE権限を取り消します。

たとえば:

```
REVOKE EXECUTE ON UTL_TCP FROM leo_dvowner;
REVOKE EXECUTE ON UTL_SMTP FROM leo_dvowner;
REVOKE EXECUTE ON UTL_MAIL FROM leo_dvowner;
```

```
REVOKE EXECUTE ON DBMS_NETWORK_ACL_ADMIN FROM leo_dvowner;
```

7. SMTP_OUT_SERVERパラメータを元の値に設定します。

   たとえば:

   ```
   ALTER SYSTEM SET SMTP_OUT_SERVER="some_value.example.com";
   ```

8. SYSOPER管理権限を持つSYSとして接続し、データベースを再起動します。

   ```
   CONNECT SYS AS SYSOPER -- Or, CONNECT SYS@hrpdb AS SYSOPER
   Enter password: password
   SHUTDOWN IMMEDIATE
   STARTUP
   ```

親トピック: チュートリアル: セキュリティ違反の電子メール・アラートの作成

5.11 チュートリアル: 二人制整合性(デュアル・キー・セキュリティ)の構成

このチュートリアルでは、Oracle Database Vaultを使用して2人のユーザーの認可を制御する方法を示します。

- このチュートリアルについて
  このチュートリアルでは、二人制整合性(TPI)を定義するルール・セットを構成します。
- ステップ1: このチュートリアル用のユーザーの作成
  このチュートリアルでは、2人のユーザーpatch_bossおよびpatch_userを作成する必要があります。
- ステップ2: ユーザーpatch_bossがログインしているかどうかをチェックするファンクションの作成
  Database Vault設定の動作は、このファンクションによって決定されます。
- ステップ3: ユーザー・アクセスを制御するためのルール、ルール・セットおよびコマンド・ルールの作成
  次に、2つのルール、それらを追加するルール・セットおよびコマンド・ルールを作成します。
- ステップ4: ユーザーのアクセスのテスト
  ルールを作成すると、テストする準備ができます。
- ステップ5: このチュートリアルのコンポーネントの削除
  コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

親トピック: ルール・セットの構成

5.11.1 このチュートリアルについて

このチュートリアルでは、二人制整合性(TPI)を定義するルール・セットを構成します。

この機能は、デュアル・キー・セキュリティ、デュアル・キー接続および二人制ルール・セキュリティとも呼ばれます。このタイプのセキュリティでは、アクションの認可に、1人ではなく2人のユーザーが必要です。

あるユーザーがタスクを開始するには、別のユーザーがそのユーザーに対するセーフティ・チェックを行います。二人制整合性では、危険を伴う可能性のあるアクションに対して、追加のセキュリティの層が用意されます。このタイプのシナリオは、データベース・パッチの更新などのタスクに使用されることが多く、このチュートリアルでもこのタスクを使用します。ユーザーpatch_userがデータベース・パッチのアップグレードを実行するにはログインが必要ですが、このユーザーがログインするにはマネージャpatch_bossがログインしている必要があります。patch_userがログイン可能かどうかを制御するファンクション、ルール、ルール・セットおよびコマンド・ルールを作成します。

親トピック: チュートリアル: 二人制整合性(デュアル・キー・セキュリティ)の構成

5.11.2 ステップ1: このチュートリアル用のユーザーの作成

このチュートリアルでは、2人のユーザーpatch_bossおよびpatch_userを作成する必要があります。

- patch_bossはスーパーバイザ・ロールとして機能します。patch_bossがログインしていない場合、patch_userユーザーはログインできません。
- patch_userは、パッチの更新の実行が割り当てられているユーザーです。ただし、このチュートリアルでは、ユーザーはpatch_user実際にはパッチの更新を実行しません。ログインを試行するのみです。

ユーザーを作成するには、次のようにします。

1. DV_ACCTMGRロールを付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus accts_admin_ace
   Enter password: password
   ```

   マルチテナント環境で、適切なプラガブル・データベース(PDB)にログインする必要があります。たとえば:

   ```
   sqlplus accts_admin_ace@hrpdb
   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。

2. 次のユーザーを作成し、CREATE SESSION権限を付与します。

   ```
   GRANT CREATE SESSION TO patch_boss IDENTIFIED BY password;
   GRANT CREATE SESSION TO patch_user IDENTIFIED BY password;
   ```

   『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

3. SYSDBA管理権限を持つユーザーSYSとして接続します。

   ```
   CONNECT SYS AS SYSDBA -- Or, CONNECT SYS@hrpdb AS SYSDBA
   Enter password: password
   ```

4. 次の権限をDV_OWNERユーザーまたはDV_ADMINユーザーに付与します。

   たとえば:

   ```
   GRANT CREATE PROCEDURE TO sec_admin_owen;
   GRANT SELECT ON V_$SESSION TO sec_admin_owen;
   ```

   V_$SESSION表はV$SESSION動的ビューの基となる表です。

実際のシナリオでは、DV_OWNERユーザーとしてもログインし、DV_PATCH_ADMINロールをpatch_userユーザーに付与します(patch_bossには付与しません)。しかし、このチュートリアルでは、実際にはデータベース・パッチの更新を実行しないため、このロールをpatch_userユーザーに付与する必要はありません。

親トピック: チュートリアル: 二人制整合性(デュアル・キー・セキュリティ)の構成

5.11.3 ステップ2: ユーザーpatch_bossがログインしているかどうかをチェックするファンクションの作成

Database Vault設定の動作は、ファンクションによって決定されます。

作成する必要があるファンクションcheck_boss_logged_inは、ユーザーpatch_userがデータベース・インスタンスへのログインを試行したときに、V$SESSIONデータ・ディクショナリ・ビューに問い合せて、ユーザーpatch_bossがログインしているかどうかを確認します。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして接続します。

   たとえば:

   ```
   CONNECT sec_admin_owen -- Or, CONNECT sec_admin_owen@hrpdb
   Enter password: password
   ```

2. check_boss_logged_inファンクションを、次のように作成します。

   ```
   CREATE OR REPLACE FUNCTION check_boss_logged_in
   return varchar2
   authid definer as

   v_session_number number := 0;
   v_allow varchar2(10)    := 'TRUE';
   v_deny varchar2(10)     := 'FALSE';

   BEGIN
     SELECT COUNT(*) INTO v_session_number
     FROM SYS.V_$SESSION
     WHERE USERNAME = 'PATCH_BOSS'; -- Enter the user name in capital letters.

    IF v_session_number > 0
     THEN RETURN v_allow;
    ELSE
     RETURN v_deny;
    END IF;
   END check_boss_logged_in;
   /
   ```

3. check_boss_logged_inファンクションのEXECUTE権限をDVSYSスキーマに付与します。

   ```
   GRANT EXECUTE ON check_boss_logged_in to DVSYS;
   ```

親トピック: チュートリアル: 二人制整合性(デュアル・キー・セキュリティ)の構成

5.11.4 ステップ3: ユーザー・アクセスを制御するためのルール、ルール・セットおよびコマンド・ルールの作成

次に、2つのルール、それらを追加するルール・セットおよびコマンド・ルールを作成します。

このルール・セットは、ユーザーpatch_userがデータベースにログインしようとしたときに、check_boss_logged_inファンクションをトリガーします。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして接続します。

   たとえば:

   ```
   CONNECT sec_admin_owen -- Or, CONNECT sec_admin_owen@hrpdb
   Enter password: password
   ```

2. patch_userユーザーがデータベースにログインしていることをチェックするCheck if Boss Is Logged Inルールを作成します。定義内のsec_admin_owenを、check_boss_logged_inファンクションを作成したDVOWNERユーザーまたはDV_ADMINユーザーに置き換えます。

   check_boss_logged_inファンクションがTRUEを返した場合(つまりpatch_bossは別のセッションにログインしている)、patch_userはログインできます。

```
BEGIN
  DBMS_MACADM.CREATE_RULE(
   rule_name => 'Check if Boss Is Logged In',
   rule_expr => 'SYS_CONTEXT(''USERENV'',''SESSION_USER'') = ''PATCH_USER'' and
sec_admin_owen.check_boss_logged_in =  ''TRUE'' ');
END;
/
```

   ユーザー名PATCH_USERを大文字で入力します(SESSION_USERパラメータはユーザー名を大文字で保存します)。

3. ログインしているユーザー(patch_user)がユーザーpatch_bossでないことを確認するAllow Connect for Other Database Usersルールを作成します。他のすべての有効なユーザーのログインの許可も行います。

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Allow Connect for Other Database Users',
  rule_expr => 'SYS_CONTEXT(''USERENV'',''SESSION_USER'') != ''PATCH_USER''');
END;
/
COMMIT;
```

4. 上司とパッチ担当者のデュアル接続のルール・セットを作成して、このルール・セットに2つのルールを追加します。

```
BEGIN
    DBMS_MACADM.CREATE_RULE_SET(
     rule_set_name     => 'Dual Connect for Boss and Patch',
     description       => 'Checks if both boss and patch users are logged in.',
     enabled           => DBMS_MACUTL.G_YES,
     eval_options      => 2,
     audit_options     => DBMS_MACUTL.G_RULESET_AUDIT_FAIL,
     fail_options      => DBMS_MACUTL.G_RULESET_FAIL_SILENT,
     fail_message      =>'',
     fail_code         => NULL,
     handler_options   => DBMS_MACUTL.G_RULESET_HANDLER_OFF,
     handler           => ''
     );
END;
/
BEGIN
    DBMS_MACADM.ADD_RULE_TO_RULE_SET(
     rule_set_name     => 'Dual Connect for Boss and Patch',
     rule_name         => 'Check if Boss Is Logged In'
   );
END;
/
BEGIN
    DBMS_MACADM.ADD_RULE_TO_RULE_SET(
     rule_set_name     => 'Dual Connect for Boss and Patch',
     rule_name         => 'Allow Connect for Other Database Users'
   );
END;
/
```

5. patch_bossがログインしている場合にのみユーザーpatch_userにデータベースへの接続を許可する、次のCONNECTコマンド・ルールを作成します。

```
BEGIN
   DBMS_MACADM.CREATE_COMMAND_RULE(
    command             => 'CONNECT',
    rule_set_name       => 'Dual Connect for Boss and Patch',
    object_owner        => '%',
    object_name         => '%',
    enabled             => DBMS_MACUTL.G_YES);
END;
/
COMMIT;
```

親トピック: チュートリアル: 二人制整合性(デュアル・キー・セキュリティ)の構成

5.11.5 ステップ4: ユーザーのアクセスのテスト

ルールを作成すると、テストする準備ができます。

1. SQL*Plusを終了します。

   ```
   EXIT
   ```

2. 2つ目のシェルを作成します。たとえば:

   ```
   xterm &
   ```

3. 最初のシェルで、ユーザーpatch_userとしてログインします。

   ```
   sqlplus patch_user -- Or, sqlplus patch_user@hrpdb
   Enter password: password
   ERROR:
   ORA-47400: Command Rule violation for CONNECT on LOGON
   Enter user-name:
   ```

   patch_bossがログインするまで、ユーザーpatch_userはログインできません。(まだEnter user-nameプロンプトを試さないでください。)

4. 2つ目のシェルで、ユーザーpatch_bossとしてログインします。

   ```
   sqlplus patch_boss -- Or, sqlplus patch_boss@hrpdb
   Enter password: password
   Connected.
   ```

   ユーザーpatch_bossはログインできます。

5. 最初のシェルに戻って、ユーザーpatch_userとしてログインを再試行します。

   ```
   Enter user_name: patch_user
   Enter password: password
   ```

   ユーザーpatch_userは有効なユーザーとみなされ、ログインできます。

親トピック: チュートリアル: 二人制整合性(デュアル・キー・セキュリティ)の構成

5.11.6 ステップ5: この例で使用したコンポーネントの削除

コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

1. ユーザーpatch_bossのセッションで、SQL*Plusを終了してシェルを閉じます。

   ```
   EXIT
   ```

2. 最初のシェルで、ユーザーDV_ACCTMGRを接続し、作成したユーザーを削除します。

```
CONNECT accts_admin_ace -- Or, CONNECT accts_admin_ace@hrpdb
Enter password: password
DROP USER patch_boss;
DROP USER patch_user;
```

3. SYSDBA管理権限を持つユーザーSYSとして接続して、DV_OWNERまたはDV_ADMINユーザーに付与した権限を取り消します。

```
CONNECT SYS AS SYSDBA -- Or, CONNECT SYS@hrpdb AS SYSDBA
Enter password: password
REVOKE CREATE PROCEDURE FROM sec_admin_owen;
REVOKE SELECT ON V_$SESSION FROM sec_admin_owen;
```

4. DV_OWNERまたはDV_ADMINユーザーとして接続し、ルール、ルール・セットおよびコマンド・ルールを示した順に削除します。

```
CONNECT sec_admin_owen -- Or, CONNECT leo_dvowner@hrpdb
Enter password: password
DROP FUNCTION check_boss_logged_in;
EXEC DBMS_MACADM.DELETE_COMMAND_RULE('CONNECT', '%', '%');
EXEC DBMS_MACADM.DELETE_RULE_FROM_RULE_SET('Dual Connect for Boss and Patch',
'Check if Boss Is Logged In');
EXEC DBMS_MACADM.DELETE_RULE_FROM_RULE_SET('Dual Connect for Boss and Patch',
'Allow Connect for Other Database Users');
EXEC DBMS_MACADM.DELETE_RULE('Check if Boss Is Logged In');
EXEC DBMS_MACADM.DELETE_RULE('Allow Connect for Other Database Users');
EXEC DBMS_MACADM.DELETE_RULE_SET('Dual Connect for Boss and Patch');
COMMIT;
```

親トピック: チュートリアル: 二人制整合性(デュアル・キー・セキュリティ)の構成

5.12 ルール・セット設計のガイドライン

Oracleでは、ルール・セット設計のガイドラインを提供しています。

- ルールは、複数のルール・セット間で共有できます。これにより、再利用可能なルール式のライブラリを作成できます。個別で式の目的が1つであるルールを設計することをお薦めします。
- 評価が静的な、すなわちユーザー・セッション中に1回のみ評価されるルール・セットを設計できます。あるいは、ルール・セットにアクセスするたびに評価できます。ルール・セットの評価が1回のみの場合、ルール・セットにアクセスするたびに、評価された値がユーザー・セッション全体で再利用されます。静的な評価の使用は、ルール・セットに複数回アクセスする必要があるが、ルール・セットが依存する条件はセッション中に変化しない場合に便利です。たとえば、ルール・セットに関連付けられているSELECTコマンド・ルールは、同じSELECT文が複数回実行される際、評価された値が再利用可能であれば、SELECTが実行されるたびにルール・セットを評価するかわりに、評価された値を再利用できます。

  ルール・セットの静的な評価を制御するには、DBMS_MACADM PL/SQLパッケージのCREATE_RULE_SETまたはUPDATE_RULE_SETプロシージャのis_staticパラメータを設定します。詳細は、「DBMS_MACADMのルール・セット・プロシージャ」を参照してください。
- 再利用性とルール式で使用される値の信頼性を実現するために、ルール式にはOracle Database Vaultファクタを使用します。ファクタにはルール式で使用可能なコンテキスト情報が用意されています。
- カスタム・イベント・ハンドラを使用して、Oracle Database Vaultセキュリティ・ポリシーを拡張し、エラー処理またはアラート通知のための外部システムを統合できます。このような統合を行うには、UTL_TCP、UTL_HTTP、UTL_MAIL、

UTL_SMTPまたはDBMS_AQなどのOracleユーティリティ・パッケージを使用すると便利です。

- 機密データを保護するレルムやコマンド・ルールにルール・セットを適用する前に、テスト・データベース、または機密データ以外のデータ用のテスト・レルムやコマンド・ルール上で、様々なアカウントやシナリオに関してルール・セットを十分にテストします。次のSQL文を使用して、ルール式を直接テストできます。

```
SQL> SELECT SYSDATE from DUAL where rule expression
```

- 単一のルールにルール式をネストできます。これにより、ルールのサブセットに論理的AND、およびその他のルールに論理的ORが必要になる複雑な状況を作成できます。この一例として、「チュートリアル: セキュリティ違反の電子メール・アラートの作成」に記載されているIs Corporate Network During Maintenanceルール・セットの定義を参照してください。
- 起動者権限プロシージャとルール式を組み合せて使用できません。定義者権限プロシージャのみルール式と組み合せて使用できます。

親トピック: ルール・セットの構成

5.13 ルール・セットのパフォーマンスへの影響

ルールの数および複雑さにより、データベースのパフォーマンスが低下する場合があります。

ルール・セットにより、特定の操作の実行パフォーマンスが管理されます。たとえば、SELECT文を制御するルール・セットに大量のルールが含まれる場合は、パフォーマンスが大幅に低下する可能性があります。

多数のルールが必要なルール・セットがある場合、すべてのルールを単一のPL/SQLスタンドアロンまたはパッケージ・ファンクションに定義されているロジックに移動すると、パフォーマンスが向上します。ただし、ルールが他のルール・セットに使用されている場合、システムのパフォーマンスにはほとんど影響ありません。

可能であれば、静的な評価を使用するようにルール・セットを設定することを検討してください(関連付けられているコマンド・ルールの使用と互換性があると想定して)。詳細は、「ルール・セット設計のガイドライン」を参照してください。

システム・パフォーマンスを確認するには、Oracle Enterprise Manager(Oracle Databaseと一緒にデフォルトでインストールされるOracle Enterprise Manager Cloud Controlを含む)、自動ワークロード・リポジトリ(AWR)およびTKPROFなどのツールを実行します。

関連項目:

- データベース・パフォーマンスの監視方法を学習するには、『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
- 個々のSQL文およびPL/SQL文の実行を監視するには、『Oracle Database SQLチューニング・ガイド』を参照してください

親トピック: ルール・セットの構成

5.14 ルール・セットとルールに関連するレポートおよびデータ・ディクショナリ・ビュー

Oracle Database Vaultには、ルール・セットおよびそれらに含まれるルールの分析に役立つ、レポートとデータ・ディクショナリ・ビューが用意されています。

表5-2では、Oracle Database Vaultレポートを示します。これらのレポートの実行方法の詳細は、「Oracle Database Vaultレポート」を参照してください。

表5-2 ルール・セットに関連するレポート

レポート	説明
「ルール・セット構成の問題」レポート	ルールが定義されていない、または有効でないルール・セットが表示されます。
「セキュア・アプリケーション構成の問題」レポート	不完全または無効なルール・セットのあるセキュア・アプリケーション・ロールが表示されます。
「コマンド・ルール構成の問題」レポート	不完全または無効なルール・セットが表示されます。

表5-3に、既存のルールおよびルール・セットに関する情報を提供するデータ・ディクショナリ・ビューを示します。

表5-3 ルールおよびルール・セットに使用されるデータ・ディクショナリ・ビュー

データ・ディクショナリ・ビュー	説明
DBA_DV_RULE ビュー	定義済のルールが表示されます。
DBA_DV_RULE_SET ビュー	作成済のルール・セットが表示されます。
DBA_DV_RULE_SET_RULE ビュー	既存のルール・セットに関連付けられているルールが表示されます。

親トピック: ルール・セットの構成

6 コマンド・ルールの構成

コマンド・ルールを作成する、またはデフォルト・コマンド・ルールを使用すると、DDL文およびDML文を保護できます。

- コマンド・ルールの概要
  コマンド・ルールは、ALTER SESSIONなどのOracle Database SQL文により、Oracle Database Vault保護を適用します。
- デフォルト・コマンド・ルール
  Oracle Database Vaultには、よく使用されるSQL文に基づいて、デフォルト・コマンド・ルールが用意されています。
- コマンド・ルールで保護できるSQL文
  コマンド・ルールを使用すると、多数のSQL文を保護できます。
- コマンド・ルールの作成
  Oracle Database Vault Administratorでコマンド・ルールを作成できます。
- コマンド・ルールの有効化ステータスの変更
  Oracle Database Vault Administratorでコマンド・ルールを有効化または無効化できます。
- コマンド・ルールの削除
  コマンド・ルールを削除する前に、それに関連するOracle Database Vaultビューに問い合せることで、そのコマンド・ルールへの様々な参照を特定できます。
- コマンド・ルールの動作
  コマンド・ルールは、一連のステップに従い、関連付けられているコンポーネントをチェックします。
- チュートリアル: コマンド・ルールを使用した、ユーザー別表作成の制御
  このチュートリアルでは、ユーザーがSCOTTスキーマに表を作成できるかどうかを制御する簡単なローカル・コマンド・ルールを作成します。
- コマンド・ルールの設計のガイドライン
  Oracleでは、コマンド・ルールを設計するためのガイドラインを提供しています。
- コマンド・ルールがパフォーマンスに与える影響
  コマンド・ルールのパフォーマンスは、そのコマンド・ルールに関連付けられているルール・セット内のルールの複雑さに依存します。
- コマンド・ルール関連のレポートおよびデータ・ディクショナリ・ビュー
  Oracle Database Vaultには、コマンド・ルールの分析に役立つ、一連のレポートとデータ・ディクショナリ・ビューが用意されています。

6.1 コマンド・ルールの概要

コマンド・ルールは、ALTER SESSIONなどのOracle Database SQL文により、Oracle Database Vault保護を適用します。

- コマンド・ルールについて
  コマンド・ルールは、1つ以上のデータベース・オブジェクトに影響するOracle Database SQL文を保護します。
- 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。
- コマンド・ルールのタイプ
  多数のSQL文のためのコマンド・ルールの他に、CONNECT、ALTER SYSTEMおよびALTER SESSION SQL文専

用のコマンド・ルールを作成できます。

親トピック: コマンド・ルールの構成

6.1.1 コマンド・ルールについて

コマンド・ルールは、1つ以上のデータベース・オブジェクトに影響するOracle Database SQL文を保護します。

これらの文は、SELECT、ALTER SYSTEM、データベース定義言語(DDL)およびデータ操作言語(DML)文を含められます。

コマンド・ルールをカスタマイズして実行するには、1つ以上のルールの集合であるルール・セットにコマンド・ルールを関連付けます。コマンド・ルールは実行時に実施されます。コマンド・ルールは、オブジェクトが存在するレルムに関係なく、コマンド・ルールによって保護されるSQL文の使用を試みる全員に影響します。

コマンド・ルールを使用して、基本的なOracle Database DDL文およびDML文の他に、幅広いSQL文を保護できます。たとえば、Oracle Flashback Technologyで使用される文を保護できます。

コマンド・ルールには、コマンド・ルールのコマンドへの関連付けに加えて、次の属性があります。

- コマンド・ルールで保護されるSQL文
- コマンド・ルールが影響するオブジェクトの所有者
- コマンド・ルールが影響するデータベース・オブジェクト
- コマンド・ルールが有効かどうか
- 関連付けられているルール・セット

コマンド・ルールは、次のように分類できます。

- 範囲がシステム全体に及ぶコマンド・ルール。このタイプでは、ほとんどの場合、データベース・インスタンスごとに1つのコマンド・ルールのみを作成できます。
- スキーマ固有のコマンド・ルール。スキーマ固有のコマンド・ルールの例は、DROP TABLE文のコマンド・ルールです。スキーマごとに1つのCONNECTコマンド・ルールのみを作成できます。
- オブジェクト固有のコマンド・ルール。コマンド・ルール定義に含まれる特定の表を使用してDROP TABLE文を作成することは、この一例です。

コマンド・ルールの影響を受ける文をユーザーが実行すると、Oracle Database Vaultによって最初にレルム認可がチェックされます。レルム違反が検出されず、関連付けられているコマンド・ルールが有効な場合は、関連付けられているルール・セットがDatabase Vaultによって評価されます。すべてのルール・セットの評価がTRUEの場合、その文は認可されてさらに処理されます。評価がFALSEのルール・セットがある場合、その文の実行は許可されずコマンド・ルール違反が生じます。

通常のステップであるユーザー認証プロセス、ファクタの初期化およびOracle Label Securityの統合が完了した後に、セッションを許可または拒否するCONNECTイベントにファクタを使用するコマンド・ルールを定義できます。たとえば、BIZAPPスキーマ内で、CREATE TABLE、DROP TABLEおよびALTER TABLEなどのDDL文が営業時間後に認可されるのは許可するが、営業時間中には許可しないというコマンド・ルールを構成できます。

Oracle Database Vaultに作成するコマンド・ルール上でレポートを実行できます。

SYSがSYS所有プロシージャを実行するのをブロックするコマンド・ルールを作成することはできません。

関連トピック

- Oracle Database Vaultコマンド・ルールのAPI

- ルール・セットの構成
- コマンド・ルールで保護できるSQL文

親トピック: コマンド・ルールの概要

6.1.2 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、共通のレルム、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、ローカルのレルム、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、次のようにします。

```
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・ビューを問い合せます。

親トピック: コマンド・ルールの概要

6.1.3 コマンド・ルールのタイプ

多数のSQL文のためのコマンド・ルールの他に、CONNECT、ALTER SYSTEMおよびALTER SESSION SQL文専用のコマンド・ルールを作成できます。

- CONNECTコマンド・ルール
  DBMS_MACADM.CREATE_CONNECT_CMD_RULEプロシージャは、ユーザー固有のCONNECTコマンド・ルールを作成します。
- ALTER SESSIONおよびALTER SYSTEMコマンド・ルール
  これらのSQL文をきめ細かく制御できる、様々な種類のALTER SESSIONおよびALTER SYSTEMコマンド・ルールを作成できます。

親トピック: コマンド・ルールの概要

6.1.3.1 CONNECTコマンド・ルール

DBMS_MACADM.CREATE_CONNECT_CMD_RULEプロシージャは、ユーザー固有のCONNECTコマンド・ルールを作成します。

このタイプのコマンド・ルールは、ユーザー、関連付けられたルール・セット、有効化ステータス、およびマルチテナント環境の場合、CONNECTコマンド・ルールの実行対象を指定します。CONNECTコマンド・ルールを有効または無効にすることや、シミュレーション・モードを使用するようそれを設定することができます。シミュレーション・モードでは、コマンド・ルールに対する違反が、ユー

ザー名や使用されたSQL文などエラーを説明する十分な情報とともに、指定されたログ表に記録されます。

マルチテナント環境では、アプリケーション・ルートまたは特定のPDB内のどちらかで、CONNECTコマンド・ルールを作成できます。関連付けられたルール・セットは、CONNECTコマンド・ルールと一致している必要があります。CONNECTコマンド・ルールがアプリケーション・ルートにある場合、ルール・セットとルールもアプリケーション・ルートにある必要があります。CDBルートから共通ユーザーとしてCONNECTコマンド・ルール・プロシージャを実行します。CONNECTコマンド・ルールがプラガブル・データベース(PDB)に対してローカルである場合は、そのPDB内でCONNECTコマンド・ルール作成コマンドを実行する必要があり、ルール・セットとルールがローカルである必要があります。

次の例では、HRユーザーのためにローカルの有効化されたCONNECTコマンド・ルールを作成する、CONNECTコマンド・ルール定義を示します。このコマンド・ルールに関連付けられているルール・セットは、現在のPDBに対してローカルです。

```
BEGIN
DBMS_MACADM.CREATE_CONNECT_COMMAND_RULE(
 rule_set_name    => 'Enabled',
 user_name        => 'HR',
 enabled          => DBMS_MACUTL.G_YES,
 scope            => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

関連トピック

- CREATE_COMMAND_RULEプロシージャ
- レルムおよびコマンド・ルール・アクティビティのログ記録のためのシミュレーション・モードの使用

親トピック: コマンド・ルールのタイプ

6.1.3.2 ALTER SESSIONおよびALTER SYSTEMコマンド・ルール

これらのSQL文をきめ細かく制御できる、様々な種類のALTER SESSIONおよびALTER SYSTEMコマンド・ルールを作成できます。

これらのタイプのコマンド・ルールを作成するプロシージャを次に示します。

- DBMS_MACADM.CREATE_COMMAND_RULEは、ALTER SESSIONの場合はADVISE、CLOSE DATABASE LINK、COMMIT IN PROCEDUREおよびSET、またはALTER SYSTEMの場合はARCHIVE_LOG、CHECK DATAFILES、CHECKPOINTおよびSETなど、対応するSQL文からの句を使用する、ALTER SESSIONおよびALTER SYSTEMコマンド・ルールを作成します。
- DBMS_MACADM.CREATE_SESSION_EVENTは、ALTER SESSION SET EVENTS SQL文固有のコマンド・ルールを作成します
- DBMS_MACADM_CREATE_SYSTEM_EVENTは、ALTER SYSTEM SET EVENTS SQL文固有のコマンド・ルールを作成します。

これらのコマンド・ルールを作成するには、適切なDatabase Vaultプロシージャを使用して、作成文で句、および該当する場合は句のパラメータを指定します。ALTER SESSIONまたはALTER SYSTEMコマンド・ルールでSET EVENTS設定が使用されている場合は、特別なパラメータを使用して、イベント、コンポーネントおよびアクションを指定できます。

たとえば、ALTER SYSTEMコマンド・ルールの場合は、ALTER SYSTEM SQL文からSECURITY句およびその RESTRICTED SESSIONパラメータを指定できます。RESTRICTED SESSIONがTRUEかFALSEかを指定するには、この順序番号の妥当性をテストできる、Database Vaultルールおよびルール・セットを作成する必要があります。

この概念がどのように機能するかを理解するには、まず、RESTRICTED SESSIONパラメータがTRUEに設定されているかどう

かをチェックするよう設計されている、次のルールおよびルール・セットを作成します。

```
EXEC DBMS_MACADM.CREATE_RULE('RESTRICTED SESSION TRUE', 'UPPER(PARAMETER_VALUE) =
''TRUE''');
BEGIN
 DBMS_MACADM.CREATE_RULE_SET(
  rule_set_name    => 'Check RESTRICTED SESSION for TRUE',
  description      => 'Checks if restricted session is true',
  enabled          => DBMS_MACUTL.G_YES,
  eval_options     => DBMS_MACUTL.G_RULESET_EVAL_ALL,
  audit_options    => DBMS_MACUTL.G_RULESET_AUDIT_FAIL +
DBMS_MACUTL.G_RULESET_AUDIT_SUCCESS,
  fail_options     => DBMS_MACUTL.G_RULESET_FAIL_SILENT,
  fail_message     => 'RESTRICTED SESSION is not TRUE',
  fail_code        => 20461,
  handler_options  => DBMS_MACUTL.G_RULESET_HANDLER_FAIL,
  handler          => '',
  is_static        => false);
END;
/
EXEC DBMS_MACADM.ADD_RULE_TO_RULE_SET(Check RESTRICTED SESSION for TRUE', 'RESTRICTED
SESSION TRUE');
```

ルールおよびルール・セットを準備したら、RESTRICTED SESSIONパラメータをチェックする、ALTER SYSTEMコマンド・ルールを作成する準備は完了です。

```
BEGIN
DBMS_MACADM.CREATE_COMMAND_RULE(
 command         => 'ALTER SYSTEM',
 rule_set_name   => 'Check RESTRICTED SESSION for TRUE',
 object_owner    => '%',
 object_name     => '%',
 enabled         => DBMS_MACUTL.G_YES,
 clause_name     => 'SECURITY',
 parameter_name  => 'RESTRICTED SESSION',
 scope           => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

この例の説明は、次のとおりです。

- rule_set_nameでは、RESTRICTED SESSIONがTRUEに設定されているかFALSEに設定されているかをチェックします。マルチテナント環境では、ルール・セットとルールを、アプリケーション・ルート、またはPDBのローカルで、同じ場所でコマンド・ルールとして作成する必要があります。
- object_ownerおよびobject_nameは、この種のALTER SESSIONまたはALTER SYSTEMコマンド・ルールに対して、必ず%に設定される必要があります。
- enabledでは、コマンド・ルールを有効または無効にすることや、シミュレーション・モードを使用してコマンド・ルールに対する違反を指定のログ表に記録することができます。ログ・データには、ユーザー名や使用されたSQL文など、エラーの説明が示されます。
- clause_nameでは、ALTER SYSTEM SQL文のSECURITY句を指定します。
- parameter_nameでは、SECURITY句からのRESTRICTED SESSIONパラメータを指定します。
- scopeでは、コマンド・ルールが現在のPDBに対してローカルになるよう設定します。関連付けられたルール・セットおよびルールも、現在のPDBに対してローカルになるようにする必要があります。アプリケーション・ルートでコマンド・ルールを作成する必要がある場合は、共通ユーザーとして、scopeをDBMS_MACUTL.G_SCOPE_COMMONに設定し、アプリケーション・ルートからプロシージャ(および、その付随するルール・セットおよびルール作成プロシージャ)を実行します。

関連項目:

- DBMS_MACADM.CREATE_COMMAND_RULEプロシージャについては、CREATE_COMMAND_RULEプロシージャを参照してください
- DVS.DBMS_MACADM.CREATE_SESSION_EVENT_CMD_RULEプロシージャについては、CREATE_SESSION_EVENT_CMD_RULEプロシージャを参照してください
- DBMS_MACADM.CREATE_SYSTEM_EVENT_CMD_RULEプロシージャの詳細は、CREATE_SYSTEM_EVENT_CMD_RULEプロシージャを参照してください
- DBA_DV_COMMAND_RULEデータ・ディクショナリ・ビューの詳細は、DBA_DV_COMMAND_RULEビューを参照してください
- ALTER SESSION 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
- ALTER SYSTEM 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: コマンド・ルールのタイプ

6.2 デフォルトのコマンド・ルール

Oracle Database Vaultには、よく使用されるSQL文に基づいて、デフォルト・コマンド・ルールが用意されています。

表6-1に、デフォルトのDatabase Vaultコマンド・ルールを示します。

表6-1 デフォルトのコマンド・ルール

SQL文	ルール・セット名
CREATE USER	アカウント/プロファイルを保守可能
ALTER USER	自分のアカウントを保守可能
DROP USER	アカウント/プロファイルを保守可能
CREATE PROFILE	アカウント/プロファイルを保守可能
ALTER PROFILE	アカウント/プロファイルを保守可能
DROP PROFILE	アカウント/プロファイルを保守可能
ALTER SYSTEM	システム・パラメータのファイングレイン・コントロールを許可
CHANGE PASSWORD	自分のアカウントを保守可能脚注 1

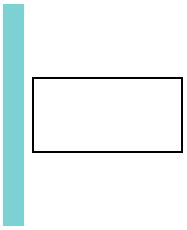
脚注1

「自分のアカウントを保守可能」ルールで参照される実際のSQL文は、PASSWORDです。

次に示す一連のコマンド・ルールは、ユーザー管理の職務分離の実現をサポートします。

- ALTER PROFILE
- ALTER USER
- CREATE PROFILE
- CREATE USER
- DROP PROFILE
- DROP USER

ユーザーにこれらのコマンドの使用権限を付与するには、ルール・セットによってチェックされるロールをユーザーに付与します。たとえば、CREATE USERコマンド・ルールは、CREATE USER文を実行しようとするユーザーにDV_ACCTMGRロールが付与されていることを確認します。



ノート:

デフォルト・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・ビューを問い合せます。

親トピック: コマンド・ルールの構成

6.3 コマンド・ルールで保護できるSQL文

コマンド・ルールを使用して、多数のSQL文を保護できます。

保護できるSQL文は、次のとおりです。

SQL文A-A	SQL文A-D	SQL文C-U
ALTER CLUSTER	ASSOCIATE STATISTICS	CREATE TABLE
ALTER DIMENSION	AUDIT	CREATE TABLESPACE
ALTER FLASHBACK ARCHIVE	AUDIT POLICY(統合監査ポリシーを監査するため)	CREATE TRIGGER
ALTER FUNCTION	CHANGE PASSWORD	CREATE TYPE
ALTER INDEX	COMMENT	CREATE TYPE BODY
ALTER INDEXTYPE	CONNECT	CREATE VIEW
ALTER JAVA	CREATE AUDIT POLICY	DELETE
ALTER LIBRARY	CREATE EDITION	DISASSOCIATE STATISTICS

SQL文A-A	SQL文A-D	SQL文C-U
ALTER OPERATOR	CREATE FLASHBACK ARCHIVE	DROP CLUSTER
ALTER OUTLINE	CREATE USER	DROP CONTEXT
ALTER MATERIALIZED VIEW	CREATE CLUSTER	DROP DATABASE LINK
ALTERMATERIALIZEDVIEWLOG	CREATE CONTEXT	DROP EDITION
ALTER PACKAGE	CREATE DATABASE LINK	DROP DIMENSION
ALTER PACKAGE BODY	CREATE DIMENSION	DROP DIRECTORY
ALTER PLUGGABLE DATABASE	CREATE DIRECTORY	DROP FLASHBACK ARCHIVE
ALTER PROCEDURE	CREATE FUNCTION	DROP FUNCTION
ALTER PROFILE	CREATE INDEX	FLASHBACK TABLE
ALTER RESOURCE COST	CREATE INDEXTYPE	EXECUTE
ALTER ROLE	CREATE JAVA	GRANT
ALTER ROLLBACK SEGMENT	CREATE LIBRARY	INSERT
ALTER SEQUENCE	CREATE OPERATOR	NOAUDIT
ALTER SESSION	CREATE OUTLINE	NOAUDIT POLICY(統合監査ポリシーのみ)
ALTER SYNONYM	CREATE PACKAGE	PURGE DBA_RECYCLEBIN
ALTER SYSTEM	CREATE PACKAGE BODY	PURGE INDEX
ALTER TABLE	CREATE PLUGGABLE DATABASE	RENAME
ALTER TABLESPACE	CREATE PROCEDURE	PURGE RECYCLEBIN
ALTER TRIGGER	CREATE PROFILE	PURGE TABLE
ALTER TYPE	CREATE ROLE	PURGE TABLESPACE
ALTER TYPE BODY	CREATE ROLLBACK SEGMENT	REVOKE

SQL文A-A	SQL文A-D	SQL文C-U
ALTER USER	CREATE SCHEMA	SELECT
ALTER VIEW	CREATE SEQUENCE	TRUNCATE CLUSTER
ANALYZE CLUSTER	CREATE MATERIALIZED VIEW	TRUNCATE TABLE
ANALYZE INDEX	CREATEMATERIALIZEDVIEWLOG	UPDATE
ANALYZE TABLE	CREATE SYNONYM	_

関連項目:

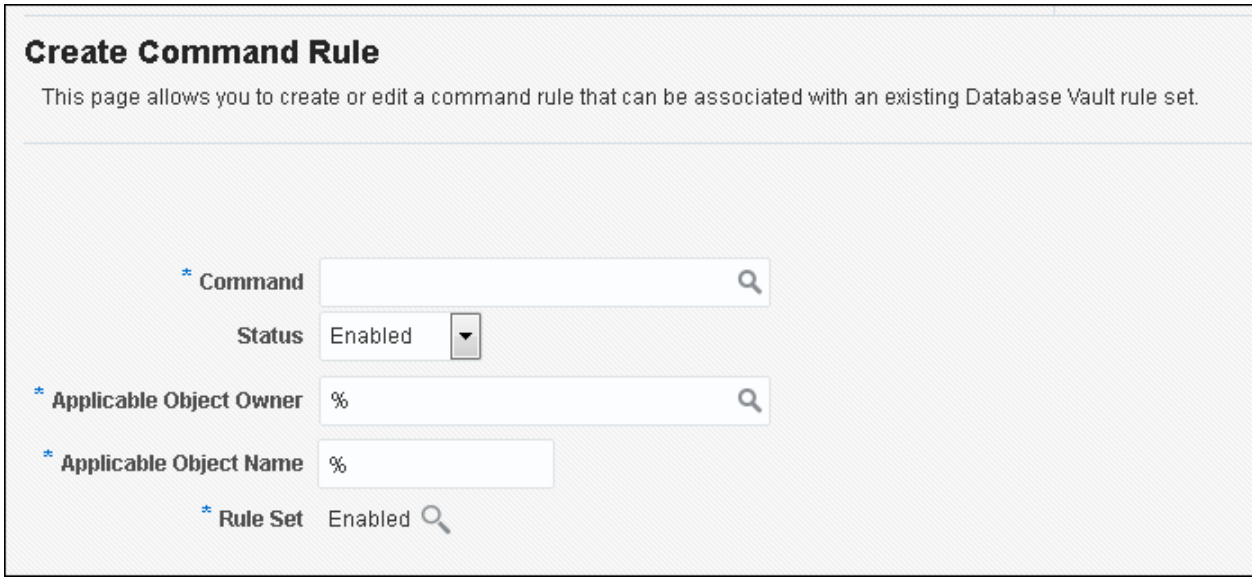
マルチテナント・コンテナ・データベース(CDB)におけるCREATE PLUGGABLE DATABASE、ALTER PLUGGABLE DATABASEおよびDROP PLUGGABLE DATABASEの使用の詳細は、「マルチテナント環境におけるコマンド・ルール」を参照してください

親トピック: コマンド・ルールの構成

6.4 コマンド・ルールの作成

Oracle Database Vault Administratorでコマンド・ルールを作成できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「コマンド・ルール」をクリックします。
3. 「コマンド・ルール」ページで次のようにします。
   - 新しいコマンド・ルールを作成するには、「作成」をクリックして、「コマンド・ルールの作成」ページを表示します。



4. 「コマンド・ルールの作成」ページで、次の設定を入力します。

- コマンド: コマンド・ルールを作成するSQL文または操作を選択します。この属性は必須です。

  統合監査ポリシー・オブジェクトに対するコマンド・ルールを作成する場合、コマンドとして、AUDITまたはNOAUDITではなく、AUDIT POLICYまたはNOAUDIT POLICYを必ず指定します。

- ステータス: 「有効」、「無効」または「シミュレーション」のいずれかを選択します。これは、実行時にコマンド・ルールに適用されます。この属性は必須です。

- 適用可能なオブジェクト所有者: リストから、コマンド・ルールが影響を与えるオブジェクトの所有者を選択します。ワイルドカード文字%を使用して、すべての所有者を選択できます。(ただし、ワイルドカード文字をテキストと一緒に使用することはできません。たとえば、EM%を使用して、名前がEMで始まるすべての所有者を選択することはできません。)この属性は、特定のスキーマ内のオブジェクトに提供するすべてのSQL文で必須です。サポートされているSQL文のリストは、「コマンド・ルールで保護できるSQL文」を参照してください。

  SELECT、INSERT、UPDATE、DELETEおよびEXECUTE文は、すべての選択(%)、またはSYSおよびDVSYSスキーマには使用できないことに注意してください。

- 適用可能なオブジェクト名: コマンド・ルールが影響を与えるデータベース・オブジェクトの名前を入力するか、%を指定してすべてのデータベース・オブジェクトを指定します。これには、表、プロシージャ、ビュー、統合監査ポリシーなどを含むことができます。「オブジェクト所有者」リストからオブジェクト所有者を選択した場合、この属性は必須です。

  コマンド・ルールが影響するオブジェクト上でOracle Database Vaultレポートを実行できます。詳細は、「コマンド・ルールに関連するレポートおよびデータ・ディクショナリ・ビュー」を参照してください。

- ルール・セット: リストから、コマンド・ルールに関連付けるルール・セットを選択します。この属性は必須です。

  ルール・セットの評価がTrueの場合、そのSQL文は成功します。評価がfalseの場合、その文は失敗し、Oracle Database Vaultによってコマンド・ルール違反が生成されます。(「Oracle Database Vaultレポート」で説明されている「コマンド・ルール構成の問題」レポートを使用することで、ルール違反を追跡できる)。ルール・セットに関連付けられている監査およびカスタム・イベント処理は、コマンド・ルール処理の一部として発生します。

  ルール・セットの詳細は、「ルール・セットの構成」を参照してください。

5. 「OK」をクリックします。

関連トピック

- 「他のデータベースへのOracle Database Vault構成の伝播」

親トピック: コマンド・ルールの構成

6.5 コマンド・ルールの有効化ステータスの変更

Oracle Database Vault Administratorでコマンド・ルールを有効または無効にできます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「コマンド・ルール」をクリックします。
3. 「コマンド・ルール」ページで、有効または無効にするコマンド・ルールを選択し、「編集」を選択します。
4. 「コマンド・ルールの編集」ページで、「ステータス」メニューから、目的のステータスを選択します。

- 有効
- 無効
- シミュレーション

5. 「OK」をクリックします。

親トピック: コマンド・ルールの構成

6.6 コマンド・ルールの削除

コマンド・ルールを削除する前に、それに関連するOracle Database Vaultビューに問い合せることで、そのコマンド・ルールへの様々な参照を特定できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. Oracle Database Vaultの「管理」ページで、「コマンド・ルール」を選択します。
3. 「コマンド・ルール」ページで、削除するコマンド・ルールを選択します。
4. 「削除」をクリックします。
5. 「確認」ウィンドウで、「はい」をクリックします。

関連トピック

- Oracle Database Vaultのデータ・ディクショナリ・ビュー

親トピック: コマンド・ルールの構成

6.7 コマンド・ルールの動作

コマンド・ルールは、一連のステップに従い、関連付けられているコンポーネントをチェックします。

「レルムの動作」では、データベース・アカウントによりレルム内のオブジェクトに影響するSELECT、DDLまたはDML文が発行された場合の動作を説明しています。

SELECT、DDLまたはDML文が発行されると、次のアクションが実行されます。

1. Oracle Database Vaultが、適用する必要のあるすべてのコマンド・ルールを問い合せます。

   SELECT、DDLおよびDML文では、オブジェクト所有者およびオブジェクト名でワイルドカード表記を使用できるため、複数のコマンド・ルールが適用される場合があります。

   ルール・セットは、コマンド・ルールおよびレルム認可の両方に関連付けることができます。Oracle Database Vaultにより、まずレルム認可ルール・セットが評価され、評価されるコマンド・タイプに適用するルール・セットが次に評価されます。

2. 適用する各コマンド・ルールに関して、Oracle Database Vaultはそれに関連付けられているルール・セットを評価します。

3. 適用可能なコマンド・ルールに関連付けられているルール・セットが1つでもFalseまたはエラーで返されると、Oracle Database Vaultによりそのコマンドの実行が阻止されます。それ以外の場合、コマンドは認可されてさらに処理されます。監査およびイベント・ハンドラに関するルール・セットの構成により、発生する監査またはカスタム処理が決定されます。

   コマンド・ルールはオブジェクト権限より優先されます。つまり、オブジェクトの所有者であっても、オブジェクトがコマンド・

ルールで保護される場合、そのオブジェクトにアクセスできません。コマンド・ルールまたはコマンドのルール・セットのいずれかを無効にすることができます。コマンド・ルールを無効にする場合、コマンド・ルールは、その処理の確認を行いません。ルール・セットを無効にする場合、ルール・セットの評価は常にTRUEになります。ただし、特定のコマンドに対してコマンド・ルールを無効にする場合、ルール・セットが他のコマンド・ルールやレルム認可に関連付けられている可能性があるため、コマンド・ルールを無効にする必要があります。

親トピック: コマンド・ルールの構成

6.8 チュートリアル: ユーザーによる表作成を制御するためのコマンド・ルールの使用方法

このチュートリアルでは、ユーザーがSCOTTスキーマに表を作成できるかどうかを制御する簡単なローカル・コマンド・ルールを作成します。

- ステップ1: 表の作成
  まず、ユーザーSCOTTが表を作成する必要があります。
- ステップ2: コマンド・ルールの作成
  SCOTTスキーマで表の作成後、コマンド・ルールを作成できます。
- ステップ3: コマンド・ルールのテスト
  次に、CREATE TABLEローカル・コマンド・ルールをテストできます。
- ステップ4: このチュートリアルのコンポーネントの削除
  コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

親トピック: コマンド・ルールの構成

6.8.1 ステップ1: 表の作成

最初に、ユーザーSCOTTとして表を作成する必要があります。

1. ユーザーSCOTTとしてデータベース・インスタンスにログインします。

```
sqlplus scott
Enter password: password
```

マルチテナント環境で、適切なPDBにログインする必要があります。たとえば:

```
sqlplus scott@hrpdb
Enter password: password
```

利用可能なプラガブル・データベース(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。

SCOTTアカウントがロックされて無効になっている場合、Database Vaultアカウント・マネージャとしてログインし、SCOTTのロックを解除し、新しいパスワードを作成します。たとえば:

```
sqlplus accts_admin_ace --Or, sqlplus accts_admin_ace@hrpdb
Enter password: password
ALTER USER SCOTT ACCOUNT UNLOCK IDENTIFIED BY password;
```

『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

```
CONNECT SCOTT --Or, sqlplus SCOTT@hrpdb
```

```
Enter password: password
```

2. ユーザーSCOTTとして、表を作成します。

```
CREATE TABLE t1 (num NUMBER);
```

3. 表を削除します。

```
DROP TABLE t1;
```

この段階で、ユーザーSCOTTは、表の作成および削除を行うことができます。SQL*Plusを終了せずに、SCOTTとして接続したままにします。後でSCOTTが別の表を作成しようとするときに使用します。

親トピック: チュートリアル: ユーザーによる表作成を制御するためのコマンド・ルールの使用方法

6.8.2 ステップ2: コマンド・ルールの作成

SCOTTスキーマで表の作成後、コマンド・ルールを作成できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. Oracle Database Vault Administratorの「管理」ページで、「コマンド・ルール」をクリックします。

   「コマンド・ルール」ページが表示されます。

3. 「作成」をクリックします。

   「コマンド・ルールの作成」ページが表示されます。

4. 次の設定を入力します。
   - コマンド: 「CREATE TABLE」を選択します。
   - ステータス: 「有効」に設定して、コマンド・ルールをアクティブにします。
   - 適用可能なオブジェクト所有者: 「SCOTT」を選択します。
   - 適用可能なオブジェクト名: %に設定して、SCOTTスキーマ内のすべてのオブジェクトに適用します。
   - ルール・セット: SCOTTスキーマに表を作成できないよう、「無効」を選択します。
5. 「OK」をクリックします。

   Database Vault Administratorを終了しないでください。

コマンド・ルールは即時有効になります。ユーザーSCOTTは、CREATE TABLEコマンド・ルールを作成する前の、先ほどまでいたユーザー・セッションにまだいるとしても、ただちに表を作成できなくなります。

親トピック: チュートリアル: ユーザーによる表作成を制御するためのコマンド・ルールの使用方法

6.8.3 ステップ3: コマンド・ルールのテスト

これで、いつでもCREATE TABLEローカル・コマンド・ルールをテストできます。

1. SQL*Plusで、ユーザーSCOTTとしてログインしていることを確認します。

```
CONNECT SCOTT --Or, CONNECT SCOTT@hrpdb
Enter password: password
```

2. 表を作成してみます。

```
CREATE TABLE t1 (num NUMBER);
```

次の出力が表示されます。

```
ORA-47400: Command Rule violation for create table on SCOTT.T1
```

この例からわかるように、SCOTTは、自身のスキーマ内でも、表を作成できなくなります。

3. Oracle Database Vault Administratorで、次の作業を実行します。
   a. 「コマンド・ルール」ページで、「CREATE TABLE」コマンド・ルールを選択して、「編集」をクリックします。
   b. 「コマンド・ルールの編集」ページで、「ルール・セット」リストから「有効」を選択します。
   c. 「OK」をクリックします。
4. SQL*Plusで、ユーザーSCOTTとして、表の作成を再試行します。

```
CREATE TABLE t1 (num NUMBER);
Table created.
```

CREATE TABLEコマンド・ルールが「有効」に設定されたので、ユーザーSCOTTは、再び表の作成が許可されるようになりました。(SQL*Plusを終了しないでください。)

親トピック: チュートリアル: ユーザーによる表作成を制御するためのコマンド・ルールの使用方法

6.8.4 ステップ4: このチュートリアルのコンポーネントの削除

コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

1. Oracle Database Vault Administratorで、CREATE TABLEコマンド・ルールを次のように削除します。
   a. 「コマンド・ルール」ページに戻ります。
   b. 「CREATE TABLE」ローカル・コマンド・ルールを選択して、「削除」をクリックします。
   c. 「確認」ウィンドウで、「はい」をクリックします。
2. ユーザーSCOTTとしてデータベース・インスタンスにログインして、t1表を削除します。

```
DROP TABLE t1;
```

3. SCOTTアカウントが使用可能である必要がなくなった場合、Database Vaultアカウント・マネージャとして接続し、次のALTER USER文を入力します。

```
CONNECT accts_admin_ace --Or, CONNECT accts_admin_ace@hrpdb
Enter password: password
ALTER USER SCOTT ACCOUNT LOCK PASSWORD EXPIRE;
```

親トピック: チュートリアル: ユーザーによる表作成を制御するためのコマンド・ルールの使用方法

6.9 コマンド・ルール設計のガイドライン

Oracleでは、コマンド・ルール設計のガイドラインを提供しています。

- 維持が簡単であるため、ファイングレイン・コマンド・ルールを作成します。

  たとえば、特定のスキーマ・オブジェクトでSELECT文が発生しないようにする場合は、スキーマ・レベルでSELECT文を

阻止する一般的なコマンド・ルールを作成するのではなく、特定のスキーマ・オブジェクトでSELECT文を阻止する複数のコマンド・ルールを設計します。

- CONNECTイベントにルールを作成する場合、必要なユーザー接続を間違えてロックアウトしないロジックを慎重に指定してください。アカウントが誤ってロックアウトされた場合、DV_ADMINまたはDV_OWNERロールを付与されたユーザーに、ロックアウト問題の原因となっているルールにログインして修正してもらうようにしてください。CONNECTコマンドは、DV_OWNERロールとDV_ADMINロールを持つユーザーには適用されません。これにより、不適切に構成されたCONNECTコマンド・ルールで完全なロックアウトが生じることはありません。

  アカウントがロックアウトされている場合は、Oracle Database Vaultを無効にして、ロックアウトの問題の原因となっているルールを修正し、再びOracle Database Vaultを有効にします。Oracle Database Vaultが無効な場合でも、Database Vault AdministratorとDatabase Vault PL/SQLパッケージはまだ使用できます。

- 管理タスクに対して有効なコマンド・ルールを一時的に緩和する必要がある場合は、コマンド・ルールをシミュレーション・モードに切り替えることを検討してください。これにより、ルール・セット基準を満たすアクティビティは取得されず、違反したアクティビティのみが取得されることに注意してください。

- コマンド・ルールの設計時には、処理が不用意に無効にされる可能性があるため、バックアップなどの自動プロセスを考慮するようにしてください。使用中のプログラム、使用中のアカウントまたはクライアント・プログラムが稼働中のコンピュータやネットワークなど、一連のOracle Database VaultファクタがTrueであることが明白な場合には、コマンドを許可するルールを作成することでこれらのタスクを把握できます。

- シミュレーション・モードを使用することで、コマンド・ルールの開発フェーズをテストできます。このモードでは、コマンド・ルールは有効になりますが、それに関する詳細情報がログ・ファイルに書き込まれます。

関連トピック

- レルムおよびコマンド・ルール・アクティビティのログ記録のためのシミュレーション・モードの使用

親トピック: コマンド・ルールの構成

6.10 コマンド・ルールのパフォーマンスへの影響

コマンド・ルールのパフォーマンスは、そのコマンド・ルールに関連付けられているルール・セット内のルールの複雑さに依存します。

たとえば、実行に5秒かかるPL/SQLファンクションを起動するルール・セットがあるとします。この場合、このルール・セットを使用するコマンド・ルールでは、実行するコマンド文へのアクセス権の付与に5秒かかります。

システム・パフォーマンスを確認するには、Oracle Enterprise Manager(Oracle Databaseと一緒にデフォルトでインストールされるOracle Enterprise Manager Cloud Controlを含む)、自動ワークロード・リポジトリ(AWR)およびTKPROFなどのツールを実行します。

関連項目:

- データベース・パフォーマンスの監視方法を学習するには、『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
- 個々のSQL文およびPL/SQL文の実行を監視するには、『Oracle Database SQLチューニング・ガイド』を参照してください

親トピック: コマンド・ルールの構成

6.11 コマンド・ルールに関連するレポートおよびデータ・ディクショナリ・ビュー

Oracle Database Vaultには、コマンド・ルールの分析に役立つ、レポートとデータ・ディクショナリ・ビューが用意されています。

表6-2に、Oracle Database Vaultレポートを示します。これらのレポートの実行方法の詳細は、「Oracle Database Vaultレポート」を参照してください。

表6-2 コマンド・ルールに関連するレポート

レポート	説明
「コマンド・ルールの監査」レポート	コマンド・ルールを処理する操作により生成された監査レコードが表示されます。
「コマンド・ルール構成の問題」レポート	コマンド・ルールに存在するその他の構成問題に加え、ルール違反が追跡されます。
オブジェクト権限レポート	コマンド・ルールが影響するオブジェクト権限が表示されます。
機密オブジェクト・レポート	コマンド・ルールが影響するオブジェクトが表示されます。
「ルール・セット構成の問題」レポート	ルールが定義されていないか、有効ではなく、それらを使用するコマンド・ルールに影響を与える可能性があるルール・セットが表示されます。

DBA_DV_COMMAND_RULEデータ・ディクショナリ・ビューを使用すると、コマンド・ルールにより保護されているSQL文を検出できます。詳細は、「DBA_DV_COMMAND_RULEビュー」を参照してください。

親トピック: コマンド・ルールの構成

7 ファクタの構成

ファクタを使用すると、Oracle Database Vault認可を決定する複雑な属性をPL/SQLで作成して使用できます。

- ファクタの概要
  ファクタは、データベースのIPアドレスなど、Oracle Database Vaultが認識できる名前付き変数または属性です。
- デフォルト・ファクタ
  Oracle Database Vaultには一連のデフォルトのファクタが用意されています。
- ファクタの作成
  通常、ファクタを作成するには、まずファクタを作成し、ファクタを編集してアイデンティティを含めます。
- ファクタへのアイデンティティの追加
  新しいファクタを作成したら、ファクタにアイデンティティを追加できます。
- ファクタの削除
  ファクタを削除する前に、そのファクタへの参照を削除する必要があります。
- ファクタの動作
  セッションが確立されると、Oracle Database Vaultではファクタが処理されます。
- チュートリアル: データベースへの非定型ツール・アクセスの阻止
  このチュートリアルでは、ファクタを使用して非定型ツール(SQL*Plusなど)がデータベースにアクセスできないようにする方法を示します。
- チュートリアル: セッション・データに基づくユーザー・アクティビティの制限
  このチュートリアルでは、ユーザーが使用しているドメインなど、セッション・データに基づいたユーザー・アクティビティを制限する方法を示します。
- ファクタ設計のガイドライン
  Oracleでは、ファクタを設計するためのガイドラインを提供しています。
- ファクタのパフォーマンスへの影響
  ファクタの複雑さは、Oracleデータベース・インスタンスのパフォーマンスに影響します。
- ファクタ関連のレポートおよびデータ・ディクショナリ・ビュー
  Oracle Database Vaultには、ファクタおよびそのアイデンティティに関する情報が表示されるレポートとデータ・ディクショナリ・ビューが用意されています。

7.1 ファクタの概要

ファクタは、データベースのIPアドレスなど、Oracle Database Vaultが認識できる名前付き変数または属性です。

ファクタは、データベースに接続するためのデータベース・アカウントの認可や、データの可視性および管理性を制限するフィルタ・ロジックの作成などのアクティビティに使用できます。

Oracle Database Vaultには、サイトのドメイン、IPアドレス、データベースなどのコンポーネントに対する制御を設定できる様々なファクタが用意されています。独自のPL/SQL取得メソッドを使用してカスタム・ファクタを作成することもできます。独自のPL/SQL取得メソッドを使用してカスタム・ファクタを作成することもできます。ただし、ほとんどの場合に、SYS_CONTEXT PL/SQLファンクションが使用できます。これにより、データベースですぐに利用できる最も一般的に使用されるファクタに対するルールを作成します。Session_User、Proxy_User、Network_Protocol、Moduleなどのファクタは、SYS_CONTEXT ファンクションから使用できます。

ファクタには、Oracle Label Securityと組み合せて使用する強力な機能があります。この機能は、コンテキストのパラメータではまだ使用できない、その他のデータベース属性のためのものです。この項では、一般に使用できるファクタを示しますが、そうし

たファクタについてはルール定義でSYS_CONTEXTファンクションを使用することをお薦めします。SYS_CONTEXTでは、まだ使用できないファクタのみを作成して使用してください。

次のことに注意してください。

- ルール・セットのルールとともにファクタを使用できます。DVFファクタ・ファンクションは、ルール式で使用できるファクタ固有のファンクションです。
- ファクタには値(アイデンティティ)があり、それぞれのファクタ・タイプによってさらに分類されます。ファクタ・タイプの詳細は、「ファクタ作成のための「一般」ページの入力」の「ファクタ・タイプ」を参照してください。
- また、Oracle Label Securityラベルを使用してファクタを統合できます。
- Oracle Database Vaultに作成するファクタ上でレポートを実行できます。詳細情報を参照してください。
- 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・ルートではなく、PDBでのみファクタを作成できます。

この章では、Oracle Database Vault Administratorを使用してファクタを構成する方法を説明します。また、Oracle Database VaultのファクタAPIを使用して、ファクタを構成することもできます。

関連トピック

- ルール・セットに追加するルールの作成
- Oracle Database SQL言語リファレンス
- Oracle Database VaultのDVF PL/SQLファクタ・ファンクション
- Oracle Database VaultファクタのAPI

親トピック: ファクタの構成

7.2 デフォルトのファクタ

Oracle Database Vaultには一連のデフォルトのファクタが用意されています。

これらのファクタごとに、ファクタの値を取得するファンクションが関連付けられています。

独自のPL/SQL取得メソッドを使用してカスタム・ファクタを作成できます。使用できる便利なPL/SQLファンクション(デフォルト・ファクタの多くに使用される)は、SYS_CONTEXT SQLファンクションで、ユーザー・セッションに関するデータを取得します。たとえば、SYS_CONTEXTのCLIENT_PROGRAM_NAME属性を使用して、データベース・セッションに使用されるプログラムの名前を検索できます。カスタム・ファクタを作成すると、デフォルト・ファクタの問合せに使用されるファンクションと同様に値を問い合せることができます。

独自のセキュリティ構成でデフォルトのファクタを使用できます。不要な場合には削除できます。(Oracle Database Vaultによる内部使用には不要です。)

デフォルト・ファクタは次のとおりです。

- Authentication_Methodは認証方式です。次に、ユーザー・タイプの後に返される方式を続けて示します。
  - パスワードで認証されるエンタープライズ・ユーザー、ローカル・データベース・ユーザー、パスワード・ファイルを使用するSYSDBAまたはSYSOPER管理権限があるユーザー(パスワードを使用するユーザー名によるプロキシ): PASSWORD
  - Kerberosで認証されるエンタープライズ・ユーザーまたは外部ユーザー(管理権限なし): KERBEROS
  - Kerberos認証済のエンタープライズ・ユーザー(管理者権限あり): KERBEROS_GLOBAL
  - Kerberos認証済の外部ユーザー(管理者権限あり): KERBEROS_EXTERNAL

- Transport Layer Security (TLS)で認証されるエンタープライズ・ユーザーまたは外部ユーザー(管理権限なし): SSL (Transport Layer SecurityでSecure Sockets Layerが置き換えられますが、SSL関連の設定はTransport Layer Securityで動作します。)
- Transport Layer Securityで認証されるエンタープライズ・ユーザー (管理権限あり): SSL_GLOBAL
- Transport Layer Securityで認証される外部ユーザー (管理権限あり): SSL_EXTERNAL
- RADIUSで認証される外部ユーザー: RADIUS
- OSで認証される外部ユーザー、またはSYSDBAまたはSYSOPER管理権限があるユーザー: OS
- 証明書付きプロキシ、DN、またはパスワードを使用しないユーザー名: NONE
- バックグラウンド・プロセス(ジョブ・キュー・スレーブ・プロセス): JOB
- パラレル問合せスレーブ・プロセス: PQ_SLAVE

非管理接続では、認証方法がPASSWORD、KERBEROSまたはSSLの場合は、Identification_Typeファクタを使用して外部ユーザーとエンタープライズ・ユーザーを区別できます。管理接続では、PASSWORD、SSL_EXTERNALおよびSSL_GLOBAL認証方式にはAuthentication_Methodファクタで十分です。

- クライアント識別子は、DBMS_SESSION.SET_IDENTIFIERプロシージャ、Oracle Call Interface (OCI)属性OCI_ATTR_CLIENT_IDENTIFIERまたはOracle Dynamic Monitoring Service (DMS)を使用してアプリケーションによって設定された識別子です。様々なOracle Databaseコンポーネントが、この属性を使用して同じデータベース・ユーザーとして認証される軽量アプリケーション・ユーザーを識別します。
- Client_IPはクライアントが接続されているコンピュータのIPアドレスです。
- Database_DomainはDB_DOMAIN初期化パラメータで指定されているデータベースのドメインです。
- Database_Hostnameは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です。
- Database_Instanceは現在のインスタンスのインスタンス識別番号です。
- Database_IPはインスタンスが実行されているコンピュータのIPアドレスです。
- Database_NameはDB_NAME初期化パラメータで指定されているデータベースの名前です。
- DBlink_Infoはデータベース・リンク・セッションのソースです。文字列の形式は、次のとおりです。

  ```
  SOURCE_GLOBAL_NAME=dblink_src_global_name,
  DBLINK_NAME=dblink_name,SOURCE_AUDIT_SESSIONID=dblink_src_audit_sessionid
  ```

  詳細は、次のとおりです。

  - dblink_src_global_name: ソース・データベースの一意のグローバル名
  - dblink_name: ソース・データベースでのデータベース・リンクの名前
  - dblink_src_audit_sessionid: dblink_nameを使用してリモート・データベースへの接続を開始したソース・データベース
- Domainは特定の機密レベルで動作するランタイム環境(ネットワーク化されたIT環境またはそのサブセットなど)の物理、構成または実装固有のファクタの名前付きコレクションです。データベースへのセキュア・アクセス・パス内にあるDatabase Vaultノードのホスト名、IPアドレスおよびデータベース・インスタンス名などのファクタを使用してドメインを識別できます。ドメインを識別するファクタ識別子の組合せを使用して、各ドメインを一意に特定できます。これらの識別ファクタやその他のファクタを使用して、ドメイン内に最大セキュリティ・ラベルを定義できます。これにより、Database

Vaultセッションに関する物理ファクタに応じて、データ・アクセスやコマンドを制限できます。必要なドメインの例として、企業機密、内部パブリック、パートナ、顧客があります。

- `Enterprise_Identity`はユーザーのエンタープライズ全体のアイデンティティです。
  - エンタープライズ・ユーザーの場合: Oracle Internet Directory識別名(DN)。
  - 外部ユーザーの場合: 外部アイデンティティ(Kerberosプリンシパル名、RADIUSおよびDCEスキーマ名、オペレーティング・システム・ユーザー名、証明書DN)。
  - ローカル・ユーザーとSYSDBAログインおよびSYSOPERログインの場合: NULL

  属性の値はプロキシ方式によって異なります。
  - DNによるプロキシの場合: クライアントのOracle Internet Directory DN。
  - 証明書によるプロキシの場合: 外部ユーザーではクライアントの証明書DN、グローバル・ユーザーではOracle Internet Directory DN。
  - ユーザー名によるプロキシの場合: クライアントがエンタープライズ・ユーザーの場合はOracle Internet Directory DN、クライアントがローカル・データベース・ユーザーの場合はNULL。
- `Identification_Type`はデータベースでユーザー・スキーマが作成された方法です。具体的には、`CREATE USER`および`ALTER USER`構文の`IDENTIFIED`句が反映されます。次に、スキーマ作成時に使用される構文の後に返される識別タイプを続けて示します。
  - `IDENTIFIED BY password`: `LOCAL`
  - `IDENTIFIED EXTERNALLY`: `EXTERNAL`
  - `IDENTIFIED GLOBALLY`: `GLOBAL SHARED`
  - `IDENTIFIED GLOBALLY AS DN`: `GLOBAL PRIVATE`
  - `GLOBAL EXCLUSIVE` (排他的なグローバル・ユーザー・マッピング)
  - `GLOBAL SHARED` (共有ユーザー・マッピング)
  - `NONE` (認証なしでスキーマを作成する場合)
- `Lang`は既存の`LANGUAGE`パラメータより短い形式の言語名のISO略称です。
- `Language`はセッションで現在使用中の言語と地域、およびデータベース文字セットです。次の形式で示されます。

  ```
  language_territory.characterset
  ```

  たとえば:

  ```
  AMERICAN_AMERICA.WE8MSWIN1252
  ```

- `Machine`は現在のセッションを確立したデータベース・クライアントのホスト名です。コンピュータがクライアントまたはサーバー・セッションに使用されていたかどうかを調べる必要がある場合には、この設定を`Database_Hostname`ファクタと比較して特定できます。
- モジュールは、`DBMS_APPLICATION_INFO` PL/SQLパッケージまたはOCIを使用して設定されたアプリケーション名(モジュール)です。
- `Network_Protocol`は接続文字列の`PROTOCOL=protocol`部分で指定されている、通信に使用されるネットワーク・プロトコルです。

- `Proxy_Enterprise_Identity`はプロキシ・ユーザーがエンタープライズ・ユーザーである場合、Oracle Internet Directory DNです。
- `Proxy_User`は`SESSION_USER`のかわりに現行セッションを開いたデータベース・ユーザーの名前です。
- `Session_User`は現行ユーザーが認証されたデータベース・ユーザー名です。この値は、セッションを通して同じです。

関連トピック

- Oracle Database VaultのDVF PL/SQLファクタ・ファンクション
- Oracle Database SQL言語リファレンス
- Oracle Databaseグローバリゼーション・サポート・ガイド

親トピック: ファクタの構成

7.3 ファクタの作成

通常、ファクタを作成するには、まずファクタを作成し、ファクタを編集してアイデンティティを含めます。

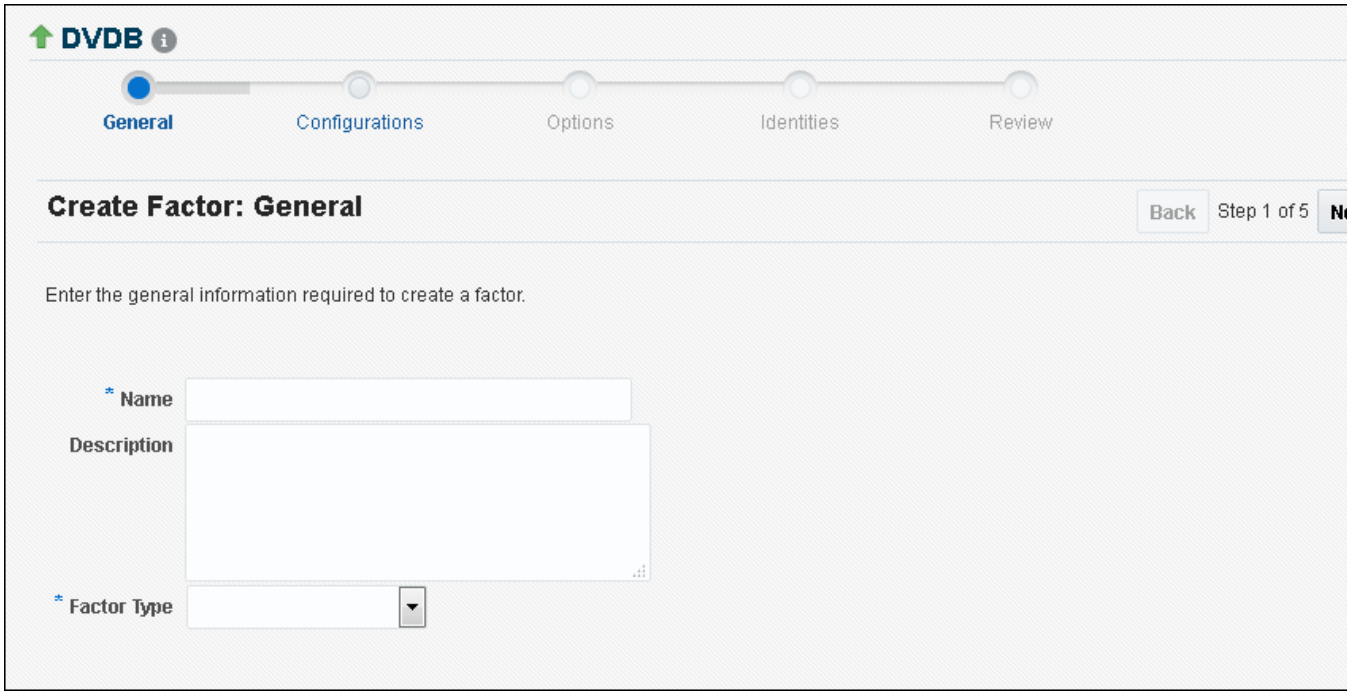
- 「ファクタの作成」ページのアクセス
  「ファクタの作成」ページを使用すると、作成するファクタの一般的な定義を含めファクタを作成できます。
- ファクタ作成のための「一般」ページの入力
  「一般」ページでは、名前などの、ファクタの一般的な識別情報を入力する必要があります。
- ファクタ作成のための「構成」ページ
  「構成」ページでは、ファクタの識別メソッドおよび評価メソッドなどの設定を定義します。
- ファクタ作成のための「オプション」ページ
  「オプション」ページでは、ルール・セットをファクタに割り当て、エラー・オプションを設定し、非統合監査について、監査オプションを設定します。

親トピック: ファクタの構成

7.3.1 「ファクタの作成」ページへのアクセス

「ファクタの作成」ページを使用すると、作成するファクタの一般的な定義を含め、ファクタを作成できます。

1. `DV_OWNER`または`DV_ADMIN`ロールおよび`SELECT ANY DICTIONARY`権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「ファクタ」をクリックします。
3. 「ファクタ」ページで、「作成」をクリックして「ファクタの作成」ページを表示します。



4. 「一般」ページ以降で次の情報を入力し、「次へ」をクリックしてそれぞれの後続ページに移動します。ファクタ定義が完了したら「完了」および「終了」をクリックします。
    - ファクタ作成のための「一般」ページの入力
    - ファクタ作成の「構成」ページ
    - ファクタ作成の「オプション」ページ
    - ファクタ・アイデンティティの作成および構成

親トピック: ファクタの作成

7.3.2 ファクタ作成のための「一般」ページの入力

「一般」ページでは、名前などの、ファクタの一般的な識別情報を入力する必要があります。

- 「一般」ページで、次の情報を入力します。
    - 名前: 28文字以内(大/小文字混在、空白なし)で名前を入力します。Oracle Database Vaultにより、選択されたファクタの名前に基づいてDVFスキーマに作成されるファクタ・ファンクションの有効なOracle識別子が作成されます。たとえば、`GetNetworkIP`という名前のファクタを作成した場合、Oracle Database VaultによりDVF.F$GETNETWORKIPファンクションが作成されます。この属性は必須です。

      名前は名詞で始まり、導出値の簡単な説明で終わることをお薦めします。

      DVFファクタ・ファンクションについては、「Oracle Database VaultのDVF PL/SQLファクタ・ファンクション」で説明しています。
    - 説明: ファクタの説明テキストを入力します。大/小文字の両方を使用して1024文字以内で指定できます。この属性はオプションです。
    - ファクタ・タイプ: リストから、ファクタのタイプまたはカテゴリを選択します。この属性は必須です。

      ファクタ・タイプには名前と説明があり、ファクタ分類の目的でのみ使用されます。ファクタ・タイプは、ファクタの分類に使用されるカテゴリ名です。デフォルトの物理ファクタ・タイプには、認証方式、ホスト名、ホストIPアドレス、インスタンス識別子およびデータベース・アカウント情報などが含まれます。時間や認証方式などのインストール

されたファクタ・タイプに加え、アプリケーション名や証明書情報などのユーザー定義のファクタ・タイプも作成できます。

特定のファクタ・タイプに関連付けられているファクタは、DBA_DV_FACTORデータ・ディクショナリ・ビューに問い合せることで参照できます。たとえば:

```
SELECT NAME FROM DBA_DV_FACTOR
WHERE FACTOR_TYPE_NAME='Authentication Method';
```

出力は次のとおりです。

```
NAME
------------------------------
Network_Protocol
Authentication_Method
Identification_Type
```

親トピック: ファクタの作成

7.3.3 ファクタ作成の「構成」ページ

「構成」ページでは、ファクタの識別メソッドおよび評価メソッドなどの設定を定義します。

- ファクタ識別情報の設定
  「ファクタの識別」で、ファクタのアイデンティティの解決方法を選択する必要があります。この属性は必須です。
- ファクタ・アイデンティティの動作
  ファクタ・アイデンティティはファクタの実際の値です(IP_Addressタイプを使用するファクタのIPアドレスなど)。
- ファクタの評価情報の設定
  「評価」で、ファクタの評価方法とアイデンティティの割当て方法を選択する必要があります。
- ファクタのOracle Label Securityラベリング情報の設定
  「ファクタ・ラベリング」で、ファクタ・アイデンティティによるOracle Label Security(OLS)ラベルの取得方法を選択する必要があります。
- ファクタの取得メソッドの設定
  「取得メソッド」に、ファクタのアイデンティティを取得するPL/SQL式または定数を入力する必要があります。
- 取得メソッドの動作方法
  「取得メソッド」により、ファクタの識別がメソッドまたは定数によって行われるファクタが識別されます。
- ファクタの検証メソッドの設定
  検証メソッドでは、PL/SQL式を使用してブール値を返し、ファクタのアイデンティティを検証します。

親トピック: ファクタの作成

7.3.3.1 ファクタの識別情報の設定

「ファクタの識別」で、ファクタのアイデンティティの解決方法を選択する必要があります。この属性は必須です。

- 「構成」ページの「ファクタの識別」で、次の情報を入力します。
  - 定数: 「取得メソッド」フィールドで検出された定数値を取得してファクタ・アイデンティティを解決します。
  - メソッド別: 「取得メソッド」フィールドに指定されたPL/SQL式を実行して、ファクタ・アイデンティティを設定します。

    たとえば、式でシステム日付を取得するとします。

    ```
    to_char(sysdate,'yyyy-mm-dd')
    ```

2015年12月15日の場合、「メソッド」オプションで次の値が返されます。

```
2015-12-15
```

- ファクタ: 子ファクタのアイデンティティを親ファクタにマップすることでファクタ・アイデンティティを特定します。親ファクタは、子ファクタと呼ばれる第2のファクタに基づいて値が解決されるファクタです。リレーションシップを確立するには、アイデンティティをマップします。(このオプションに取得メソッド式を指定する必要はありません。)

  アイデンティティのマップの詳細は、「他のファクタを使用するアイデンティティを構成するためのアイデンティティ・マップの使用方法」を参照してください。

親トピック: ファクタ作成の「構成」ページ

7.3.3.2 ファクタの識別の動作

ファクタ・アイデンティティはファクタの実際の値です(IP_Addressタイプを使用するファクタのIPアドレスなど)。

取得メソッドやアイデンティティ・マップ・ロジックに応じて、1つのファクタに複数のアイデンティティが存在する場合もあります。たとえば、Oracle Real Application Clusters環境ではDatabase_Hostnameなどのファクタには、複数のアイデンティティが存在することがあります。RDBMS環境では、Client_IPのようなファクタには複数のアイデンティティが存在する場合があります。取得メソッドはデータベース・セッションに基づいているため、これらのタイプのファクタの取得メソッドでは異なる値が返される場合があります。複数のレポートを使用してファクタ・アイデンティティ構成を追跡できます。

次のようにしてファクタの割当てを構成できます。

- データベース・セッションの確立時にファクタを割り当てます。
- 個々のリクエストを構成してファクタのアイデンティティを取得します。

Oracle Label Security統合を使用すると、Oracle Label Securityラベルでアイデンティティをラベル付けできます。また、アイデンティティに信頼レベルを割り当てることもできます。信頼レベルは、同じファクタの別のアイデンティティと比較した信頼の度合いを示す数値です。一般に、信頼レベルの数値が高く設定されているほど信頼の度合いも高くなります。信頼レベルの数値が負の場合は信頼できません。

データベース・セッション内では、Oracle Database Vault、および次のようなDVFスキーマ(ファクタ値を取得するファンクションを含む)に存在するパブリックからアクセス可能なPL/SQLファンクションのあるアプリケーションで、ファクタに割り当てられたアイデンティティを使用できます。

```
dvf.f$factor_name
```

これにより、(PL/SQL、SQL、Oracle仮想プライベート・データベース、トリガーなどを使用して)Oracleデータベース内からファクタのアイデンティティにグローバルにアクセスできます。たとえば、SQL*Plusでは次のようにします。

```
CONNECT sec_admin_owen
Enter password: password
SELECT DVF.F$DATABASE_IP FROM DUAL;
```

次のような出力が表示されます。

```
SELECT DVF.F$DATABASE_IP FROM DUAL;
F$DATABASE_IP
------------------------------------------------------------
192.0.2.1
```

GET_FACTORファンクションを使用して、パブリック・アクセスが可能になっているファクタのアイデンティティを見つけることもできます。たとえば:

```
SELECT GET_FACTOR('DATABASE_IP') FROM DUAL;
```

次のような出力結果が表示されます。

```
GET_FACTOR('DATABASE_IP')
------------------------------------------------------------
192.0.2.1
```

関連トピック

- ファクタへのアイデンティティの追加
- ファクタに関連するレポートおよびデータ・ディクショナリ・ビュー

親トピック: ファクタ作成の「構成」ページ

7.3.3.3 ファクタの評価情報の設定

「評価」で、ファクタの評価方法とアイデンティティの割当て方法を選択する必要があります。

セッション・ファクタのパフォーマンスへの影響の詳細は、「ファクタのパフォーマンスへの影響」を参照してください。この属性は必須です。

- 「構成」ページの「評価」で、次の情報を入力します。
  - セッション: データベース・セッションの作成時にファクタを評価します。
  - アクセス: データベース・セッションが初めて作成された際や、ファクタがアクセスされる(アプリケーションによる参照など)たびにファクタが評価されます。
  - 起動時: データベース・セッションの起動時にファクタを評価します。

親トピック: ファクタ作成の「構成」ページ

7.3.3.4 ファクタのOracle Label Securityラベル付け情報の設定

「ファクタ・ラベリング」で、ファクタ・アイデンティティによるOracle Label Security(OLS)ラベルの取得方法を選択する必要があります。

Oracle Label Security統合を使用する場合は、この設定が適用されます。OLSラベルを使用する場合、この属性は必須です。

- 「構成」ページの「ファクタ・ラベリング」で、次の情報を入力します。
  - 自己: Oracle Label Securityポリシーに関連付けられているラベルから直接ファクタのアイデンティティをラベル付けします。
  - ファクタ: 子ファクタ・ラベルが複数ある場合は、適用可能なOracle Label Securityポリシーに関連付けられているOracle Label Securityのアルゴリズムを使用してOracle Database Vaultによりラベルがマージされます。適用可能なそれぞれのOracle Label Securityポリシーに対して、ファクタ・アイデンティティはラベルを割り当てることができます。

関連トピック

- Oracle Database VaultとOracle Label Securityの統合

親トピック: ファクタ作成の「構成」ページ

7.3.3.5 ファクタの取得メソッドの設定

「取得メソッド」に、ファクタのアイデンティティを取得するPL/SQL式または定数を入力する必要があります。

- 「構成」ページの「取得メソッド」で、PL/SQLの取得メソッドを入力します。大/小文字混在で最大255文字まで使用できます。

次の取得メソッドでは、ユーザー・セッションのUSERENV名前空間からデータベース名(DB_NAME)を取得することで、DB_NAMEファクタの値が設定されます。

```
UPPER(SYS_CONTEXT('USERENV','DB_NAME'))
```

親トピック: ファクタ作成の「構成」ページ

7.3.3.6 取得メソッドの動作

「取得メソッド」により、ファクタの識別がメソッドまたは定数によって行われるファクタが識別されます。

ファクタの識別がファクタによって行われる場合、Oracle Database Vaultはアイデンティティ・マップによってファクタを識別します。独自のPL/SQL取得メソッドを作成するか、Oracle Database Vaultに用意されているファンクションを使用できます。取得メソッドの作成に使用可能な、ファクタに固有の一般的なユーティリティ・ファンクションについては、次の項を参照してください。

- Oracle Database VaultのDVF PL/SQLファクタ・ファンクション
- DBMS_MACADMファクタのプロシージャおよびファンクション
- Oracle Database VaultユーティリティのAPI

取得メソッドの例として、Oracle Database Vaultが提供するデフォルトのファクタも参照してください。これらのファクタの説明は、「デフォルトのファクタ」を参照してください。

「ファクタの識別」で次の設定を選択した場合、「取得メソッド」フィールドは必須です。

- メソッド: 「取得メソッド」フィールドにメソッドを入力します。
- 定数: 「取得メソッド」フィールドに定数を入力します。

ファクタ・アイデンティティとして返される値は、VARCHAR2文字列またはこの型に変換可能である必要があります。

式には、パッケージ・ファンクションまたはスタンドアロン・ファンクションを含めることができます。式がschema.function_nameなどの完全修飾ファンクションであることを確認してください。完全なSQL文は含めないでください。アプリケーション・パッケージまたはファンクションを使用している場合は、オブジェクトのEXECUTE権限のあるDVSYSを指定する必要があります。

次の書式を使用してファンクション・シグネチャを記述します。

```
FUNCTION GET_FACTOR RETURN VARCHAR2
```

親トピック: ファクタ作成の「構成」ページ

7.3.3.7 ファクタの検証メソッドの設定

検証メソッドでは、PL/SQL式を使用してブール値を返し、ファクタのアイデンティティを検証します。

「検証メソッド」で、ブール値(TRUEまたはFALSE)を返すPL/SQL式を入力し、(GET_FACTORファンクションで)取得されるファクタのアイデンティティまたは(SET_FACTORファンクションで)ファクタに割り当てられる値を検証する必要があります。

取得または割り当てられる値に対してメソッドがFalseと評価されると、ファクタ・アイデンティティはNULLに設定されます。このオプションの機能により、ファクタが正しく取得および設定されることがさらに確実になります。このフィールドには、大/小文字混在で最大で255文字まで入力できます。

式には、パッケージ・ファンクションまたはスタンドアロン・ファンクションを含めることができます。式がschema.function_nameなどの完全修飾ファンクションであることを確認してください。完全なSQL文は含めないでください。アプリケーション・パッケージまた

はファンクションを使用している場合は、オブジェクトのEXECUTE権限のあるDVSYSを指定する必要があります。

- 「構成」ページの「検証メソッド」で、次の形式のいずれかを使用するファンクションを作成します。
  - `FUNCTION IS_VALID RETURN BOOLEAN`

    この書式では、ファンクション・ロジック内の`DVF.F$factor_name`ファンクションを使用できます。セッションによって評価されるファクタに適しています。
  - `FUNCTION IS_VALID(p_factor_value VARCHAR2) RETURN BOOLEAN`

    この書式では、ファクタ値が検証ファンクションに直接渡されます。これは、アクセスごとに評価するファクタに適しています。また、セッションごとに評価するファクタにも有効です。

関連トピック

- DBMS_MACADMファクタのプロシージャおよびファンクション
- Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクション
- Oracle Database VaultのDVF PL/SQLファクタ・ファンクション
- Oracle Database VaultユーティリティのAPI

親トピック: ファクタ作成の「構成」ページ

7.3.4 ファクタ作成の「オプション」ページ

「オプション」ページでは、ルール・セットをファクタに割り当て、エラー・オプションを設定し、非統合監査について、監査オプションを設定します。

- ファクタへのルール・セットの割当て
  ファクタ・アイデンティティの設定をルール・セットによって制御するかどうかを、「割当てルール・セット」でルール・セットを選択できます。
- ファクタのエラー・オプションの設定
  「エラー・オプション」で、ファクタ・アイデンティティが解決されない場合に発生する処理を設定します。
- ファクタの監査オプションの設定
  統合監査環境を使用していない場合、「監査オプション」で、監査証跡を生成できます。
- ファクタ監査の動作
  統合監査が有効になっているかどうかは、監査がファクタに対してどのように処理されるかに影響します。

親トピック: ファクタの作成

7.3.4.1 ファクタへのルール・セットの割当て

ファクタ・アイデンティティの設定をルール・セットによって制御する場合は、「割当てルール・セット」でルール・セットを選択できます。

たとえば、ルール・セットを使用して、既知のアプリケーション・サーバーまたはプログラムからデータベース・セッションが発生する時期を決定できます。

- 「オプション」ページの「割当てルール・セット」で、リストからルール・セットを選択します。

この属性は、JDBC接続プールを使用するWebアプリケーションなどのデータベース・アプリケーションで、現在のデータベース・セッションに対するファクタ・アイデンティティを動的に設定する必要がある場合に特に有用です。たとえば、Webアプリケーションで、そのWebアプリケーションにログインするデータベース・アカウントの地理的位置を割り当てる場合があります。これを行うには、WebアプリケーションではJDBCコール可能文またはOracle Data Provider for .NET (ODP.NET)を使用して、PL/SQLファンクションSET_FACTORを実行できます。たとえば:

```
BEGIN
 SET_FACTOR('GEO_STATE','VIRGINIA');
END;
```

その後、GEO_STATEファクタの割当てルールを作成し、その他のファクタまたはルール式に基づいてGEO_STATEファクタの設定を許可または禁止できます。

関連トピック

- ルール・セットの構成
- ファクタの設定

親トピック: ファクタ作成の「オプション」ページ

7.3.4.2 ファクタのエラー・オプションの設定

「エラー・オプション」で、ファクタ・アイデンティティが解決されない場合に発生する処理を設定します。

- 「オプション」ページの「エラー・オプション」で、次の値から選択します。
  - エラー・メッセージの表示: データベース・セッションに対するエラー・メッセージを表示します。
  - エラー・メッセージを表示しない: エラー・メッセージは表示されません。

    「エラー・メッセージを表示しない」を選択して監査を有効にする利点は、潜在的な侵入者のアクティビティを追跡できるということです。監査レポートにより侵入者のアクティビティを把握できますが、エラー・メッセージが表示されないため、侵入者は監査が行われていることに気付きません。

新規ファクタを作成すると、アイデンティティを構成できます。これを実行するには、ファクタを編集してアイデンティティを追加します。

親トピック: ファクタ作成の「オプション」ページ

7.3.4.3 ファクタの監査オプションの設定

統合監査環境を使用していない場合、「監査オプション」で、監査証跡を生成できます。

- 「オプション」ページの「監査オプション」で、次の値から選択します。
  - 行わない: 監査は実行されません。
  - 常時: ファクタの評価時には、常に監査レコードが作成されます。次に説明する条件から選択できます。
  - 検証がFalse: 検証メソッド(存在する場合)で`False`が返された場合に、監査レコードが作成されます。
  - 取得エラー: エラー(`No data found`、`Too many rows`など)のため、ファクタのアイデンティティを解決および割当てできない場合に、監査レコードを作成します。
  - 信頼レベルがNULL: ファクタの解決されたアイデンティティに割り当てられている信頼レベルが`NULL`の場合に、監査レコードが作成されます。

    信頼レベルの詳細は、「ファクタ・アイデンティティの作成および構成」を参照してください。
  - 信頼レベルがゼロ未満: ファクタの解決されたアイデンティティに割り当てられている信頼レベルがゼロ未満の場合に、監査レコードが作成されます。
  - 検証エラー: 検証メソッド(存在する場合)でエラーが返された場合に、監査レコードが作成されます。

親トピック: ファクタ作成の「オプション」ページ

7.3.4.4 ファクタの監査の動作

統合監査が有効になっているかどうかは、監査がファクタに対してどのように処理されるかに影響します。

非統合監査環境では、Oracle Database Vaultは、監査証跡をDVSYS.AUDIT_TRAIL$表に書き込みます(「Oracle Database Vaultの監査」を参照)。

統合監査が有効な場合、この設定では監査レコードは取得されません。かわりに、『Oracle Databaseセキュリティ・ガイド』の説明に従い、この情報を取得する監査ポリシーを作成できます。

ファクタの監査レポートを使用して、生成された監査レコードを表示できます。(詳細は、「ファクタに関連するレポートおよびデータ・ディクショナリ・ビュー」を参照)。また、一度に複数の監査オプションを選択できます。各オプションはビット・マスクに変換され、集計の動作を決定するために追加されます。ファクタにエラーがないかぎり、監査のパフォーマンスへの影響はほとんどありません。

親トピック: ファクタ作成の「オプション」ページ

7.4 ファクタへのアイデンティティの追加

新しいファクタを作成したら、ファクタにアイデンティティを追加できます。

- ファクタ・アイデンティティについて
  アイデンティティは、IP_Addressファクタ・アイデンティティの192.0.2.4などファクタの実際の値です。
- 信頼レベルについて
  信頼レベルを使用することにより、信頼できるかどうかの尺度を示す数値を割り当てることができます。
- ラベル・アイデンティティについて
  ファクタ・アイデンティティにOracle Label Security(OLS)ラベルを割り当てることができます。
- ファクタ・アイデンティティの作成および構成
  Oracle Database Vault Administratorで、ファクタ・アイデンティティを作成および構成できます。
- ファクタ・アイデンティティの削除
  ファクタ・アイデンティティを削除する場合、ファクタに関連するOracle Database Vaultビューに問い合せることで、そのファクタ・アイデンティティへの様々な参照を特定できます。
- 他のファクタを使用するアイデンティティを構成するためのアイデンティティ・マッピングの使用方法
  アイデンティティ・マッピングを使用すると、ファクタのグループを使用してアイデンティティ値を管理できます。

親トピック: ファクタの構成

7.4.1 ファクタ・アイデンティティについて

アイデンティティは、IP_Addressファクタ・アイデンティティの192.0.2.4などファクタの実際の値です。

指定されたデータベース・セッションのファクタ・アイデンティティは、「ファクタの作成」で説明されている「ファクタの識別」および「取得メソッド」フィールドを使用して実行時に割り当てられます。次のような場合には、さらにアイデンティティを構成できます。

- ファクタの既知のアイデンティティを定義する場合
- ファクタ・アイデンティティに信頼レベルを追加する場合
- ファクタ・アイデンティティにOracle Label Securityラベルを追加する場合
- アイデンティティ・マップを使用して子ファクタによりファクタ・アイデンティティを解決する場合

関連トピック

- チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

親トピック: ファクタへのアイデンティティの追加

7.4.2 信頼レベルについて

信頼レベルを使用することにより、信頼できるかどうかの尺度を示す数値を割り当てることができます。

信頼値1は信頼度が低いことを意味します。値が大きければ信頼度も高くなります。負の値またはゼロは信頼できないことを意味します。ファクタ取得メソッドにより返されたファクタ・アイデンティティがアイデンティティに定義されていない場合は、Oracle Database Vaultによりそのアイデンティティに自動的に負の信頼レベルが割り当てられます。

実行時にファクタ・アイデンティティの信頼レベルを特定するために、DVSYSスキーマのGET_TRUST_LEVELおよびGET_TRUST_LEVEL_FOR_IDENTITYファンクションを使用できます。

たとえば、Networkという名前のファクタを作成したとします。Networkファクタに次のようなアイデンティティを作成できます。

- Intranet(信頼レベル10)
- VPN(仮想プライベート・ネットワーク)(信頼レベル5)
- Public(信頼レベル1)

ポリシー決定の基準を信頼レベルに置くルール式(またはカスタム・アプリケーション・コード)を作成できます。たとえば、GET_TRUST_LEVELファンクションを次のように使用して、5より大きい信頼レベルを検出できます。

```
GET_TRUST_LEVEL('Network') > 5
```

または、次のようにDBA_DV_IDENTITYデータ・ディクショナリ・ビューでSELECT文を使用して、信頼レベルが5以上のNetworkファクタを検出できます。

```
SELECT VALUE, TRUST_LEVEL FROM DBA_DV_IDENTITY
   WHERE TRUST_LEVEL >= 5
   AND FACTOR_NAME='Network'
```

次のような出力が表示されます。

```
F$NETWORK GET_TRUST_LEVEL('NETWORK')
-----------------------------------
VPN                                5
INTRANET                          10
```

前の例では、VPNのNetworkファクタ・アイデンティティは信頼されており(値が5)、INTRANETドメインのアイデンティティはより信頼度の高い10です。

関連トピック

- Oracle Database VaultレルムのAPI

親トピック: ファクタへのアイデンティティの追加

7.4.3 ラベル・アイデンティティについて

ファクタ・アイデンティティにOracle Label Security(OLS)ラベルを割り当てることができます。

簡単に説明すると、ラベルはデータベース表の行に権限を割り当てるために行の識別子の役割を果します。ファクタの「ファクタ・ラベリング」属性により、ファクタが「自己」または「ファクタ」のいずれにラベル付けされるかが決まります。「ファクタ・ラベリング」属性に「自己」を設定すると、OLSラベルをファクタ・アイデンティティに関連付けられます。「ファクタ・ラベリング」属性に「ファクタ」を設定すると、Oracle Database Vaultにより子ファクタ・アイデンティティのラベルからファクタ・アイデンティティ・ラベルが導出されま

す。ラベルのある子ファクタ・アイデンティティが複数ある場合は、適用可能なファクタのOracle Label Securityポリシーに関連付けられているOLSアルゴリズムを使用して、Oracle Database Vaultによりラベルがマージされます。

関連項目:

ラベルの詳細は、『Oracle Label Security管理者ガイド』を参照してください

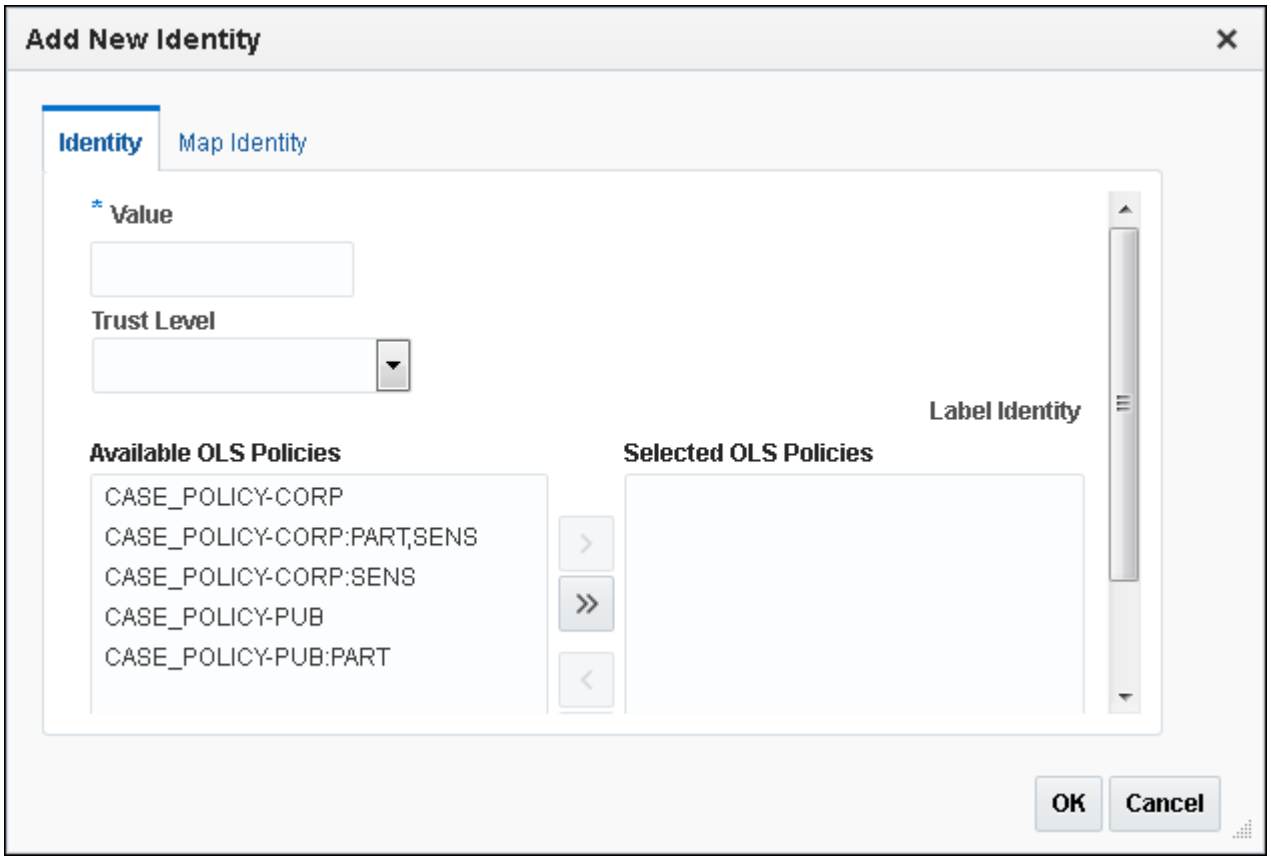
親トピック: ファクタへのアイデンティティの追加

7.4.4 ファクタ・アイデンティティの作成および構成

Oracle Database Vault Administratorで、ファクタ・アイデンティティを作成または構成できます。

1. 「ファクタの作成」ページのアイデンティティの選択ページで、「新規アイデンティティの追加」を選択します。

   「新規アイデンティティの追加」ウィンドウが表示されます。



2. 「アイデンティティ」サブページで、次の値を入力します。
   - 値: 大/小文字混在で1024文字以内でアイデンティティの値を入力します。この属性は必須です。
   - 信頼レベル: 次のいずれかの信頼レベルを選択します。
     - 信頼度高: 信頼レベル値10が割り当てられます。
     - 信頼: 信頼レベル値5が割り当てられます。
     - 信頼度低: 信頼レベル値1が割り当てられます。
     - 信頼されない: 信頼レベル値-1が割り当てられます。
     - 信頼レベルが未定義: 信頼レベル値NULLが割り当てられます(デフォルト)。

信頼レベルの詳細は、「信頼レベルについて」を参照してください。

- ラベル・アイデンティティ: オプションで、使用可能なOracle Label Securityポリシーのリストから選択し、「移動」ボタンをクリックして、ポリシーを「選択したOLSポリシー」リストに移動します。

  リストには、サイトのOracle Label Securityインストールのデータ・ラベルが表示されます。詳細は、『Oracle Label Security管理者ガイド』を参照してください。

  ラベル・アイデンティティの詳細は、「ラベル・アイデンティティについて」を参照してください。

3. 「OK」をクリックすると、「ファクタの作成」:「アイデンティティ」ページに戻ります。
4. 「次へ」をクリックしてファクタ設定を表示します。
5. 「終了」をクリックします。

親トピック: ファクタへのアイデンティティの追加

7.4.5 ファクタ・アイデンティティの削除

ファクタ・アイデンティティを削除する場合、ファクタに関連するOracle Database Vaultビューに問い合せることで、そのファクタ・アイデンティティへの参照を特定できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「ファクタ」をクリックします。
3. アイデンティティを削除するファクタを選択してから、「編集」をクリックします。
4. 「ファクタの編集」ページで、「アイデンティティ」ページになるまで「次へ」をクリックします。
5. 移動するファクタ・アイデンティティを選択します。
6. 「削除」をクリックします
7. 「完了」、「終了」の順にクリックします。

関連トピック

- Oracle Database Vaultのデータ・ディクショナリ・ビュー

親トピック: ファクタへのアイデンティティの追加

7.4.6 他のファクタを使用するアイデンティティを構成するためのアイデンティティ・マップの使用方法

アイデンティティ・マッピングを使用してファクタのグループを使用すると、アイデンティティ値を管理できます。

- アイデンティティ・マッピングについて
  ファクトリ・アイデンティティを作成している場合、それをマップできます。
- ファクタへのアイデンティティのマッピング
  2つのファクタに親子関係を作成すると、ファクタにアイデンティティをマップできます。

親トピック: ファクタへのアイデンティティの追加

7.4.6.1 アイデンティティ・マッピングについて

ファクタ・アイデンティティを作成する間に、そのアイデンティティをマップできます。

アイデンティティ・マッピングは、他(子)のファクタを使用してファクタを識別するプロセスです。これはファクタの組合せをファクタの論理アイデンティティに変換する方法です。また、連続するアイデンティティ値(温度など)や連続しない大きなアイデンティティ値(IPアドレスの範囲など)を論理セットに変換する方法でもあります。アイデンティティのマッピングにおける構成の問題を確認するには、「アイデンティティ構成の問題」レポートを実行します。

親ファクタの別のアイデンティティを構成ファクタの別のアイデンティティにマップできます。たとえば、INTRANETアイデンティティは192.0.2.1から192.0.2.24の範囲のIPアドレスにマップします。REMOTEアイデンティティは、192.0.2.1から192.0.2.24の範囲のアドレスを除くIPアドレスにマップします。

アイデンティティ・マップに基づいて、セキュリティ・ポリシーを作成できます。たとえば、企業ネットワーク(INTRANET)内から接続している従業員とは対照的に、VPN(REMOTE)経由で接続している従業員には少ない権限を定義できます。

関連トピック

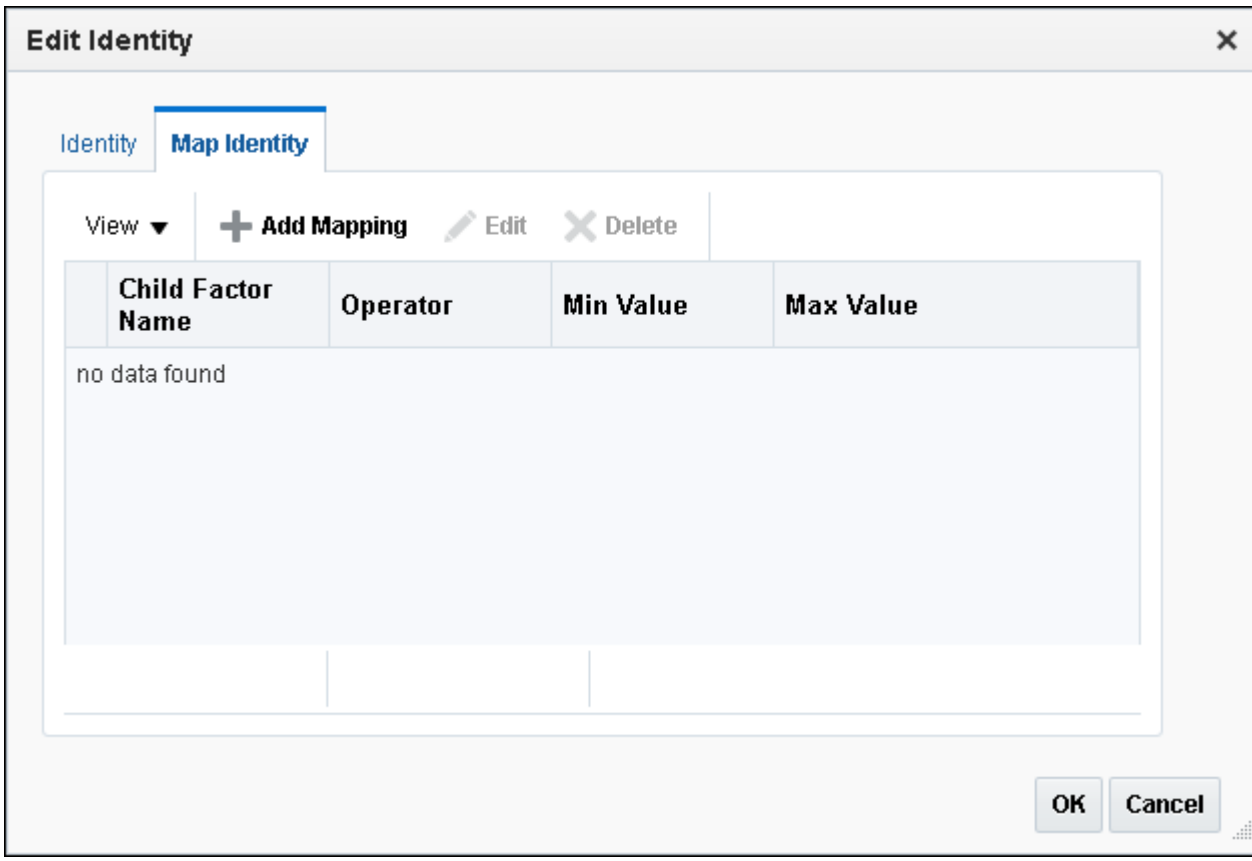
- チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

親トピック: 他のファクタを使用するアイデンティティを構成するためのアイデンティティ・マップの使用方法

7.4.6.2 ファクタへのアイデンティティのマッピング

2つのファクタに親子関係を作成すると、ファクタにアイデンティティをマップできます。

1. 「ファクタの作成」の説明に従い、親ファクタを作成し、「ファクタ」に属性「ファクタの識別」を設定します。
2. 「アイデンティティ」ページで、「ファクタ・アイデンティティの作成および構成」の説明に従い、親ファクタのアイデンティティを作成します。
3. 親ファクタのファクタとアイデンティティの組合せを、子のファクタとアイデンティティの組合せにマップします。次の手順を実行します。
    a. 「アイデンティティ」ページで、既存のアイデンティティを選択して「編集」をクリックするか、「新規アイデンティティの追加」をクリックして新しいアイデンティティを作成します。
    b. 「識別情報の編集」ウィンドウ(または「新規アイデンティティの追加」ウィンドウ)で、少なくとも「アイデンティティ」サブページの「値」フィールドが入力されていることを確認します。
    c. 「アイデンティティのマップ」タブをクリックします。



d. 「マッピングの追加」をクリックします。

e. 次の情報を入力します。

子ファクタ名: リストから子ファクタ名を選択します。

演算子: リストから演算子を選択します。

最小値: 最小値を入力します。

最大値: 最大値を入力します。

たとえば、ファクタ・ネットワークへの構成がClient_IP、「演算子」が「Between」、「最小値」が192.0.2.1、「最大値」が192.0.2.24に設定されているシナリオを想定します。この場合、クライアントIPアドレスが192.0.2.1から192.0.2.24の指定されたアドレスの範囲内である場合は、親ファクタが事前に定義されたアイデンティティ(INTRANETなど)と評価されます。

f. 「OK」をクリックして、「新規のアイデンティティ・マッピングの追加」ウィンドウを終了します。

g. 「OK」をクリックして、「新規のアイデンティティとマッピングの追加」ウィンドウを終了します。

4. 「完了」、「終了」の順にクリックします。

このプロセスを繰り返して、親ファクタ・アイデンティティの構成ファクタをさらに追加します。たとえば、ProgramファクタがOracle General Ledgerに解決され、Client_IPが192.0.2.1から192.0.2.24の間の場合には、値ACCOUNTING-SENSITIVEに解決するようNetworkファクタを構成できます。そのため、IPアドレスが192.0.2.12のクライアント上で稼働している、認可済の経理金融アプリケーション・プログラムがデータベースにアクセスすると、NetworkファクタはACCOUNTING-SENSITIVEに解決されます。Networkの値がACCOUNTING-SENSITIVEのデータベース・セッションには、Networkの値がINTRANETのデータベース・セッションよりも多くのアクセス権があります。

親トピック: 他のファクタを使用するアイデンティティを構成するためのアイデンティティ・マップの使用方法

7.5 ファクタの削除

ファクタを削除する前に、そのファクタへの参照を削除する必要があります。

ファクタに関連するOracle Database Vaultビューを問い合せることで、ファクタとそのアイデンティティへの様々な参照を確認できます。詳細は、「Oracle Database Vaultのデータ・ディクショナリ・ビュー」を参照してください。

1. ルール・セット、ファクタ・アイデンティティおよびOracle Label Securityポリシーの関連付けなど、ファクタへの参照を削除します。

   これを行うには、ファクタを編集します。「オプション」ページからルール・セットを、「アイデンティティ」ページからOracle Label Securityポリシーの関連付けおよびアイデンティティを見つけて削除できます。

2. Oracle Database Vaultの「管理」ページで、「ファクタ」を選択します。
3. 「ファクタ」ページで削除するファクタを選択します。
4. 「削除」をクリックします。
5. 「確認」ウィンドウで、「はい」をクリックします。

親トピック: ファクタの構成

7.6 ファクタの動作

セッションが確立されると、Oracle Database Vaultではファクタが処理されます。

- セッションの確立時のファクタの処理方法
  Oracle Database Vaultは、セッションの開始時に基づいてファクタを評価します。
- ファクタの取得方法
  データベース・セッション内のファクタは、DVFファクタ・ファンクションまたはGET_FACTORファンクションを使用していつでも取得できます。
- ファクタの設定方法
  データベース・セッション中はいつでもファクタにアイデンティティを割り当てられますが、ファクタ割当てルール・セットがTrueと評価される場合にかぎります。

親トピック: ファクタの構成

7.6.1 セッション確立時のファクタの処理

セッションが開始した時刻に基づいて、Oracle Database Vaultではファクタが評価されます。

データベース・セッションが確立されると、次のアクションが発生します。

1. 各データベース・セッションの開始時に、Oracle Database Vaultは、データベース・インスタンス内のデフォルトおよびユーザー作成のすべてのファクタの評価を開始します。

   適用可能な場合、評価はセッションの通常のデータベース認証、およびOracle Label Securityセッション情報の初期化後に開始されます。

2. ファクタの評価段階において、ファクタ初期化プロセスがメソッドまたは定数によって識別されるすべてのファクタの取得メソッドを実行し、セッションのファクタ・アイデンティティを解決します。

   ファクタのエラー・オプション設定は、ファクタ初期化プロセスには影響しません。

3. ファクタに検証メソッドが定義されている場合は、Oracle Database Vaultによりその検証メソッドが実行され、ファク

タのアイデンティティ(値)が検証されます。検証メソッドが失敗するかFalseが返された場合、ファクタのアイデンティティは未定義(NULL)です。

4. ファクタにアイデンティティが定義されている場合、Oracle Database Vaultは定義されているアイデンティティに基づいてファクタの信頼レベルを解決します。ファクタのアイデンティティが定義済のアイデンティティのリストに定義されている場合、Oracle Database Vaultは構成されている信頼レベルを割り当てます。そうでない場合は-1が設定されます。ファクタにアイデンティティが定義されていない場合、信頼レベルは未定義(NULL)になります。
5. ファクタ評価、ファクタ検証および信頼レベル解決の結果により、Database Vaultはファクタ監査構成の指示に従って評価の詳細を監査します。
6. メソッドまたは定数によって識別されるすべてのファクタの評価が完了すると、ファクタ構成アイデンティティに定義されているアイデンティティ・マップを使用して、その他のファクタによって識別されるファクタが解決されます。

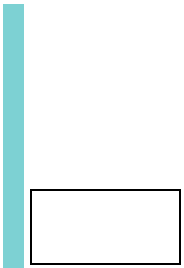
   ファクタ構成アイデンティティの評価順序は、アイデンティティ値のASCIIソートにより決まります。Oracle Database Vaultは、アルファベット順で最初にソートされたアイデンティティ・マップを使用して評価します。ファクタTESTにXおよびYというアイデンティティがあるとします。さらに、アイデンティティXおよびYに、ファクタA、B、Cのアイデンティティに依存するアイデンティティ・マップがある場合、次のマップが行われます。

   - A=1およびB=1の時はXがマップされます。
   - A=1、B=1およびC=2の時はYがマップされます。

   この場合、最初に評価されるのはXです。Yは評価されませんが、TESTファクタの成功に必要な条件にCのマップが一致した場合はどうなるでしょうか。Xの前にYをマップして、A、BおよびCが最初に評価されるように、逆にマップする必要があります。逆にマップするには、YをVという名前(またはXの前にソートされるアルファベット値)に変更します。これにより適切に解決されます。

   このアルゴリズムはASCIIソートの順序が適切な場合に機能し、アイデンティティは同レベルの同じ番号のファクタをマップします。
7. ファクタの初期化が終了すると、Oracle Database VaultのOracle Label Securityとの統合が行われます。

このプロセスが終了すると、Oracle Database Vaultはコマンド・ルールがCONNECTイベントと関連付けられていることを確認します。ルール・セットがCONNECTイベントと関連付けられている場合は、ルール・セットが評価されます。ルール・セットがFalseと評価されるかエラーが戻されると、セッションは終了します。セッションが終了する前に、ルール・セットに関連付けられた監査またはコール・ハンドラが実行されます。



ノート:

不用意に、他のユーザーをデータベースからロックアウトする可能性があるため、コマンド・ルールを CONNECT イベントに関連付ける際は注意してください。通常、CONNECT のコマンド・ルールを作成する場合は、関連付けられたルール・セットの評価オプションを「いずれか True」に設定します。

不用意に他のユーザーをロックアウトした場合は、一時的に Oracle Database Vault を無効にして、CONNECT コマンド・ルールを無効にし、Oracle Database Vault を再び有効にして、問題の原因となっているファクタ・コードを修正します。これを実行する方法の例は、「テストが失敗した場合」で説明しています。

親トピック: ファクタの動作

7.6.2 ファクタの取得

データベース・セッション内のファクタは、DVFファクタ・ファンクションまたはGET_FACTORファンクションを使用していつでも取得できます。

使用可能なファクタのリストを確認するには、「DBA_DV_FACTORビュー」で説明したDBA_DV_FACTORデータ・ディクショナリ・ビューに問い合せます。

例7-1に、GET_FACTORファンクションの使用例を示します。

例7-1 GET_FACTORを使用したファクタの取得

```
SELECT GET_FACTOR('client_ip') FROM DUAL;
```

DVFファクタ・ファンクションまたはGET_FACTORから取得されたファクタ値は、次に示す方法で使用できます。

- Oracle Database Vaultルール式
- Oracle Database Vault環境のすべてのデータベース・セッションで使用可能なカスタム・アプリケーション・コード

DVFファクタ・ファンクションについては、「Oracle Database VaultのDVF PL/SQLファクタ・ファンクション」で詳しく説明しています。

「セッション確立時のファクタの処理」で説明されているように、ファクタ評価をセッションに設定した場合は、Oracle Database Vaultにより、確立したセッション・コンテキストから値が取得されます。

「セッション確立時のファクタの処理」で説明されているように、ファクタ評価を「アクセス」に設定した場合は、ファクタが取得されるたびに、Oracle Database Vaultによりステップ2からステップ5(またはステップ6)が実行されます。

ファクタにエラー・オプションを定義し、エラーが発生した場合には、エラー・メッセージが表示されます。

親トピック: ファクタの動作

7.6.3 ファクタの設定

データベース・セッション中はいつでもファクタにアイデンティティを割り当てられますが、ファクタ割当てルール・セットがTrueと評価される場合にかぎります。

SET_FACTORファンクションを使用することにより、アプリケーション・コード内でこれを実行できます。Javaコードでは、JDBCクラス`java.sql.CallableStatement`を使用してこの値を設定できます。たとえば:

```
java.sql.Connection connection ;
...
java.sql.CallableStatement statement =
   connection.prepareCall("{call SET_FACTOR('FACTOR_X', ?)}");
statement.setString(1, "MyValue");
boolean result = statement.execute();
...
```

Oracle Data Provider for .NET(ODP.NET)を使用して記述されたアプリケーションなど、Oracle PL/SQLファンクションの実行が可能なアプリケーションは、このプロシージャを使用できます。

この概念は、ファクタ値の設定時期をルール・セットで制御する機能が追加された標準のOracle DBMS_SESSION.SET_IDENTIFIERプロシージャに似ています。ルール・セットの評価がTrueの場合、「セッション確立時のファクタの処理」のステップ2から5が実行されます。

ファクタに割当てルール・セットを関連付けていない、またはルール・セットでFalse(またはエラー)が返された場合、

SET_FACTORファンクションを使用してファクタを設定しようとすると、Oracle Database Vaultによってエラー・メッセージが送信されます。

親トピック: ファクタの動作

7.7 チュートリアル: データベースへの非定型ツール・アクセスの阻止

このチュートリアルでは、ファクタを使用して非定型ツール(SQL*Plusなど)がデータベースにアクセスできないようにする方法を示します。

- このチュートリアルについて
  多くのデータベース・アプリケーションには、ユーザーのアクションを明示的に制御する機能が含まれています。
- ステップ1: HRおよびOEユーザー・アカウントの有効化
  後でこのチュートリアルでOracle Database VaultコンポーネントをテストするときにHRアカウントとOEアカウントを使用する必要があります。
- ステップ2: ファクタの作成
  HRおよびOEアカウントがアクティブであることを確認後、ファクタを作成します。
- ステップ3: ルール・セットおよびルールの作成
  ファクタを作成後、ファクタとともに使用するルール・セットおよびルールを作成します。
- ステップ4: CONNECTコマンド・ルールの作成
  CONNECTコマンド・ルールは、CONNECT SQL文を制御します。
- ステップ5: 非定期ツール・アクセス制限のテスト
  Oracle Database Vaultの変更を有効にするために、SQL*Plusセッションを再起動する必要はありません。
- ステップ6: このチュートリアルのコンポーネントの削除
  コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

親トピック: ファクタの構成

7.7.1 このチュートリアルについて

多くのデータベース・アプリケーションには、ユーザーのアクションを明示的に制御する機能が含まれています。

ただし、非定型問合せツール(SQL*Plusなど)には、これらの制御機能がないことがあります。このため、ユーザーは非定型ツールを使用して、通常はデータベース・アプリケーションで実行できないアクションを、データベースで実行できる場合があります。Oracle Database Vaultのファクタ、ルール・セットおよびコマンド・ルールを組み合せて使用すると、非定型問合せツールによるデータベースへの不正アクセスを阻止できます。

次のチュートリアルでは、ユーザーHRおよびOEがSQL*Plusを使用するのを阻止します。これを実行するには、システム上でアプリケーションを検索するファクタと、これら4ユーザーにSQL*Plusを制限するためのルールおよびルール・セットを作成する必要があります。次に、ルール・セットに関連付けられるCONNECT SQL文のコマンド・ルールを作成します。このファクタ、Client_Prog_NameはSYS_CONTEXT SQLファンクションのUSERENV名前空間のCLIENT_PROGRAM_NAME属性を使用して、Oracle Databaseの現在のインスタンスへのアクセスに使用されるアプリケーションの名前を検索します。SYS_CONTEXT SQLファンクションには、ユーザー・セッションの状態を検出するための便利なメソッドが多数用意されています。SYS_CONTEXTは、カスタム・ファクタを作成するための貴重なツールです。

関連項目:

SYS_CONTEXTファンクションの詳細は、『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: チュートリアル: データベースへの非定型ツール・アクセスの阻止

7.7.2 ステップ1: HRおよびOEユーザー・アカウントの有効化

後でこのチュートリアルのためにOracle Database Vaultコンポーネントをテストするときに、HRおよびOEアカウントを使用する必要があります。

1. DV_ACCTMGRロールを付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus accts_admin_ace
   Enter password: password
   ```

   マルチテナント環境で、適切なプラガブル・データベース(PDB)に接続する必要があります。

   たとえば:

   ```
   sqlplus accts_admin_ace@hrpdb
   Enter password: password
   ```

   使用可能なPDBを見つけるには、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. HRアカウントのステータスを確認します。

   ```
   SELECT USERNAME, ACCOUNT_STATUS FROM DBA_USERS WHERE USERNAME = 'HR';
   ```

3. HRが無効になり、ロックされている場合、次の文を入力してアクティブにします。

   ```
   ALTER USER HR ACCOUNT UNLOCK IDENTIFIED BY password;
   ```

   『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

4. OEアカウントに対して、これらのステップを繰り返します。

親トピック: チュートリアル: データベースへの非定型ツール・アクセスの阻止

7.7.3 ステップ2: ファクタの作成

HRおよびOEアカウントがアクティブであることを確認後、ファクタを作成します。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして接続します。

   たとえば:

   ```
   CONNECT leo_dvowner --Or, CONNECT leo_dvowner@hrpdb
   Enter password: password
   ```

2. ファクタを作成します。

   ```
   BEGIN
    DBMS_MACADM.CREATE_FACTOR(
     factor_name       => 'Client_Prog_Name',
     factor_type_name  => 'Application',
     description       => 'Stores client program name that connects to database',
     rule_set_name     => NULL,
     validate_expr     => NULL,
     get_expr          =>
   'UPPER(SYS_CONTEXT(''USERENV'',''CLIENT_PROGRAM_NAME''))',
   ```

```
  identify_by       => DBMS_MACUTL.G_IDENTIFY_BY_METHOD,
  labeled_by        => DBMS_MACUTL.G_LABELED_BY_SELF,
  eval_options      => DBMS_MACUTL.G_EVAL_ON_SESSION,
  audit_options     => DBMS_MACUTL.G_AUDIT_ON_GET_ERROR,
  fail_options      => DBMS_MACUTL.G_FAIL_SILENTLY);
END;
/
```

詳細は、次のとおりです。

- `factor_type_name`は、これがアプリケーション・ベース・ファクタであることを指定します。
- `get_expr`は、ファクタの式を定義します。この式は、USERENV名前空間と`CLIENT_PROGRAM_NAME`属性を使用して`SYS_CONTEXT`ファンクションをコールして、Oracle Databaseにログインしているプログラムを検索します。
- `identify_by`は、メソッドによってファクタを識別します。
- `labeled_by`は、Oracle Label Securityポリシーに関連付けられているラベルから直接ファクタのアイデンティティをラベル付けします(デフォルト)。
- `eval_options`は、データベース・セッションの作成時にファクタを評価します。
- `audit_options`は、`get_expr`がエラーを返した場合に監査します。
- `fail_silently`は、ファクタのエラー・メッセージを表示しません。

親トピック: チュートリアル: データベースへの非定型ツール・アクセスの阻止

7.7.4 ステップ3: ルール・セットとルールの作成

ファクタを作成後、ファクタとともに使用するルール・セットおよびルールを作成します。

1. Limit SQL*Plus Accessルール・セットを次のように作成します。

```
BEGIN
 DBMS_MACADM.CREATE_RULE_SET(
  rule_set_name     => 'Limit SQL*Plus Access',
  description       => 'Limits access to SQL*Plus for Apps Schemas',
  enabled           => DBMS_MACUTL.G_YES,
  eval_options      => DBMS_MACUTL.G_RULESET_EVAL_ANY,
  audit_options     => DBMS_MACUTL.G_RULESET_AUDIT_OFF,
  fail_options      => DBMS_MACUTL.G_RULESET_FAIL_SHOW,
  fail_message      => 'SQL*Plus access not allowed for Apps Schemas',
  fail_code         => 20461,
  handler_options   => DBMS_MACUTL.G_RULESET_HANDLER_OFF,
  handler           => NULL,
  is_static         => FALSE);
END;
/
```

詳細は、次のとおりです。

- `fail_options`は`fail_message`によって設定されたエラー・メッセージおよび`fail_code`によって設定されたエラー・コードを有効にして、エラーの場合に表示します。
- `is_static`は、ユーザー・セッション中にルール・セットを1度評価します。その後、値は再利用されます。

2. `CLIENT_PROGRAM_NAME`属性から返された内容に基づいて、ポリシーに適用するコンピュータの正確な設定を検索します。

```
SELECT SYS_CONTEXT('USERENV', 'CLIENT_PROGRAM_NAME') FROM DUAL;
```

出力は次のようになります。

```
SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')
----------------------------------------------------------------
sqlplus@nemosity (TNS V1-V3)
```

このチュートリアルの場合、コンピュータの名前はnemosityです。(TN V1-V3)出力は、TNSコネクタのバージョンを示します。

3. 次のルール・セットを作成します。

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name  => 'Prevent Apps Schemas Access to SQL*Plus',
  rule_expr  =>'UPPER (DVF.F$CLIENT_PROG_NAME) != ''SQLPLUS@NEMOSITY (TNS V1-
V3)'' AND DVF.F$SESSION_USER IN (''HR'', ''OE'')');
END;
/
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name  => 'Allow Non-Apps Schemas Access to SQL*Plus',
  rule_expr  =>'DVF.F$SESSION_USER NOT IN (''HR'', ''OE'')');
END;
/
```

このルールは次のように変換されます:「ユーザーHRおよびOEのSQL*Plusへのログインは阻止するが、他のユーザーのアクセスは許可する」

4. ルールをLimit SQL*Plus Accessルール・セットに追加します。

```
BEGIN
 DBMS_MACADM.ADD_RULE_TO_RULE_SET(
  rule_set_name => 'Limit SQL*Plus Access',
  rule_name     => 'Prevent Apps Schemas Access to SQL*Plus',
  rule_order    => 1);
END;
/
BEGIN
 DBMS_MACADM.ADD_RULE_TO_RULE_SET(
  rule_set_name => 'Limit SQL*Plus Access',
  rule_name     => 'Allow Non-Apps Schemas Access to SQL*Plus',
  rule_order    => 1);
END;
/
```

プロシージャが機能するには、rule_order設定が必要です。

親トピック: チュートリアル: データベースへの非定型ツール・アクセスの阻止

7.7.5 ステップ4: CONNECTコマンド・ルールの作成

CONNECTコマンド・ルールは、CONNECT SQL文を制御します。

このコマンド・ルールは、コマンドラインまたはSQL*Plusへのアクセスにサイトで使用されるその他のツールからSQL*Plusにログインする場合にも適用されます。

- CONNECTコマンド・ルールを次のように作成します。

```
BEGIN
 DBMS_MACADM.CREATE_COMMAND_RULE(
  command         => 'CONNECT',
  rule_set_name   => 'Limit SQL*Plus Access',
  object_owner    => '%',
```

```
  object_name     => '%',
  enabled         => DBMS_MACUTL.G_YES);
END;
/
```

詳細は、次のとおりです。

- `rule_set_name`は、Limit SQL*Plus Accessルール・セットとCONNECTコマンド・ルールを関連付けます。
- コマンド・ルールがすべてのユーザーに適用されるように、`object_owner`は%に設定されます。
- コマンド・ルールがすべてのオブジェクトに適用されるように、`object_name`は%に設定されます。
- `enabled`はコマンド・ルールをただちに使用できるように有効化します。

親トピック: チュートリアル: データベースへの非定型ツール・アクセスの阻止

7.7.6 ステップ5: 非定期ツール・アクセス制限のテスト

Oracle Database Vaultの変更を有効にするために、SQL*Plusセッションを再起動する必要はありません。

1. SQL*Plusで、ユーザーHRとして接続を試行します。

   ```
   CONNECT HR --Or, CONNECT HR@hrpdb
   Enter password: password
   ```

   次の出力が表示されます。

   ```
   ERROR:
   ORA-47306: 20461: Limit SQL*Plus Access rule set failed
   ```

   ユーザーHRは、SQL*Plusの使用を阻止されます。

2. 次に、ユーザーOEとして接続を試行します。

   ```
   CONNECT OE --Or, CONNECT OE@hrpdb
   Enter password: password
   ```

   次の出力が表示されます。

   ```
   ERROR:
   ORA-47306: 20461: Limit SQL*Plus Access rule set failed
   ```

   ユーザーOEも、SQL*Plusの使用を阻止されます。

3. ここで、ユーザーSYSTEMとして接続を試行します。

   ```
   CONNECT SYSTEM --Or, CONNECT SYSTEM@hrpdb
   Enter password: password
   Connected.
   ```

   ユーザーSYSTEMはデータベース・インスタンスにログインできる必要があります。SYS、Database Vault所有者アカウントおよびDatabase Vaultアカウント・マネージャ・アカウントもログインできます。

テストが失敗した場合

SYSTEMとして(またはルール式で指定されているその他の管理ユーザーのいずれかとして)データベース・インスタンスにログインできない場合、SQL*Plusは使用できません。

この問題は、次の方法で対処できます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   CONNECT sec_admin_owen --Or, CONNECT sec_admin_owen@hrpdb for a PDB
   Enter password: password
   ```

2. 次の文を入力して、CONNECTコマンド・ルールを削除します。

   ```
   EXEC DBMS_MACADM.DELETE_COMMAND_RULE ('CONNECT', '%', '%');
   ```

   Oracle Database Vaultを無効にしても、そのPL/SQLパッケージとDatabase Vault Administratorはまだ使用できます。

3. エラーがあるかどうかポリシー・コンポーネントを確認し、エラーを修正します。CONNECTコマンド・ルールを再作成し、テストします。

親トピック: チュートリアル: データベースへの非定型ツール・アクセスの阻止

7.7.7 ステップ6: このチュートリアルのコンポーネントの削除

コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

1. CONNECTコマンド・ルールを削除します。

   ```
   EXEC DBMS_MACADM.DELETE_COMMAND_RULE ('CONNECT', '%', '%');
   ```

2. Client_Prog_Nameファクタを削除します。

   ```
   EXEC DBMS_MACADM.DELETE_FACTOR('Client_Prog_Name');
   ```

3. Limit SQL*Plus Accessルール・セットを削除します。

   ```
   EXEC DBMS_MACADM.DELETE_RULE_SET('Limit SQL*Plus Access');
   ```

4. ルールを削除します。

   ```
   EXEC DBMS_MACADM.DELETE_RULE('Prevent Apps Schemas Access to SQL*Plus');
   EXEC DBMS_MACADM.DELETE_RULE('Allow Non-Apps Schemas Access to SQL*Plus');
   ```

5. 必要に応じて、DBV_ACCTMGRロールを付与されているユーザーとしてHRおよびOEアカウントをロックします。

   ```
   CONNECT accts_admin_ace --Or, CONNECT amalcolumn_dbacctmgr@hrpdb
   Enter password: password
   ALTER USER HR ACCOUNT LOCK;
   ALTER USER OE ACCOUNT LOCK;
   ```

親トピック: チュートリアル: データベースへの非定型ツール・アクセスの阻止

7.8 チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

このチュートリアルでは、ユーザーが使用しているドメインなど、セッション・データに基づいたユーザー・アクティビティを制限する方法を示します。

- このチュートリアルについて
  ファクタ・アイデンティティ・マップを使用して、データベース・アクティビティのセッションベースのユーザー制限を設定できます。
- ステップ1: 管理ユーザーの作成
  このチュートリアルを使用するには、管理ユーザーを作成する必要があります。

- ステップ2: ドメイン・ファクタへのアイデンティティの追加
  次に、デフォルト・ファクタであるDomainファクタにアイデンティティを追加する必要があります。
- ステップ3: DomainファクタのアイデンティティのClient_IPファクタへのマップ
  アイデンティティをDomainファクタに追加後、これをClient_IPファクタにマップします。
- ステップ4: 時間を設定するルール・セットの作成およびファクタ・アイデンティティの選択
  変更したファクタで使用できるルール・セットを作成する必要があります。
- ステップ5: ルール・セットを使用するコマンド・ルールの作成
  作成したルール・セットを使用するコマンド・ルールを作成する必要があります。
- ステップ6: ファクタ・アイデンティティの設定のテスト
  システム・クロックを再設定して、mwaldron管理ユーザーとしてログインし、表を作成することにより、設定をテストします。
- ステップ7: このチュートリアルのコンポーネントの削除
  コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

親トピック: ファクタの構成

7.8.1 このチュートリアルについて

ファクタ・アイデンティティ・マップを使用して、データベース・アクティビティのセッションベースのユーザー制限を設定できます。

たとえば、次の基準を使用して、データベースへの管理アクセスを制御するとします。

- 管理者が正しいIPアドレスからデータベースにアクセスしていることを確認する。
- データベース・アクセスを管理者の標準勤務時間に制限する。

このような構成は、様々なタイプの管理者(ローカルの内部管理者だけでなく、海外および契約管理者も含む)を制限する場合に便利です。

このチュートリアルでは、管理者が使用しているコンピュータのIPアドレスに基づく、セキュアおよび非セキュアなネットワーク・アクセスのアイデンティティが含まれるように、Domainファクタを変更します。管理者が標準勤務時間外に、あるいは異なるIPアドレスからアクションを実行しようとすると、Oracle Database Vaultはそれを阻止します。

親トピック: チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

7.8.2 ステップ1: 管理者ユーザーの作成

このチュートリアルを使用する前に、管理ユーザーを作成する必要があります。

1. SQL*Plusで、DV_ACCTMGRロールを付与されているユーザーとしてログインし、ユーザー・アカウントmwaldronを作成します。

   たとえば:

   ```
   sqlplus accts_admin_ace
   Enter password: password
   CREATE USER mwaldron IDENTIFIED BY password;
   ```

   『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

   マルチテナント環境で、適切なプラガブル・データベース(PDB)に接続する必要があります。

   たとえば:

```
sqlplus accts_admin_ace@hrpdb
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。

2. CREATE SESSION権限およびDBAロールを付与する権限を持つユーザーとして接続し、ユーザーmwaldronにこれらの権限を付与します。また、このユーザーは、Oracleシステム権限およびロール管理レルムの所有者として認可されている必要があります。

   たとえば:

```
CONNECT dba_psmith -- Or, CONNECT dba_psmith@hrpdb
Enter password: password
GRANT CREATE SESSION, DBA TO mwaldron;
```

親トピック: チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

7.8.3 ステップ2: Domainファクタへのアイデンティティの追加

次に、アイデンティティをDomainファクタ(デフォルトのファクタ)に追加する必要があります。

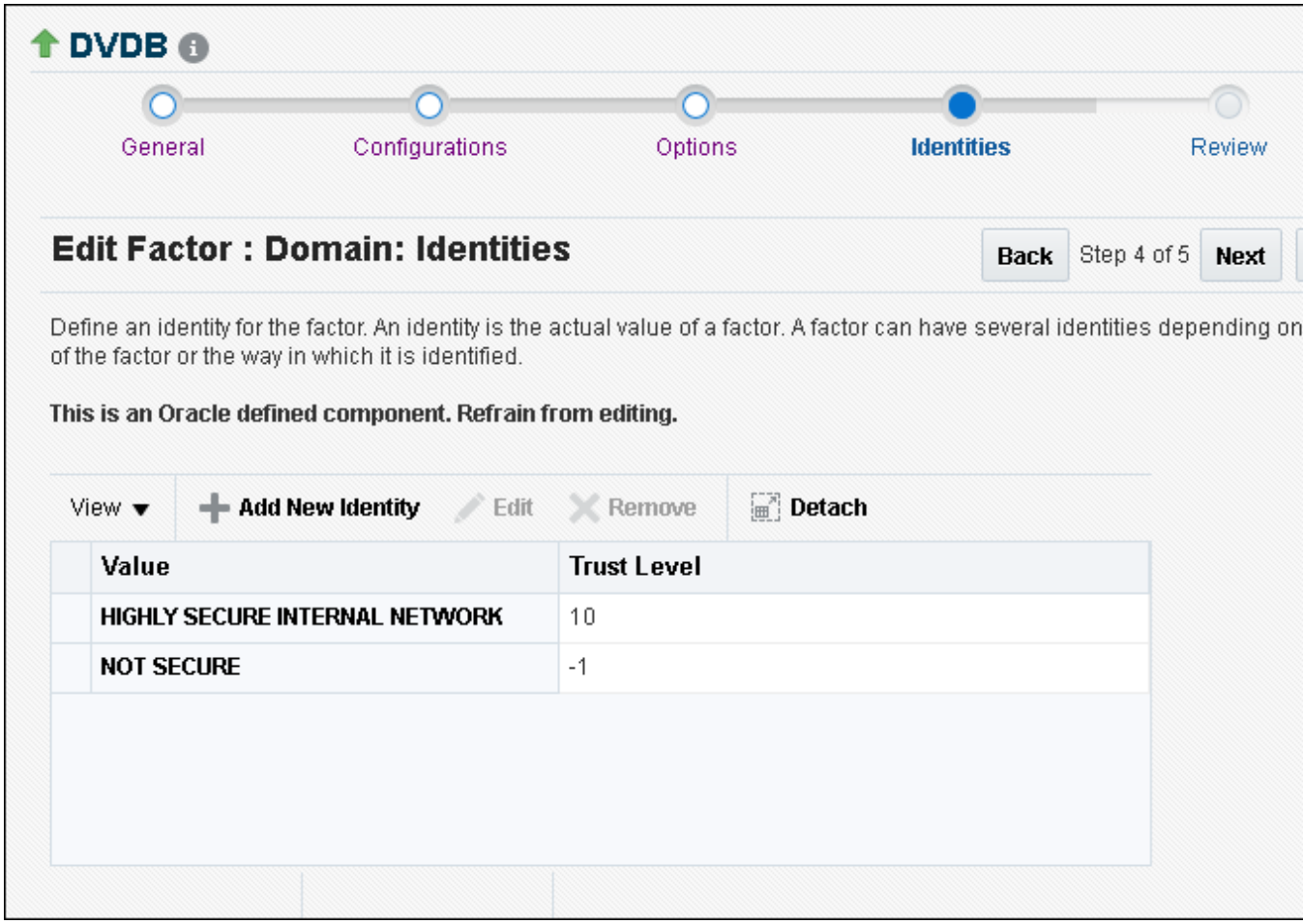
1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「ファクタ」をクリックします。

   「ファクタ」ページが表示されます。

3. 「Oracle定義のファクタの表示」チェック・ボックスを選択してデフォルトのファクタを表示します。
4. Domainファクタを選択し、「編集」を選択します。

   「ドメイン」ファクタが親ファクタになります。

5. 「アイデンティティ」ページになるまで、「次へ」ボタンをクリックします。
6. 「新規アイデンティティの追加」ボタンを選択します。
7. 「新規のアイデンティティとマッピングの追加」ページの「アイデンティティ」タブで、次の情報を入力します。
   - 値: HIGHLY SECURE INTERNAL NETWORKと入力します。
   - 信頼レベル: 「信頼度高」を選択します。
8. 「OK」をクリックします。
9. これらのステップを繰り返してNOT SECUREというもう1つのアイデンティティを作成し、その信頼レベルを「信頼されない」に設定します。



親トピック: チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

7.8.4 ステップ3: Domainファクタ・アイデンティティのClient_IPファクタへのマップ

アイデンティティをDomainファクタに追加後、これをClient_IPファクタにマップします。

Client_IPファクタはデフォルト・ファクタです。

1. 「アイデンティティ」ページで、HIGHLY SECURE INTERNAL NETWORKアイデンティティを選択して「編集」を選択します。
2. 「新規のアイデンティティとマッピングの追加」ウィンドウで、「アイデンティティのマップ」サブページを選択します。
3. 「アイデンティティのマップ」タブを選択して、「マッピングの追加」を選択します。
4. 「新規のアイデンティティ・マッピングの追加」ページで、次の情報を入力します。
   - 子ファクタ: 子ファクタとなる「Client_IP」を選択します。
   - 演算子: 「次と等しい」を選択します。
   - 最小値: 仮想マシン(`192.0.2.12`など)のIPアドレスを入力します。(これは、ユーザー`mwaldron`が使用するコンピュータです。このチュートリアルでは、自身のコンピュータのIPアドレスを入力できます。Microsoft Windowsを使用している場合は、ループバック・アダプタに割り当てられたIPアドレスを使用します。)
   - 最大値: このフィールドは空白のままにします。
5. 「OK」をクリックし、再度「OK」をクリックして「アイデンティティ」ページに戻ります。

NOT SECUREアイデンティティ用に次の2つのアイデンティティ・マップを作成します。作成するには、このアイデンティティを編集します。

子ファクタ	演算子	最小値	最大値
Client_IP	より小さい	192.0.2.5	(空白のままにする)
Client_IP	より大きい	192.0.2.20	(空白のままにする)

NOT SECUREアイデンティティでのアイデンティティ・マップは、ユーザーmwaldronによって使用されるIPアドレス(192.0.2.12)以外の範囲のIPアドレスにあります。ここでのIPアドレスは、mwaldronのIPアドレス以外のいずれかの範囲にある必要があります。

このアイデンティティ・マップにより、ユーザーが正しいIPアドレスからログインすると、Oracle Database VaultではHIGHLY SECURE INTERNAL NETWORKアイデンティティにより、その接続がセキュアであると判断する、という条件が作成されます。しかし、ユーザーが192.0.2.5未満または192.0.2.20より大きいIPアドレスからログインすると、NO SECUREアイデンティティにより、その接続はセキュアではないと判断されます。

7. 「OK」をクリックします。
8. 「完了」、「終了」の順にクリックします。
9. ファクタ・アイデンティティをテストします。

最初に、SQL*Plusにユーザーmwaldronとして接続しますが、データベース・インスタンスは指定しません。

```
CONNECT mwaldron -- Or, CONNECT mwaldron@hrpdb
Enter password: password
SELECT DVF.F$CLIENT_IP FROM DUAL;
```

次の出力が表示されます。

```
F$CLIENT_IP
------------------------------------
```

続いて次のように入力します。

```
SELECT DVF.F$DOMAIN FROM DUAL;
```

次の出力が表示されます。

```
F$DOMAIN
------------------------------------
NOT SECURE
```

ユーザーmwaldronはデータベース・インスタンスに直接接続していないので、Oracle Database Vaultではユーザーの接続元であるIPアドレスが認識されません。この場合、Oracle DatabaseではIPCプロトコルを使用して、IP値をNULLに設定する接続を実行します。したがって、この接続のアイデンティティはNOT SECUREに設定されます。

ここで、データベース・インスタンス(たとえば、orcl)を指定してSQL*Plusに接続し、再びファクタ・アイデンティティを確認します。

```
CONNECT mwaldron@orcl
Enter password: password
SELECT DVF.F$CLIENT_IP FROM DUAL;
```

次の出力が表示されます。

```
F$CLIENT_IP
------------------------------------
192.0.2.12
```

続いて次のように入力します。

```
SELECT DVF.F$DOMAIN FROM DUAL;
```

次の出力が表示されます。

```
F$DOMAIN
-----------------------------------
HIGHLY SECURE INTERNAL NETWORK
```

ユーザーmwaldronはorclデータベース・インスタンスに接続しているので、そのIPアドレスが認識されます。これはデータベースでTCPプロトコルが使用されていて、ホストIP値を適切に移入できるようになったからです。IPアドレスは正しい範囲内にあるため、ファクタ・アイデンティティはHIGHLY SECURE INTERNAL NETWORKに設定されます。

親トピック: チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

7.8.5 ステップ4: 時間を設定するルール・セットの作成およびファクタ・アイデンティティの選択

変更したファクタで使用できるルール・セットを作成する必要があります。

1. 「管理」ページの「Database Vaultコンポーネント」で、「ルール・セット」を選択します。
2. 「ルール・セット」ページで「作成」を選択します。
3. 「ルール・セットの作成」ページで、次の設定を入力します。
   - 名前: `Internal DBA Standard Working Hours`と入力します。
   - ステータス: 「有効」を選択します。
   - 評価オプション: 「すべてTrue」を選択します。

   残りの設定はデフォルトのままにします。
4. 「次へ」をクリックして、「ルールとの関連付け」ページを表示します。
5. 「ルールの作成」を選択します。
6. 「ルールの作成」ウィンドウで、次の情報を入力します。
   - 名前: `Internal DBA`
   - 式: `DVF.F$SESSION_USER='MWALDRON'`

     (ユーザー名を含む式を作成する場合、ユーザー名は大文字で入力します。データベースではユーザー名が大文字で格納されるためです。)
7. 「OK」をクリックします。
8. 「ルールの作成」ページを使用して、次のルールをさらに作成します。
   - 名前: `Internal Network Only`

     ルール式: `DVF.F$DOMAIN='HIGHLY SECURE INTERNAL NETWORK'`
   - 名前: `Week Day`

     ルール式: `TO_CHAR(SYSDATE, 'D') BETWEEN '2' AND '6'`
   - 名前: `Week Working Day Hours`

     ルール式: `TO_CHAR(SYSDATE, 'HH24') BETWEEN '08' AND '19'`
9. 「完了」、「終了」の順にクリックします。

親トピック: チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

7.8.6 ステップ5: ルール・セットを使用するコマンド・ルールの作成

作成したルール・セットを使用するコマンド・ルールを作成する必要があります。

1. 「管理」ページで「コマンド・ルール」を選択します。
2. 「コマンド・ルール」ページで「作成」を選択します。
3. 「コマンド・ルールの作成」ページで、次の設定を入力します。
   - コマンド: リストから「CREATE TABLE」を選択します。
   - ステータス: 「有効」を選択します。
   - 適用可能なオブジェクト所有者: %(デフォルト)に設定されていることを確認します。
   - 適用可能なオブジェクト名: %(デフォルト)に設定されていることを確認します。
   - ルール・セットの評価: リストから「内部DBA標準勤務時間」を選択します。
4. 「OK」をクリックします。

親トピック: チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

7.8.7 ステップ6: ファクタ・アイデンティティの設定のテスト

システム・クロックを再設定して、mwaldron管理ユーザーとしてログインし、表を作成することにより、設定をテストします。

1. システム時間を午後9時に設定します。

   UNIX: rootとしてログインし、dateコマンドを使用して時間を設定します。たとえば、今日の日付が2013年8月15日だとすると、次のように入力します。

   ```
   su root
   Password: password
   date --set="15 AUG 2013 21:00:00"
   ```

   Windows: 通常画面の右下隅にある時計アイコンをダブルクリックします。「日付と時刻のプロパティ」ウィンドウで、時刻を午後9時に設定し、「OK」をクリックします。

2. SQL*Plusで、ユーザーmwaldronとして接続し、表の作成を試行します。次の文で、orclを使用するデータベース・インスタンスの名前に置き換えます。

   ```
   CONNECT mwaldron@orcl
   Enter password: password
   CREATE TABLE TEST (num number);
   ```

   次の出力が表示されます。

   ```
   ORA-47400: Command Rule violation for create table on MWALDRON.TEST
   ```

   ユーザーmwaldronは勤務時間外に表を作成するため、Database Vaultにより阻止されます。

3. システム時間をローカル時間に再設定します。
4. SQL*Plusで、ユーザーmwaldronとして、表の作成を再試行します。

   ```
   CREATE TABLE TEST (num number);
   Table created.
   DROP TABLE TEST;
   Table dropped.
   ```

ここで、ユーザーmwaldronはローカル時間に、HIGHLY SECURE INTERNAL NETWORKアイデンティティに関連付けられたIPアドレスから作業を行っているので、表を作成できます。

5. ユーザーmwaldronとして再接続し、ここで接続コマンドにデータベース・インスタンス名を追加せずに、再び表を作成してみます。

```
CONNECT mwaldron -- Or, CONNECT mwaldron@hrpdb
Enter password: password
CREATE TABLE TEST (num number);
```

次の出力が表示されます。

```
ORA-47400: Command Rule violation for create table on MWALDRON.TEST
```

ユーザーmwaldronは正しい時間に表を作成しようとしていますが、orclデータベース・インスタンスに直接ログインしていないため、作成できません。Oracle Database Vaultでは、ユーザーがNOT SECUREアイデンティティを使用しているものと判断し、アクセスを拒否します。

親トピック: チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

7.8.8 ステップ7: この例で使用したコンポーネントの削除

コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

1. DV_ACCTMGRユーザーとしてデータベース・インスタンスにログインして、ユーザーmwaldronを削除します。

```
sqlplus accts_admin_ace -- Or, CONNECT accts_admin_ace@hrpdb
Enter password: password
DROP USER mwaldron CASCADE;
```

2. CREATE TABLEコマンド・ルールを削除します。

「管理」ページに戻り、「コマンド・ルール」を選択します。「CREATE TABLE」コマンド・ルールを選択して、「削除」をクリックします。「確認」ウィンドウで、「はい」を選択します。

3. Internal DBA Standard Working Hoursルール・セットを削除します。

「管理」ページで「ルール・セット」を選択します。「ルール・セット」ページで、「内部DBA標準勤務時間」ルール・セットを選択し、「削除」を選択します。「確認」ウィンドウで「ルール・セットに関連付けられたルールの削除」チェック・ボックスを選択して、「はい」をクリックします。

4. 「内部DBA標準勤務時間」ルール・セットに関連付けられたルールを削除します。

「管理」ページで、「ルール」を選択します。「ルール」ページで、「内部DBA、内部ネットワークのみ、平日、平日勤務時間」ルールを選択して、「削除」を選択します。「確認」ウィンドウで「はい」を選択します。

5. DomainファクタからHIGHLY SECURE INTERNAL NETWORKおよびNOT SECUREファクタ・アイデンティティを削除します。

「管理」ページで「ファクタ」を選択します。「ドメイン」ファクタを選択し、「編集」を選択します。「アイデンティティ」ページになるまで、「次へ」をクリックします。「HIGHLY SECURE INTERNAL NETWORK」および「NOT SECURE」ファクタ・アイデンティティを選択し、「削除」をクリックしてそれぞれを削除します。(複数の項目を選択するには、[Ctrl]キーを押しながらクリックします。)「確認」ウィンドウで、「はい」を選択します。「完了」、「終了」の順にクリックします。

親トピック: チュートリアル: セッション・データに基づくユーザー・アクティビティの制限

7.9 ファクタ設計のガイドライン

Oracleでは、ファクタ設計のガイドラインを提供しています。

- セキュリティまたは外部システムからのセッションに関するその他のコンテキスト情報を統合するには、UTL_TCP、UTL_HTTP、DBMS_LDAPおよびDBMS_PIPEなどのOracleユーティリティ・パッケージを使用できます。
- ファクタの識別が「ファクタによる識別」に設定されている場合は、取得メソッドを指定しないでください。取得メソッドが必要なのは、ファクタを「メソッド」または「定数」に設定した場合のみです。
- ファクタに割当てルール・セットがある場合は、検証メソッドの使用を検討します。これにより、無効なアイデンティティが発行されないことを検証できます。
- 指定されている値は、クライアント・ソフトウェアが信頼されていて、クライアント・ソフトウェアからの通信チャネルが安全であることがわかっている場合にのみ信頼できるため、クライアント指定のProgram、OS Userおよびその他のファクタは注意して使用します。
- 時間ベースのファクタなど、同じセッション内のある起動と次の起動で、取得メソッドによって返される値が変わる可能性がある場合には、「アクセス」評価オプションのみを指定します。
- 従来のSQLおよびPL/SQLの最適化技術を使用して、ファクタ取得メソッドに使用されるファンクションの内部ロジックを最適化します。パフォーマンスと最適化の詳細は、Oracle Database SQLチューニング・ガイドを参照してください。
- 取得メソッドによって返される離散値がわかっている場合は、各値にアイデンティティを定義し、信頼レベルを割り当てられるようにします。ファクタに基づくアプリケーション・ロジックに信頼レベルを使用するにつれ、信頼レベルによりファクタに値が追加されます。
- 通常、より多くのファクタに基づくセキュリティ・ポリシーは、少ないファクタに基づくセキュリティ・ポリシーよりも強力です。別のファクタによって識別される新しいファクタを作成し、アイデンティティ・マップを使用してファクタの組合せを論理グループに保存できます。これにより、ファクタをOracle Label Securityラベルと統合する際の、親ファクタのラベル付けもより簡単になります。(詳細は、「Oracle Database VaultとOracle Label Securityの統合」を参照)。
- Oracle Label Securityを統合する際は、ファクタとラベル付けされているファクタよりも、自己とラベル付けされているファクタを構成してデバッグする方が簡単です。
- 1つ以上のセキュリティ、エンドユーザーまたは環境属性を関連付けられたデータベース・セッションで使用できるように、それらの属性を渡すデータベース・クライアント・アプリケーションを設計できます。これを行うには、属性ごとに1つのファクタを作成し、割当てルール・セットを使用してこれらの属性が割り当てられる場合(特定のWebアプリケーションを指定された名前付きアプリケーション・サーバー・コンピュータで使用するときのみ、など)を制御します。この方法で使用されるOracle Database Vaultファクタは、OracleプロシージャDBMS_SESSION.SET_IDENTIFIERに非常によく似ていますが、設定可能な場合を制御する機能も含まれています。DBMS_SESSION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。

親トピック: ファクタの構成

7.10 ファクタのパフォーマンスへの影響

ファクタの複雑さは、Oracleデータベース・インスタンスのパフォーマンスに影響します。

各ファクタには、検証メソッドや信頼レベルのような処理される要素があります。セッションによって評価されるDatabase_HostnameおよびProxy_Userのようなファクタの場合は、Oracle Database Vaultによりセッションの初期化中にこのプロセスが実行され、その値に対する後続のリクエスト用に結果がキャッシュされます。

「デフォルトのファクタ」に示されているデフォルトのファクタは、典型的なセキュリティ・ポリシーで使用される可能性が高いためキャッシュされます。ただし、ルール・セットやその他のコンポーネントなどで5つのファクタしか使用しない場合、別のことに使用できるリソースが残りのファクタにより消費されます。このような場合は、不要なファクタを削除する必要があります。(Oracle Database Vaultでは、これらのファクタを内部的に使用しないため、不要な場合は削除できます。)

ユーザー数が多い場合やアプリケーション・サーバーで接続の作成や切断を頻繁に行う場合、使用されるリソースがシステムのパフォーマンスに影響を与える可能性があります。不要なファクタは削除できます。

システム・パフォーマンスを確認するには、Oracle Enterprise Manager(Oracle Databaseと一緒にデフォルトでインストールされるOracle Enterprise Manager Cloud Controlを含む)、自動ワークロード・リポジトリ(AWR)およびTKPROFなどのツールを実行します。

関連項目:

- データベース・パフォーマンスの監視方法を学習するには、『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
- 個々のSQL文およびPL/SQL文の実行を監視するには、『Oracle Database SQLチューニング・ガイド』を参照してください

親トピック: ファクタの構成

7.11 ファクタに関連するレポートおよびデータ・ディクショナリ・ビュー

Oracle Database Vaultには、ファクタおよびそのアイデンティティに関する情報が表示されるレポートとデータ・ディクショナリ・ビューが用意されています。

表7-1では、Oracle Database Vaultレポートを示します。これらのレポートの実行方法の詳細は、「Oracle Database Vaultレポート」を参照してください。

表7-1 ファクタおよびアイデンティティに関連するレポート

レポート	説明
「ファクタの監査」レポート	評価に失敗したファクタの検出など、ファクタが監査されます。
「ファクタ構成の問題」レポート	無効なルール・セットまたは不完全のルール・セットなどの構成問題の表示、またはファクタに影響を与える可能性のある問題の監査が行われます。
「アイデンティティのないファクタ」レポート	アイデンティティが割り当てられていないファクタが表示されます。
「アイデンティティ構成の問題」レポート	無効なラベル・アイデンティティがあるファクタ、またはアイデンティティがマップされていないファクタが表示されます。
「ルール・セット構成の問題」レポート	ルールが定義されていないか、有効ではなく、それらを使用するファクタに影響を与える可能性があるルール・セットが表示されます。

表7-2に、既存のファクタおよびファクタ・アイデンティティに関する情報を提供するデータ・ディクショナリ・ビューを示します。

表7-2 ファクタおよびファクタ・アイデンティティに使用されるデータ・ディクショナリ・ビュー

データ・ディクショナリ・ビュー	説明
DBA_DV_FACTOR ビュー	現行のデータベース・インスタンス内の既存のファクタが表示されます。
DBA_DV_FACTOR_LINK ビュー	子ファクタの関連によりアイデンティティが決定される各ファクタの関係が表示されます。
DBA_DV_FACTOR_TYPE ビュー	システムで使用されているファクタ・タイプの名前および説明が表示されます。
DBA_DV_IDENTITY ビュー	各ファクタのアイデンティティが表示されます。
DBA_DV_IDENTITY_MAP ビュー	各ファクタのアイデンティティのマップが表示されます。

親トピック: ファクタの構成

8 Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

セキュア・アプリケーション・ロールを使用すると、ユーザーのアプリケーションに対するアクセス権限を制御できます。

- Oracle Database Vaultのセキュア・アプリケーション・ロールの概要
  Oracle Database Vaultでは、Oracle Database Vaultルール・セットで有効にするセキュア・アプリケーション・ロールを作成できます。
- Oracle Database Vaultセキュア・アプリケーション・ロールの作成
  Database Vault AdministratorでDatabase Vaultセキュア・アプリケーション・ロールを作成できます。
- Oracle Database Vaultで使用するためのOracle Databaseセキュア・アプリケーション・ロールの有効化
  既存のセキュア・アプリケーション・ロールを変更できるのは、それがOracle Database Vaultで作成された場合のみです。
- Oracle Database Vaultセキュア・アプリケーション・ロールのセキュリティ
  データベース管理権限を持つユーザーは、DROP ROLE文を使用して、Oracle Database Vaultセキュア・アプリケーション・ロールを削除する場合があります。
- Oracle Database Vaultセキュア・アプリケーション・ロールの削除
  Oracle Database Vault AdministratorでOracle Database Vaultセキュア・アプリケーション・ロールを削除できます。
- Oracle Database Vaultセキュア・アプリケーション・ロールの動作
  Oracle Database Vaultセキュア・アプリケーション・ロールのプロセス・フローは、セキュア・アプリケーション・ロールを作成した後に開始されます。
- チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与
  このチュートリアルでは、セキュア・アプリケーション・ロールを作成して、作業時間中にユーザーのOE.ORDERS表へのアクセスを制御する方法を示します。
- セキュア・アプリケーション・ロールのパフォーマンスへの影響
  Oracle Enterprise Manager Cloud Controlによってシステム・パフォーマンスを確認できます。
- セキュア・アプリケーション・ロールに関連するレポートおよびデータ・ディクショナリ・ビュー
  Oracle Database Vaultには、Oracle Database Vaultセキュア・アプリケーション・ロールの分析に使用できる、レポートとデータ・ディクショナリ・ビューが用意されています。

8.1 Oracle Database Vaultのセキュア・アプリケーション・ロールの概要

Oracle Database Vaultでは、Oracle Database Vaultルール・セットで有効にするセキュア・アプリケーション・ロールを作成できます。

通常のOracle Databaseセキュア・アプリケーション・ロールはカスタムPL/SQLプロシージャによって有効になります。セキュア・アプリケーション・ロールは、ユーザーによるアプリケーション外からのデータへのアクセスを阻止します。これにより、ユーザーはロールに付与されているアプリケーション権限のフレームワーク内での作業を強制されます。

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・ルートではなく、PDBでのみセキュア・アプリケーション・ロールを作成できます。

ルール・セットを基準にしてロールのデータベース・アクセスを行う利点は、データベース・セキュリティ・ポリシーをすべてのアプリケーションに保存するのではなく、中央の1つの場所に保存できることです。ルール・セットに基づいたロールにより、一貫性と柔軟性

を兼ね備えたメソッドを作成し、ロールが提供するセキュリティ・ポリシーを実行できます。この方法では、アプリケーション・ロールのセキュリティ・ポリシーを更新する必要がある場合、1つの場所、つまりルール・セットで行うことができます。さらに、ユーザーがどのようにしてデータベースに接続しても、ルール・セットがロールにバインドされているため、結果は同じです。ユーザーは、ロールを作成して、それをルール・セットに関連付けるのみです。関連付けられているルール・セットでは、ロールの有効化を試行するユーザーが検証されます。

関連トピック

- Oracle Database Vaultセキュア・アプリケーション・ロールのAPI

親トピック: Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

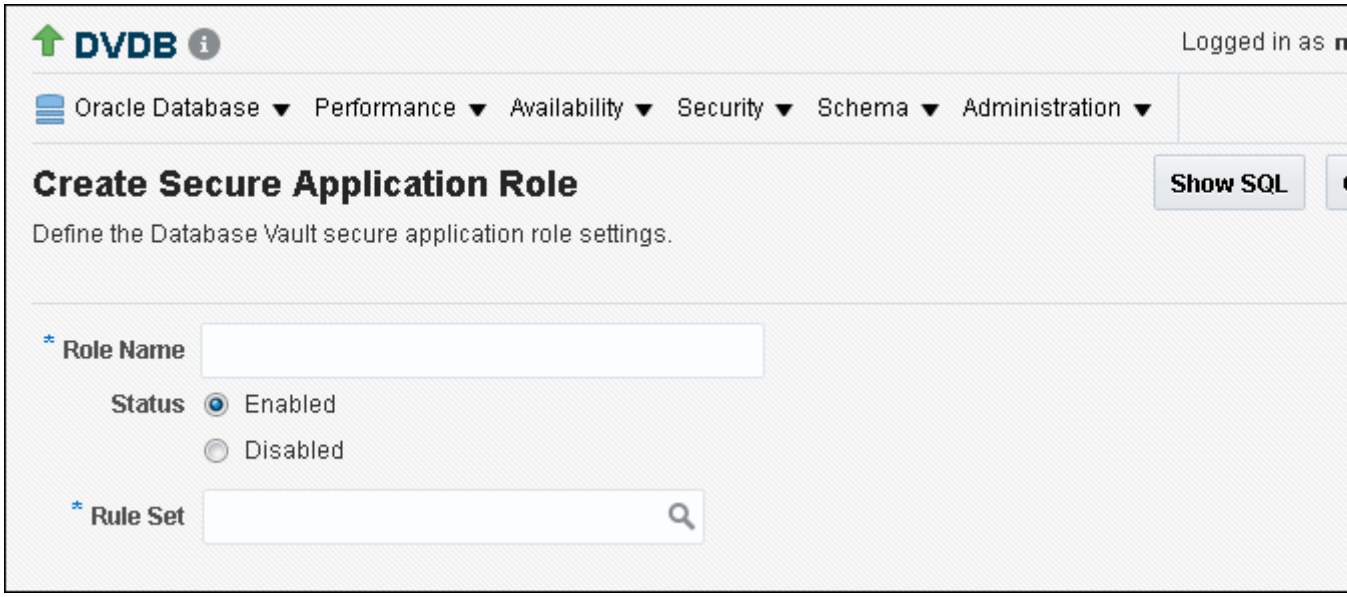
8.2 Oracle Database Vaultセキュア・アプリケーション・ロールの作成

Database Vault AdministratorでDatabase Vaultセキュア・アプリケーション・ロールを作成できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. ユーザーにロールの有効化を許可または禁止する条件を設定する1つ以上のルールが含まれるルール・セットを作成します。

   ルール・セットの基礎となるルールを作成する場合、ルールでロールの有効化を試行するユーザーを検証する必要があります。

3. 「管理」ページの「Database Vaultコンポーネント」で、「セキュア・アプリケーション・ロール」をクリックします。
4. 「セキュア・アプリケーション・ロール」ページで、「作成」を作成します。



5. 「セキュア・アプリケーション・ロールの作成」ページで、次の設定を入力します。
   - ロール名: 空白は使用せず、30文字以内の名前を入力します。この名前が、『Oracle Database SQL言語リファレンス』で説明されている、CREATE ROLE文を使用したロール作成の標準のOracleネーミング規則に準拠していることを確認してください。この属性は必須です。
   - ステータス: 「有効」または「無効」のいずれかを選択し、実行時にセキュア・アプリケーション・ロールを有効または無効にします。この属性は必須です。

- 有効: 使用できるようにロールを有効にします。すなわち、ユーザーはDBMS_MACSEC_ROLES.SET_ROLEファンクションを呼び出してロールの有効化を試行できます。ロールを有効化できるかどうかは、関連付けられているルール・セットの評価結果によって決まります。
- 無効: 使用できないようにロールを無効にします。DBMS_MACSEC_ROLES.SET_ROLEファンクションでロールを有効化できません。

- ルール・セット: リストから、セキュア・アプリケーション・ロールに関連付けるルール・セットを選択します。この属性は必須です。

  DBMS_MACSEC_ROLES.SET_ROLEを呼び出す際に、ルール・セットがTrueと評価されると、Oracle Database Vaultによりデータベース・セッションに対してそのロールが有効になります。ルール・セットがFalseと評価されると、ロールは有効になりません。

6. 「OK」をクリックします。

関連トピック

- ルール・セットの構成
- SET_ROLEプロシージャ
- 「他のデータベースへのOracle Database Vault構成の伝播」

親トピック: Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

8.3 Oracle Database Vaultで使用するためのOracle Databaseセキュア・アプリケーション・ロールの有効化

既存のセキュア・アプリケーション・ロールを変更できるのは、それがOracle Database Vaultで作成された場合のみです。

Oracle Database Vault以外を使用して作成されたセキュア・アプリケーション・ロールやデータベース・ロールは変更できません。ただし、Oracle Database Vault以外のロールをOracle Database Vaultで動作するようにすることはできます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとしてデータベースに接続します。
   たとえば:

   ```
   CONNECT sec_admin_owen
   Enter password: password
   ```

   マルチテナント環境でPDBを接続します。たとえば:

   ```
   CONNECT sec_admin_owen@pdb_name
   Enter password: password
   ```

   使用可能なPDBを確認するには、DBA_PDBSデータ・ディクショナリ・ビューのPDB_NAME列を問い合せます。現在のコンテナを確認するには、show con_nameコマンドを実行します。

2. Oracle Database Vaultで新しいセキュア・アプリケーション・ロールを作成して、そのセキュア・アプリケーション・ロールに既存のロールを付与します。
   たとえば:

   ```
   GRANT myExistingDBrole TO myDVrole;
   ```

3. この新しいロールに使用するコードを変更します。
   これを実行するには、アプリケーション・コードにDBMS_MACSEC_ROLES.SET_ROLEを使用します。

関連トピック

- SET_ROLEプロシージャ

親トピック: Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

8.4 Oracle Database Vaultセキュア・アプリケーション・ロールのセキュリティ

データベース管理権限を持つユーザーは、DROP ROLE文を使用して、Oracle Database Vaultセキュア・アプリケーション・ロールを削除する場合があります。

Oracle Database Vaultセキュア・アプリケーション・ロールが作成されると、必ずDatabase Vaultによりセキュア・アプリケーション・ロールがOracle Database Vaultレルムに追加されます。これにより、データベース管理者はDROP ROLE文を使用してセキュア・アプリケーション・ロールを削除できません。

親トピック: Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

8.5 Oracle Database Vaultセキュア・アプリケーション・ロールの削除

Oracle Database Vault AdministratorでOracle Database Vaultセキュア・アプリケーション・ロールを削除できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 必要に応じて、ロールに関連するOracle Database Vaultビューに問い合せることで、そのセキュア・アプリケーション・ロールへの様々な参照を特定します。
3. 削除するセキュア・アプリケーション・ロールを使用している可能性のあるアプリケーションを確認して変更します。
4. 「管理」ページの「Database Vaultコンポーネント」で、「セキュア・アプリケーション・ロール」をクリックします。
5. 「セキュア・アプリケーション・ロール」ページで、削除するロールを選択します。
6. 「削除」をクリックします。
7. 「確認」ウィンドウで、「はい」をクリックします。

関連トピック

- Oracle Database Vaultのデータ・ディクショナリ・ビュー

親トピック: Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

8.6 Oracle Database Vaultセキュア・アプリケーション・ロールの動作

Oracle Database Vaultセキュア・アプリケーション・ロールのプロセス・フローは、セキュア・アプリケーション・ロールを作成した後に開始されます。

1. Oracle Database Vault Administrator、またはDBMS_MACADMパッケージのセキュア・アプリケーション・ロール固有のファンクションを使用してロールを作成または更新します。

   詳細は、「DBMS_MACADMセキュア・アプリケーション・ロールのプロシージャ」を参照してください。

2. DBMS_MACSEC_ROLES.SET_ROLEファンクションを使用して、そのロールを呼び出すようにアプリケーションを変更します。

詳細は、「SET_ROLEプロシージャ」を参照してください。

3. Oracle Database Vaultにより、セキュア・アプリケーション・ロールに関連付けられているルール・セットが評価されます。

   ルール・セットがTrueと評価されると、ロールが現行のセッションに対して有効になります。ルール・セットがFalseと評価されると、ロールは有効になりません。どちらの場合も、Oracle Database Vaultにより、セキュア・アプリケーション・ロールに関連付けられているルール・セットの関連する監査およびカスタム・イベント・ハンドラが処理されます。

親トピック: Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

8.7 チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与

このチュートリアルでは、作業時間中にユーザーのOE.ORDERS表へのアクセスを制御するセキュア・アプリケーション・ロールを作成する方法を示します。

親トピック: Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

8.7.1 このチュートリアルについて

このチュートリアルでは、OEスキーマのORDERS表に対するSELECT文を特定のユーザーに制限します。

さらに、これらのユーザーは、リモート接続からではなく、オフィス内でのみOE.ORDERS表に対してこれらの文を実行できます。これを実行するには、Oracle Database Vaultセキュア・アプリケーション・ロールを作成し、これに関連付けたルール・セットによって実施されるチェックに合格した場合のみ、ユーザーにこのロールが有効化されるようにします。

親トピック: チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与

8.7.2 ステップ1: このチュートリアル用のユーザーの作成

最初に、チュートリアル用のユーザーを作成する必要があります。

1. DV_ACCTMGRロールを付与されているユーザーとしてSQL*Plusにログインします。

   たとえば:

   ```
   sqlplus accts_admin_ace
   Enter password: password
   ```

   マルチテナント環境で、適切なプラガブル・データベース(PDB)に接続する必要があります。

   たとえば:

   ```
   sqlplus accts_admin_ace@hrpdb
   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。

2. 次のユーザー・アカウントを作成します。

   ```
   GRANT CREATE SESSION TO eabel IDENTIFIED BY password;
   GRANT CREATE SESSION TO ahutton IDENTIFIED BY password;
   GRANT CREATE SESSION TO ldoran IDENTIFIED BY password;
   ```

   『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

親トピック: チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与

8.7.3 ステップ2: OEユーザー・アカウントの有効化

OEスキーマがこのチュートリアル用に使用されます。

1. DV_ACCTMGRユーザーとしてSQL*Plusに接続します。

   たとえば:

   ```
   CONNECT accts_admin_ace -- Or, CONNECT accts_admin_ace@hrpdb
   Enter password: password
   ```

2. OEアカウントのアカウント・ステータスを確認します。

   ```
   SELECT USERNAME, ACCOUNT_STATUS FROM DBA_USERS WHERE USERNAME = 'OE';
   ```

3. OEアカウントがロックされて無効になっている場合、ロックを解除し、新しいパスワードを割り当てます。

   ```
   ALTER USER OE ACCOUNT UNLOCK IDENTIFIED BY password;
   ```

親トピック: チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与

8.7.4 ステップ3: ルール・セットとそのルールの作成

ルール・セットおよびルールは、OE.ORDERS表の順序を変更できるユーザーを制限します。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページで、「ルール・セット」を選択します。

   「ルール・セット」ページが表示されます。

3. 「作成」をクリックします。

   「ルール・セットの作成」ページが表示されます。

4. 次の情報を入力します。
   - 名前: `Can Modify Orders`と入力します。
   - 説明: `Rule set to control who can modify orders in the OE.ORDERS table`と入力します。
   - ステータス: 「有効」を選択します。
   - 評価オプション: 「すべてTrue」を選択します。
5. 残りの設定はデフォルトのままにし、「次へ」をクリックして「ルールとの関連付け」ページに移動します。
6. 「ルールの作成」をクリックして、「ルールの作成」ダイアログ・ボックスで次の設定を入力します。
   - 名前: `Check IP Address`
   - 式: `DVF.F$CLIENT_IP = 'your_IP_address'`

   Check IP Addressルールでは、`your_IP_address`を自身のコンピュータのIPアドレスに置き換えます。実際には、アクセスを許可されるユーザーのすべてのIPアドレスを含む式を作成します。

   このルールでは、デフォルトのファクタClient_IPが使用されます。このファクタが削除されている場合、かわりに次のルール式を使用できます。

   ```
   UPPER(SYS_CONTEXT('USERENV','IP_ADDRESS')) = 'your_IP_address'
   ```

7. 「OK」をクリックします。
8. 「ルールの作成」を再度クリックして、「ルールの作成」ダイアログ・ボックスで次の設定を入力します。
   - 名前: `Check Session User`
   - 式: `DVF.F$SESSION_USER IN ('EABEL','AHUTTON')`

   このルールでは、デフォルトのファクタSession_Userが使用されます。このファクタが削除または変更されている場合、かわりに次のルール式を使用できます。

   ```
   UPPER(SYS_CONTEXT('USERENV','SESSION_USER')) IN ('EABEL','AHUTTON')
   ```

9. 「OK」をクリックします。
10. 「完了」、「終了」の順にクリックします。

親トピック: チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与

8.7.5 ステップ4: Database Vaultセキュア・アプリケーション・ロールの作成

Database Vaultセキュア・アプリケーション・ロールは、ルール・セット条件が満たされた場合に、設定されます。

1. Oracle Database Vaultで、「管理」ページに戻ります。
2. 「管理」で「セキュア・アプリケーション・ロール」を選択します。

   「セキュア・アプリケーション・ロール」ページが表示されます。

3. 「作成」をクリックします。

   「ロールの作成」ページが表示されます。

4. 「ロール」ボックスで、`ORDERS_MGMT`と入力し、ロールに名前を付けます。

5. 「ルール・セット」で、「Can Modify Orders」を選択します。
6. 「OK」をクリックします。

この段階で、Database Vaultセキュア・アプリケーション・ロールとそれに関連付けられたルール・セットが作成されますが、ロールにはまだ権限がありません。

親トピック: チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与

8.7.6 ステップ5: セキュア・アプリケーション・ロールへのSELECT権限の付与

セキュア・アプリケーション・ロールにはSELECT権限を付与する必要があります。

1. SQL*PlusでユーザーOEとして接続します。

```
CONNECT OE -- Or, CONNECT OE@hrpdb
Enter password: password
```

2. ORDERS_MGMT Database Vaultセキュア・アプリケーション・ロールにSELECT権限を付与します。

```
GRANT SELECT ON ORDERS TO ORDERS_MGMT;
```

親トピック: チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与

8.7.7 ステップ6: Database Vaultセキュア・アプリケーション・ロールのテスト

すべてのコンポーネントを準備したら、Database Vaultセキュア・アプリケーション・ロールをテストできます。

1. SQL*Plusで、ユーザーeabelとしてデータベースに直接接続します。

```
connect eabel@orcl
Enter password: password
```

   orclを使用するデータベース・インスタンスの名前に置き換えます。

2. ORDERS_MGMTロールを設定します。

```
EXEC DBMS_MACSEC_ROLES.SET_ROLE('ORDERS_MGMT');
```

   通常、このコールはユーザーがログインするアプリケーションに埋め込みます。

3. OE.ORDERS表から選択します。

```
SELECT COUNT(*) FROM OE.ORDERS;
```

   次の出力が表示されます。

```
  COUNT(*)
----------
       105
```

   ユーザーeabelは、正しいIPアドレスからデータベースに直接ログインし、有効セクション・ユーザーとしてリストにあるため、OE.ORDERS表から選択できます。ユーザーahuttonが同様にSQL*Plusにログインした場合も、OE.ORDERS表から選択できます。

4. データベース・インスタンスを指定せずに、ユーザーeabelとして再接続し、再びOE.ORDERS表からの選択を試行します。

```
CONNECT eabel
Enter password: password
```

```
EXEC DBMS_MACSEC_ROLES.SET_ROLE('ORDERS_MGMT');
```

次の出力が表示されます。

```
Error at line 1:
ORA-47305: Rule Set Violation on SET ROLE (Can Modfiy Orders)
...
```

続いて次のように入力します。

```
SELECT COUNT(*) FROM OE.ORDERS;
```

次の出力が表示されます。

```
ERROR at line 1:
ORA-00942: table or view does not exist
```

ユーザーeabelは、有効なユーザーであっても、ルール・セットのCheck IP Addressルールに違反しているため、ORDERS_MGMTロールを有効にできません。IPアドレスが認識される唯一の方法は、ユーザーeabelがステップ1で行ったように、データベース・インスタンスを指定して接続することです。(これがどのように機能するかは、「ファクタの構成」の「ステップ3: Domainファクタ・アイデンティティのClient_IPファクタへのマップ」のステップ9を参照)。

5. ユーザーldoranとして接続します。

```
CONNECT ldoran -- Or, CONNECT ldoran@hrpdb
Enter password: password
```

6. 次の文を入力します。

```
EXEC DBMS_MACSEC_ROLES.SET_ROLE('ORDERS_MGMT');
SELECT COUNT(*) FROM OE.ORDERS;
```

ユーザーldoranは有効なユーザーではないため、ORDERS_MGMTロールを有効にすることはできません。したがって、OE.ORDERS表から選択できません。

親トピック: チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与

8.7.8 ステップ7: この例で使用したコンポーネントの削除

コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. ORDERS_MGMTセキュア・アプリケーション・ロールを削除します。「セキュア・アプリケーション・ロール」ページでORDERS_MGMTセキュア・アプリケーション・ロールを選択して「削除」をクリックし、「確認」ダイアログ・ボックスで「はい」をクリックします。
3. 「ルール・セット」ページで、「Can Modify Orders」ルール・セットを選択して、「削除」をクリックします。
4. 「確認」ダイアログ・ボックスで、「はい」を選択してルール・セットを削除します。
5. 「ルール」ページで、「Check IP Address」ルールと「Check Session User」ルールを選択して、「削除」を選択します。「確認」ボックスで「はい」を選択します。

   複数のルールを選択するには、[Ctrl]キーを押しながらクリックします。

6. Database VaultアカウントマネージャとしてSQL*Plusに接続し、ユーザーを削除します。

たとえば:

```
CONNECT bea_dvacctmgr -- Or, CONNECT bea_dvacctmgr@hrpdb
Enter password: password
DROP USER eabel;
DROP USER ahutton;
DROP USER ldoran;
```

7. 必要がなければ、OEユーザー・アカウントをロックし、無効にします。

```
ALTER USER OE ACCOUNT LOCK PASSWORD EXPIRE;
```

親トピック: チュートリアル: Database Vaultセキュア・アプリケーション・ロールによるアクセス権限の付与

8.8 セキュア・アプリケーション・ロールのパフォーマンスへの影響

Oracle Enterprise Manager Cloud Controlによってシステム・パフォーマンスを確認できます。

他に使用できるツールとして、自動ワークロード・リポジトリ(AWR)およびTKPROFがあります。

関連項目:

- データベース・パフォーマンスの監視方法を学習するには、『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
- 個々のSQL文およびPL/SQL文の実行を監視するには、『Oracle Database SQLチューニング・ガイド』を参照してください

親トピック: Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

8.9 セキュア・アプリケーション・ロールに関連するレポートおよびデータ・ディクショナリ・ビュー

Oracle Database Vaultには、Oracle Database Vaultセキュア・アプリケーション・ロールの分析に使用できる、レポートとデータ・ディクショナリ・ビューが用意されています。

表8-1に、Oracle Database Vaultレポートを示します。これらのレポートの実行方法の詳細は、「Oracle Database Vaultレポート」を参照してください。

表8-1 セキュア・アプリケーション・ロールに関連するレポート

レポート	説明
「セキュア・アプリケーション・ロールの監査」レポート	Oracle Database Vault セキュア・アプリケーション・ロールを有効にする操作によって生成された監査レコードが表示されます。 このタイプの監査レコードを生成するには、ロールに関連付けられているルール・セットの監査を有効にします。

レポート	説明
「セキュア・アプリケーション構成の問題」レポート	存在しないデータベース・ロールや、不完全または無効なルール・セットのあるセキュア・アプリケーション・ロールが表示されます。
「ルール・セット構成の問題」レポート	ルールが定義されていないか、有効ではなく、それらを使用するセキュア・アプリケーション・ロールに影響を与える可能性のあるルール・セットが表示されます。
強力なデータベース・アカウントおよびロールのレポート	強力な権限のあるデータベース・アカウントおよびロールに関する情報が示されます。

DBA_DV_ROLEデータ・ディクショナリ・ビューを使用すれば、権限管理で使用されるOracle Database Vaultセキュア・アプリケーション・ロールを検索できます。詳細は、「DBA_DV_ROLEビュー」を参照してください。

親トピック: Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

9 Oracle Database Vaultポリシーの構成

Oracle Database Vaultポリシーを使用して、よく使用されるレルムおよびコマンド・ルール設定を実装できます。

- Database Vaultポリシーの概要
  Oracle Database Vaultポリシーにより、ローカルのレルムおよびコマンド・ルールを、必要に応じて有効または無効にできる、名前付きポリシーにグループ化できます。
- デフォルトのOracle Database Vaultポリシー
  Oracle Database Vaultには、ユーザー・アカウントとシステム権限をよりしっかりと保護するために使用できる、2つのデフォルト・ポリシーが用意されています。
- Oracle Databaseポリシーの作成
  通常、Oracle Database Vaultポリシーを作成するには、そのポリシーを取り巻くレルムおよびコマンド・ルールを指定する、コンテナ・ポリシーを作成します。
- Oracle Database Vaultポリシーの変更
  Enterprise Manager Cloud Controlを使用してOracle Database Vaultポリシーを変更できます。
- Oracle Database Vaultポリシーの削除
  Enterprise Manager Cloud Controlを使用してOracle Database Vaultポリシーを削除できます。
- 関連するデータ・ディクショナリ・ビュー
  Oracle Database Vaultには、Database Vaultポリシーの分析に便利なデータ・ディクショナリ・ビューが用意されています。

9.1 Database Vaultポリシーの概要

Oracle Database Vaultポリシーにより、ローカルのレルムおよびコマンド・ルールが、必要に応じて有効または無効にできる、名前付きポリシーにグループ化されます。

- Oracle Database Vaultポリシーについて
  Oracle Database Vaultポリシーを使用してレルムおよびコマンド・ルールの定義を1つのポリシーにグループ化でき、その後、まとめて有効または無効にできます。
- 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。

親トピック: Oracle Database Vaultポリシーの構成

9.1.1 Oracle Database Vaultポリシーについて

Oracle Database Vaultポリシーを使用してレルムおよびコマンド・ルールの定義を1つのポリシーにグループ化でき、その後、まとめて有効または無効にできます。

Database Vaultポリシーにより、DVADMロールおよびDVOWNERロールで提供される強力な権限を付与することなく、限定されたレルム管理権限をデータベース・ユーザーに委任できます。Oracle Database Vaultには、デフォルト・ポリシーが用意されています。

たとえば、レルムおよびいくつかのコマンド・ルールなど、特定のアプリケーションに関連する一連のOracle Database Vaultオブジェクトがあるとします。Database Vaultポリシーを使用して、これらのオブジェクトを1つのポリシーにグループ化できます。その後、このアプリケーションのためとポリシーの有効化または無効化のためにレルムへのユーザーの追加を管理するポリシー管理者

を指定できます。プライマリ・アプリケーションが1つのみの場合は、含まれているDatabase Vaultオブジェクトごとにコマンドを発行するのではなく、ユーザーがすべての関連オブジェクトを1つのコマンドで有効化、無効化またはシミュレート(シミュレーション・モードを使用)できるという、管理容易性のためにそれを使用できます。

ポリシーのポリシー状態の設定内容に応じて個々のレルムおよびコマンド・ルールの有効化がどのように機能するかを、次に示します。

- 完全有効化モード(DBMS_MACADM.G_ENABLED)では、関連付けられたレルムおよびコマンド・ルールの個々の有効化設定より優先されるよう、ポリシーが設定されます。たとえば、ポリシーの関連オブジェクトが個別に無効にされている場合、それらは、ポリシーが有効になると有効になります。(反対に、埋込みのセキュリティ・オブジェクトでそれら固有の有効化、無効化またはシミュレーション・モードを設定できるよう、DBMS_MACADM.G_PARTIALを設定できます)。
- 部分有効化モード(DBMS_MACADM.G_PARTIAL)では、関連付けられたレルムおよびコマンド・ルールを様々なステータス設定(ENABLED、DISABLEDおよびSIMULATION)にできます。他のポリシー・ステータスを選択すると、関連するすべての制御が、強制的に、ポリシーによって決定された同じステータスになります。ポリシー・ステータスを部分にすると、各レルムおよびコマンド・ルールで、必要に応じてステータスを変更できます。
- シミュレーション・モード(DBMS_MACACM.G.SIMULATION)では、ポリシーが有効になりますが、レルムまたはコマンド・ルールに対する違反が、ユーザー名や使用されたSQL文など、違反のタイプに関する情報とともに、指定したログ表に書き込まれます。シミュレーションにより、ポリシー内のすべてのセキュリティ・オブジェクトが強制的にシミュレーション・モードになります。
- 無効化モード(DBMS_MACADM.G_DISABLED)では、ポリシーはその作成後に無効になります。

通常、Database Vaultポリシーを作成するには、次のステップを実行します。

1. ポリシー内で使用する、必要なレルムおよびコマンド・ルールを作成します。
2. Database Vaultポリシーを作成します。

   DBMS_MACADM.CREATE_POLICYプロシージャを使用してポリシーを作成できます。
3. 1つ以上のレルムをポリシーに追加します。

   DBMS_MACADM.ADD_REALM_TO_POLICYプロシージャを使用してレルムをポリシーに追加できます。
4. 1つ以上のコマンド・ルールをポリシーに追加します。

   DBMS_MACADM.ADD_CMD_TO_POLICYプロシージャを使用してコマンド・ルールをポリシーに追加できます。
5. 1人以上のデータベース・ユーザーをポリシーの所有者として追加します。

   DBMS_MACADM.ADD_OWNER_TO_POLICYプロシージャを使用してユーザーをポリシーに追加できます。その後、このユーザーにDV_POLICY_OWNERロールを付与します。このユーザーは、ポリシー状態の変更、レルムでの認可の追加または削除、および一連のDVSYS.POLICY_OWNER*データ・ディクショナリ・ビューに対するSELECT権限の保持といった、限定された一連の作業を実行できます。デフォルトでは、DVOWNERユーザーがポリシーを所有します。

ポリシーは、作成後すぐに使用できます。

この項では、Oracle Enterprise Manager Cloud ControlでOracle Database Vault Administratorページを使用してポリシーを構成する方法について説明します。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ポリシーを構成するには、DBMS_MACADM PL/SQLパッケージを使用する必要があります。

関連トピック

- デフォルトのOracle Database Vaultポリシー

- Oracle Database VaultポリシーのAPI
- DV_POLICY_OWNER Database Vault所有者ロール

親トピック: Database Vaultポリシーの概要

9.1.2 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。

つまり、PDBでポリシーを作成した場合は、ローカルのレルムおよびコマンド・ルールのみをそれに追加できます。共通レルム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: Database Vaultポリシーの概要

9.2 デフォルトのOracle Database Vaultポリシー

Oracle Database Vaultには、ユーザー・アカウントとシステム権限をよりしっかりと保護するために使用できる、2つのデフォルト・ポリシーが用意されています。

独自のセキュリティ構成でデフォルトのポリシーを使用できます。それらは、Oracle Database Vaultによる内部使用には必要ないため、不要な場合は削除できます。

デフォルト・ポリシーは、次のとおりです。

- Oracleアカウント管理コントロールは、Oracle Database Vault内のユーザー関連操作を強制的に制御します。権限のない特権ユーザーによるその場かぎりのユーザー・アカウント作成、ユーザー削除、およびその他のユーザー・アカウント関連操作を防ぐために使用されます。それには、CREATE USERなどのSQL文のためのDatabase Vaultアカウント管理レルムおよびユーザー・アカウント管理コマンド・ルールが含まれています。
- Oracleシステム保護コントロールは、デフォルトのOracle Database環境に関連付けられている、重要なデータベース・スキーマ、権限およびロールを強制的に制御します。それには、システム管理SQL文ALTER SYSTEMのための、Oracleデフォルト・スキーマ保護レルムなどのレルム、およびコマンド・ルールが含まれています。

関連トピック

- DBA_DV_POLICY_OBJECTビュー

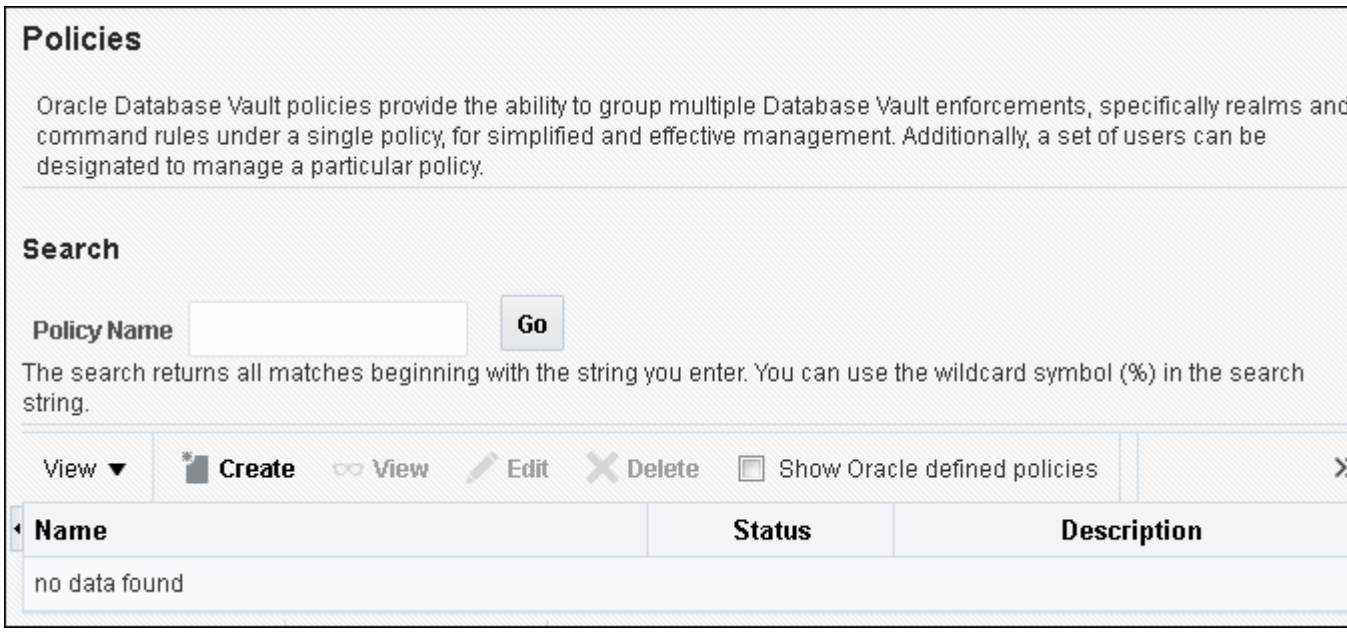
親トピック: Oracle Database Vaultポリシーの構成

9.3 Oracle Databaseポリシーの作成

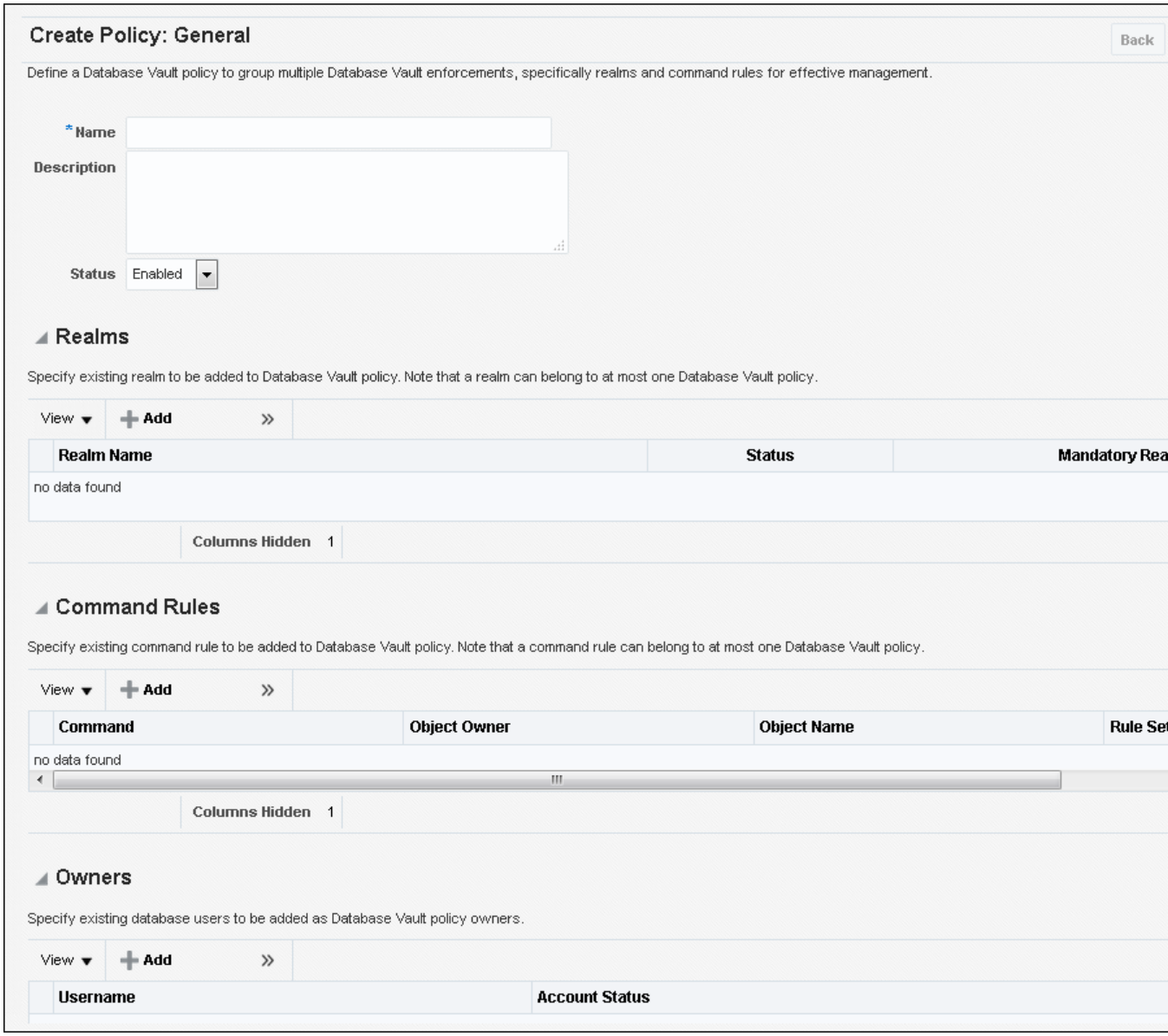
Oracle Database Vaultポリシーを作成するには、そのポリシーを取り巻くレルムおよびコマンド・ルールを指定する、コンテナ・ポリシーを作成します。

ポリシーは作成時に有効にするか、後で有効にできます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「レルムの作成」および「ルール・セットの作成」を使用して、ポリシーに関連付ける必要があるレルムおよびコマンド・ルールを作成します。
3. 「管理」ページの「Database Vaultコンポーネント」の下で、「ポリシー」をクリックして「ポリシー」ページを表示します。



4. 「ポリシー」ページで、「作成」をクリックして「ポリシーの作成」ページを表示します。



5. 「ポリシーの作成」ページの「一般」で、次の設定を入力します。
   - 名前：ポリシー名を最大128文字で入力します。

- 説明: ポリシーの説明を最大4000文字で入力します。
- ステータス: 次の中から選択します。
  - 「有効」では、ポリシーがその作成後に有効になります。
  - 「無効」では、ポリシーがその作成後に無効になります。
  - 「シミュレーション」では、ポリシーがシミュレーション・モードに設定されます。シミュレーション・モードでは、ポリシー内で使用されるレルムまたはコマンド・ルールに対する違反が、ユーザー名や使用されたSQL文などエラーを説明する十分な情報とともに、指定されたログ表に記録されます。
  - 「部分」では、ポリシーに関連付けられているレルムまたはコマンド・ルールの強制状態を個別に変更できます。

6. 「レルム」の下で「追加」をクリックし、レルムを選択してポリシーに追加します。次に、「OK」をクリックします。
7. 「コマンド・ルール」の下で「追加」をクリックし、コマンド・ルールを選択してポリシーに追加します。次に、「OK」をクリックします。
8. 「所有者」の下で「追加」をクリックし、所有者をポリシーに追加します。次に、「OK」をクリックします。
9. 「次へ」をクリックします。
10. 確認ページで、「終了」をクリックします。
11. Database Vaultポリシー所有者にポリシーに関連するビューの問合せと許可されたプロシージャの実行が可能になるよう、このユーザーにDV_POLICY_OWNERロールを付与します。
    たとえば:

```
GRANT DV_POLICY_OWNER TO psmith;
```

親トピック: Oracle Database Vaultポリシーの構成

9.4 Oracle Database Vaultポリシーの変更

Enterprise Manager Cloud Controlを使用してOracle Database Vaultポリシーを変更できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「管理」ページの「Database Vaultコンポーネント」で、「ポリシー」をクリックします。
3. 変更するポリシーの行を選択します。
4. 「編集」をクリックします。
5. 「ポリシーの編集」ページで、必要に応じて設定を変更します。
6. 「次へ」をクリックして、「完了」をクリックします。

親トピック: Oracle Database Vaultポリシーの構成

9.5 Oracle Database Vaultポリシーの削除

Enterprise Manager Cloud Controlを使用してOracle Database Vaultポリシーを削除できます。

Oracle Database Vaultポリシーを削除する場合、基礎となるレルムおよびコマンド・ルールは保持され、それら個別の有効化ステータスが維持されます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、

Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。

2. 「管理」ページの「Database Vaultコンポーネント」で、「ポリシー」をクリックします。
3. 削除するポリシーの行を選択し、「削除」をクリックしてから、確認ダイアログ・ボックスで「はい」をクリックします。

親トピック: Oracle Database Vaultポリシーの構成

9.6 関連するデータ・ディクショナリ・ビュー

Oracle Database Vaultには、Database Vaultポリシーの分析に便利なデータ・ディクショナリ・ビューが用意されています。

表9-1に、既存のOracle Database Vaultポリシーについて情報を提供するデータ・ディクショナリ・ビューを示します。

表9-1 Oracle Database Vaultポリシーのために使用されるデータ・ディクショナリ・ビュー

データ・ディクショナリ・ビュー	説明
DBA_DV_POLICY ビュー	Database Vault ポリシー、説明およびそれらの状態を示します。
DBA_DV_POLICY_OBJECT ビュー	関連付けられているレルムおよびコマンド・ルールなど、ポリシーに関する詳細情報を提供します。
DBA_DV_POLICY_OWNER ビュー	Database Vault ポリシーの所有者を示します。
DBA_DV_REALM_AUTH ビュー	DV_POLICY_OWNER ロールを付与されたユーザーが、レルム名、権限受領者および関連付けられたルール・セットなど、Database Vault ポリシーに関連付けられているレルムに付与された認可について情報を確認できます。
DVSYS.POLICY_OWNER_COMMAND_RULE ビュー	DV_POLICY_OWNER ロールを付与されたユーザーが、コマンド・ルール名など、Database Vault ポリシーに関連付けられているコマンド・ルールについて情報を確認できます。
DVSYS.POLICY_OWNER_POLICY ビュー	DV_POLICY_OWNER ロールを付与されたユーザーが、他のポリシー所有者によって作成されたポリシーを含め、現在のデータベース・インスタンス内の既存のポリシーの名前、説明および状態などの情報を確認できます。
DVSYS.POLICY_OWNER_POLICY ビュー	DV_POLICY_OWNER ロールを付与されたユーザーが、他のポリシー所有者によって作成されたポリシーを含め、現在のデータベース・インスタンス内の既存のポリシーの名前、説明および状態などの情報を確認できます。

データ・ディクショナリ・ビュー	説明
DVSYS.POLICY_OWNER_REALM ビュー	DV_POLICY_OWNER ロールを付与されたユーザーが、レルム名、監査オプションまたはタイプなど、Database Vault ポリシーに関連付けられているレルムについて情報を確認できます。
DVSYS.POLICY_OWNER_REALM_OBJECT ビュー	DV_POLICY_OWNER ロールを付与されたユーザーが、レルム名、権限受領者および関連付けられたルール・セットなど、Database Vault ポリシーに関連付けられているレルムに追加されたオブジェクトについて情報を確認できます。
DVSYS.POLICY_OWNER_RULE ビュー	DV_POLICY_OWNER ロールを付与されたユーザーが、ルール名とその式など、Database Vault ポリシー内のルール・セットに関連付けられているルールについて情報を確認できます。
DVSYS.POLICY_OWNER_RULE_SET ビュー	DV_POLICY_OWNER ロールを付与されたユーザーが、ルール・セットの名前、そのハンドラ情報、およびそれが有効になっているかどうかなど、Database Vault ポリシーに関連付けられているルール・セットについて情報を確認できます。
DVSYS.POLICY_OWNER_RULE_SET_RULE ビュー	DV_POLICY_OWNER ロールを付与されたユーザーが、ルール・セットの名前、およびそれが有効になっているかどうかなど、Database Vault ポリシーで使用されるルールを含むルール・セットについて情報を確認できます。

親トピック: Oracle Database Vaultポリシーの構成

10 レルムおよびコマンド・ルール・アクティビティのログ記録のためのシミュレーション・モードの使用

シミュレーション・モードでは、新規および変更されたOracle Database Vaultの制御を迅速にテストするために、SQLの実行を防止するかわりにシミュレーション・ログに違反を書き込みます。

- シミュレーション・モードについて
  シミュレーション・モードを使用すると、Oracle Database Vaultのレルムおよびコマンド・ルールによってSQL実行をブロックするかわりに、シミュレーション・ログに違反を記録できます。
- シミュレーション・モードの使用例
  シミュレーション・モードは、新しいレルムおよびコマンド・ルールの開発構成をテストするために役立ちます。
- シミュレーション・モードでのレルムのログ記録
  通常のレルムおよび必須レルムの両方をシミュレーション・モードに設定できます。
- チュートリアル: シミュレーション・モードの使用によるレルムに対する違反の追跡
  このチュートリアルでは、シミュレーション・モードを使用するレルムを作成してから、レルムに対する違反をテストする方法を示します。

10.1 シミュレーション・モードについて

シミュレーション・モードでは、Oracle Database Vaultのレルムおよびコマンド・ルールによってSQL実行をブロックするかわりに、シミュレーション・ログに違反を記録できます。

シミュレーション・モードでは、簡単に分析できるよう、1つの場所で取得されたエラーが格納されます。シミュレーション・モードを使用するには、レルムまたはコマンド・ルールの作成または更新時に、レルムまたはコマンド・ルールを有効または無効にするのではなく、それをシミュレーション・モードに設定します。レルムまたはコマンド・ルールはまだ有効になっていますが、違反はブロックされず、かわりにシミュレーション・ログ・ファイルに記録されるため、本番環境に対して有効にする前に、潜在的なエラーを見つけるためにテストできます。シミュレーション・モードを有効にすると、レポートに複数のレルムまたはコマンド・ルールに対する違反が含められることがあります。ユーザーのSQL文のソースをより正確に識別できる詳細なレポートが必要な場合は、PL/SQLコール・スタックを含めるようにシミュレーション・モードを構成します。このコール・スタックは、Database Vault監査レコードのより優れたトラブルシューティングのために、プロシージャおよびファンクションのコールを再帰的にキャプチャします。コール・スタック情報は、DVSYS.DBA_DV_SIMULATION_LOGデータ・ディクショナリ・ビューのPL_SQL_STACK列に格納されます。

たとえば、次のレルム作成文では、シミュレーション・モードが有効になり、PL/SQLコース・スタックが生成されます。

```
BEGIN
 DBMS_MACADM.CREATE_REALM(
 realm_name    => 'HR Apps',
 description   => 'Realm to protect the HR realm',
 enabled       => DBMS_MACUTL.G_SIMULATION,
 audit_options => DBMS_MACUTL.G_REALM_AUDIT_FAIL,
 realm_type    => 1,
 realm_scope   => DBMS_MACUTL.G_SCOPE_LOCAL,
 pl_sql_stack  => TRUE);
END;
/
```

この時点では、レルムまたはコマンド・ルールに違反するSQL文はまだ実行可能ですが、これらのアクティビティはDBA_DV_SIMULATION_LOGデータ・ディクショナリ・ビューに記録されます。たとえば、次の問合せは、HR Appsレルムおよびシミュレーション・モードに設定されているその他すべてのレルムまたはコマンド・ルールに対する違反を検出します。

```
SELECT USERNAME, COMMAND, SQLTEXT, VIOLATION_TYPE
FROM DBA_DV_SIMULATION_LOG, TABLE(DBA_DV_SIMULATION_LOG.REALM_NAME) RN
WHERE RN.COLUMN_VALUE = "HR APPS";
USERNAME  COMMAND    SQLTEXT                          VIOLATION_TYPE

--------  ---------- -------------------------------  --------------
DGRANT    SELECT     SELECT SALARY FROM HR.EMPLOYEES; Realm Violation
```

レルムまたはコマンド・ルールのテストの完了後、DV_ADMINまたはDV_OWNERロールを付与されているユーザーは、このビューDVSYS.SIMULATION_LOG$の基礎となる表の内容を削除することで、.DBA_DV_SIMULATION_LOGデータ・ディクショナリ・ビューをクリアできます。

たとえば:

```
DELETE FROM DVSYS.SIMULATION_LOG$;
```

または

```
DELETE FROM DVSYS.SIMULATION_LOG$ WHERE COMMAND = 'SELECT';
```

親トピック: レルムおよびコマンド・ルール・アクティビティのログ記録のためのシミュレーション・モードの使用

10.2 シミュレーション・モードの使用例

シミュレーション・モードは、新しいレルムおよびコマンド・ルールの開発構成をテストするために役立ちます。

使用例を次に示します。

- アプリケーション認証

  アプリケーションを認証している場合、アプリケーション・テスト環境で、次のようにシミュレーション・モードを使用できます。

  1. アプリケーションのすべてのスキーマを、シミュレーション・モードが有効になっている必須レルムに配置します。
  2. フル・リグレッション・テストを実行します。
  3. DBA_DV_SIMULATION_LOGデータ・ディクショナリ・ビューを問い合せてこれらのスキーマにアクセスできるユーザーを確認することで、シミュレーション・モード・ログを分析します。
  4. レルムを新しい認可で更新し、レルムを有効にします(つまり、シミュレーション・モードは使用しない)。
  5. リグレッション・テストを再実行します。

- 新しいコマンド・ルールの導入

  Oracle Database Vaultが有効になっている本番データベースでシミュレーション・モードを使用できます。

  1. 新しいコマンド・ルールを、必要な数週間の間、シミュレーション・モードで本番に配置します。
  2. DBA_DV_SIMULATION_LOGを問い合せて、コマンド・ルールが正しく動作しているかどうかを判断することで、シミュレーション・モード・ログを分析します。
  3. 必要に応じて、コマンド・ルールに変更を加えます。
  4. コマンド・ルールを有効にします。

- シミュレーション・モードでの本番データベースへの新しいレルムの配置。

  この方法は、ルール・セット内の信頼できるパスのルールを設定し、レルムの認可ユーザーを見つけるために必要な、システム・コンテキスト情報を確認するために役立ちます。

1. レルムを必須モードで作成し、保護されたオブジェクトを追加します。
2. 認可ユーザーは追加しないでください。
3. 使用される通常のIPアドレスからアプリケーションおよび開発操作を実行します。
4. 両方の認可ユーザーのシミュレーション・ログ・ファイル、および信頼できるパスの作成に使用できるシステム・コンテキスト情報を確認します。
5. 信頼できるパスを作成してから、認可ユーザーを追加します。
6. シミュレーション・ログをクリアし、アプリケーションおよび開発操作タスクを再度実行します。
7. 一定期間の経過後、シミュレーション・ログを確認します。すべての制御が正しく更新された場合、シミュレーション・ログは空になっています。シミュレーション・モードでのログ・エントリには、レルムおよびルール・セットに加える必要がある追加変更が示されます。または、ログ・エントリに、悪意ある使用が示される場合があります。

親トピック: レルムおよびコマンド・ルール・アクティビティのログ記録のためのシミュレーション・モードの使用

10.3 シミュレーション・モードでのレルムのログ記録

通常のレルムおよび必須レルムの両方をシミュレーション・モードに設定できます。

- シミュレーション・モードでレルムのログを記録する場合の考慮事項
  シミュレーション・モードでレルムを使用する場合、考慮する必要があるいくつかのユースケースがあります。
- ユースケース: すべての新規レルムがシミュレーション・モード
  このユースケースでは、すべてのレルムが必須または通常のいずれかで、ユーザーが作成するレルムはすべてシミュレーション・モードになります。
- ユースケース: 既存レルムへの新規レルムの導入
  このユースケースでは、既存のレルムを持つデータベースに新規レルムを追加します。
- ユースケース: レルムへの新規オブジェクトの追加のテスト
  このユースケースでは、既存のレルムに新規オブジェクトを追加してから、現在のレルム保護を削除せずにシミュレーション・モードを使用してこれをテストします。
- ユースケース: レルムからのオブジェクトの削除のテスト
  このユースケースでは、既存レルムからのオブジェクトの削除をテストします。
- ユースケース: 認可されたユーザーのレルムへの追加のテスト
  このユースケースでは、ユーザーを追加することでセキュリティ制御を緩和します。単に認可されたユーザーを追加する場合は、何もシミュレートする必要はありません。
- ユースケース: 認可されたユーザーのレルムからの削除のテスト
  このユースケースでは、認可されたユーザーを削除し、シミュレーション・モードを使用してそのユーザーがまだレルムにアクセスする必要があるかどうかを確認します。
- ユースケース: レルムを使用した新規ファクタのテスト
  このユースケースでは、ファクタに対する変更をテストします。
- ユースケース: 既存のコマンド・ルールへの変更のテスト
  このユースケースでは、既存のコマンド・ルールに対する変更を、元のコマンド・ルールを有効にしたままテストします。

親トピック: レルムおよびコマンド・ルール・アクティビティのログ記録のためのシミュレーション・モードの使用

10.3.1 シミュレーション・モードでレルムのログを記録する場合の考慮事項

シミュレーション・モードでレルムを使用する場合、考慮する必要があるいくつかのユースケースがあります。

- すべての新規Database Vault制御によるアプリケーションのテスト: すべてのレルムがシミュレーション・モード
- 既存の使用中のDatabase Vault制御に対するレルムの追加: レルムのサブセットのみがシミュレーション・モード
- 有効化されている既存レルムに新規オブジェクトを追加し、既存の制御を無効にしないでシミュレーション・モードを使用して差異をテスト
- 有効化されている既存レルムから1つ以上の既存オブジェクトを削除し、既存の制御を無効にしないでシミュレーション・モードを使用して差異をテスト
- 有効化されている既存レルムに新しい認可ユーザーを追加し、既存の制御を無効にしないでシミュレーション・モードを使用して差異をテスト
- 有効化されている既存レルムから1つ以上の既存の認可ユーザーを削除し、既存の制御を無効にしないでシミュレーション・モードを使用して差異をテスト
- 有効化されている既存レルムでファクタを追加するか変更し、既存の制御を無効にしないでシミュレーション・モードを使用して差異をテスト
- 元のコマンド・ルールを有効にしたまま、コマンド・ルールへの変更を本番でテスト

ユーザーがSQL文を実行し、それが失敗した場合、失敗の原因は有効化されているレルム、シミュレーションされているレルムまたはそれらの両方のいずれかです。必須レルム、通常のレルムまたはその両方が存在する可能性があります。これらの条件によって、シミュレーション・ログに記録されるデータが決まります。

次の各項で説明するユースケースを作成し、通常と必須の両方のタイプのレルムがオブジェクトを保護している場合、通常のレルムは必須レルムによって完全に無力化されます。必須レルムと通常のレルムが同じオブジェクトを保護しているすべてのケースで、シミュレーション・ログに関して通常のレルムを無視できます。必須レルムの失敗のみがシミュレーション・ログに記録されます。通常のレルムの失敗がシミュレーション・ログに記録されるのは、オブジェクトのすべてのレルムが通常のレルムである場合のみとなります。さらに、通常のレルムがシミュレーション・ログに書き込まれるには、次のことに該当する必要があります。

- シミュレーション・モードの通常のレルムがすべて失敗している。かつ
- 有効化されている通常のレルムもすべて失敗している

有効化されたまたはシミュレーションの通常のレルムが1つ以上成功している場合は、シミュレーションの通常のレルムはいずれもログに記録されません。

親トピック: シミュレーション・モードでのレルムのログ記録

10.3.2 ユースケース: すべての新規レルムがシミュレーション・モード

このユースケースでは、すべてのレルムが必須または通常のいずれかで、ユーザーが作成するレルムはすべてシミュレーション・モードです。

次に例を示します。

- 必須レルムのみで、すべてがシミュレーション・モード
    - ユーザーは、すべての必須レルムでSQL文の実行を認可されます。シミュレーション・ログ表には何も記録されません。

- ユーザーは、1つ以上の必須レルム・チェックに失敗します。すべてのレルム・チェックの失敗が、シミュレーション・ログに記録されます。ユーザーのSQL文が成功した必須レルム・チェックはログに記録されません。

  この例では、3つの必須レルムがあります。ユーザーのSQL文は1つのレルムで成功し、他の2つでは失敗します。シミュレーション・ログには、失敗した2つのレルム・チェックのみが記録されます。

- 通常のレルムのみで、すべてがシミュレーション・モード
  - ユーザーは、少なくとも1つの通常のレルムでSQL文の実行を認可されます。ユーザーはデータへのアクセス権があるため、シミュレーション・ログには何も記録されません。
  - ユーザーは、すべての通常のレルムでSQL文の実行を認可されません。シミュレーション・ログには、レルム認可の失敗がすべて記録されます。これにより、ユーザーは、どのレルムでユーザーが認可される必要があるかを選択できます。SQLが機能するには、1つの通常のレルムで認可されることのみが必要となり、SQLを認可するために通常のレルムすべてを更新する必要はありません。
- 必須および通常のレルムが混在し、すべてがシミュレーション・モード
  - この場合、ユーザーが拒否されたときに主要レルムを取得します。必須および通常のレルムが混在している場合、必須レルムが主要レルムとなります。ユーザーがアクセス権を取得するには、すべての必須レルムが認可チェックに合格する必要があります。実際は、必須レルムがオブジェクトを保護している場合、通常のレルムは必要ないとみなすことができます。そのため、必須レルムと通常のレルムの両方が同じオブジェクトを保護している場合は、必須レルムのみが、SQL文がブロックされるか実行を許可されるかを制御します。ユーザーが通常のレルムに対して認可されているかどうかは関係ありません。この例では、最初のシナリオである、シミュレーション・モードの必須レルムのルールに従います。
  - ユーザーは、すべての必須レルムでSQL文の実行を認可されます。シミュレーション・ログ表には何も記録されません。ユーザーは1つ以上の通常のレルムで成功する場合も失敗する場合もありますが、通常のレルムの失敗に関しては何も記録されません。
  - ユーザーは、1つ以上の必須レルム・チェックに失敗します。すべてのレルム・チェックの失敗が、シミュレーション・ログに記録されます。ユーザーのSQL文が成功した必須レルム・チェックはログに記録されません。

    たとえば、必須レルムが3つあるとします。ユーザーのSQL文は1つのレルムで成功し、他の2つでは失敗します。シミュレーション・ログには、失敗した2つのレルム・チェックのみが記録されます。

    シミュレーション・ログに記録する必要があるのは必須レルムのみであるため、通常のレルムを記録する必要はありません。

親トピック: シミュレーション・モードでのレルムのログ記録

10.3.3 ユースケース: 既存レルムへの新規レルムの導入

このユースケースでは、既存のレルムを持つデータベースに新規レルムを追加します。

既存のレルムは有効化され、動作しています。新規レルムはシミュレーション・モードになっています。このユースケースは、シミュレーション・モードのレルムと有効化されているレルムの両方が同じオブジェクトを保護している場合にのみ適用されます。

例:

- シミュレーション・モードの新しい必須レルムと有効化されている既存の必須レルムがあります。このユースケースでは、オブジェクトに対する追加の必須レルムを示します。これにより、既存のオブジェクトのセキュリティが強化されます。
  - 有効化されている必須レルムとシミュレーション・モードの必須レルムがすべて、ユーザーのSQL文で成功: こ

の場合、SQLは正常に実行され、何も記録されません

- 有効化されている必須レルム(1つ以上)が失敗し、シミュレーション・モードの必須レルムはすべて成功: SQLがブロックされ、シミュレーション・ログには何も書き込まれません
- 有効化されている必須レルム(1つ以上)が失敗し、シミュレーション・モードの必須レルムの1つ以上が失敗: SQLがブロックされ、失敗したシミュレーション・モードの必須レルムがすべてシミュレーション・ログに書き込まれます
- 有効化されている必須レルムはすべて成功し、シミュレーション・モードの必須レルムの1つ以上が失敗: SQLはブロックされず、失敗したシミュレーション・モードの必須レルムがすべてシミュレーション・ログに書き込まれます

- シミュレーション・モードの新しい通常のレルムと、有効化されている既存の通常のレルムがある場合: 通常のレルムがセキュリティ・オブジェクトに追加され、ユーザーが機密データにアクセスするための新しい方法が提供されます
  - 有効化されている通常のレルム(1つ以上)とシミュレーション・モードの通常のレルム(1つ以上)が成功: ユーザーのSQLは正常に実行され、シミュレーション・ログには何も書き込まれません
  - 有効化されている通常のレルム(1つ以上)が成功し、シミュレーション・モードの通常のレルムはすべて失敗: ユーザーのSQLは正常に実行され、シミュレーション・ログには何も書き込まれません
  - 有効化されている通常のレルムがすべて失敗し、シミュレーション・モードの通常のレルムがすべて失敗: ユーザーのSQLがブロックされ、シミュレーション・モードの通常のレルムがすべてシミュレーション・ログに書き込まれます。必要に応じて、どの通常のレルムを認可するかをユーザーが評価する必要があります。現在の実装では、SQLがブロックされ、シミュレーション・モードの通常のレルムはシミュレーション・ログに追加されません。これは、有効化されている通常のレルムによって、いずれにしろSQLはブロックされるためです。このユースケースでは、ユーザーがSQLを認可するための新しいレルムを追加している可能性があるため、これを変更する必要があります。新しいSQLが、動作したと考えられるものの、シミュレーション・モードの通常のレルムすべてによってブロックされる場合(シミュレーション・モードの通常のレルムのいずれかが、SQLの動作を許可するように設定されている場合)、何が発生したかを確認する方法はありません。これは、この状況での監査ログへの入力と同様です。
  - 有効化されている通常のレルムがすべて失敗し、シミュレーション・モードの通常のレルム(1つ以上)が成功: ユーザーのSQLがブロックされ、シミュレーション・ログには書き込まれません。

- 新しい通常のレルムと、有効化されている既存の必須レルムがある場合: この状況では何も行う必要はありません。有効化されている必須レルムが引き続きオブジェクトを制御し、シミュレーション・モードの新しい通常のレルムは、有効化されているかどうかに関係なく、影響を及ぼしません。この場合、シミュレーション・ログは生成されません。

- シミュレーション・モードの新しい必須レルムと、有効化されている既存の通常のレルムがある場合: 現時点では有効化されている通常のレルムがオブジェクトを制御していますが、シミュレーション・モードの新しい必須レルムが有効化されると、それらの必須レルムがオブジェクトを完全に制御するようになり、有効化されている古い通常のレルムによる制御は失われます。そのため、シミュレーション・ログはすべての必須レルムについて作成されます。これは、新しい必須レルムと有効化されている既存の必須レルムがあるシナリオと同じです。

- シミュレーション・モードの新しい通常のレルムと、有効化されている既存の必須レルムおよび通常のレルムがある場合: 有効化されている必須レルムが、システム内に既存の有効化されている通常のレルムに、シミュレーション・モードの新しい通常のレルムを追加するかどうかを決定するレルムになります。これは、すべてがシミュレーション・モードの必須レルムと通常のレルムが混在しているシナリオと同じです。シミュレーション・ログには何も書き込まれません。

- シミュレーション・モードの新しい必須レルムと、有効化されている必須レルムおよび通常のレルムがある場合: 有効化

されている通常のレルムは無視できます。これは、新しい必須レルムと有効化されている既存の必須レルムがあるシナリオと同じです。

- シミュレーション・モードの新しい必須レルムおよび通常のレルムと、有効化されている既存の必須レルムおよび通常のレルムが混在している場合: 有効化されている必須レルムと通常のレルムはすべて無視できます。この場合は、単に、既存オブジェクトに必須レルムが追加されます。これは、新しい必須レルムと有効化されている既存の必須レルムがあるシナリオと同じです。

親トピック: シミュレーション・モードでのレルムのログ記録

10.3.4 ユースケース: レルムへの新規オブジェクトの追加のテスト

このユースケースでは、既存のレルムに新規オブジェクトを追加してから、現在のレルム保護を削除せずにシミュレーション・モードを使用してこれをテストします。

同じ認可されたユーザーおよびルール・セットを使用して、新規オブジェクトに対しシミュレーション・モードの重複レルムを作成することをお薦めします。こうすることで、新規オブジェクトのテスト中に、既存のレルムが既存オブジェクトへの保護を提供し続けることができます。

親トピック: シミュレーション・モードでのレルムのログ記録

10.3.5 ユースケース: レルムからのオブジェクトの削除のテスト

このユースケースでは、既存のレルムからのオブジェクトの削除をテストします。

既存オブジェクトに対するセキュリティ制御を削除するため、シミュレーション・モードを使用する必要はありません。単に、オブジェクトをレルムから削除してください。

親トピック: シミュレーション・モードでのレルムのログ記録

10.3.6 ユースケース: 認可されたユーザーのレルムへの追加のテスト

このユースケースでは、ユーザーを追加することでセキュリティ制御を緩和します。単に認可されたユーザーを追加する場合は、何もシミュレートする必要はありません。

レルム内のデータにアクセスする新機能を追加しようとしているが、レルムに対して認可する新規データベース・ユーザーが不明な場合は、単に、新機能をテストとして実行してください(認可されていない場合は、機能がブロックされます)。Database Vault 監査ログをレビューして、レルム・データにアクセスしようとしたユーザー名を確認し、新規データベース・ユーザーを認可して追加します。

親トピック: シミュレーション・モードでのレルムのログ記録

10.3.7 ユースケース: 認可されたユーザーのレルムからの削除のテスト

このユースケースでは、認可されたユーザーを削除し、シミュレーション・モードを使用してそのユーザーがまだレルムにアクセスする必要があるかどうかを確認します。

認可されたユーザーが認可されたアクティビティのためにレルムにアクセスしているかどうかを確認する必要があるため、このユーザーを削除してよいか不明な場合があります。

データが通常のレルムによってのみ保護されている場合は、認可されているユーザーのみが異なるレルムのクローンを作成できます。元のレルムから削除対象のユーザーを削除し、このユーザーをレルムのクローンに追加します。次に、`audit on`

successを取得するようにレルムのクローンの監査設定を変更します。こうすることで、削除したユーザーが一定期間にわたりレルムにアクセスしていた場合、監査レコードにそのユーザーが表示されます。この場合、監査ポリシーも使用できます。必須レルムによって保護されているデータの場合、最適な方法は監査ポリシーを作成することです。

親トピック: シミュレーション・モードでのレルムのログ記録

10.3.8 ユースケース: レルムを使用した新規ファクタのテスト

このユースケースでは、ファクタに対する変更をテストします。

ファクタが変更されるシナリオには、次の2つがあります。

- アプリケーションまたはインフラストラクチャに対する変更によって、ファクタが強制的に変更される場合

  この場合、元のファクタを保持する必要はありません。ただし、新規ファクタのテスト中、オブジェクトおよび認可されたユーザーを有効化されたままにしておくことができる必要があります。有効化されたレルムを使用して、認可されたユーザーからファクタを削除できます。同時に、認可されたユーザーなしで、保護されている同じオブジェクトの必須レルムをシミュレーション・モードで作成します。通常のレルムが認可されていないユーザーからオブジェクトを保護し、シミュレーション・レルムがすべてのアクセスをファクタ情報とともに取得します。ユーザーごとに、シミュレーション・ログを調べてシミュレーション・モードの必須レルムに追加できる新規ファクタを見つけ、元の通常のレルムに移行する前にそのファクタがクリーンであることを確認できます。

- アプリケーションおよびインフラストラクチャの変更は行われないが、新規ファクタの追加やファクタの削除などの変更は行われる場合

  ファクタを追加したら、元のレルムから、新規ファクタが追加された2つ目のシミュレーション・レルムのクローンを作成する必要があります。シミュレーション・ログでファクタの使用に問題がないことが示されている場合は、元のレルムに新規ファクタを安全に導入できます。

  ファクタを削除するとセキュリティ・プロファイルが低下するため、単に、ルール・セットからファクタを削除できます。テストを実行する必要はありません。

親トピック: シミュレーション・モードでのレルムのログ記録

10.3.9 ユースケース: 既存のコマンド・ルールへの変更のテスト

このユースケースでは、既存のコマンド・ルールに対する変更を、元のコマンド・ルールを有効にしたままテストします。

コマンド・ルールには更新が必要な場合があり、本番で変更を有効にする前にテストすることが理想的です。既存のコマンド・ルールに追加する新規コマンド・ルールの場合は、作成時にその新規コマンド・ルールをシミュレーション・モードに設定します。その他の既存のコマンド・ルールはすでに有効化され、保護を提供しています。

既存のコマンド・ルールを変更する必要がある場合、既存の保護を保持したまま新しい変更をテストする方法はありません。元のコマンド・ルールの実行内容を取得する監査ポリシーを作成し、それに対するアラートを設定することをお薦めします。監査はコマンド・ルールとは異なり、SQLの実行を阻止しませんが、少なくともアクションに関するアラートを受け取ることはできます。続けて、更新された新しいコマンド・ルールをシミュレーション・モードに設定してテストできます。

親トピック: シミュレーション・モードでのレルムのログ記録

10.4 チュートリアル: シミュレーション・モードの使用によるレルムに対する違反の追跡

このチュートリアルでは、シミュレーション・モードを使用するレルムを作成してから、レルムに対する違反をテストする方法を示します。

- このチュートリアルについて
  このチュートリアルでは、HR.EMPLOYEES表に対するレルムを作成し、それに対する違反をテストします。
- ステップ1: このチュートリアル用のユーザーの作成
  このチュートリアル用に3つのユーザーを作成する必要があります。
- ステップ2: レルムおよびOracle Database Vaultポリシーの作成
  次に、HR.EMPLOYEES表の周囲にレルムを作成し、このレルムをOracle Database Vaultポリシーに追加します。
- ステップ3: レルムおよびポリシーのテスト
  ユーザーtjones_dbaがレルムに対して違反をコミットし、レルムおよびポリシーをテストします。
- ステップ4: DBA_DV_SIMULATION_LOGビューでの違反の問合せ
  これで、ユーザーtjones_dbaが犯した違反のシミュレーション・モード・ログを確認できるようになります。
- ステップ5: レルムの有効化および再テスト
  違反を取得したので、ユーザーpsmithは、HR.EMPLOYEES_polポリシーを更新できます。
- ステップ6: このチュートリアルのコンポーネントの削除
  コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

親トピック: レルムおよびコマンド・ルール・アクティビティのログ記録のためのシミュレーション・モードの使用

10.4.1 このチュートリアルについて

このチュートリアルでは、HR.EMPLOYEES表に対するレルムを作成し、それに対する違反をテストします。

HR.EMPLOYEES表には、従業員の給与などの機密データが含まれています。レルムをテストするために、管理者tjones_dbaは、別の従業員smavrisの給与を参照して変更します。Database Vault管理者sec_admin_owenは、シミュレーション・モードを使用して、HR.EMPLOYEES表に対する違反を追跡します。これを達成するために、ユーザーsec_admin_owenは、委任された管理者であるユーザーpsmithが所有することになる、Database Vaultポリシーを作成します。ユーザーpsmithは、その後、DV_OWNERロールやDV_ADMINロールを必要とせずに、制限された変更をポリシーに加えることができるようになります。

親トピック: チュートリアル: シミュレーション・モードの使用によるレルムに対する違反の追跡

10.4.2 ステップ1: このチュートリアル用のユーザーの作成

このチュートリアル用に3つのユーザーを作成する必要があります

各ユーザーは、Database Vaultポリシー所有者であるpsmith、HR.EMPLOYEES表での違反をコミットするtjones_dba、給与がtjones_dbaの違反を受けるsmavrisとなります。

1. DV_ACCTMGRロールを付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus accts_admin_ace
   Enter password: password
   ```

マルチテナント環境で、適切なプラガブル・データベース(PDB)にログインする必要があります。たとえば:

```
CONNECT accts_admin_ace@hrpdb
Enter password: password
```

使用可能なPDBを見つけるには、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. 次のユーザーを作成し、CREATE SESSION権限を付与します。

```
GRANT CREATE SESSION TO psmith IDENTIFIED BY password;
GRANT CREATE SESSION TO tjones_dba IDENTIFIED BY password;
GRANT CREATE SESSION TO smavris IDENTIFIED BY password;
```

『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

3. DV_OWNERロールを付与されたユーザーとして接続します。
   たとえば:

```
CONNECT sec_admin_owen -- Or, sec_admin_owen@hrpdb
Enter password: password
```

4. ユーザーpsmithにDV_POLICY_OWNERロールを付与します。これにより、psmithは、Database Vaultポリシーを管理できるようになります。

```
GRANT DV_POLICY_OWNER TO psmith;
```

5. SYSDBA管理権限を持つユーザーSYSとして接続します。

```
CONNECT SYS AS SYSDBA -- Or, CONNECT SYS@hrpdb AS SYSDBA
Enter password: password
```

6. DBAロールをユーザーtjones_dbaに付与します。

```
GRANT DBA TO tjones_dba;
```

7. HRスキーマ所有者として接続します。

```
CONNECT HR -- Or, HR@hrpdb
Enter password: password
```

8. HR.EMPLOYEES表に対するSELECT権限をユーザーsmavrisに付与します。

```
GRANT SELECT ON HR.EMPLOYEES TO smavris;
```

この段階で、ユーザーはすべて、適切な権限を作成および付与されています。

親トピック: チュートリアル: シミュレーション・モードの使用によるレルムに対する違反の追跡

10.4.3 ステップ2: レルムおよびOracle Database Vaultポリシーの作成

次に、HR.EMPLOYEES表周りのレルムを作成し、このレルムをOracle Database Vaultポリシーに追加します。

1. DV_OWNERロールを付与されたユーザーとして接続します。

   たとえば:

```
CONNECT sec_admin_owen -- Or, sec_admin_owen@hrpdb
Enter password: password
```

2. 次のように、HR.EMPLOYEES表周りのレルムを作成します。

   これらのプロシージャは、HR.EMPLOYEES_realmレルムを作成し、HR.EMPLOYEES表をこのレルムに追加し、HRを所有者として認証し、ユーザーpsmithを参加者として認証して、レルムをシミュレーション・モードに設定します。

```
BEGIN
 DBMS_MACADM.CREATE_REALM(
  realm_name    => 'HR.EMPLOYEES_realm',
  description   => 'Realm to protect HR.EMPLOYEES',
  enabled       => DBMS_MACUTL.G_SIMULATION,
  audit_options => DBMS_MACUTL.G_REALM_AUDIT_FAIL,
  realm_type    => 0);
END;
/
BEGIN
 DBMS_MACADM.ADD_OBJECT_TO_REALM(
  realm_name   => 'HR.EMPLOYEES_realm',
  object_owner => 'HR',
  object_name  => 'EMPLOYEES',
  object_type  => 'TABLE');
END;
/
```

3. HR.EMPLOYEES_pol Database Vaultポリシーを作成し、シミュレーション・モードになるよう設定します。

   これらのプロシージャは、HR.EMPLOYEES_polポリシーを作成し、たった今作成したレルムをポリシーに追加してから、ユーザーpsmithをポリシーの所有者として追加します。

```
BEGIN
 DBMS_MACADM.CREATE_POLICY(
  policy_name  => 'HR.EMPLOYEES_pol',
  description  => 'Policy to protect HR.EMPLOYEES',
  policy_state => DBMS_MACADM.G_SIMULATION);
END;
/
BEGIN
 DBMS_MACADM.ADD_REALM_TO_POLICY(
  policy_name  => 'HR.EMPLOYEES_pol',
  realm_name   => 'HR.EMPLOYEES_realm');
END;
/
BEGIN
 DBMS_MACADM.ADD_OWNER_TO_POLICY(
  policy_name  => 'HR.EMPLOYEES_pol',
  owner_name   => 'PSMITH');
END;
/
```

この時点では、レルムおよびポリシーはテストを受ける準備ができています。

親トピック: チュートリアル: シミュレーション・モードの使用によるレルムに対する違反の追跡

10.4.4 ステップ3: レルムおよびポリシーのテスト

ユーザーtjones_dbaがレルムに対して違反をコミットし、レルムおよびポリシーをテストします。

1. ユーザーtjones_dbaとして接続します。

```
CONNECT tjones_dba -- Or, tjones_dba@hrpdb
Enter password: password
```

2. HR.EMPLOYEES表にsmavrisの給与を問い合せます。

```
SELECT SALARY FROM HR.EMPLOYEES WHERE EMAIL = 'SMAVRIS';
```

次のような出力が表示されます。

```
    SALARY
----------
      6500
```

3. smavrisの給与を半分に減らします。

```
UPDATE HR.EMPLOYEES
SET SALARY = SALARY / 2
WHERE EMAIL = 'SMAVRIS';
1 row updated.
```

4. ユーザーsmavrisとして接続します。

```
CONNECT smavris -- Or, smavris@hrpdb
```

5. smavrisの給与を問い合せます。

```
SELECT SALARY FROM HR.EMPLOYEES WHERE EMAIL = 'SMAVRIS';
```

次のような出力が表示されます。

```
    SALARY
----------
      3250
```

この時点では、tjones_dbaの違反は、DBA_DV_SIMULATION_LOGデータ・ディクショナリ・ビューに記録されています。

親トピック: チュートリアル: シミュレーション・モードの使用によるレルムに対する違反の追跡

10.4.5 ステップ4: DBA_DV_SIMULATION_LOGビューでの違反の問合せ

これで、ユーザーtjones_dbaが犯した違反のシミュレーション・モード・ログを確認できるようになります。

1. DV_OWNERロールを付与されたユーザーとして接続します。

   たとえば:

```
CONNECT sec_admin_owen -- Or, leo_dvowner@hrpdb
Enter password: password
```

2. DBA_DV_SIMULATION_LOGデータ・ディクショナリ・ビューを問い合せます。

```
SELECT USERNAME, COMMAND, SQLTEXT, VIOLATION_TYPE
FROM DBA_DV_SIMULATION_LOG, TABLE(DBA_DV_SIMULATION_LOG.REALM_NAME) RN
WHERE RN.COLUMN_VALUE = 'HR.EMPLOYEES_realm';
```

次のような出力が表示されます。

```
USERNAME
--------------------------------------------------------------------------------
-
COMMAND
--------------------------------------------------------------------------------
-
SQLTEXT
--------------------------------------------------------------------------------
-
VIOLATION_TYPE
```

```
--------------------------------------------------------------------------------
-
TJONES_DBA
UPDATE
UPDATE HR.EMPLOYEES SET SALARY = SALARY / 2 WHERE EMAIL = 'SMAVRIS'
Realm Violation
USERNAME
--------------------------------------------------------------------------------
-
COMMAND
--------------------------------------------------------------------------------
-
SQLTEXT
--------------------------------------------------------------------------------
-
VIOLATION_TYPE
--------------------------------------------------------------------------------
-
TJONES_DBA
SELECT
SELECT SALARY FROM HR.EMPLOYEES WHERE EMAIL = 'SMAVRIS'
Realm Violation
```

出力に、ユーザーtjones_dbaが2つの違反をコミットしたことが示されます。まず、別の従業員の給与を参照し、それのみでなく、半分に減らしました。違反タイプはレルム違反です。自分の給与の参照は正当であるため、smavrisによる問合せは取得されませんでした。

親トピック: チュートリアル: シミュレーション・モードの使用によるレルムに対する違反の追跡

10.4.6 ステップ5: レルムの有効化および再テスト

違反を取得したので、ユーザーpsmithは、HR.EMPLOYEES_polポリシーを更新できます。

これは、HR.EMPLOYEES_realmレルムを有効化できるようにするためです。その後、もう一度違反をテストできます。

1. ユーザーpsmithとして接続します。

```
CONNECT psmith -- Or, psmith@hrpdb
Enter password: password
```

2. ポリシーを有効になるよう更新します。

```
BEGIN
 DBMS_MACADM.UPDATE_POLICY_STATE(
  policy_name   => 'HR.EMPLOYEES_pol',
  policy_state  => 1);
END;
/
```

3. ユーザーtjones_dbaとして接続します。

```
CONNECT tjones_dba --Or, tjones_dba@hrpdb
```

4. smavrisの給与をさらに低く減らすことを試みます。

```
UPDATE HR.EMPLOYEES
SET SALARY = SALARY / 2
WHERE EMAIL = 'SMAVRIS';
```

次のような出力が表示されます。

```
ERROR at line 1:
ORA-01031: insufficient privileges
```

ポリシー(現在は有効になっている)により、レルムでHR.EMPLOYEES表を保護できます。smavrisの給与は、これ以上減らせません。

親トピック: チュートリアル: シミュレーション・モードの使用によるレルムに対する違反の追跡

10.4.7 ステップ6: このチュートリアルのコンポーネントの削除

コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

1. DV_OWNERロールを付与されたユーザーとして接続します。
   たとえば:

```
CONNECT sec_admin_owen -- Or, leo_dvowner@hrpdb
Enter password: password
```

2. HR.EMPLOYEES_pol Database Vaultポリシーを削除します。

```
EXEC DBMS_MACADM.DROP_POLICY('HR.EMPLOYEES_pol');
```

   まずポリシーを削除してからでないと、その内容を削除できません。

3. HR.EMPLOYEES_realmレルムを削除します。

```
EXEC DBMS_MACADM.DELETE_REALM('HR.EMPLOYEES_realm');
```

4. 累積されたシミュレーション・モード・ログ・データを削除します。

   シミュレーション・モード・ログではユーザーtjones_dbaに関する情報のみが取得されるため、このユーザーに関連する行のみを削除できます。

```
DELETE FROM DVSYS.SIMULATION_LOG$ WHERE USERNAME = 'TJONES_DBA';
```

5. ユーザーHRとして接続します。

```
CONNECT HR -- Or, CONNECT HR@hrpdb
Enter password: password
```

6. smavrisの給与をその違反前の状態に戻します。

```
UPDATE HR.EMPLOYEES
SET SALARY = 6500
WHERE EMAIL = 'SMAVRIS';
```

7. DV_ACCTMGRロールを付与されているユーザーとして接続します。

   たとえば:

```
CONNECT accts_admin_ace -- Or, accts_admin_ace@hrpdb
Enter password: password
```

8. ユーザーpsmith、smavrisおよびtjones_dbaを削除します。

```
DROP USER psmith;
DROP USER smavris;
DROP USER tjones_dba;
```

親トピック: チュートリアル: シミュレーション・モードの使用によるレルムに対する違反の追跡

11 Oracle Database Vaultとその他のOracle製品の統合

Oracle Database Vaultは、Oracle Enterprise User Securityなど別のOracle製品と統合できます。

- Oracle Database Vaultとエンタープライズ・ユーザー・セキュリティの統合について
  Oracle Database Vaultは、Oracle Enterprise User Securityと統合できます。
- Oracle Database Vaultと透過的データ暗号化の統合
  データがデータベースのセキュアな範囲外にある場合のデータ保護を提供するという意味で、透過的データ暗号化はOracle Database Vaultを補完するものです。
- Oracle Virtual Private Databaseへのファクタの追加
  Oracle Virtual Private Databaseにはファクタを追加できます。
- Oracle Database VaultとOracle Label Securityの統合
  Oracle Database VaultとOracle Label Securityを統合すると、レポートおよびデータ・ディクショナリ・ビューとの統合を確認できます。
- Oracle Database VaultとOracle Data Guardの統合
  Oracle Database VaultとOracle Data Guardの統合では、まずプライマリ・データベースを構成し、次にスタンバイ・データベースを構成します。
- Oracle Database Configuration Assistantを使用したOracle Internet Directoryの登録
  Oracle Database Vault対応データベースでOracle Internet Directoryを使用できます。

11.1 Oracle Database Vaultとエンタープライズ・ユーザー・セキュリティの統合

Oracle Database Vaultは、Oracle Enterprise User Securityと統合できます。

- Oracle Database Vaultとエンタープライズ・ユーザー・セキュリティの統合について
  エンタープライズ・ユーザー・セキュリティでは、データベース・ユーザーと認可が1箇所で集中管理されます。
- エンタープライズ・ユーザー認可の構成
  エンタープライズ・ユーザー認可を構成するには、Oracle Database Vaultルール・セットを作成して、ユーザー・アクセスを制御する必要があります。
- Oracle Database Vaultアカウントをエンタープライズ・ユーザー・アカウントとして構成
  既存のOracle Database Vaultユーザー・アカウントをエンタープライズ・ユーザー・アカウントとして構成できます。

親トピック: Oracle Database Vaultとその他の Oracle製品の統合

11.1.1 Oracle Database Vaultとエンタープライズ・ユーザー・セキュリティの統合について

エンタープライズ・ユーザー・セキュリティでは、データベース・ユーザーと認可が1箇所で集中管理されます。

Oracle Identity Managementと組み合せ、Oracle Database Enterprise Editionで使用できます。

通常、Oracle Database VaultをOracle Enterprise User Securityと統合するには、適切なレルムを構成して、保護の対象となるデータベース内のデータを保護します。

必要に応じてOracle Database Vaultレルムを定義した後に、エンタープライズ・ユーザーに対してアクセスを許可または禁止するルール・セットを作成できます。

関連項目:

エンタープライズ・ユーザー・セキュリティの詳細は、『Oracle Databaseエンタープライズ・ユーザー・セキュリティ管理者ガイド』を参照してください

親トピック: Oracle Database Vaultとエンタープライズ・ユーザー・セキュリティの統合

11.1.2 エンタープライズ・ユーザー認可の構成

エンタープライズ・ユーザー認可を構成するには、Oracle Database Vaultルール・セットを作成して、ユーザー・アクセスを制御する必要があります。

1. ユーザー・アクセスを許可または禁止するルールを作成します。

   「ルール・セットに追加するルールの作成」の説明に従って新しいルールを作成します。「ルールの作成」ページで、次のPL/SQLを「ルール式」フィールドに入力します。

   ```
   SYS_CONTEXT('USERENV','AUTHENTICATED_IDENTITY') = 'user_domain_name'
   ```

   `user_domain_name`をドメインに置き換えます。たとえば、次のようになります。

   ```
   SYS_CONTEXT('USERENV','AUTHENTICATED_IDENTITY') = 'myserver.us.example.com'
   ```

2. このルールを新しいルール・セットに追加します。

   新しいルール・セットの作成方法は、それに対する既存のルールの追加方法も含め、「ルール・セットの作成」で説明されています。

3. このルール・セットを保護対象のデータのレルム認可に追加します。

   レルム認可の作成方法は、「レルム認可について」で説明されています。「認可ルール・セット」リストで、ステップ2で作成したルール・セットを選択します。後で、レルム認可がすべてのユーザーに適用されます。

親トピック: Oracle Database Vaultとエンタープライズ・ユーザー・セキュリティの統合

11.1.3 Oracle Database Vaultアカウントをエンタープライズ・ユーザー・アカウントとして構成

既存のOracle Database Vaultユーザー・アカウントをエンタープライズ・ユーザー・アカウントとして構成できます。

1. `CREATE ROLE`システム権限を付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus system
   Enter password: password
   ```

2. マルチテナント環境で、適切なプラガブル・データベース(PDB)に接続します。

   たとえば:

   ```
   CONNECT SYSTEM@hrpdb
   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。

3. DV_OWNERロールのグローバル・ロールとDV_ACCTMGRロールのグローバル・ロールを作成します。

   たとえば:

```
CREATE ROLE g_dv_owner IDENTIFIED GLOBALLY;
CREATE ROLE g_dv_acctmgr IDENTIFIED GLOBALLY;
```

4. DV_OWNERロールを付与されたユーザーとして接続します。

   たとえば:

```
CONNECT sec_admin_owen -- Or, CONNECT sec_admin_owen@hrpdb
Enter password: password
```

5. DV_OWNERロールをグローバルDV_OWNERロールに付与します。

```
GRANT DV_OWNER TO g_dv_owner;
```

6. DV_ACCTMGRロールを付与されているユーザーとして接続します。

   たとえば:

```
CONNECT dbv_acctmgr -- Or, CONNECT dbv_acctmgr@hrpdb
Enter password: password
```

7. DV_ACCTMGRロールをグローバルDV_ACCTMGRロールに付与します。

```
GRANT DV_ACCTMGR TO g_dv_acctmgr;
```

8. SYSDBA管理権限を持つユーザーSYSとして接続します。

```
CONNECT SYS AS SYSDBA -- Or, CONNECT SYS@hrpdb AS SYSDBA
Enter password: password
```

9. Database VaultユーザーをOIDにインポートするDV_ACCTMGRユーザーに、一時的にCREATE TABLE権限とSELECT_CATALOG_ROLEロールを付与します。

```
GRANT CREATE TABLE, SELECT_CATALOG_ROLE TO dbv_acctmgr;
```

10. コマンド・ラインで、ユーザー移行ユーティリティ(UMU)を実行してDatabase VaultアカウントをOracle Internet Directory (OID)にインポートします。

    次に例では、Database Vaultアカウントleo_dvownerおよびbea_dvacctmgrをOIDにインポートします。DBADMIN設定にDV_ACCTMGRユーザーが指定されます。

```
$ORACLE_HOME/rdbms/bin/umu PHASE=ONE
DBADMIN=dbv_acctmgr:password
ENTADMIN=cn=jane_ent_admin,dc=example,dc=com:password
USERS= LIST
DBLOCATION=example.com:7777:orcl
DIRLOCATION=example.com:636
USERSLIST=leo_dvowner:bea_dvacctmgr
MAPSCHEMA=PRIVATE
CONTEXT=CONTEXT="c=Users, c=us"
KREALM=EXAMPLE.COM
$ORACLE_HOME/rdbms/bin/umu PHASE=TWO
DBADMIN=dbv_acctmgr:password
ENTADMIN=cn=jane_ent_admin,dc=example,dc=com:password
DBLOCATION=example.com:7777:orcl
```

```
DIRLOCATION=example.com:636
```

デフォルトでは、$ORACLE_HOME/network/log/umu.logファイルにエラーが書き込まれます。

11. Oracle Internet Directoryセルフ・サービス・コンソール(http://hostname:port/oiddas/)で、グローバルDV_OWNERおよびDV_ACCTMGRロール(g_dv_ownerやg_dv_acctmgrなど)をエンタープライズ・ユーザーのDatabase Vaultアカウントに付与します。

グローバル・ロールからエンタープライズ・ロールを作成し、このロールをユーザーに付与する方法については、『Oracle Databaseエンタープライズ・ユーザー・セキュリティ管理者ガイド』のエンタープライズ・ユーザーの作成例に関する項を参照してください。

12. SQL*Plusで、SYSDBA管理権限を持つユーザーSYSとして、CREATE TABLEおよびSELECT_CATALOG_ROLEロールをDV_ACCTMGRユーザーから取り消します。

```
REVOKE CREATE TABLE, SELECT_CATALOG_ROLE FROM dbv_acctmgr;
```

関連項目:

ユーザー移行ユーティリティの詳細は、『Oracle Databaseエンタープライズ・ユーザー・セキュリティ管理者ガイド』を参照してください

親トピック: Oracle Database Vaultとエンタープライズ・ユーザー・セキュリティの統合

11.2 Oracle Database Vaultと透過的データ暗号化の統合

データがデータベースのセキュアな範囲外にある場合のデータ保護を提供するという意味で、透過的データ暗号化はOracle Database Vaultを補完するものです。

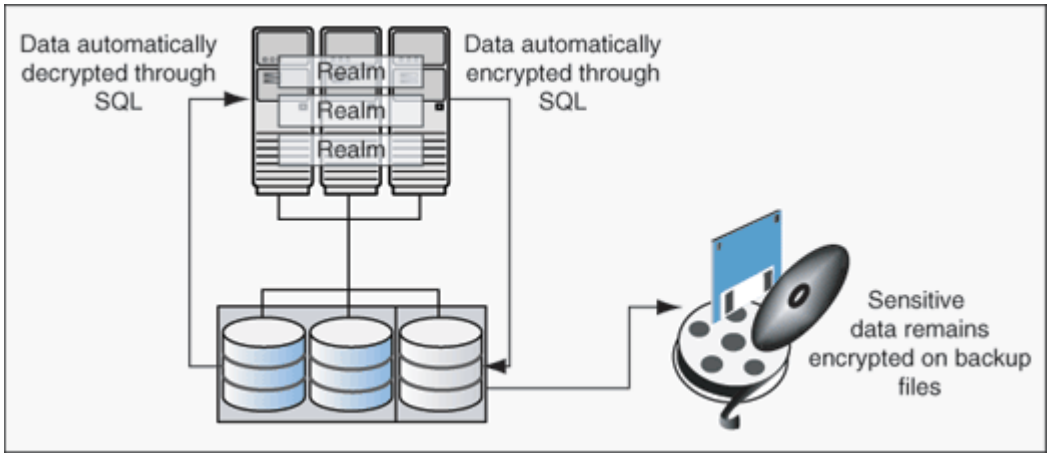
透過的データ暗号化を使用して、データベース管理者またはデータベース・セキュリティ管理者は、アプリケーション表の機密情報の列のみを暗号化したり、アプリケーション表領域全体を暗号化したりできます。アプリケーションを変更する必要はありません。

ユーザーが認証チェックと認可チェックを通ると、透過的データ暗号化によりユーザーの情報は自動的に暗号化および復号化されます。このように、アプリケーションを変更しなくても暗号化を実装できます。

透過的データ暗号化ユーザーに適切な権限を付与したら、透過的データ暗号化を通常どおり管理でき、Database Vaultを補完するものとして使用できます。

図11-1に、暗号化されたデータがOracle Database Vaultレルムでどのように処理されるかを示します。

図11-1 暗号化されたデータとOracle Database Vault



関連項目:

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

親トピック: Oracle Database Vaultとその他の Oracle製品の統合

11.3 Oracle Virtual Private Databaseへのファクタの追加

Oracle Virtual Private Databaseにはファクタを追加できます。

1. PL/SQLファンクションまたはPL/SQL式であるVirtual Private Databaseポリシー述語を定義します。
2. 各ファンクションまたは式に対して、ファクタごとに作成されるPL/SQLファンクションDVF.F$を使用します。

関連項目:

Oracle Databaseセキュリティ・ガイド Oracle Virtual Private Database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: Oracle Database Vaultとその他の Oracle製品の統合

11.4 Oracle Database VaultとOracle Label Securityの統合

Oracle Database VaultとOracle Label Securityを統合すると、レポートおよびデータ・ディクショナリ・ビューとの統合を確認できます。

- Oracle Database VaultとOracle Label Securityの統合方法
  Oracle Database VaultとOracle Label Securityの統合により、OLSラベルをDatabase Vaultファクタ・アイデンティティに割当てできます。
- Oracle Database VaultをOracle Label Securityとともに使用するための要件
  Oracle Database VaultとOracle Label Securityを使用する前に、特定の要件を満たす必要があります。
- Oracle Label SecurityポリシーでのOracle Database Vaultファクタの使用
  セキュリティを強化するには、Oracle Database VaultのファクタとOracle Label Securityポリシーを統合します。
- チュートリアル: Oracle Database VaultとOracle Label Securityの統合
  Oracle Database VaultとOracle Label Securityの統合により、同じ権限を持つ2人の管理ユーザーに異なるレベルのアクセス権を付与できます。
- 関連するレポートおよびデータ・ディクショナリ・ビュー
  Oracle Database Vaultには、Oracle Database VaultとOracle Label Securityの統合に関する情報が示されるレポートおよびデータ・ディクショナリ・ビューが用意されています。

親トピック: Oracle Database Vaultとその他の Oracle製品の統合

11.4.1 Oracle Database VaultとOracle Label Securityの統合方法

Oracle Database VaultとOracle Label Securityの統合により、OLSラベルをDatabase Vaultファクタ・アイデンティティに割当てできます。

Oracle Label Securityでは、データベース表またはPL/SQLプログラムのレコードへのアクセスを制限できます。たとえば、アクセスを特定の管理者に限定する必要があるレコードを含むEMPLOYEE表の、HIGHLY SENSITIVEラベル(Oracle Label

Securityラベル)で保護されたデータをMaryは参照できます。もう1つのラベルを、このデータへのよりオープンなアクセスを許可するPUBLICとすることができます。

Oracle Database Vaultでは、データベース・セッションが発生するネットワーク用に、次のアイデンティティを指定してNetworkというファクタを作成できます。

- Intranet: 従業員が会社のイントラネット内の場所で作業している場合に使用します。
- Remote: 従業員がVPN接続から在宅で作業している場合に使用します。

次に、両方のアイデンティティに最大のセッション・ラベルを割り当てます。たとえば:

- IntranetアイデンティティをOracle Label SecurityラベルのHIGHLY SENSITIVEに割り当てます。
- RemoteアイデンティティをPUBLICラベルに割り当てます。

つまり、MaryがVPN接続を使用して在宅で作業している場合は、PUBLICアイデンティティのもとに保護される、限定された表データにのみアクセスできます。しかし、職場にいる場合は、Intranetアイデンティティを使用しているため、HIGHLY SENSITIVEデータにアクセスできます。「チュートリアル: Oracle Database VaultとOracle Label Securityの統合」に、このような統合を実現する方法の例を示します。

非統合監査環境では、Oracle Label Securityとの統合は、「Label Security統合の監査」レポートを使用して監査できます。Oracle Database Vaultは、監査証跡を`DVSYS.AUDIT_TRAIL$`表に書き込みます。統合監査が有効な場合、『Oracle Databaseセキュリティ・ガイド』の説明に従い、この情報を取得する監査ポリシーを作成できます。

関連項目:

- 「Label Security統合の監査」レポート
- Database VaultとOracle Label Securityを統合するために使用できるDatabase Vault APIの詳細は、「Oracle Database Vault Oracle Label SecurityのAPI」を参照してください
- Oracle Database VaultとOracle Label Securityの統合で実行できるレポートの詳細は、「関連するレポートおよびデータ・ディクショナリ・ビュー」を参照してください
- Oracle Label Securityのラベルの詳細は、『Oracle Label Security管理者ガイド』を参照してください

親トピック: Oracle Database VaultとOracle Label Securityの統合

11.4.2 Oracle Database VaultをOracle Label Securityとともに使用するための要件

Oracle Database VaultとOracle Label Securityを使用する前に、特定の要件を満たす必要があります。

- Oracle Label Securityは別個にライセンス許可されます。それを使用するためのライセンスを購入済であることを確認してください。
- Oracle Database Vaultをインストールする前に、Oracle Label Securityのインストールを済ませておく必要があります。
- Oracle Label Securityのインストール・プロセスで`LBACSYS`ユーザー・アカウントが作成されます。`DV_ACCTMGR`ロールを付与されているユーザーとして、このアカウントをロック解除し、新しいパスワードを付与します。たとえば:

```
sqlplus accts_admin_ace -- Or, sqlplus accts_admin_ace@hrpdb for a PDB
Enter password: password
```

```
ALTER USER LBACSYS ACCOUNT UNLOCK IDENTIFIED BY password;
```

『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

- Oracle Enterprise ManagerでLBACSYSユーザー・アカウントを使用する場合、SYSDBA管理権限を持つユーザーSYSとしてEnterprise Managerにログインし、このユーザーにSELECT ANY DICTIONARYおよびSELECT_CATALOG_ROLEシステム権限を付与します。
- 適切なOracle Label Securityポリシーが定義されていることを確認してください。詳細は、『Oracle Label Security管理者ガイド』を参照してください。
- Oracle Label SecurityポリシーをDatabase Vaultポリシーと統合する場合、Oracle Label Securityのポリシー名が24文字未満であることを確認します。ALL_SA_POLICIESデータ・ディクショナリ・ビューのPOLICY_NAME列を問い合せることで、Oracle Label Securityポリシーの名前がチェックできます。

親トピック: Oracle Database VaultとOracle Label Securityの統合

11.4.3 Oracle Label SecurityポリシーでのOracle Database Vaultファクタの使用方法

セキュリティを強化するには、Oracle Database VaultのファクタとOracle Label Securityポリシーを統合します。

- Oracle Label SecurityポリシーでのOracle Database Vaultファクタの使用
  Oracle Database VaultとOracle Label Securityの統合により、データベース・セッションの最大セキュリティ・チェックを制御できます。
- Oracle Label Securityポリシーと連携するファクタの構成
  Oracle Label Securityポリシーの最大許容データ・ラベルに含めるファクタを定義できます。

親トピック: Oracle Database VaultとOracle Label Securityの統合

11.4.3.1 Oracle Label SecurityポリシーでのOracle Database Vaultファクタの使用

Oracle Database VaultとOracle Label Securityの統合により、データベース・セッションの最大セキュリティ・チェックを制御できます。

Oracle Database Vaultでは、Oracle Label Securityポリシーに関連付けられているOracle Database Vaultファクタのラベルをマージすることによって、データベース・セッションにおける各ラベルの最大許容データをマージして、データベース・セッションの最大セキュリティ・チェックを制御します。

つまり、ラベルは、データベース表の行のアクセス権限に対する識別子として機能します。ポリシーは、表の行へのアクセスを管理するラベル、ルールおよび認可と関連付けられた名前です。

関連項目:

行ラベルおよびポリシーの詳細は、『Oracle Label Security管理者ガイド』を参照してください

親トピック: Oracle Label SecurityポリシーでのOracle Database Vaultファクタの使用方法

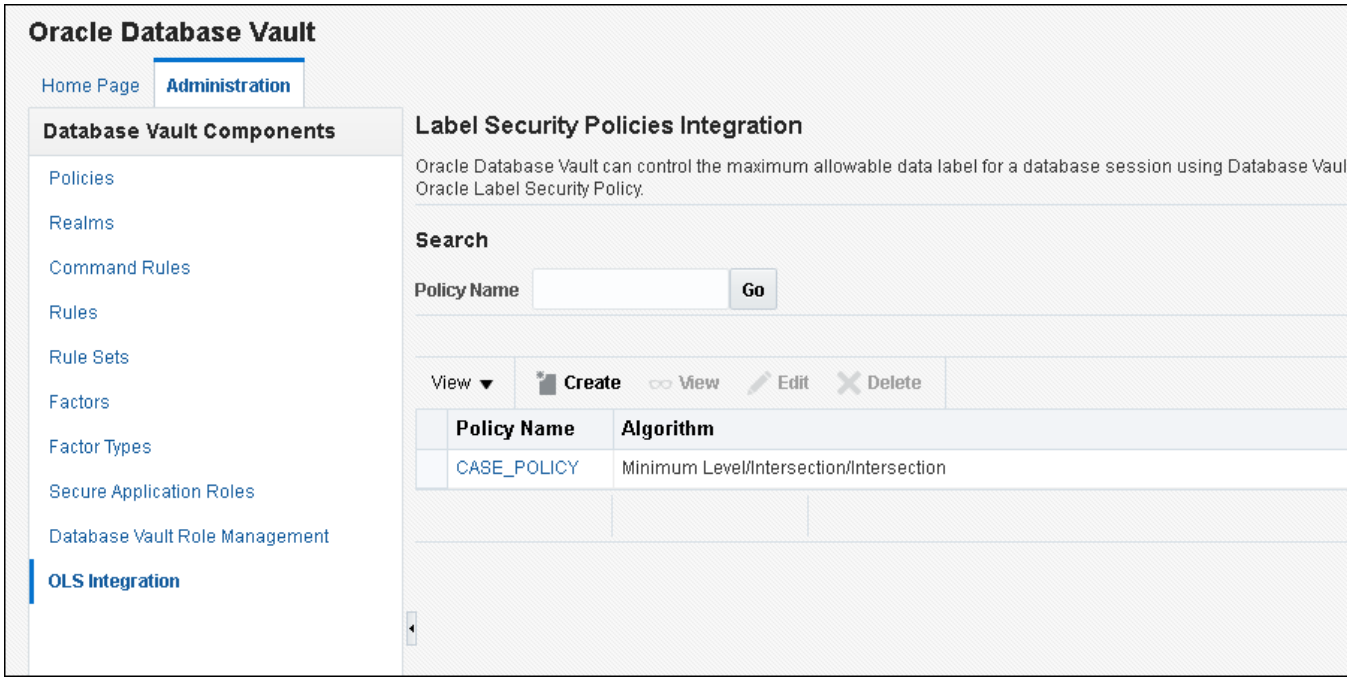
11.4.3.2 Oracle Label Securityポリシーと連携するファクタの構成

Oracle Label Securityポリシーの最大許容データ・ラベルに含めるファクタを定義できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. ユーザー・アカウントLBACSYSを、Label Securityポリシーが適用されているスキーマを含むレルムの所有者にします。

   これにより、レルム内で保護されているすべてのデータへのアクセス権がLBACSYSアカウントに付与され、データを適切に分類できるようになります。

   LBACSYSアカウントは、Oracle Universal Installerのカスタム・インストール・オプションを使用してOracle Label Security内に作成されます。Oracle Database Vaultとともに使用するOracle Label Securityポリシーを作成するには、LBACSYSを使用予定のレルムの所有者にする必要があります。詳細は、「レルム認可について」を参照してください。
3. レルムの参加者または所有者として(ラベル・セキュリティ・ポリシーが適用されている)スキーマの所有者を認可します。
4. 「管理」ページの「Database Vaultコンポーネント」で、「OLS統合」をクリックします。



5. 「ラベル・セキュリティ・ポリシー統合」ページで、次のようにします。
   - 新しいラベル・セキュリティ・ポリシーをDatabase Vaultに登録するには、「作成」をクリックします。
   - Database Vaultに登録されている既存のラベル・セキュリティ・ポリシーを編集するには、リストでそのポリシーを選択して「編集」をクリックします。
6. 次の設定を入力します。
   - Label Securityポリシー: リストから使用するOracle Label Securityポリシーを選択します。
   - アルゴリズム: Oracle Label Securityで2つのラベルをマージしている場合、ラベルマージ・アルゴリズムを必要に応じて変更します。ほとんどの場合、「LII - 最小レベル/論理積/論理積」を選択します。この設定は、Oracle Label Security管理者が2つのラベルをマージする際に最も一般的に使用する方法です。この設定により、ラベルが異なる2つのデータ・セットを組み合せる際に必要な結果のラベルをアプリケーションで特定する必要がある場合に、最適な柔軟性が提供されます。これは、データ・ラベルが異なる行で結合を使用して問合せを実行する必要がある場合にも必要です。

     マージ・アルゴリズムの指定にDBMS_MACADMパッケージを使用する場合、使用可能なマージ・アルゴリズムを

すべて示すリストについては、表19-2を参照してください。

- ラベル・セキュリティ・ポリシー・ファクタ: 「使用可能なファクタ」リストの「ラベル・セキュリティ・ポリシー・ファクタ」で、Oracle Label Securityポリシーと関連付けるファクタを選択します。「移動」をクリックして、ファクタを「選択したファクタ」リストに移動します。[Ctrl]キーを押しながら必要な各ファクタをクリックすると、複数のファクタを選択できます。

7. 「OK」をクリックします。

   ポリシーは「ラベル・セキュリティ・ポリシー統合」ページにリストされます。

8. ラベルのポリシーを使用して、ファクタ・アイデンティティのラベル付けを行います。

   詳細は、「ファクタへのアイデンティティの追加」を参照してください。

ノート:

Oracle Label Security ポリシーをファクタと関連付けない場合、Oracle Database Vault ではポリシーに対する Oracle Label Security のデフォルト動作が維持されます。

親トピック: Oracle Label SecurityポリシーでのOracle Database Vaultファクタの使用方法

11.4.4 チュートリアル: Oracle Database VaultとOracle Label Securityの統合

Oracle Database VaultとOracle Label Securityの統合により、同じ権限を持つ2人の管理ユーザーに異なるレベルのアクセス権を付与できます。

- このチュートリアルについて
  Oracle Database VaultファクタをOracle Label SecurityおよびOracle Virtual Private Database(VPD)とともに使用すると、機密データへのアクセスを制限できます。
- ステップ1: このチュートリアル用のユーザーの作成
  このチュートリアル用に2つの管理ユーザーを作成する必要があります。
- ステップ2: Oracle Label Securityポリシーの作成
  次に、Oracle Label Securityポリシーを作成し、ユーザーに適切な権限を付与できます。
- ステップ3: OLS認可を制御するためのOracle Database Vaultルールの作成
  Oracle Label Securityポリシーの作成後、これと連携するDatabase Vaultルールを作成できます。
- ステップ4: ルール・セットを使用するためのALTER SYSTEMコマンド・ルールの更新
  ルール・セットを使用する前に、デフォルト・コマンド・ルールであるALTER SYSTEMコマンド・ルールを更新する必要があります。
- ステップ5: 認可のテスト
  すべてのコンポーネントの準備ができたら、認可をテストする準備ができます。
- ステップ6: このチュートリアルのコンポーネントの削除
  コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

親トピック: Oracle Database VaultとOracle Label Securityの統合

11.4.4.1 このチュートリアルについて

Oracle Database VaultファクタをOracle Label SecurityおよびOracle Virtual Private Database(VPD)とともに使用すると、機密データへのアクセスを制限できます。

このようなデータを制限して、セキュリティ管理者が任意のデータ・セッションに対して定義するファクタの適切な組合せが存在する場合にのみデータベース・セッションに公開されるようにすることができます。

親トピック: チュートリアル: Oracle Database VaultとOracle Label Securityの統合

11.4.4.2 ステップ1: このチュートリアル用のユーザーの作成

このチュートリアル用に2つの管理ユーザーを作成する必要があります。

1. DV_ACCTMGRロールを付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus accts_admin_ace
   Enter password: password
   ```

   マルチテナント環境で、適切なプラガブル・データベース(PDB)に接続する必要があります。

   たとえば:

   ```
   sqlplus accts_admin_ace@hrpdb
   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。

2. 次のローカル・ユーザーを作成します。

   ```
   GRANT CREATE SESSION TO mdale IDENTIFIED BY password CONTAINER = CURRENT;
   GRANT CREATE SESSION TO jsmith IDENTIFIED BY password CONTAINER = CURRENT;
   ```

   『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

3. システム権限を付与でき、Oracleシステム権限およびロール管理レルムの所有者認可を付与されているユーザーとして接続し、ユーザーmdaleとユーザーjsmithに管理権限を付与します。

   ```
   CONNECT dba_psmith -- Or, CONNECT dba_psmith@hrpdb
   Enter password: password
   GRANT DBA TO mdale, jsmith;
   ```

   この段階で、ユーザーmdaleとユーザーjsmithは、同じ管理権限を持ちます。

親トピック: チュートリアル: Oracle Database VaultとOracle Label Securityの統合

11.4.4.3 ステップ2: Oracle Label Securityポリシーの作成

次に、Oracle Label Securityポリシーを作成し、ユーザーに適切な権限を付与します。

1. SQL*Plusで、Oracle Label Security管理者LBACSYSとして接続します。

   ```
   CONNECT LBACSYS -- Or, CONNECT LBACSYS@hrpdb
   Enter password: password
   ```

   ユーザーLBACSYSがロックされて無効になっている場合、Database Vaultアカウント・マネージャとして接続し、

LBACSYSアカウントのロックを解除して有効にしてから、LBACSYSとして再びログインします。

たとえば:

```
CONNECT accts_admin_ace -- Or, CONNECT bea_dvaccmgr@hrpdb
Enter password: password
ALTER USER LBACSYS ACCOUNT UNLOCK IDENTIFIED BY password;
CONNECT LBACSYS
Enter password: password
```

2. 新規のOracle Label Securityポリシーを作成します。

```
EXEC SA_SYSDBA.CREATE_POLICY('PRIVACY','PRIVACY_COLUMN','NO_CONTROL');
```

3. PRIVACYポリシーに次のレベルを作成します。

```
EXEC SA_COMPONENTS.CREATE_LEVEL('PRIVACY',2000,'S','SENSITIVE');
EXEC SA_COMPONENTS.CREATE_LEVEL('PRIVACY',1000,'C','CONFIDENTIAL');
```

4. PII区分を作成します。

```
EXEC SA_COMPONENTS.CREATE_COMPARTMENT('PRIVACY',100,'PII','PERS_INFO');
```

5. ユーザーmdaleとユーザーjsmithに次のラベルを付与します。

```
EXEC SA_USER_ADMIN.SET_USER_LABELS('PRIVACY','mdale','S:PII');
EXEC SA_USER_ADMIN.SET_USER_LABELS('PRIVACY','jsmith','C');
```

ユーザーmdaleは、PII区分を含むより機密性の高いラベル、Sensitiveを付与されます。ユーザーjsmithは、機密性の低いConfidentialラベルを取得します。

親トピック: チュートリアル: Oracle Database VaultとOracle Label Securityの統合

11.4.4.4 ステップ3: OLS認可を制御するためのOracle Database Vaultルールの作成

Oracle Label Securityポリシーの作成後、これと連携するDatabase Vaultルールを作成できます。

1. Database Vault所有者としてSQL*Plusに接続します。

たとえば:

```
CONNECT sec_admin_owen -- Or, CONNECT leo_dvowner@hrpdb
Enter password: password
```

2. 次のルール・セットを作成します。

```
EXEC DBMS_MACADM.CREATE_RULE_SET('PII Rule Set', 'Protect PII data from
privileged users','Y',1,0,2,NULL,NULL,0,NULL);
```

3. PII Rule Setにルールを作成します。

```
EXEC DBMS_MACADM.CREATE_RULE('Check OLS Factor',
'dominates(sa_utl.numeric_label(''PRIVACY''),
char_to_label(''PRIVACY'',''S:PII'')) = ''1''');
```

この例のように、二重引用符ではなく、必ず一重引用符を使用してください。

4. Check OLS FactorルールをPII Rule Setに追加します。

```
EXEC DBMS_MACADM.ADD_RULE_TO_RULE_SET('PII Rule Set', 'Check OLS Factor');
```

親トピック: チュートリアル: Oracle Database VaultとOracle Label Securityの統合

11.4.4.5 ステップ4: ルール・セットを使用するためのALTER SYSTEMコマンド・ルールの更新

ルール・セットを使用する前に、デフォルト・コマンド・ルールであるALTER SYSTEMコマンド・ルールを更新する必要があります。

1. Database Vault所有者として、ALTER SYSTEMコマンド・ルールの現行値を確認します。このコマンド・ルールは、Oracle Database Vaultインストール時のデフォルト・コマンド・ルールの1つです。

   ```
   SELECT * FROM DBA_DV_COMMAND_RULE WHERE COMMAND = 'ALTER SYSTEM';
   ```

2. 後で元の値に戻せるように、これらの設定をノートにとります。

   デフォルトのインストールでは、「ALTER SYSTEM」コマンド・ルールは「システム・パラメータのファイングレイン・コントロールを許可」ルール・セットを使用し、有効になっています。

3. PIIルール・セットに関連付けられる「ALTER SYSTEM」コマンド・ルールを更新します。

   ```
   EXEC DBMS_MACADM.UPDATE_COMMAND_RULE('ALTER SYSTEM', 'PII Rule Set', '%', '%',
   'Y');
   ```

   このコマンドでは、PII Rule SetをALTER SYSTEMコマンド・ルールに追加し、すべてのオブジェクト所有者およびオブジェクト名に適用し、コマンド・ルールを有効にします。

親トピック: チュートリアル: Oracle Database VaultとOracle Label Securityの統合

11.4.4.6 ステップ5: 認可のテスト

すべてのコンポーネントの準備ができたら、認可をテストする準備ができます。

1. ユーザーmdaleとしてSQL*Plusにログインします。

   ```
   CONNECT mdale -- Or, CONNECT mdale@hrpdb
   Enter password: password
   ```

2. AUDIT_TRAIL初期化パラメータの現行設定を確認します。

   ```
   SHOW PARAMETER AUDIT_TRAIL
   NAME                                 TYPE        VALUE
   ------------------------------------ ----------- ----------------------
   audit_trail                          string      DB
   ```

   後で元の設定に戻せるように、これらの設定をノートにとります。

3. ユーザーmdaleとして、ALTER SYSTEM文を使用し、CPU_COUNTパラメータを変更します。

   ```
   ALTER SYSTEM SET CPU_COUNT = 4;
   System altered.
   ```

   ユーザーmdaleはPII区分を含むSensitiveラベルを割り当てられたので、ALTER SYSTEM文を使用して、AUDIT_TRAILシステム・パラメータを変更できます。

4. CPU_COUNTパラメータを元の値に戻します。

   たとえば:

   ```
   ALTER SYSTEM SET CPU_COUNT = 2;
   ```

5. ユーザーjsmithとしてログインし、同じALTER SYSTEM文を発行します。

   ```
   CONNECT jsmith -- Or, CONNECT jsmith@hrpdb
   Enter password: password
   ALTER SYSTEM SET CPU_COUNT = 14;
   ```

次の出力が表示されます。

```
ERROR at line 1:
ORA-01031: insufficient privileges
```

ユーザーjsmithはConfidentialラベルしか割り当てられていないので、ALTER SYSTEM文を実行できません。

親トピック: チュートリアル: Oracle Database VaultとOracle Label Securityの統合

11.4.4.7 ステップ6: このチュートリアルのコンポーネントの削除

コンポーネントが不要になった場合、このチュートリアルで作成したコンポーネントを削除できます。

1. Oracle Label Security管理者として接続し、ラベル・ポリシーとそのコンポーネントを削除します。

```
CONNECT LBACSYS -- Or, CONNECT LBACSYS@hrpdb
Enter password: password
EXEC SA_SYSDBA.DROP_POLICY('PRIVACY', TRUE);
```

2. Oracle Database Vault所有者として接続し、次のコマンドを示した順序で発行し、ALTER SYSTEMコマンド・ルールを以前の設定に戻して、ルール・セットを削除します。

   たとえば:

```
CONNECT accts_admin_ace
Enter password: password
EXEC DBMS_MACADM.UPDATE_COMMAND_RULE('ALTER SYSTEM', 'Allow System
Parameters','%', '%', 'Y');
EXEC DBMS_MACADM.DELETE_RULE_FROM_RULE_SET('PII Rule Set', 'Check OLS Factor');
EXEC DBMS_MACADM.DELETE_RULE('Check OLS Factor');
EXEC DBMS_MACADM.DELETE_RULE_SET('PII Rule Set');
COMMIT;
```

3. Database Vaultアカウント・マネージャとして接続し、ユーザーmdaleとユーザーjsmithを削除します。

```
CONNECT accts_admin_ace -- Or, CONNECT accts_admin_ace@hrpdb
Enter password: password
DROP USER mdale;
DROP USER jsmith;
```

親トピック: チュートリアル: Oracle Database VaultとOracle Label Securityの統合

11.4.5 関連するレポートおよびデータ・ディクショナリ・ビュー

Oracle Database Vaultには、Oracle Database VaultとOracle Label Securityの統合に関する情報が示されるレポートおよびデータ・ディクショナリ・ビューが用意されています。

表11-1では、Oracle Database Vaultレポートを示します。これらのレポートの実行方法の詳細は、「Oracle Database Vaultレポート」を参照してください。

表11-1 Oracle Database Vault-Oracle Label Security統合に関連するレポート

レポート	説明
「ファクタ構成の問題」レポート	Oracle Label Security ポリシーが存在しないファクタが表示されます。

レポート	説明
「アイデンティティ構成の問題」レポート	無効なラベル・アイデンティティ(このアイデンティティの Oracle Label Security ラベルが削除されていて、すでに存在しない)が表示されます。
「セキュリティ・ポリシー除外」レポート	EXEMPT ACCESS POLICY システム権限が付与されているアカウントおよびロールが表示されます。この権限を持つアカウントは、すべての Virtual Private Database のポリシー・フィルタと、Oracle Virtual Private Database を間接的に使用する Oracle Label Security ポリシーを無視できます。

表11-2に、Oracle Database Vaultで使用される既存のOracle Label Securityポリシーに関する情報を提供するデータ・ディクショナリ・ビューを示します。

表11-2 Oracle Label Securityに使用されるデータ・ディクショナリ・ビュー

データ・ディクショナリ・ビュー	説明
DBA_DV_MAC_POLICY ビュー	定義されている Oracle Label Security ポリシーが表示されます。
DBA_DV_MAC_POLICY_FACTOR ビュー	Oracle Label Security ポリシーに関連付けられているファクタが表示されます。
DBA_DV_POLICY_LABEL ビュー	各ポリシーの DBA_DV_IDENTITY ビューの各ファクタ識別子に対する Oracle Label Security ラベルが表示されます。

親トピック: Oracle Database VaultとOracle Label Securityの統合

11.5 Oracle Database VaultとOracle Data Guardの統合

Oracle Database VaultとOracle Data Guardの統合では、まずプライマリ・データベースを構成し、次にスタンバイ・データベースを構成します。

- ステップ1: プライマリ・データベースの構成
  DGMGRLユーティリティを実行し、Database Vaultを構成して有効化し、次にALTER SYSTEM文を実行して、プライマリ・データベースを構成する必要があります。
- ステップ2: スタンバイ・データベースの構成
  スタンバイ・データベースに使用されるデータベース内でスタンバイ・データベース構成を実行できます。
- Oracle Database VaultとOracle Active Data Guardの統合後の監査の動作
  Oracle Database VaultをOracle Active Data Guardと統合した後は、監査の構成内容によって、監査レコードがどのように生成されるかが変わります。
- Oracle Data Guard環境でのOracle Database Vaultの無効化
  Oracle Data Guard環境でOracle Database Vaultを無効にする場合は、まずプライマリ・データベースで、次に

スタンバイ・データベースでプロシージャを実行する必要があります。

親トピック: Oracle Database Vaultとその他の Oracle製品の統合

11.5.1 ステップ1: プライマリ・データベースの構成

DGMGRLユーティリティを実行し、Database Vaultを構成して有効化し、次にALTER SYSTEM文を実行して、プライマリ・データベースを構成する必要があります。

1. LinuxおよびUNIXシステムの場合、Oracle Database Vaultをインストールするノードのデータベースに/etc/oratabエントリがあることを確認します。
2. Data Guard Brokerを使用している場合は、コマンド・プロンプトから次のように構成を無効化します。

```
dgmgrl sys
Enter password: password
DGMGRL> disable configuration;
```

3. プライマリ・サーバーでOracle Database Vaultを構成して有効化します。

   Oracle Database Vaultは、デフォルトで、Oracle Databaseの一部としてインストールされます。この登録のステータスは、DBA_DV_STATUSデータ・ディクショナリ・ビューを問い合せることで確認できます。

4. SYSDBA管理権限を持つユーザーSYSとして、データベース・インスタンスにログインします。

```
sqlplus sys as sysdba
Enter password: password
```

5. 次のALTER SYSTEM文を実行します。

```
ALTER SYSTEM SET AUDIT_SYS_OPERATIONS=TRUE SCOPE=SPFILE;
ALTER SYSTEM SET OS_ROLES=FALSE SCOPE=SPFILE;
ALTER SYSTEM SET RECYCLEBIN='OFF' SCOPE=SPFILE;
ALTER SYSTEM SET REMOTE_LOGIN_PASSWORDFILE='EXCLUSIVE' SCOPE=SPFILE;
ALTER SYSTEM SET SQL92_SECURITY=TRUE SCOPE=SPFILE;
ALTER SYSTEM SET REMOTE_OS_AUTHENT=FALSE SCOPE=SPFILE;
ALTER SYSTEM SET REMOTE_OS_ROLES=FALSE SCOPE=SPFILE;
```

6. 各データベース・インスタンスでALTER SYSTEM文を実行して、ステップ5に示すとおりにパラメータを設定します。
7. 各データベース・インスタンスを再起動します。

```
CONNECT SYS AS SYSOPER
Enter password: password
SHUTDOWN IMMEDIATE
STARTUP
```

関連トピック

- Oracle Database Vaultの開始

親トピック: Oracle Database VaultとOracle Data Guardの統合

11.5.2 ステップ2: スタンバイ・データベースの構成

スタンバイ・データベースに使用されるデータベース内でスタンバイ・データベース構成を実行できます。

1. SYSDBA管理権限を持つユーザーSYSとして、データベース・インスタンスにログインします。

```
sqlplus sys as sysdba
Enter password: password
```

2. マルチテナント環境で、適切なPDBに接続します。

   たとえば:

   ```
   CONNECT bea_dvacctmgr@hrpdb
   Enter password: password
   ```

   使用可能なPDBを見つけるには、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。

3. スタンバイ・データベース・インスタンスをマウントします。

   ```
   ALTER DATABASE MOUNT STANDBY DATABASE;
   ```

4. 次のALTER SYSTEM文を実行します。

   ```
   ALTER SYSTEM SET AUDIT_SYS_OPERATIONS=TRUE SCOPE=SPFILE;
   ALTER SYSTEM SET OS_ROLES=FALSE SCOPE=SPFILE;
   ALTER SYSTEM SET RECYCLEBIN='OFF' SCOPE=SPFILE;
   ALTER SYSTEM SET REMOTE_LOGIN_PASSWORDFILE='EXCLUSIVE' SCOPE=SPFILE;
   ALTER SYSTEM SET SQL92_SECURITY=TRUE SCOPE=SPFILE;
   ALTER SYSTEM SET REMOTE_OS_AUTHENT=FALSE SCOPE=SPFILE;
   ALTER SYSTEM SET REMOTE_OS_ROLES=FALSE SCOPE=SPFILE;
   ```

5. データベース・インスタンスを再起動またはマウントします。

   たとえば:

   ```
   SHUTDOWN IMMEDIATE
   STARTUP
   ```

6. 次のスタンバイ・インスタンスをマウントします。
7. 管理リカバリを次のように再起動します。

   ```
   ALTER DATABASE RECOVER MANAGED STANDBY DATABASE;
   ```

8. Data Guard Brokerを使用している場合は、コマンド・ラインから構成を再有効化します。

   ```
   dgmgrl sys
   Enter password: password
   DGMGRL> enable configuration;
   ```

   このコマンドにより、変更内容がプライマリ・データベースのOracle Database Vaultインストールで作成された物理スタンバイ・データベースに適用されます。

9. 各物理スタンバイ・データベースで、物理スタンバイ・インストール・プロセスを繰り返します。たとえば、物理スタンバイ・データベースが3つある場合は、各スタンバイ・データベースでこれらの手順を実行します。

親トピック: Oracle Database VaultとOracle Data Guardの統合

11.5.3 Oracle Database VaultとOracle Active Data Guardの統合後の監査の動作

Oracle Database VaultをOracle Active Data Guardと統合した後は、監査の構成内容によって、監査レコードがどのように生成されるかが変わります。

読取り専用の問合せ用にActive Data Guardフィジカル・スタンバイ・データベースを使用する場合は、混合モードではなく純粋な統合監査を使用する必要があります。混合モードを使用すると、Oracle Database Vault監査レコードを生成する

Active Data Guardフィジカル・スタンバイにおける問合せがブロックされます。Oracle Database Vaultでは、従来のDatabase Vault監査表(DVSYS.AUDIT_TRAILS$)に書き込むことができません。統合監査により、Database Vaultの監査データがOracle Active Data Guardフィジカル・スタンバイ・データベースのオペレーティング・システムのログ・ファイルに確実に書き込まれます。これらのログ・ファイルのデータは統合監査証跡に移動できます。Database Vaultのアクティビティを監査するには、Database Vaultの従来の監査設定が統合監査に適用されないため、統合監査ポリシーを作成する必要があることに注意してください。

親トピック: Oracle Database VaultとOracle Data Guardの統合

11.5.4 Oracle Data Guard環境でのOracle Database Vaultの無効化

Oracle Data Guard環境でOracle Database Vaultを無効にする場合は、まずプライマリ・データベースで、次にスタンバイ・データベースでプロシージャを実行する必要があります。

次の順序で、プライマリ・データベースとスタンバイ・データベースでOracle Database Vaultの無効化を実行します。

1. プライマリ・データベースでOracle Database Vaultを無効にします。
2. セカンダリ・データベースでOracle Database Vaultを無効にします。
3. プライマリ・データベースを再起動します。
4. 各スタンバイ・データベースを再起動します。

関連トピック

- ステップ1: Oracle Database Vaultの無効化

親トピック: Oracle Database VaultとOracle Data Guardの統合

11.6 Oracle Database Configuration Assistantを使用したOracle Internet Directoryの登録

Oracle Database Vault対応データベースでOracle Internet Directoryを使用できます。

ただし、Oracle Database Configuration Assistant (DBCA)を使用して、Oracle Internet Directory (OID)を登録する場合、まずOracle Database Vaultを無効にする必要があります。

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultとその他の Oracle製品の統合

12 Oracle Database Vault環境でのDBA操作

データベース管理者は、Oracle Data Pumpなどの製品とのDatabase Vaultの使用など、Oracle Database Vault環境で操作を実行できます。

- Oracle Database VaultでのDDL操作の実行
  Oracle Database Vaultでのデータ定義言語(DDL)操作は、スキーマの所有権やパッチ・アップグレードなどの状況による影響を受ける場合があります。
- Oracle Database VaultのOracle Enterprise Managerとの使用
  Oracle Database Vault管理者は、他のデータベースへのポリシーの伝播など、Oracle Enterprise Manager Cloud Controlでタスクを実行できます。
- Oracle Database VaultでのOracle Data Pumpの使用
  データベース管理者は、Oracle Data PumpユーザーにDatabase Vault環境で作業する認可を付与します。
- Oracle Database VaultでのOracle Schedulerの使用
  データベース・ジョブのスケジュールを担当するユーザーは、Oracle Database Vault固有の認可を有している必要があります。
- Oracle Database Vaultでの情報ライフサイクル管理の使用
  Oracle Database Vault対応データベースで情報ライフサイクル管理操作を実行するユーザーは、これらの操作を実行するために認可を受けている必要があります。
- Oracle Database VaultでのOracle Database Replayの使用
  データベース管理者は、Oracle Database ReplayユーザーにDatabase Vault環境で作業する認可を付与できます。
- Oracle Database Vaultでのプリプロセッサ・プログラムの実行
  外部表からプリプロセッサ・プログラムを実行するユーザーには、Oracle Database Vault固有の認可が必要です。
- 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
  インフラストラクチャ・データベース管理者などのCDBルート共通ユーザーでPDBアクセスを制御できます。
- Oracle Recovery ManagerとOracle Database Vault
  Oracle Database Vault環境ではRecovery Manager (RMAN)を使用できます。
- Oracle Database VaultでXStreamを使用するための権限
  Oracle Database Vault環境でXStreamを使用する場合、適切な権限が必要です。
- Oracle Database VaultでOracle GoldenGateを使用するための権限
  Oracle Database Vault環境でOracle GoldenGateを使用する場合、適切な権限が必要です。
- Oracle Database Vault環境でのデータ・マスキングの使用
  Oracle Database Vault環境でデータ・マスキングを実行するには、正しい認可が必要です。
- 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。
- Oracle Database Vault環境でのORADEBUGユーティリティの使用
  ORADEBUGユーティリティは、主にOracleサポートがOracle Databaseで生じる問題を診断する場合に使用します。
- Oracle Database Vault環境でのパッチ操作の実行
  ユーザーSYSがOracle Database Vault対応データベースでパッチ操作を実行するには、DV_PATCH_ADMINロールが必要です。

12.1 Oracle Database VaultでのDDL操作の実行

Oracle Database Vaultでのデータ定義言語(DDL)操作は、スキーマの所有権やパッチ・アップグレードなどの状況による影響を受ける場合があります。

- Oracle Database VaultでのDDL操作の実行に関する制限
  Oracle Database Vault構成によっては、DDL操作が制限されてOracle Database Vault環境でDDL認可が必要になる場合があります。
- DDL操作におけるDV_PATCH_ADMINロールの影響
  DV_PATCH_ADMINロールを付与されたオブジェクト所有者およびユーザーは、DDL認可要件から除外されます。

親トピック: Oracle Database Vault 環境でのDBA操作

12.1.1 Oracle Database VaultでのDDL操作の実行に関する制限

Oracle Database Vault構成によっては、DDL操作が制限されてOracle Database Vault環境でDDL認可が必要になる場合があります。

具体的には、次のいずれかの特性を持つスキーマでDDL操作を実行するには、DDL認可が必要です。

- スキーマが、有効なレルムによって保護されるオブジェクトの所有者である。
- スキーマが、有効なレルムに直接またはロールを介して認可される。
- スキーマにオブジェクト権限が直接、または有効なレルムによって保護されているオブジェクトのロールを介して付与される。
- スキーマにOracle Database Vaultのロールが直接またはロールを介して付与される。

DV_PATCH_ADMINロールを付与されたオブジェクト所有者およびユーザーは、DDL認可要件から除外されます。DBMS_MACADM.AUTHORIZE_DDLプロシージャを使用して、特定のスキーマに対してDDL操作を実行するユーザーを認可できます。ただし、DDL認可では、権限受領者がレルムで保護されたオブジェクトまたはスキーマに対してDDL操作を実行できるわけではありません。このような操作を有効化するには、レルムに対してユーザーを認可する必要があります。この認可を与えられているユーザーについて情報を確認するには、DBA_DV_DDL_AUTHデータ・ディクショナリ・ビューに問い合せます。

Oracle Database VaultがOracle Database 21cより古い以前のリリースからアップグレードされた場合、デフォルトのDDL認可(%, %)が存在する可能性があり、これにより、ユーザーは明示的なDDL認可なしで任意のスキーマに対してDDL操作を実行できます。セキュリティを強化するために、Oracleでは、DBMS_MACADM.UNAUTHORIZE_DDL('%', '%')を実行してデフォルトのDDL認可を削除し、DDL操作を実行する必要があるユーザーにのみ必要なDDL認可を付与することをお薦めします。

関連トピック

- AUTHORIZE_DDLプロシージャ

親トピック: Oracle Database VaultでのDDL操作の実行

12.1.2 DDL操作におけるDV_PATCH_ADMINロールの影響

DV_PATCH_ADMINロールを付与されたオブジェクト所有者およびユーザーは、DDL認可要件から除外されます。

DBMS_MACADM.AUTHORIZE_DDLプロシージャを使用して、特定のスキーマに対してDDL操作を実行するユーザーを認可できます。ただし、DDL認可では、権限受領者がレルムで保護されたオブジェクトまたはスキーマに対してDDL操作を実行することはできません。このような操作を許可するには、レルムに対してユーザーを認可する必要があります。この認可を与えられてい

るユーザーについて情報を確認するには、DBA_DV_DDL_AUTHデータ・ディクショナリ・ビューに問い合せます。

関連トピック

- AUTHORIZE_DDLプロシージャ

親トピック: Oracle Database VaultでのDDL操作の実行

12.2 Oracle Database VaultのOracle Enterprise Managerとの使用

Oracle Database Vault管理者は、他のデータベースへのポリシーの伝播など、Oracle Enterprise Manager Cloud Controlでタスクを実行できます。

- 他のデータベースへのOracle Database Vault構成の伝播
  Database Vault構成(レルム構成など)を、Database Vaultで保護された他のデータベースに伝播できます。
- Oracle Database Vaultポリシーに対するEnterprise Manager Cloud Controlアラート
  Oracle Database Vaultアラートを表示するには、DV_OWNER、DV_ADMINまたはDV_SECANALYSTロールが付与されている必要があります。
- Enterprise Manager Cloud ControlにおけるOracle Database Vault固有レポート
  Database Vaultホーム・ページから、違反に関する情報を確認できます。

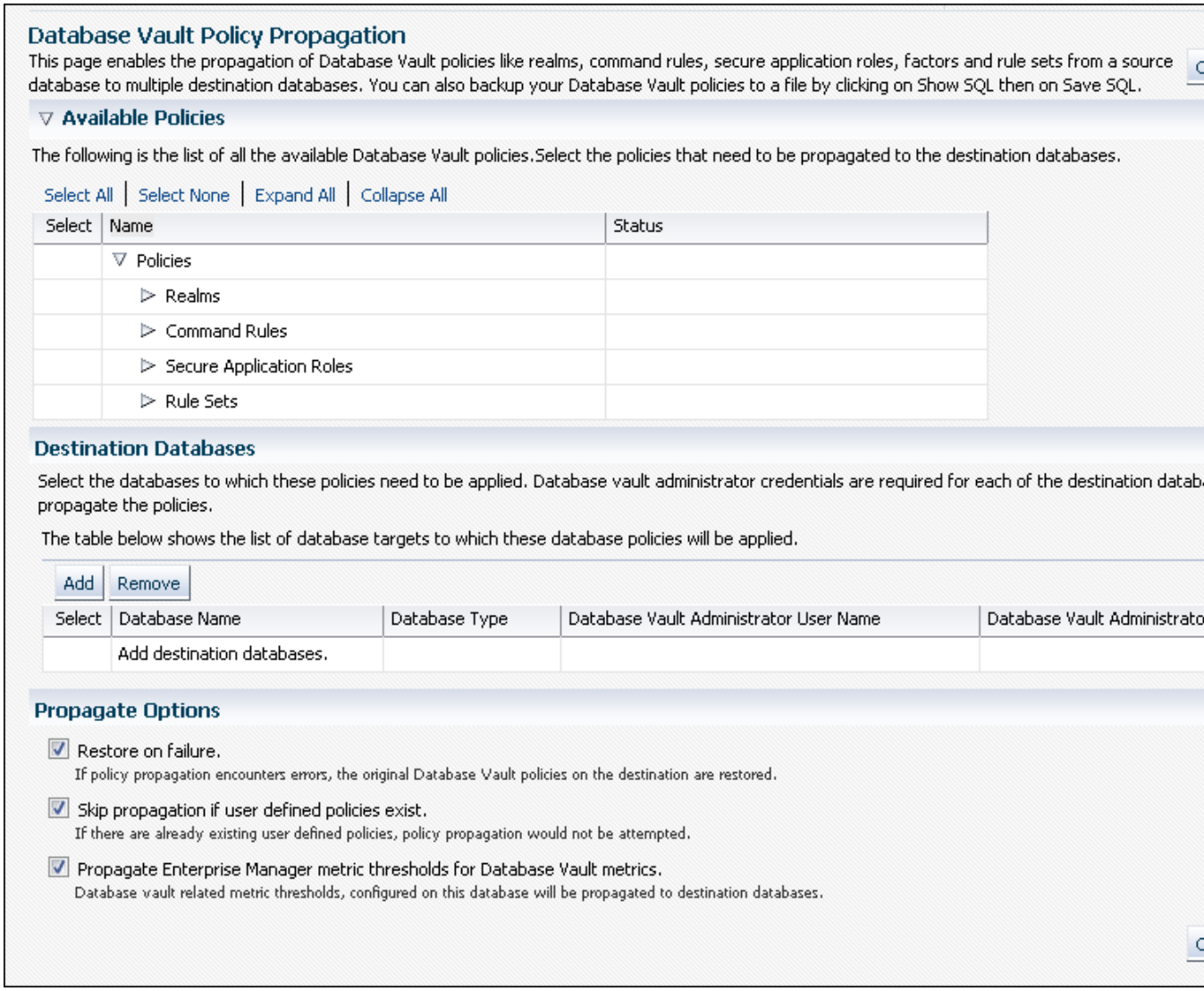
親トピック: Oracle Database Vault 環境でのDBA操作

12.2.1 「他のデータベースへのOracle Database Vault構成の伝播」

Database Vault構成(レルム構成など)を、Database Vaultで保護された他のデータベースに伝播できます。

1. DV_OWNERまたはDV_ADMINロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「Database Vault」ホームページの「Database Vaultポリシー伝播」で、「Database Vaultポリシー伝播」を選択します。

   「ポリシー伝播」サブページの「使用可能なポリシー」領域に、現在のデータベースのために作成されたOracle Database Vault構成(つまり、レルム、コマンド・ルール、ルール・セットおよびセキュア・アプリケーション・ロールのために作成された構成)のサマリーが表示されます。Oracle Database release 12c (12.2)で導入されたOracle Database Vaultポリシーは表示されません。ここから、これらの構成を他のデータベースに伝播できます。
3. 「使用可能なポリシー」で、他のデータベースに伝播する各構成を選択します。



4. 「宛先データベース」で「追加」ボタンをクリックします。
5. 「検索と選択: Database Vaultに対応した宛先データベース」で宛先データベースを検索し、構成の伝播先となる各データベースを選択します。「選択」ボタンをクリックします。
6. 「宛先データベース」で次の処理を行います。
    a. 「宛先データベースに資格証明を適用」で、伝播する構成を含むDatabase Vaultデータベースの管理者のユーザー名とパスワードを入力します。

        この機能により、Database Vault管理者のユーザー名とパスワードが、選択したすべての宛先データベースに適用されます。

    b. 構成の伝播先となる各データベースを選択します。
    c. 各データベースのDatabase Vault管理者のユーザー名とパスワードを入力します。
    d. 「適用」ボタンをクリックします。
7. 「伝播オプション」ページで、次のオプションのオン/オフを選択します。

    シードされたレルム、コマンド・ルール、ルール・セットなどに加えられた変更は、宛先データベースに伝播されません。カスタム作成されたデータのみが伝播されます。

    a. 失敗時にリストアします。: 伝播操作でエラーが発生した場合に、伝播がロールバックされます。つまり、宛先データベースの元のポリシーがリストアされます。このオプションを選択しない場合、宛先データベースでポリシー

の伝播が続けられ、エラーは無視されます。

b. ユーザー定義ポリシーが存在する場合は、伝播をスキップします。: 宛先データベースにユーザー定義構成がすでにある場合、伝播操作は試行されません。このオプションを選択しない場合、ユーザー定義ポリシーが宛先データベースにあるかどうかに関係なく、既存の構成はすべてクリアされ、ソース・データベースからの構成が宛先データベースに適用されます。

c. Database VaultメトリックのEnterprise Managerメトリックしきい値を伝播します。: ソース・データベースにOracle Database Vaultメトリックしきい値が設定されている場合、これらのしきい値も宛先データベースに伝播されます。このオプションを選択しない場合、構成のみが伝播され、Oracle Database Vaultしきい値は伝播されません。

8. 「OK」ボタンをクリックします。
9. 「確認」ウィンドウで「OK」をクリックします。

   成功または失敗を示すメッセージが表示されます。伝播が成功した場合、構成は宛先データベースでただちにアクティブになります。

親トピック: Oracle Database VaultのOracle Enterprise Managerとの使用

12.2.2 Oracle Database Vaultポリシーに対するEnterprise Manager Cloud Controlアラート

Oracle Database Vaultアラートを表示するには、DV_OWNER、DV_ADMINまたはDV_SECANALYSTロールが付与されている必要があります。

アラートは次のとおりです。

- Database Vaultレルム違反未遂。このアラートにより、Oracle Database Vaultセキュリティ分析者(DV_SECANALYSTロール)はDatabase Vaultデータベースでの違反試行を監視できます。アラートの影響を受けるレルムを選択し、エラー・コードを使用して様々な試行タイプに基づいてレルムをフィルタリングできます。このメトリックは、メトリックおよびポリシーの設定ページで有効化できます。デフォルトでは、レルム違反未遂は24時間ごとに収集されます。
- Database Vaultコマンド・ルール違反未遂。このアラートの機能は、対象がコマンド・ルール違反であることを除き、Database Vaultレルム違反未遂と同じです。
- Database Vaultレルム構成の問題。このメトリックは、レルムの構成を追跡し、構成が正しくない場合にアラートを生成します。このメトリックは、Oracle Database Vaultのインストール時に有効化され、デフォルトで1時間ごとにデータを収集します。
- Database Vaultコマンド・ルール構成の問題。このアラートの機能は、対象がコマンド・ルールに対する構成変更であることを除き、Database Vaultレルム構成の問題と同じです。
- Database Vaultポリシー変更。このメトリックでは、Database Vaultポリシー(レルムやコマンド・ルールに対するポリシー)に変更があると、アラートが生成されます。詳細なポリシー変更レポートが提供されます。

親トピック: Oracle Database VaultのOracle Enterprise Managerとの使用

12.2.3 Enterprise Manager Cloud ControlにおけるOracle Database Vault固有レポート

Database Vaultホーム・ページから、違反に関する情報を確認できます。

これらの違反は、次のとおりです。

- レルムおよびコマンド・ルール違反未遂トップ5
- データベース・ユーザーおよびクライアント・ホストによる違反未遂トップ5
- 違反未遂の詳細分析の時系列グラフィック・レポート

Database Vaultレポートの完全なアクセス権を持つには、DV_OWNER、DV_ADMINまたはDV_SECANALYSTロールが付与されたユーザーとしてDatabase Vault Administratorにログインする必要があります。

関連トピック

- Oracle Database Vaultレポート

親トピック: Oracle Database VaultのOracle Enterprise Managerとの使用

12.3 Oracle Database VaultでのOracle Data Pumpの使用

データベース管理者は、Oracle Data PumpユーザーにDatabase Vault環境で作業する認可を付与します。

- Oracle Database VaultでのOracle Data Pumpの使用について
  Database Vault環境でOracle Data Pumpを使用しているデータベース管理者には、データのエクスポートおよびインポートのためのDatabase Vault固有の認可が必要です。
- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可
  Database Vault環境でOracle Data Pumpエクスポート操作およびインポート操作を実行する管理者に、様々なタイプの認可を使用できます。
- ユーザーまたはロールへのData Pumpのトランスポータブル・エクスポート操作およびトランスポータブル・インポート操作の認可
  Oracle Data Pumpのトランスポータブル操作を実行するユーザーに、直接またはロールを通じて様々な認可レベルを付与できます。
- Database Vault環境でのデータのエクスポートまたはインポートのガイドライン
  Oracle Data Pumpデータベース管理者に適切な認可を付与すると、必要なエクスポートまたはインポート操作を実行できるようになります。

親トピック: Oracle Database Vault 環境でのDBA操作

12.3.1 Oracle Database VaultでのOracle Data Pumpの使用について

Database Vault環境でOracle Data Pumpを使用しているデータベース管理者には、データのエクスポートおよびインポートのためのDatabase Vault固有の認可が必要です。

このタイプのユーザーは、Oracle Data Pumpの標準の権限に加えて、Database Vault権限を持っている必要があります。これらのユーザーがOracle Data Pumpのトランスポータブル表領域操作を実行する場合、特殊な認可が必要です。DBA_DV_DATAPUMP_AUTHデータ・ディクショナリ・ビューを問い合せることで、Oracle Database Vault環境でのData Pumpの使用に関するユーザーの認可を確認できます。この認可は、個々のユーザーまたはデータベース・ロールに付与できます。

関連項目:

- Oracle Data Pumpの詳細は、『Oracle Databaseユーティリティ』を参照してください。
- トランスポータブル表領域の詳細は、『Oracle Database管理者ガイド』を参照してください。
- DBA_DV_DATAPUMP_AUTHビュー

親トピック: Oracle Database VaultでのOracle Data Pumpの使用

12.3.2 ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

Database Vault環境でOracle Data Pumpエクスポート操作およびインポート操作を実行する管理者に、様々なタイプの認可を使用できます。

- Oracle Data Pumpの通常操作のユーザーまたはロールへの認可について
  Oracle Data Pump認可を持つユーザーは、Database Vault環境で通常のOracle Data Pump操作を実行できます。
- Oracle Data Pumpの通常操作に対するDatabase Vault認可のレベル
  Oracle Database Vaultでは、Database Vault環境でのOracle Data Pumpの通常操作に必要な認可にいくつかのレベルがあります。
- Database VaultにおけるOracle Data Pumpの通常操作をユーザーまたはロールに認可
  データベース管理者またはロールに、Oracle Database Vault環境で通常操作にData Pumpを使用する認可を付与できます。
- ユーザーまたはロールからのOracle Data Pump認可の取消し
  通常操作のためにOracle Data Pumpを使用するデータベース管理者またはロールの認可を取り消すことができます。

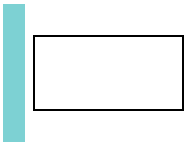
親トピック: Oracle Database VaultでのOracle Data Pumpの使用

12.3.2.1 Oracle Data Pumpの通常操作のユーザーまたはロールへの認可について

Oracle Data Pump認可を持つユーザーは、Database Vaul環境で通常のOracle Data Pump操作を実行できます。

次のタイプのOracle Data Pump認可を実行できます。

- 保護されたスキーマおよびオブジェクトをインポートできるようにユーザーまたはロールを認可します
- ユーザーまたはロールに対して、インポート操作中に行われるアクティビティ(ユーザーの作成、Oracle Database Vaultで保護されたロールおよびシステム権限の付与、特定のOracle Databaseロールの付与、Oracle Databaseシステム権限の付与)の実行を認可します



ノート:

完全レベル Data Pump 認可では、トランスポータブル・エクスポートおよびインポート操作も実行できます。

関連トピック

- ユーザーまたはロールへのData Pumpのトランスポータブル・エクスポート操作およびトランスポータブル・インポート操作の認可

親トピック: ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

12.3.2.2 Oracle Data Pumpの通常操作に対するDatabase Vault権限のレベル

Oracle Database Vaultでは、Database Vault環境でのOracle Data Pumpの通常操作に必要な認可にいくつかのレベルがあります。

表12-1に、これらのレベルを示します。

表12-1 Oracle Data Pumpの通常操作のための認可のレベル

シナリオ	必要な認可
データベース管理者は、データを別のスキーマにインポートします。	このユーザー(またはロール)には、BECOME USER システム権限とIMP_FULL_DATABASE ロールを付与する必要があります。脚注 1 ユーザーに付与されている権限を確認するには、USER_SYS_PRIVS データ・ディクショナリ・ビューを問い合せます。
データベース管理者は、Database Vault 保護なしのスキーマでデータをエクスポートまたはインポートします。	このユーザー(またはロール)には、標準の Oracle Data Pump 権限のみを付与する必要があります。これは、EXP_FULL_DATABASE ロールとIMP_FULL_DATABASE ロールです。ユーザーがデータをインポートする場合は、このユーザーに BECOME USER システム権限を付与します。
データベース管理者は、保護スキーマでデータをエクスポートまたはインポートします。	EXP_FULL_DATABASE ロールと IMP_FULL_DATABASE ロールの他に、DBMS_MACADM.AUTHORIZE_DATAPUMP_USER プロシージャを使用してDatabase Vault 固有の権限もこのユーザー(またはロール)に付与する必要があります。この認可は、EXPDP と IMPDP の両方のユーティリティに適用されます。後から、DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER プロシージャを使用してこの権限を取り消すことができます。 ユーザーがデータをインポートする場合は、このユーザーに BECOME USER システム権限も付与します。
データベース管理者は、データベース全体の内容をエクスポートまたはインポートします。	EXP_FULL_DATABASE ロールと IMP_FULL_DATABASE ロール、およびDBMS_MACADM.AUTHORIZE_DATAPUMP_USER プロシージャで付与される権限の他に、このユーザー(またはロール)には DV_OWNER ロールも付与する必要があります。ユーザーがデータをインポートする場合は、このユーザーに BECOME USER システム権限を付与します。

脚注1

デフォルトではBECOME USER権限はIMP_FULL_DATABASEロールに含まれますが、Oracle Database Vault環境ではこの権限は取り消されます。

親トピック: ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

12.3.2.3 Database VaultにおけるOracle Data Pumpの通常操作をユーザーまたはロールに認可

データベース管理者またはロールに、Oracle Database Vault環境で通常操作にData Pumpを使用することを認可できます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。
2. 認可を付与するユーザーまたはロールに、Oracle Data Pumpの使用に必要なEXP_FULL_DATABASEおよびIMP_FULL_DATABASEロールが付与されていることを確認します。

```
SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS WHERE GRANTED_ROLE LIKE
'%FULL%';
```

3. 保護されたスキーマおよびオブジェクトをインポートするために、このユーザーまたはロールにOracle Database Vaultの認可を付与します。

   たとえば、Data PumpユーザーDP_MGRに、データベース表EMPLOYEESのオブジェクトをエクスポートおよびインポートする権限を付与するには、次のように入力します。

```
EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('DP_MGR', 'HR', 'EMPLOYEES');
```

   DP_MGRのアクティビティを特定のスキーマに制限するには、次のプロシージャを入力します。

```
EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('DP_MGR', 'HR');
```

   DP_MGR_ROLEロールを付与されているユーザーに、データベース全体のオブジェクトのエクスポートおよびインポートを認可するには、次のように入力します。

```
EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('DP_MGR_ROLE');
```

   DBMS_MACADM.AUTHORIZE_DATAPUMP_USERプロシージャの実行後、DBA_DV_DATAPUMP_AUTHデータ・ディクショナリ・ビューに問い合せてユーザーまたはロールの認可を確認できます。

   ユーザーまたはロールに完全な認可を付与した(schema、object、typeおよびactionパラメータに%を使用した)場合は、次のステップを省略できます。ただし、認可が特定のスキーマのみの場合(たとえば、schemaがHRに設定され、残りのパラメータが引き続き%に設定されている場合)は、次のステップを実行する必要があります。

4. 必要に応じて、インポート操作中に次のアクティビティを実行するようにユーザーまたはロールに認可を付与します。
   a. インポート中にユーザーを作成します。例:

```
EXEC DBMS_MACADM.AUTH_DATAPUMP_CREATE_USER('DP_MGR');
```

   b. インポート中にOracle Database Vaultで保護されたロールおよびシステム権限を付与します。例:

```
EXEC DBMS_MACADM.AUTH_DATAPUMP_GRANT('DP_MGR');
```

   c. インポート中に特定のロールを付与します。例:

```
EXEC DBMS_MACADM.AUTH_DATAPUMP_GRANT_ROLE('DP_MGR', 'DBA');
```

   d. インポート中にシステム権限を付与します。例:

```
EXEC DBMS_MACADM.AUTH_DATAPUMP_GRANT_SYSPRIV('DP_MGR');
```

5. ユーザーまたはロールがデータベース全体をエクスポートする必要がある場合は、DV_OWNERロールを付与します。
   たとえば、ロールの場合は次のように入力します。

```
GRANT DV_OWNER TO DP_MGR_ROLE;
```

関連トピック

- AUTHORIZE_DATAPUMP_USERプロシージャ
- DBA_DV_DATAPUMP_AUTHビュー

親トピック: ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

12.3.2.4 ユーザーまたはロールからのOracle Data Pump認可の取消し

通常操作のためにOracle Data Pumpを使用するデータベース管理者またはロールの認可を取り消すことができます。

1. ユーザーまたはロールにDV_OWNERロールが付与されている場合は、オプションでDV_OWNERロールを取り消します。

   ```
   REVOKE DV_OWNER FROM DP_MGR;
   ```

2. DBA_DV_DATAPUMP_AUTHデータ・ディクショナリ・ビューに問い合せて、Oracle Data Pumpの認可が付与されているユーザーまたはロールを確認します。

   ```
   SELECT GRANTEE, SCHEMA, OBJECT FROM DBA_DV_DATAPUMP_AUTH;
   ```

3. 前のステップで収集した情報を使用して、DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USERコマンドを作成します。

   たとえば:

   ```
   EXEC DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER('DP_MGR', 'HR', 'EMPLOYEES');
   ```

   この権限取消しが、元の権限付与アクションを補完するものであることを確認します。すなわち、最初にDP_MGRにデータベース全体に対する権限を付与した場合、次のコマンドは機能しません。

   ```
   EXEC DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER('DP_MGR', 'HR');
   EXEC DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER('DP_MGR', 'HR', 'EMPLOYEES');
   ```

4. インポート操作中にユーザーの作成またはその他のアクティビティを実行する認可をユーザーまたはロールに付与した場合は、これらを取り消します。
   たとえば:

   ```
   EXEC DBMS_MACADM.UNAUTH_DATAPUMP_CREATE_USER('DP_MGR');
   EXEC DBMS_MACADM.UNAUTH_DATAPUMP_GRANT('DP_MGR');
   EXEC DBMS_MACADM.UNAUTH_DATAPUMP_GRANT_ROLE('DP_MGR', 'DBA');
   EXEC DBMS_MACADM.UNAUTH_DATAPUMP_GRANT_SYSPRIV('DP_MGR');
   ```

   ユーザーの認可を確認するには、DBA_DV_DATAPUMP_AUTHデータ・ディクショナリ・ビューを問い合せます。

関連トピック

- UNAUTHORIZE_DATAPUMP_USERプロシージャ
- DBA_DV_DATAPUMP_AUTHビュー

親トピック: ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

12.3.3 ユーザーまたはロールへのData Pumpのトランスポータブル・エクスポート操作およびトランスポータブル・インポート操作の認可

Oracle Data Pumpのトランスポータブル操作を実行する必要があるユーザーには、直接またはロールを通じて、様々な認可

レベルを付与できます。

- Oracle Data Pumpのトランスポータブル操作のユーザーへの認可について
  様々なレベルのトランスポータブル操作認可をユーザーに(直接またはロールを通じて)付与できます。
- Data Pumpのトランスポータブル操作に対するDatabase Vault権限のレベル
  Oracle Database Vaultでは、Database Vault環境でのエクスポートおよびインポート・トランスポータブル操作を実行する必要があるユーザーに必要な認可にいくつかのレベルがあります。
- Database VaultにおけるData Pumpのトランスポータブル操作をユーザーまたはロールに認可
  ユーザーまたはロールがDatabase Vault環境でOracle Data Pumpのトランスポータブル・エクスポート操作またはトランスポータブル・インポート操作を実行することを認可できます。
- トランスポータブル表領域の認可のユーザーまたはロールからの取消し
  Data Pumpを使用するデータベース管理者の認可を取り消すことができます。

親トピック: Oracle Database VaultでのOracle Data Pumpの使用

12.3.3.1 Oracle Data Pumpのトランスポータブル操作のユーザーへの認可について

様々なレベルのトランスポータブル操作の認可をユーザーに(直接またはロールを通じて)付与できます。

トランスポータブル・エクスポートとトランスポータブル・インポートの操作を実行する認可のみをユーザーに付与する場合は、タスクに基づいて、ユーザーまたはロールに適切な認可を付与する必要があります。

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: ユーザーまたはロールへのData Pumpのトランスポータブル・エクスポート操作およびトランスポータブル・インポート操作の認可

12.3.3.2 Data Pumpのトランスポータブル操作に対するDatabase Vault権限のレベル

Oracle Database Vaultでは、Database Vault環境でトランスポータブル・エクスポートとトランスポータブル・インポートの操作を実行する必要があるユーザーに必要な認可にいくつかのレベルがあります。

表12-2に、これらのレベルを示します。

表12-2 Oracle Data Pumpのトランスポータブル操作に対する権限のレベル

シナリオ	必要な認可
データベース管理者は、Database Vault 保護なしの表領域または表のトランスポータブル・エクスポートを実行します。	このユーザー(またはロール)には、標準の Oracle Data Pump 権限のみを付与する必要があります。これは、EXP_FULL_DATABASE ロールと IMP_FULL_DATABASE ロールです。
データベース管理者は、Database Vault 保護ありの表領域のトランスポータブル・エクスポートを実行します(たとえば、その表領域に存在する表オブジェクトのレルムやコマンドルール)。	EXP_FULL_DATABASE ロールと IMP_FULL_DATABASE ロールの他に、DBMS_MACADM.AUTHORIZE_TTS_USER プロシージャを使用して Database Vault 固有のトランスポータブル表領域の認可もこのユーザー(またはロール)に付与する必要があります。後から、DBMS_MACADM.UNAUTHORIZE_TTS_USER プロシージャを使用してこの権限を取り消すことができます。

シナリオ	必要な認可
	完全データベース・レベルの Oracle Data Pump 権限を付与されたユーザーも、(DBMS_MACADM.AUTHORIZE_DATAPUMP_USER プロシージャを介して)、これらの操作を実行できます。
データベース管理者は、Database Vault 保護ありの表領域内で表のトランスポータブル・エクスポートを実行します(たとえば、エクスポートする表を含む表領域に存在する表オブジェクトのレルムやコマンド・ルール)。	EXP_FULL_DATABASE ロールと IMP_FULL_DATABASE ロールの他に、DBMS_MACADM.AUTHORIZE_TTS_USER プロシージャを使用して、エクスポートされる表を含む表領域に対する Database Vault 固有のトランスポータブル表領域の認可もこのユーザー(またはロール)に付与する必要があります。 完全データベース・レベルの Oracle Data Pump 権限を付与されたユーザーも、(DBMS_MACADM.AUTHORIZE_DATAPUMP_USER プロシージャで)、これらの操作を実行できます。
データベース管理者は、データベース全体の内容のトランスポータブル・エクスポートを実行します。	DV_OWNER、EXP_FULL_DATABASE、IMP_FULL_DATABASE の各ロールの他に、DBMS_MACADM.AUTHORIZE_DATAPUMP_USER プロシージャを使用して Database Vault 固有の完全データベース・レベルの Oracle Data Pump の認可もこのユーザー(またはロール)に付与する必要があります。このユーザーに対して DBMS_MACADM.AUTHORIZE_TTS_USER プロシージャを実行する必要はありません。
データベース管理者は、ネットワーク・リンクを使用して、Database Vault 保護なしの表領域または表のトランスポータブル・インポートを実行します。	データベース管理者と接続ユーザー両方の EXP_FULL_DATABASE ロールと IMP_FULL_DATABASE ロールの他に、ネットワーク・リンクで指定されている接続ユーザー(またはロール)に DV_DATAPUMP_NETWORK_LINK ロールも付与する必要があります。
データベース管理者は、ネットワーク・リンクを使用して、Database Vault 保護ありの表領域のトランスポータブル・インポートを実行します(たとえば、その表領域に存在する表オブジェクトのレルムやコマンド・ルール)。	EXP_FULL_DATABASE ロールと IMP_FULL_DATABASE ロールの他に、DBMS_MACADM.AUTHORIZE_TTS_USER プロシージャを使用して、ネットワーク・リンクで指定されている接続ユーザー(またはロール)に、その表領域の Database Vault 固有トランスポータブル表領域の認可も付与する必要があります。接続ユーザーに、DV_DATAPUMP_NETWORK_LINK ロールも付与する必要があります。 Database Vault 固有の完全データベース・レベルの Oracle Data Pump の認可を付与されたユーザーも、(DBMS_MACADM.AUTHORIZE_DATAPUMP_USER プロシージャを介して)、これらの操作を実行できます。
データベース管理者は、ネットワーク・リンクを使用して、Database Vault 保護ありのトランス	EXP_FULL_DATABASE ロールと IMP_FULL_DATABASE ロールの他に、DBMS_MACADM.AUTHORIZE_TTS_USER プロシージャを使用して、

シナリオ	必要な認可
ポータブル表領域内で表をインポートします(たとえば、エクスポートする表を含む表領域に存在する表オブジェクトのレルムやコマンド・ルール)。	エクスポートされる表を含む表領域に対する Database Vault 固有のトランスポータブル表領域の認可も接続ユーザー(またはロール)に付与する必要があります。ネットワーク・リンクで指定されている接続ユーザー(またはロール)に、DV_DATAPUMP_NETWORK_LINK ロールも付与する必要があります。 Database Vault 固有の完全データベース・レベルの Oracle Data Pump 権限を付与されたユーザーも、(DBMS_MACADM.AUTHORIZE_DATAPUMP_USER プロシージャを介して)、この操作を実行できます。
データベース管理者は、ネットワーク・リンクを使用してデータベース全体の内容のトランスポータブル・インポートを実行します。	DV_OWNER ロールの他に、DBMS_MACADM.AUTHORIZE_DATAPUMP_USER プロシージャを使用して Database Vault 固有の完全データベース・レベルの Oracle Data Pump の認可も、接続ユーザー(またはロール)に付与する必要があります。このユーザーに対して DBMS_MACADM.AUTHORIZE_TTS_USER プロシージャを実行する必要はありません。ネットワーク・リンクで指定されている接続ユーザー(またはロール)に、DV_DATAPUMP_NETWORK_LINK ロールも付与する必要があります。

親トピック: ユーザーまたはロールへのData Pumpのトランスポータブル・エクスポート操作およびトランスポータブル・インポート操作の認可

12.3.3.3 Database VaultにおけるData Pumpのトランスポータブル操作をユーザーまたはロールに認可

ユーザーまたはロールがDatabase Vault環境でOracle Data Pumpのトランスポータブル・エクスポートおよびインポート操作を実行することを認可できます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。
2. 認可を付与するユーザーまたはロールに、Oracle Data Pumpの使用に必要なEXP_FULL_DATABASEおよびIMP_FULL_DATABASEロールが付与されていることを確認します。

```
SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS
 WHERE GRANTED_ROLE LIKE '%FULL%';
```

3. トランスポータブル・エクスポートを実行する、またはネットワーク・リンクを使用してデータベースの内容全体のトランスポータブル・インポートを実行する場合は、DBMS_MACADM.AUTHORIZE_DATAPUMP_USERプロシージャを使用して、完全データベース・レベルのOracle Data Pump認可をユーザーまたはロールに付与します。それ以外の場合、このステップは無視してください。

   たとえば:

```
EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('DP_MGR');
```

4. Database Vault固有のトランスポータブル表領域認可のみが必要な場合は、このユーザーまたはロールにその認可を付与します。

たとえば:

```
EXEC DBMS_MACADM.AUTHORIZE_TTS_USER('DP_MGR', 'HR_TS');
```

5. トランスポータブル・インポート操作を実行するユーザーが、ネットワーク・リンクを使用して操作を実行する場合は、このユーザーまたはロールにDV_DATAPUMP_NETWORK_LINKロールを付与します。

   たとえば:

   ```
   GRANT DV_DATAPUMP_NETWORK_LINK TO DP_MGR;
   ```

6. トランスポータブル・エクスポートを実行するユーザー、またはネットワーク・リンクを使用してデータベース全体のトランスポータブル・インポートを実行する場合は、このユーザーまたはロールにDV_OWNERロールを付与します。

   ```
   GRANT DV_OWNER TO DP_MGR;
   ```

関連トピック

- AUTHORIZE_TTS_USERプロシージャ
- AUTHORIZE_DATAPUMP_USERプロシージャ
- DV_DATAPUMP_NETWORK_LINK Data Pumpネットワーク・リンク・ロール

親トピック: ユーザーまたはロールへのData Pumpのトランスポータブル・エクスポート操作およびトランスポータブル・インポート操作の認可

12.3.3.4 トランスポータブル表領域認可のユーザーまたはロールからの取消し

Data Pumpを使用するデータベース管理者の権限を取り消すことができます。

1. ユーザーまたはロールにDV_OWNERロールが付与されている場合は、オプションでこのロールを取り消します。

   ```
   REVOKE DV_OWNER FROM DP_MGR;
   ```

2. DBA_DV_TTS_AUTHデータ・ディクショナリ・ビューに問い合せて、Oracle Data Pumpの認可が付与されているユーザーまたはロールを確認します。

   ```
   SELECT GRANTEE, TSNAME FROM DBA_DV_TTS_AUTH;
   ```

3. 前のステップで収集した情報を使用して、DBMS_MACADM.UNAUTHORIZE_TTS_USER文を作成します。

   たとえば:

   ```
   EXEC DBMS_MACADM.UNUTHORIZE_TTS_USER('DP_MGR', 'HR_TS');
   ```

4. トランスポータブル・エクスポートを実行した、またはネットワーク・リンクを使用してデータベースの内容全体のトランスポータブル・インポートを実行した場合は、完全データベース・レベルのOracle Data Pump認可をユーザーまたはロールから取り消します。

   たとえば:

   ```
   EXEC DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER('DP_MGR');
   ```

5. すでにトランスポータブル・インポート操作を実行したユーザーが、ネットワーク・リンクを使用して操作を実行した場合は、このユーザーまたはロールからDV_DATAPUMP_NETWORK_LINKロールを取り消します。

   たとえば:

   ```
   REVOKE DV_DATAPUMP_NETWORK_LINK FROM DP_MGR;
   ```

関連トピック

- UNAUTHORIZE_TTS_USERプロシージャ
- UNAUTHORIZE_DATAPUMP_USERプロシージャ
- DV_DATAPUMP_NETWORK_LINK Data Pumpネットワーク・リンク・ロール

親トピック: ユーザーまたはロールへのData Pumpのトランスポータブル・エクスポート操作およびトランスポータブル・インポート操作の認可

12.3.4 Database Vault環境でのデータのエクスポートまたはインポートのガイドライン

Oracle Data Pumpデータベース管理者に適切な認可を付与すると、必要なエクスポートまたはインポート操作を実行できるようになります。

作業を開始する前に、次のガイドラインに従う必要があります。

- データベースのデータファイルの完全バックアップを作成します。これにより、新しくインポートしたデータが気に入らなかった場合、容易にデータベースを元の状態に戻すことができます。このガイドラインは、侵入者が自分のポリシーを使用するようにOracle Data Pumpエクスポート・データを変更した場合に特に有用です。
- 複数のスキーマまたは表をエクスポートおよびインポートする方法を決定します。DBMS_MACADM.AUTHORIZE_DATAPUMP_USERプロシージャでは複数のスキーマまたは表は指定できませんが、次のいずれかの方法でこの作業を行うことできます。
  - それぞれのスキーマまたは表に対してDBMS_MACADM.AUTHORIZE_DATAPUMP_USERプロシージャを実行し、次にEXPDPユーティリティとIMPDPユーティリティのSCHEMASパラメータまたはTABLESパラメータのオブジェクトのリストを指定します。
  - 全データベースのエクスポートまたはインポート操作を実行します。この場合、次のガイドランを参照してください。
- データベース全体のエクスポートまたはインポート操作を実行する場合は、EXPDPまたはIMPDP FULLオプションをYに設定します。この設定によりDVSYSスキーマが取得されるため、認可したユーザーまたはロールにDV_OWNERロールが付与されていることを確認してください。

次のことに注意してください。

- Oracle Database Vaultが有効になっている場合、ダイレクト・パス・オプション(direct=y)でレガシーのEXPユーティリティとIMPユーティリティは使用できません。
- 直接の付与またはロールを介した付与のいずれかによって、DBMS_MACADM.AUTHORIZE_DATAPUMP_USERプロシージャを通じてDatabase Vault固有のOracle Data Pump認可を付与されている、またはDBMS_MACADM.AUTHORIZE_TTS_USERプロシージャを通じてトランスポータブル表領域の認可を付与されているユーザーは、データベース・オブジェクトのエクスポートとインポートを実行できますが、通常はアクセスがないスキーマ表に対するSELECT問合せなど、他のアクティビティは実行できません。同様に、指定されたデータベース・オブジェクト外部のオブジェクトに対してData Pump操作を実行することもできません。
- データベース全体をエクスポートまたはインポートするユーザーにDV_OWNERロールを付与する必要があります。これは、データベース全体のエクスポートには、Oracle Database Vaultポリシーを格納するDVSYSスキーマへのアクセスが必要になるためです。ただし、DVSYSスキーマ自体をエクスポートすることはできません。Data Pumpは、保護定義のみをエクスポートします。インポート・プロセスを開始する前に、ターゲット・データベースにはDVSYSスキーマが必要であり、Database Vaultが有効になっている必要があります。)逆に、インポートされたポリシーをターゲット・データベースに適用するData Pumpインポート操作では、内部的にDBMS_MACADM PL/SQLパッケージが使用され、ここでData

PumpユーザーにDV_OWNERロールが必要になります。

関連項目:

Oracle Data Pump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: Oracle Database VaultでのOracle Data Pumpの使用

12.4 Oracle Database VaultでのOracle Schedulerの使用

データベース・ジョブのスケジュールを担当するユーザーは、Oracle Database Vault固有の認可を有している必要があります。

- Oracle Database VaultでのOracle Schedulerの使用について
  付与する必要がある認可レベルは、タスクを実行する管理者のスキーマによって異なります。
- ジョブ・スケジュール管理者へのDatabase Vaultの認可の付与
  Database Vault環境で、ユーザーにデータベース・ジョブをスケジュールする権限を付与できます。
- ジョブ・スケジュール管理者からの権限の取消し
  ユーザーからデータベース・ジョブをスケジュールする権限を取り消すことができます。

親トピック: Oracle Database Vault 環境でのDBA操作

12.4.1 Oracle Database VaultでのOracle Schedulerの使用について

付与する必要がある認可レベルは、タスクを実行する管理者のスキーマによって異なります。

次のシナリオが想定されます。

- 管理者が、独自のスキーマでジョブをスケジュールする場合。スキーマがレルムで保護されている場合を除き、データベース・ジョブをスケジュールする権限が付与されている管理者は、Oracle Database Vault固有の権限がなくても作業を続行できます。スキーマがレルムで保護されている場合は、このユーザーがレルムへのアクセスを許可されていることを確認してください。
- 管理者は別のスキーマでジョブを実行するが、このジョブでOracle Database Vaultのレルムまたはコマンド・ルールで保護されたオブジェクトにアクセスしない場合。この場合、このユーザーには、ジョブに関連するシステム権限のみが必要で、Oracle Database Vault権限は必要ありません。
- 管理者がデータベースまたはリモート・データベースの任意のスキーマを含む、他のユーザーのスキーマでジョブを実行する場合。このジョブがOracle Database Vaultレルムまたはコマンド・ルールで保護されているオブジェクトにアクセスする場合、DBMS_MACADM.AUTHORIZE_SCHEDULER_USERプロシージャを使用して、このユーザーにDatabase Vault固有の権限を付与する必要があります。この権限は、バックグラウンド・ジョブおよびフォアグラウンド・ジョブの両方に適用されます。フォアグラウンド・ジョブの場合、権限はジョブを作成または変更した最後のユーザーに適用されます。また、スキーマ所有者(ジョブが作成された保護スキーマ)がレルムに対して認可されていることを確認してください。

  後から、DBMS_MACADM.UNAUTHORIZE_SCHEDULER_USERプロシージャを使用してこの権限を取り消すことができます。スキーマがレルムで保護されていない場合、ユーザーにDBMS_MACADM.AUTHORIZE_SCHEDULER_USERプロシージャを実行する必要はありません。

レルムによって保護されているOracle Schedulerジョブを有効または無効にするには、自身がそのレルムに対して認可されているか(DBMS_MACADM.ADD_AUTH_TO_REALMを使用)、またはジョブ所有者スキーマに対するOracle Scheduler認可(DBMS_MACADM.AUTHORIZE_SCHEDULER_USERを使用)が必要です。

関連トピック

- レルム認可について

親トピック: Oracle Database VaultでのOracle Schedulerの使用

12.4.2 ジョブ・スケジュール管理者へのDatabase Vaultの認可の付与

Database Vault環境で、ユーザーにデータベース・ジョブをスケジュールする権限を付与できます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。

   これらのどちらかのロールを付与されているユーザーのみ、必要な権限を付与できます。

2. 権限を付与するユーザーに、データベース・ジョブをスケジュールするシステム権限が付与されていることを確認してください。

   この権限には、CREATE JOB、CREATE ANY JOB、CREATE EXTERNAL JOB、EXECUTE ANY PROGRAM、EXECUTE ANY CLASS、MANAGE SCHEDULERが含まれます。DBAおよびSCHEDULER_ADMINロールがこれらの権限を提供しますが、Oracle Database Vaultが有効な場合、権限はこれらのロールから取り消されます。

   たとえば:

   ```
   SELECT GRANTEE, PRIVILEGE FROM DBA_SYS_PRIVS
    WHERE PRIVILEGE IN ('CREATE JOB', 'CREATE ANY JOB');
   ```

3. このユーザーにOracle Database Vaultに対する権限を付与します。

   たとえば、ユーザーjob_mgrにデータベースの任意のスキーマのジョブをスケジュールする権限を付与するには、次のようになります。

   ```
   EXEC DBMS_MACADM.AUTHORIZE_SCHEDULER_USER('JOB_MGR');
   ```

   オプションで、job_mgrのアクティビティを特定のスキーマに制限することもできます。

   ```
   EXEC DBMS_MACADM.AUTHORIZE_SCHEDULER_USER('JOB_MGR', 'HR');
   ```

4. 次のようにDBA_DV_JOB_AUTHデータ・ディクショナリ・ビューを問い合せることで、ユーザーに権限が付与されていることを確認してください。

   ```
   SELECT GRANTEE,SCHEMA FROM DBA_DV_JOB_AUTH WHERE GRANTEE = 'user_name';
   ```

関連トピック

- AUTHORIZE_SCHEDULER_USERプロシージャ
- DBA_DV_JOB_AUTHビュー

親トピック: Oracle Database VaultでのOracle Schedulerの使用

12.4.3 ジョブ・スケジュール管理者からの権限の取消し

ユーザーからデータベース・ジョブをスケジュールする権限を取り消すことができます。

1. DBA_DV_JOB_AUTHデータ・ディクショナリ・ビューを問い合せて、ユーザーの認可を確認します。

   ```
   SELECT GRANTEE, SCHEMA FROM DBA_DV_JOB_AUTH WHERE GRANTEE='username';
   ```

2. 前のステップで収集した情報を使用して、DBMS_MACADM.UNAUTHORIZE_SCHEDULER_USERコマンドを作成し

ます。

たとえば:

```
EXEC DBMS_MACADM.UNAUTHORIZE_SCHEDULER_USER('JOB_MGR');
```

この権限取消しが、元の権限付与アクションを補完するものであることを確認します。すなわち、最初にjob_mgrにデータベース全体に対する権限を付与した場合、次のコマンドは機能しません。

```
EXEC DBMS_MACADM.UNAUTHORIZE_SCHEDULER_USER('JOB_MGR', 'HR');
```

関連トピック

- UNAUTHORIZE_SCHEDULER_USERプロシージャ

親トピック: Oracle Database VaultでのOracle Schedulerの使用

12.5 Oracle Database Vaultでの情報ライフサイクル管理の使用

Oracle Database Vault対応データベースで情報ライフサイクル管理操作を実行するユーザーは、これらの操作を実行するために認可を受けている必要があります。

- Oracle Database Vaultでの情報ライフサイクル管理の使用について
  Oracle Database Vaultのレルムおよびコマンド・ルールで保護されたオブジェクトに対して情報ライフサイクル管理(ILM)操作を実行する役割を担うユーザーに認可を付与できます。
- ユーザーへのDatabase VaultでのILM操作の認可
  ユーザーにOracle Database Vault環境での情報ライフサイクル管理(ILM)操作の実行を認可できます。
- ユーザーからの情報ライフサイクル管理認可の取消し
  Oracle Database Vault環境で情報ライフサイクル管理(ILM)操作を実行できないよう、ユーザーから認可を取り消すことができます。

親トピック: Oracle Database Vault 環境でのDBA操作

12.5.1 Oracle Database Vaultでの情報ライフサイクル管理の使用について

Oracle Database Vaultのレルムおよびコマンド・ルールで保護されたオブジェクトに対して情報ライフサイクル管理(ILM)操作を実行する役割を担うユーザーに認可を付与できます。

ユーザーがDatabase Vault対応データベースでのILM操作のために次のSQL文を実行できるようにするには、まずユーザーに認可を与える必要があります。

- ALTER TABLE
  - ILM
  - FLASHBACK ARCHIVE
  - NO FLASHBACK ARCHIVE
- ALTER TABLESPACE
  - FLASHBACK MODE

親トピック: Oracle Database Vaultでの情報ライフサイクル管理の使用

12.5.2 ユーザーへのDatabase VaultでのILM操作の認可

ユーザーにOracle Database Vault環境での情報ライフサイクル管理(ILM)操作の実行を認可できます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。
   これらのどちらかのロールを付与されているユーザーのみ、必要な権限を付与できます。
2. DBMS_MACADM.AUTHORIZE_MAINTENANCE_USERを使用してユーザーに認可を与えます。
   たとえば、HR.EMPLOYEES表でILM操作を実行するための認可をユーザーに与えるには、次のようにします。

   ```
   EXEC DBMS_MACADM.AUTHORIZE_MAINTENANCE_USER ('PSMITH', 'HR', 'EMPLOYEES',
   'TABLE', 'ILM');
   ```

   ユーザーpsmithにデータベース全体のILM認可を与える必要がある場合は、次のようなプロシージャを入力します。

   ```
   EXEC DBMS_MACADM.AUTHORIZE_MAINTENANCE_USER ('PSMITH', '%', '%', '%', '%');
   ```

3. DBA_DV_MAINTENANCE_AUTHデータ・ディクショナリ・ビューを問い合せることで、ユーザーが認可されていることを確認してください。

関連トピック

- AUTHORIZE_MAINTENANCE_USERプロシージャ
- DBA_DV_MAINTENANCE_AUTHビュー

親トピック: Oracle Database Vaultでの情報ライフサイクル管理の使用

12.5.3 ユーザーからの情報ライフサイクル管理認可の取消し

Oracle Database Vault環境で情報ライフサイクル管理(ILM)操作を実行できないよう、ユーザーから認可を取り消すことができます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。
   これらのどちらかのロールを付与されているユーザーのみ、必要な権限を付与できます。
2. DBA_DV_MAINTENANCE_AUTHデータ・ディクショナリ・ビューを問い合せて、ILMユーザーに与えられた認可の種類を確認します。
3. DBMS_MACADM.UNAUTHORIZE_MAINTENANCE_USERを使用してユーザーから認可を取り消します。
   たとえば:

   ```
   EXEC DBMS_MACADM.UNAUTHORIZE_MAINTENANCE_USER ('PSMITH', 'HR', '%', 'TABLE',
   'ILM');
   ```

関連トピック

- DBA_DV_MAINTENANCE_AUTHビュー
- UNAUTHORIZE_MAINTENANCE_USERプロシージャ

親トピック: Oracle Database Vaultでの情報ライフサイクル管理の使用

12.6 Oracle Database VaultにおけるOracle Database Replayの使用

データベース管理者は、Oracle Database ReplayユーザーにDatabase Vault環境で作業する認可を付与できます。

- Oracle Database VaultでのDatabase Replayの使用について
  Oracle Database Replayでワークロード取得操作およびワークロード・リプレイ操作の両方を実行するDatabase Vault認可をユーザーに付与できます。
- ユーザーに対するDatabase Replay操作の認可

ワークロード取得操作およびワークロード・リプレイ操作の両方をOracle Database Replayユーザーに認可できます。

- ユーザーからのDatabase Replay認可の取消し
  Oracle Database Replayのワークロード取得操作およびワークロード・リプレイ操作の両方に対する認可を取り消すことができます。

親トピック: Oracle Database Vault 環境でのDBA操作

12.6.1 Oracle Database VaultでのDatabase Replayの使用について

Oracle Database Replayでワークロード取得操作およびワークロード・リプレイ操作の両方を実行するDatabase Vault認可をユーザーに付与できます。

親トピック: Oracle Database VaultにおけるOracle Database Replayの使用

12.6.2 ユーザーへのDatabase Replay操作の認可

ワークロード取得操作およびワークロード・リプレイ操作の両方をOracle Database Replayユーザーに認可できます。

- ユーザーへのワークロード取得操作の認可
  Oracle Database Vault環境でOracle Database Replayのワークロード取得操作を実行する認可をユーザーに付与できます。
- ユーザーへのワークロード・リプレイ操作の認可
  Oracle Database Vault環境でOracle Database Replayのワークロード・リプレイ操作を実行する認可をユーザーに付与できます。

親トピック: Oracle Database VaultにおけるOracle Database Replayの使用

12.6.2.1 ユーザーへのワークロード取得操作の認可

Oracle Database Vault環境でOracle Database Replayのワークロード取得操作を実行する認可をユーザーに付与できます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。
   これらのロールのいずれかを付与されているユーザーのみが、この認可を付与できます。
2. DBMS_MACADM.AUTHORIZE_DBCAPTUREプロシージャを使用して、ユーザーを認可します。
   たとえば:

```
EXEC DBMS_MACADM.AUTHORIZE_DBCAPTURE ('PFITCH');
```

3. DBA_DV_DBCAPTURE_AUTHデータ・ディクショナリ・ビューを問い合せて、ユーザーが認可されていることを確認します。

関連トピック

- AUTHORIZE_DBCAPTUREプロシージャ
- DBA_DV_DBCAPTURE_AUTHビュー

親トピック: ユーザーへのDatabase Replay操作の認可

12.6.2.2 ユーザーへのワークロード・リプレイ操作の認可

Oracle Database Vault環境でOracle Database Replayのワークロード・リプレイ操作を実行する認可をユーザーに付与できます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。

これらのロールのいずれかを付与されているユーザーのみが、この認可を付与できます。

2. DBMS_MACADM.AUTHORIZE_DBREPLAYプロシージャを使用して、ユーザーを認可します。
   たとえば:

```
EXEC DBMS_MACADM.AUTHORIZE_DBREPLAY ('PFITCH');
```

3. DBA_DV_DBREPLAY_AUTHデータ・ディクショナリ・ビューを問い合せて、ユーザーが認可されていることを確認します。

関連トピック

- AUTHORIZE_DBREPLAYプロシージャ
- DBA_DV_DBREPLAYビュー

親トピック: ユーザーへのDatabase Replay操作の認可

12.6.3 ユーザーからのDatabase Replay認可の取消し

Oracle Database Replayのワークロード取得操作およびワークロード・リプレイ操作の両方に対する認可を取り消すことができます。

- ワークロード取得権限の取消し
  ユーザーから認可を取り消して、Oracle Database Vault環境でOracle Database Replayのワークロード取得操作を実行できないようにすることができます。
- ワークロード・リプレイ権限の取消し
  ユーザーから認可を取り消して、Oracle Database Vault環境でOracle Database Replayのワークロード・リプレイ操作を実行できないようにすることができます。

親トピック: Oracle Database VaultにおけるOracle Database Replayの使用

12.6.3.1 ワークロード取得権限の取消し

ユーザーから認可を取り消して、Oracle Database Vault環境でOracle Database Replayのワークロード取得操作を実行できないようにすることができます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。
   これらのロールのいずれかを付与されているユーザーのみが、この認可を付与できます。
2. DBA_DV_DBCAPTURE_AUTHデータ・ディクショナリ・ビューを問い合せて、ワークロード取得認可を取り消すユーザーを見つけます。
3. DBMS_MACADM.UNAUTHORIZE_DBCAPTUREプロシージャを使用して、ユーザーから認可を取り消します。
   たとえば:

```
EXEC DBMS_MACADM.UNAUTHORIZE_DBCAPTURE ('PFITCH');
```

関連トピック

- DBA_DV_DBCAPTURE_AUTHビュー
- UNAUTHORIZE_DBCAPTUREプロシージャ

親トピック: ユーザーからのDatabase Replay認可の取消し

12.6.3.2 ワークロード・リプレイ権限の取消し

ユーザーから認可を取り消して、Oracle Database Vault環境でOracle Database Replayのワークロード・リプレイ操作を実行できないようにすることができます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。これらのロールのいずれかを付与されているユーザーのみが、この認可を付与できます。
2. DBA_DV_DBREPLAY_AUTHデータ・ディクショナリ・ビューを問い合せて、ワークロード・リプレイ認可を取り消すユーザーを見つけます。
3. DBMS_MACADM.UNAUTHORIZE_DBDBREPLAYプロシージャを使用して、ユーザーから認可を取り消します。たとえば:

```
EXEC DBMS_MACADM.UNAUTHORIZE_DBREPLAY ('PFITCH');
```

関連トピック

- DBA_DV_DBREPLAYビュー
- UNAUTHORIZE_DBREPLAYプロシージャ

親トピック: ユーザーからのDatabase Replay認可の取消し

12.7 Oracle Database Vaultでのプリプロセッサ・プログラムの実行

外部表からプリプロセッサ・プログラムを実行するユーザーには、Oracle Database Vault固有の認可が必要です。

- Oracle Database Vaultでのプリプロセッサ・プログラムの実行について
  ユーザーが外部表からプリプロセッサ・プログラムを実行するDatabase Vaultの認可を付与したり、取り消すことができます。
- ユーザーに対するプリプロセッサ・プログラム実行の認可
  DBMS_MACADM.AUTHORIZE_PREPROCESSORプロシージャは、外部表からプリプロセッサ・プログラムを実行する認可をユーザーに付与します。
- ユーザーからのプリプロセッサ実行認可の取消し
  DBMS_MACADM.UNAUTHORIZE_PREPROCESSORプロシージャは、ユーザーから認可を取り消して、Oracle Database Vault環境でユーザーが外部表からプリプロセッサ・プログラムを実行できないようにします。

親トピック: Oracle Database Vault 環境でのDBA操作

12.7.1 Oracle Database Vaultでのプリプロセッサ・プログラムの実行について

ユーザーが外部表からプリプロセッサ・プログラムを実行するDatabase Vaultの認可を付与したり、取り消すことができます。

親トピック: Oracle Database Vaultでのプリプロセッサ・プログラムの実行

12.7.2 ユーザーへのプリプロセッサ・プログラム実行の認可

DBMS_MACADM.AUTHORIZE_PREPROCESSORプロシージャは、外部表からプリプロセッサ・プログラムを実行する認可をユーザーに付与します。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。これらのロールのいずれかを付与されているユーザーのみが、この認可を付与できます。
2. DBMS_MACADM.AUTHORIZE_PREPROCESSORプロシージャを使用して、ユーザーを認可します。たとえば:

```
EXEC DBMS_MACADM.AUTHORIZE_PREPROCESSOR ('PFITCH');
```

3. DBA_DV_PREPROCESSOR_AUTHデータ・ディクショナリ・ビューを問い合せて、ユーザーが認可されていることを確認

します。

親トピック: Oracle Database Vaultでのプリプロセッサ・プログラムの実行

12.7.3 ユーザーからのプリプロセッサ実行認可の取消し

DBMS_MACADM.UNAUTHORIZE_PREPROCESSORプロシージャは、ユーザーから認可を取り消して、Oracle Database Vault環境でユーザーが外部表からプリプロセッサ・プログラムを実行できないようにします。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。これらのロールのいずれかを付与されているユーザーのみが、この認可を付与できます。
2. DBMS_MACADM.UNAUTHORIZE_PREPROCESSORプロシージャを使用して、ユーザーから認可を取り消します。たとえば:

```
EXEC DBMS_MACADM.UNAUTHORIZE_PREPROCESSOR ('PFITCH');
```

3. DBA_DV_PREPROCESSOR_AUTHデータ・ディクショナリ・ビューを問い合せて、ユーザーが認可されていないことを確認します。

親トピック: Oracle Database Vaultでのプリプロセッサ・プログラムの実行

12.8 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

インフラストラクチャ・データベース管理者などのCDBルート共通ユーザーでPDBアクセスを制御できます。

- Database Vault操作の制御の使用について
  共通ユーザーが自律型で通常のクラウドまたはオンプレミス環境でプラガブル・データベース(PDB)のローカル・データにアクセスすることを自動的に制限できます。
- 例外リストへの共通ユーザーおよびパッケージの追加の動作
  共通ユーザーまたはパッケージを例外リストに追加する前に、特別な要件を満たす必要があります。
- Database Vault操作の制御の有効化
  Database Vault操作の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。
- 例外リストへの共通ユーザーおよびパッケージの追加
  PDB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およびアプリケーションを例外リストに追加できます。
- 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
  PDB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を例外リストから削除できます。
- Database Vault操作の制御の無効化ま
  Database Vault操作の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。

親トピック: Oracle Database Vault 環境でのDBA操作

12.8.1 Database Vault操作の制御の使用について

共通ユーザーが自律型で通常のクラウドまたはオンプレミス環境でプラガブル・データベース(PDB)のローカル・データにアクセスすることを自動的に制限できます。

これは、CDB共通ユーザーに適用されるOracle Database Vault操作の制御を使用して実現できます。

Database Vault操作の制御は、データベース管理者がCDBルートに高い権限を持つユーザーとしてログインする必要があるが、PDB顧客データにアクセスできない場合に便利です。データベース操作の制御では、PDBデータベース管理者はブロックされません。これらのユーザーをブロックするには、PDBでOracle Database Vaultを有効にし、レルム制御などのDatabase Vault機能を使用してこれらのユーザーをブロックします。

操作の制御では、共通ユーザーがPDBローカル・データ(ローカルPDBユーザーが所有するデータ)にアクセスできない設計になっています。しかし、操作の制御を使用する場合は、アプリケーション・ルート、アプリケーションPDBおよび通常のPDBにある共通ユーザーのオブジェクトのデータに共通ユーザーが引き続きアクセスして、共通の操作を実行する必要があると考えられます。

操作の制御が有効で、Oracle DatabaseインストールでALLOW_COMMON_OPERATIONパラメータを使用できない場合は、PDB、アプリケーション・ルートまたはアプリケーションPDBのローカル・データへの共通ユーザーのアクセスがブロックされます。ただし、CDB$ROOT以外のコンテナ(アプリケーション・ルート、アプリケーションPDBおよび通常のPDB)で次のシナリオが実行された場合、CDBで共通に作成されていないDatabase Vaultレルムおよびコマンド・ルールがデフォルトのDatabase Vaultレルムおよびコマンド・ルールでない場合は無視されます。

- CDB共通ユーザーが所有するオブジェクトにCDB共通ユーザーがアクセスする。
- PDB、アプリケーション・ルートまたはアプリケーションPDBにCDB共通ユーザーが接続する。
- ALTER SYSTEM文またはALTER SESSION文をCDB共通ユーザーが実行する。

最近のOracle Databaseのパッチには、共通ユーザーがPDBの共通ユーザー・オブジェクトにアクセスして共通コマンドを実行できるかどうかを制御するALLOW_COMMON_OPERATIONパラメータが含まれています。ALLOW_COMMON_OPERATIONにより、共通ユーザーのオブジェクトおよび共通の操作コマンドへのアクセス機能が操作の制御から分離され、共通の操作に対してローカルDatabase Vaultの制御を強制するかどうかをユーザーがより柔軟に制御できるようになります。

ALLOW_COMMON_OPERATIONパラメータは、Oracle Databaseリリース19Cに対するパッチとして追加されました。DV_MONITORロールまたはDV_SECANALYSTロールがある場合は、次の問合せを実行して、インストール内にあるかどうかを確認できます。

```
SELECT * FROM DVSYS.DBA_DV_COMMON_OPERATION_STATUS;
```

ALLOW_COMMON_OPERATIONパラメータが存在しない場合は、「ORA-00942: 表またはビューが存在しません。」のエラーになります。ALLOW_COMMON_OPERATIONが存在し、FALSEに設定されている場合は、すべてのDatabase Vaultの制御が操作の制御で尊重されます(無視されません)。そのような場合、PDB Database Vaultの制御で、共通ユーザーのオブジェクトへのアクセスがブロックされ、共通操作もブロックされることがあります。

共通ユーザーまたはアプリケーションがPDBのローカル・データにアクセスする必要があるタスクを実行する必要がある場合、共通ユーザーおよびパッケージに対するDatabase Vault操作の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で使用するCTXSYSアプリケーション・アカウントがあります。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完全なアクセス権を提供するかわりに、より細かい制御を適用できます。

Database Vault操作の制御を使用する一般的なプロセスは、次のとおりです。

1. Database Vault操作の制御を有効にして、本番環境に対してこれを有効にしたままにします。
2. この段階では、Database Vault操作の制御は、PDBでDatabase Vaultが有効になっているかどうかに関係なく、環境内のすべてのPDBに適用されます。
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、こ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にアクセス権が不要になった場合、例外リストから削除できます。たとえば、データベースでOracle Textを使用している場合、CTXSYS管理ユーザー・アカウントおよびパッケージを例外リストに追加できます。

親トピック: 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

12.8.2 例外リストへの共通ユーザーおよびパッケージの追加の動作

共通ユーザーまたはパッケージを例外リストに追加する前に、特別な要件を満たす必要があります。

パッケージがユーザー・アカウント内の唯一のオブジェクトであり、PDBローカル・データにアクセスする必要がある場合、例外リストにユーザー・パッケージを追加できます。これにより、例外リストに含まれる内容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およびパッケージの種類は、PDBの機能に必要な種類です。たとえば、Oracle Spatialを使用している場合、例外リストにMDSYSアカウントを追加する必要があります。MDSYSでは、Oracle Spatialの機能用の顧客PDBデータにアクセスする必要があります。

操作の制御の例外リストのPL/SQLプロシージャは、共通ユーザーがPL/SQLプロシージャを実行するためのシステム権限または直接オブジェクト権限を持っている場合は、すべての共通ユーザーが実行できます。(定義者権限プロシージャのみ例外リストに追加でき、実行者権限では追加できません。)

操作の制御の例外リスト(ユーザー、%例外)上のユーザーのみが、PL/SQLプロシージャを変更する権限を持っている場合にのみ、例外リスト上のPL/SQLプロシージャを変更できます。たとえば、プロシージャが操作の制御の例外リストにあるがUser Xが例外リストにない場合、User Xは自身のUser X PL/SQLプロシージャを変更できません。User Yが例外リスト(Y, %)に存在し、User YがUser Xプロシージャを変更する権限を持っている場合、User YはUser Xのプロシージャを変更できます。

共通ユーザーおよびパッケージをDatabase Vault操作の制御の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します。既存の例外を確認するには、DBA_DV_APP_EXCEPTIONデータ・ディクショナリ・ビューを問い合せます。

親トピック: 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

12.8.3 Database Vault操作の制御の有効化

Database Vault操作の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対してDatabase Vault操作の制御を使用する場合、Database Vault操作の制御をフル・タイムで有効にしたままにすることをお薦めします。

ほとんどの場合、特定のPDBだけでなく、CDB全体に対してデータベース操作の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の目的など)、PDBでDBMS_MACADM.DISABLE_APP_PROTECTIONプロシージャを実行できます。PDBのトラブルシューティングが終了したら、このトピックの例に示すように、Database Vault操作の制御を再有効化します。
Database Vault操作の制御を有効にする前に、CDBルートでDatabase Vaultを有効にして構成する必要があります。ただし、Database VaultはPDBで有効にする必要はありません。

1. DV_OWNERロールを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。
   たとえば:

   ```
   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操作を制御するには、次のようにします。

     ```
     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

PBB (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がDatabase Vault操作の制御に対して有効になります。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定の操作を実行する必要がある場合、DBMS_MACADM.ADD_APP_EXCEPTIONプロシージャを使用して、ユーザーまたはパッケージをDatabase Vault操作の制御の例外リストに追加できます。

関連トピック

- 例外リストへの共通ユーザーおよびパッケージの追加

親トピック: 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

12.8.4 例外リストへの共通ユーザーおよびパッケージの追加

PDB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およびアプリケーションを例外リストに追加できます。

1. DV_OWNERロールを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。
   たとえば:

   ```
   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。
   - ユーザーが作成したパッケージは、定義者権限プロシージャを使用して作成する必要があります。

   ユーザーが作成したパッケージの詳細は、DBA_OBJECTSデータ・ディクショナリ・ビューを問い合せると確認できます。

3. DBMS_MACADM.ADD_APP_EXCEPTIONプロシージャを実行します。
   たとえば:

   ```
   DBMS_MACADM.ADD_APP_EXCEPTION ('MDSYS', 'PATCH_APP');
   ```

親トピック: 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

12.8.5 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

PDB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を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操作の制御の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。既存の例外を確認するには、DBA_DV_APP_EXCEPTIONデータ・ディクショナリ・ビューを問い合せます。

1. DV_OWNERロールを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。
   たとえば:

   ```
   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。
   たとえば:

```
DBMS_MACADM.DELETE_APP_EXCEPTION ('MDSYS', 'PATCH_APP');
```

親トピック: 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

12.8.6 Database Vault操作の制御の無効化

Database Vault操作の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。

ほとんどの場合は、Database Vault操作の制御を有効にしたままにする必要があります。トラブルシューティングでDatabase Vault操作の制御からPDBを削除する必要がある場合は、PDBに対するDatabase Vault操作の制御を一時的に無効にすることをお薦めします(残りのPDBの運用制御を維持します)。トラブルシューティングが完了したら、Database Vault操作の制御を再度有効にする必要があります。

1. DV_OWNERロールを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。
   たとえば:

   ```
   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操作の制御を無効にするには、次のようにします。

     ```
     EXEC DBMS_MACADM.DISABLE_APP_PROTECTION;
     ```

   - 特定のPBB (HRPDBなど)に対するDatabase Vault操作の制御を無効にするには、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

12.9 Oracle Recovery ManagerとOracle Database Vault

Oracle Database Vault環境ではRecovery Manager(RMAN)を使用できます。

Oracle Database VaultでRMANを使用する場合の機能は、標準のOracle Database環境での機能とほぼ同じです。ただし、エクスポート操作を試行する場合、RMANのリカバリ表および表パーティション機能は、レルムで保護された表と連携しないことに注意してください。エクスポート操作を実行するには、全表リカバリを実行してから、Database Vault認可ユーザーが実際に保護されている保護表のエクスポートを実行する必要があります。

表をリカバリしようとしたときにRMANの表および表パーティション・リカバリ機能とレルム保護された表が連携しないことに注意してください。その表をリカバリするには、データベース全体のリカバリを実行してから、Database Vault認可ユーザーが、レルムで保護された表をエクスポートし既存のデータベースにインポートする必要があります。

関連トピック

- 『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 Recovery Managerリファレンス』

親トピック: Oracle Database Vault 環境でのDBA操作

12.10 Oracle Database VaultでXStreamを使用するための権限

Oracle Database Vault環境でOracle Streamsを使用する場合、適切な権限が必要です。

これらの権限は次のとおりです。

- XStreamを構成するには、DV_XSTREAM_ADMINロールが付与されている必要があります。
- レルムで保護された表に変更を適用するには、そのレルムへのアクセスが認可されている必要があります。たとえば:

```
EXEC DBMS_MACADM.ADD_AUTH_TO_REALM('realm_name','username');
```

- DBMS_XSTREAM_AUTH.GRANT_ADMIN_PRIVILEGEプロシージャを実行する前に、DV_ACCTMGRロールが付与されている必要があります。

関連トピック

- DV_XSTREAM_ADMIN XStream管理ロール
- ADD_AUTH_TO_REALMプロシージャ

親トピック: Oracle Database Vault 環境でのDBA操作

12.11 Oracle Database VaultでOracle GoldenGateを使用するための権限

Oracle Database Vault環境でOracle GoldenGateを使用する場合、適切な権限が必要です。

これらの権限は次のとおりです。

- Oracle GoldenGateを構成するには、ユーザーにDV_GOLDENGATE_ADMINロールが付与されている必要があります。
- ユーザーがOracle GoldenGateのTRANLOGOPTIONS DBLOGREADERメソッドを使用してREDOログにアクセスする必要がある場合は、ユーザーにDV_GOLDENGATE_REDO_ACCESSロールが付与されている必要があります。

  たとえば、DV_GOLDENGATE_ADMINロールとDV_GOLDENGATE_REDO_ACCESSロールをgg_adminというユーザーに付与するには、次のようにします。

```
GRANT DV_GOLDENGATE_ADMIN, DV_GOLDENGATE_REDO_ACCESS TO gg_admin;
```

- ユーザーにDV_ACCTMGRロールが付与されていないと、レプリケートされた側でこのユーザーがユーザーを作成することはできません。
- ユーザーは、手続き型レプリケーションを実行する前に、トリガーなしモードで抽出操作を実行する必要があります。
- レルムで保護された表に変更を適用するには、そのレルムへのアクセスが認可されている必要があります。たとえば:

```
EXEC DBMS_MACADM.ADD_AUTH_TO_REALM('realm_name','username');
```

- SYSユーザーは、次のように、SYSTEMスキーマでデータ定義言語(DDL)を実行する認可を与えられている必要があります。

```
EXECUTE DVSYS.DBMS_MACADM.AUTHORIZE_DDL('SYS', 'SYSTEM');
```

- ユーザーは、Oracleデフォルト・コンポーネント保護レルムへの認可を与えられている必要があります。たとえば、このレルム認可をgg_adminというユーザーに与えるには、次のようにします。

```
BEGIN
 DVSYS.DBMS_MACADM.ADD_AUTH_TO_REALM(
   REALM_NAME    => 'Oracle Default Component Protection Realm',
   GRANTEE       => 'gg_admin',
   AUTH_OPTIONS  => 1);
```

```
END;
/
```

ノート:

Oracle GoldenGate は、SYS、SYSTEM および GoldenGate 関連スキーマ内のオブジェクトを問い合せ、更新および管理します。いずれかのスキーマが Oracle Database Vault レルムによって保護されている場合、GoldenGate Extract 操作は失敗する可能性があります。Oracle Database Vault では、ディクショナリ関連オブジェクトが Oracle のデフォルト・コンポーネント保護レルムで保護されます。カスタム Oracle Database Vault レルムまたはカスタム Oracle Database Vault コマンド・ルールを使用して、SYS や SYSTEM などのデフォルト・スキーマを保護しないことをお薦めします。

関連トピック

- DV_GOLDENGATE_ADMIN GoldenGate管理ロール
- DV_GOLDENGATE_REDO_ACCESS GoldenGate REDOログ・ロール
- ADD_AUTH_TO_REALMプロシージャ

親トピック: Oracle Database Vault 環境でのDBA操作

12.12 Oracle Database Vault環境でのデータ・マスキングの使用

Oracle Database Vault環境でデータ・マスキングを実行するには、正しい認可が必要です。

- Oracle Database Vaultが有効なデータベースでのデータ・マスキングについて
  Oracle Database Vaultが有効なデータベースでは、Database Vaultの認可を受けたユーザーのみがDatabase Vaultで保護されたデータベース・オブジェクトのデータをマスクできます。
- データ・ディクショナリ・レルム認可へのデータ・マスキング・ユーザーの追加
  データ・マスキング・ユーザーをOracleデフォルト・コンポーネント保護レルムに追加するとデータ・ディクショナリ・レルム認可を付与できます。
- マスクする表またはスキーマへのアクセス権のユーザーへの付与
  マスクする表またはスキーマへのアクセス権をユーザーに付与するには、適切なレルムに対してユーザーを認可する必要があります。
- データ・マスキングの権限を制御するコマンド・ルールの作成
  Oracle Database Vault環境でデータ・マスキングを使用するには、表、パッケージおよびトリガーを管理する権限が必要です。

親トピック: Oracle Database Vault 環境でのDBA操作

12.12.1 Oracle Database Vaultが有効なデータベースでのデータ・マスキングについて

Oracle Database Vaultが有効なデータベースでは、Database Vaultの認可を受けたユーザーのみがDatabase Vaultで保護されたデータベース・オブジェクトのデータをマスクできます。

Database Vault以外の環境では、SELECT_CATALOG_ROLEおよびDBAロールを付与されたユーザーがデータ・マスキングを行えます。ただし、Database Vaultを使用する場合、ユーザーに追加の権限が必要です。この項では、ユーザーがDatabase Vaultで保護されたオブジェクトのデータをマスクできるようにするために使用できる3つの方法について説明します。

ユーザーに適切な権限がない場合、マスキング定義の作成時またはジョブの実行時に次のエラーが発生します。

```
ORA-47400: Command Rule violation for string on string
ORA-47401: Realm violation for string on string.
ORA-47408: Realm violation for the EXECUTE command
ORA-47409: Command Rule violation for the EXECUTE command
ORA-01301: insufficient privileges
```

親トピック: Oracle Database Vault環境でのデータ・マスキングの使用

12.12.2 データ・ディクショナリ・レルム認可へのデータ・マスキング・ユーザーの追加

データ・マスキング・ユーザーをOracleデフォルト・コンポーネント保護レルムに追加するとデータ・ディクショナリ・レルム認可を付与できます。

Oracleデータ・ディクショナリでは、SYSおよびSYSTEMなどのOracle Databaseカタログ・スキーマへのアクセスが制御されます。(これらのスキーマをすべて示すリストについては、「デフォルトのレルム」を参照)。システム権限およびデータベース管理者ロールを付与する機能も制御されます。ユーザーをOracleデフォルト・コンポーネント保護レルムに追加する場合、これらのユーザーに、Oracleデータ・ディクショナリに関連付けられている権限がすでにあれば、これらのユーザーはDatabase Vault環境で同じ権限を持ちます。したがって、ユーザーをこのレルムに追加する場合、このユーザーが信頼できるユーザーであることを確認します。

- Oracleデフォルト・コンポーネント保護レルムにユーザーを追加するには、DBMS_MACADM.ADD_AUTH_TO_REALMプロシージャを使用します。

たとえば:

```
BEGIN
 DBMS_MACADM.ADD_AUTH_TO_REALM(
realm_name    => 'Oracle Default Component Protection Realm',
grantee       => 'DBA_JSMITH',
auth_options => DBMS_MACUTL.G_REALM_AUTH_PARTICIPANT);
END;
/
```

親トピック: Oracle Database Vault環境でのデータ・マスキングの使用

12.12.3 マスクする表またはスキーマへのアクセス権のユーザーへの付与

マスクする表またはスキーマへのアクセス権をユーザーに付与するには、適切なレルムに対してユーザーを認可する必要があります。

データ・マスクする表または表のスキーマがレルム内にある場合、データ・マスキングを行うユーザーを参加者または所有者としてレルム認可に追加する必要があります。表またはスキーマに、他のレルムで保護された表内にある依存オブジェクトがある場合、それらのレルムに対する参加者または所有者認可もユーザーに付与する必要があります。

- データ・マスクするオブジェクトを保護するレルムに対して、データ・マスキング・ユーザーを認可するには、DBMS_MACADM.ADD_AUTH_TO_REALMプロシージャを使用します。

次の例では、ユーザーDBA_JSMITHに、Business Apps Realmと呼ばれるレルムで保護されたHR.EMPLOYEES表に対する認可を付与する方法を示します。

```
BEGIN
 DBMS_MACADM.ADD_AUTH_TO_REALM(
realm_name    => 'Business Apps Realm',
grantee       => 'DBA_JSMITH',
auth_options => DBMS_MACUTL.G_REALM_AUTH_PARTICIPANT;
END;
```

```
/
```

親トピック: Oracle Database Vault環境でのデータ・マスキングの使用

12.12.4 データ・マスキングの権限を制御するコマンド・ルールの作成

Oracle Database Vault環境でデータ・マスキングを使用するには、表、パッケージおよびトリガーを管理する権限が必要です。

データ・マスキングを行うには、ユーザーは、マスキング・オブジェクトに対するCREATE TABLE、SELECT TABLE、ALTER TABLEおよびDROP TABLE権限を持っている必要があります。作成する依存オブジェクトがある場合は、CREATE PACKAGE、CREATE TRIGGERなどの適切な権限を持っている必要があります。

データ・マスキングの権限を粒度レベルで制御するためのコマンド・ルールを作成できます。これを行うには、データ・マスクする必要のあるオブジェクトへのユーザー・アクセスを禁止または許可するコマンド・ルールを作成します。たとえば、ユーザーがデータ・マスキングを行うユーザーのリスト内にあるかどうかをチェックするAllow Data Maskingと呼ばれるコマンド・ルールを作成できます。ログインするユーザーがこれらのユーザーのいずれかの場合、コマンド・ルールはtrueと評価され、ユーザーは、保護されたオブジェクトに対してデータ・マスクを作成することが許可されます。

データ・マスキング権限を制御するコマンド・ルールを作成するには:

1. ルール・セット・ルールを作成します。

   たとえば:

   ```
   BEGIN
    DBMS_MACADM.CREATE_RULE(
   rule_name  => 'Is HDRISCOLL or DBA_JSMITH User',
   rule_expr  =>'USER IN(''HDRISCOLL'',''DBA_JSMITH'')';
   END;
   /
   ```

2. ルール・セットを作成し、ルールを追加します。

   ```
   BEGIN
    DBMS_MACADM.CREATE_RULE_SET(
   rule_set_name    => 'Allow Data Masking',
   description      => 'Allows users HDRISCOLL and DBA_JSMITH access',
   enabled          => 'Y',
   eval_options     => 1,
   audit_options    => 1,
   fail_options     => 1,
   fail_message     => 'You do not have access to this object.',
   fail_code        => 20461,
   handler_options  => 0,
   is_static        => TRUE);
   END;
   /
   BEGIN
    DBMS_MACADM.ADD_RULE_TO_RULE_SET(
   rule_set_name => 'Allow Data Masking',
   rule_name     => 'Is HDRISCOLL or DBA_JSMITH User'),
   rule_order    => 1);
   END;
   /
   ```

3. コマンド・ルールを作成し、このルールを追加します。

   ```
   BEGIN
    DBMS_MACADM.CREATE_COMMAND_RULE(
   command         => 'CREATE TABLE',
   ```

```
rule_set_name   => 'Allow Data Masking',
object_owner    => 'HR',
object_name     => 'EMPLOYEES',
enabled         => DBMS_MACUTL.G_YES);
END;
/
```

親トピック: Oracle Database Vault環境でのデータ・マスキングの使用

12.13 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。

   たとえば:

   ```
   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。

   たとえば:

   ```
   CONNECT SYS AS SYSOPER
   Enter password: password
   ```

4. データベースを読取り専用モードで再起動します。

   たとえば:

   ```
   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。

   たとえば:

   ```
   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。

たとえば:

```
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。

たとえば:

```
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を読取り/書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。

たとえば:

```
SHUTDOWN IMMMEDIATE
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;
```

親トピック: Oracle Database Vault 環境でのDBA操作

12.14 Oracle Database Vault環境でのORADEBUGユーティリティの使用

ORADEBUGユーティリティは、主にOracleサポートがOracle Databaseで生じる問題を診断する場合に使用します。

ユーザーがOracle Database Vaultが有効な環境でORADEBUGユーティリティを実行できるかどうかを制御できます。従来型の監査環境では、AUDIT_SYS_OPERATIONS初期化パラメータをTRUEに設定することで、ORADEBUGの使用を監査できます。統合監査環境では、ORADBUGコマンドが強制的に監査されます。この制御は、特権OSユーザー(Oracleサーバー・プロセスと同じOSユーザーIDを持つOSユーザー)には適用されません。このようなユーザーは他の手段(たとえば、デバッガ)を使用してOracleプロセスを完全に制御および調査できるため、この例外が発生します。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。
2. 必要に応じて、ORADEBUGがすでに無効か有効かを確認します。

```
SELECT * FROM DBA_DV_ORADEBUG;
```

3. 次のいずれかのプロシージャを実行します。
   - ORADEBUGの使用を無効にするには:

     ```
     EXEC DBMS_MACADM.DISABLE_ORADEBUG;
     ```

   - ORADEBUGの使用を有効にするには:

     ```
     EXEC DBMS_MACADM.ENABLE_ORADEBUG;
     ```

関連トピック

- DBA_DV_ORADEBUGビュー
- DISABLE_ORADEBUGプロシージャ
- ENABLE_ORADEBUGプロシージャ

親トピック: Oracle Database Vault 環境でのDBA操作

12.15 Oracle Database Vault環境でのパッチ操作の実行

ユーザーSYSがOracle Database Vault対応データベースでパッチ操作を実行するには、DV_PATCH_ADMINロールが必要です。

DV_PATCH_ADMINが付与されているユーザーもデータを表示できます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、CDBまたはアプリケーション・ルートに接続します。
2. SYSユーザーに、一時的にDV_PATCH_ADMINロールを付与します。

   ```
   GRANT DV_PATCH_ADMIN TO SYS CONTAINER=ALL;
   ```

   単一のPDBにパッチを適用する場合、すべてのコンテナでDV_PATCH_ADMINをSYSに付与する必要はありません。
3. SYSユーザーがパッチ操作を実行した後、パッチのreadmeファイルの指示に慎重に従って、ユーザーSYSからDV_PATCH_ADMINを取り消します。

   ```
   REVOKE DV_PATCH_ADMIN FROM SYS CONTAINER=ALL;
   ```

親トピック: Oracle Database Vault 環境でのDBA操作

13 Oracle Database Vaultのスキーマ、ロールおよびアカウント

Oracle Database Vaultには、Database Vaultオブジェクトが含まれるスキーマや、特定のタスクの職務分離を提供するロール、およびデフォルト・ユーザーのアカウントが用意されています。

- Oracle Database Vaultスキーマ
  Oracle Database Vaultスキーマ(DVSYSおよびDVF)は、Oracle Database Vaultの管理およびランタイム処理をサポートしています。
- Oracle Database Vaultロール
  Oracle Database Vaultには、実行する必要がある特定のユーザー・タスクに基づいていて、職務分離の概念に従うデフォルト・ロールが用意されています。
- 登録中に作成されるOracle Database Vaultアカウント
  登録プロセスの間に、Oracle Database Vault所有者およびOracle Database Vaultアカウント・マネージャのためのアカウントを作成する必要があります。
- バックアップOracle Database Vaultアカウント
  ベスト・プラクティスとして、DV_OWNERロールとDV_ACCTMGRロールのバックアップ・アカウントを保持することをお薦めします。

13.1 Oracle Database Vaultスキーマ

Oracle Database Vaultスキーマ(DVSYSおよびDVF)は、Oracle Database Vaultの管理およびランタイム処理をサポートしています。

- DVSYSスキーマ
  DVSYSスキーマには、Oracle Database Vaultデータベース・オブジェクトが含まれます。
- DVFスキーマ
  DVFスキーマは、Oracle Database Vault DBMS_MACSEC_FUNCTION PL/SQLパッケージの所有者です。

親トピック: Oracle Database Vaultのスキーマ、ロールおよび アカウント

13.1.1 DVSYSスキーマ

DVSYSスキーマには、Oracle Database Vaultデータベース・オブジェクトが含まれます。

このオブジェクトは、Oracle Database Vault構成情報を格納し、Oracle Database Vaultの管理およびランタイム処理をサポートしています。

デフォルト・インストールでは、DVSYSスキーマはロックされています。DVSYSスキーマは、AUDIT_TRAIL$表も所有します。

マルチテナント・データベース環境で、DVSYSスキーマは共通スキーマとみなされます(すなわち、DVSYS内のオブジェクト(表、ビュー、PL/SQLパッケージなど)は任意の子プラガブル・データベース(PDB)で自動的に利用できます)。また、DVSYSスキーマ・アカウントは、ALTER SESSION文を使用して他のコンテナに切り替えることはできません。

Oracle Database Vaultでは、保護スキーマ設計を使用してDVSYSスキーマを保護します。保護スキーマ設計により、スキーマはシステム権限(SELECT ANY TABLE、CREATE ANY VIEW、DROP ANYなど)の不正使用から保護されます。

DVSYSスキーマは、Oracle Database Vaultによって次のように保護されます。

- DVSYS保護スキーマとその管理ロールは削除できません。デフォルトで、DVSYSアカウントはロックされています。
- デフォルトでは、ユーザーはDVSYSアカウントに直接ログインできません。ユーザーがこのアカウントに直接ログインできるように制御するには、DBMS_MACADM.DISABLE_DV_DICTIONARY_ACCTSプロシージャを実行してユーザーがログインできないようにして、DBMS_MACADM.ENABLE_DV_DICTIONARY_ACCTSプロシージャを実行してログインを許可します。
- CREATE USER、ALTER USER、DROP USER、CREATE PROFILE、ALTER PROFILE、DROP PROFILEなどの文は、DV_ACCTMGRロールを持つユーザーのみが発行できます。SYSDBA管理権限でログインしたユーザーは、「アカウント/プロファイルを保守可能」ルール・セットを変更して許可された場合のみこれらの文を発行できます。
- データベース定義言語(DDL)およびデータ操作言語(DML)のコマンドに対する強力なANYシステム権限は、保護スキーマではブロックされます。つまり、DVSYSスキーマのオブジェクトは、スキーマ・アカウント自体によって作成される必要があります。また、スキーマ・オブジェクトへのアクセスは、オブジェクト権限の付与により認可される必要があります。
- DVSYSスキーマのオブジェクト権限は、スキーマのDatabase Vault管理ロールにのみ付与できます。つまり、ユーザーは、事前定義済管理ロールによってのみ保護スキーマにアクセスできます。
- スキーマのDatabase Vault事前定義済管理ロールに対してALTER ROLE文を発行できるのは、保護されたスキーマ・アカウントDVSYSのみです。Oracle Database Vaultの事前定義済管理ロールの詳細は、「Oracle Database Vaultのロール」で説明します。
- SQL文の実行に、SYS.DBMS_SYS_SQL.PARSE_AS_USERプロシージャを保護スキーマDVSYSのかわりに使用することはできません。

ノート:

データベース・ユーザーは、Oracle Database Vault の管理ロール(たとえば DV_ADMIN および DV_OWNER)に、追加のオブジェクト権限およびロールを付与できます。ただし、付与に十分な権限がある場合にかぎります。

親トピック: Oracle Database Vaultスキーマ

13.1.2 DVFスキーマ

DVFスキーマは、Oracle Database Vault DBMS_MACSEC_FUNCTION PL/SQLパッケージの所有者です。

このパッケージには、ファクタ・アイデンティティを取得するファンクションが含まれます。Oracle Database Vaultのインストール後、DVFアカウントをより確実に保護するために、このアカウントがインストール処理によってロックされます。新しいファクタを作成すると、Oracle Database Vaultでそのファクタの取得ファンクションが新たに作成され、このスキーマに保存されます。

マルチテナント環境で、DVFユーザーはALTER SESSION文を使用して他のコンテナに切り替えることはできません。

デフォルトでは、ユーザーはDVFアカウントに直接ログインできません。ユーザーがこのアカウントに直接ログインできるように制御するには、DBMS_MACADM.DISABLE_DV_DICTIONARY_ACCTSプロシージャを実行してユーザーがログインできないようにして、DBMS_MACADM.ENABLE_DV_DICTIONARY_ACCTSプロシージャを実行してログインを許可します。

親トピック: Oracle Database Vaultスキーマ

13.2 Oracle Database Vaultロール

Oracle Database Vaultには、実行する必要がある特定のユーザー・タスクに基づいていて、職務分離の概念に従うデフォルト・ロールが用意されています。

- Oracle Database Vaultロールについて
  Oracle Database Vaultには、Oracle Database Vaultの管理に必要な一連のロールが用意されています。
- Oracle Database Vaultロールの権限
  Oracle Database Vaultのロールは、職務分離の利点が最大限に高まるよう設計されています。
- ユーザーへのOracle Database Vaultのロールの付与
  Enterprise Manager Cloud Controlを使用して、ユーザーにOracle Database Vaultのロールを付与できます。
- DV_OWNER Database Vault所有者ロール
  DV_OWNERロールでは、Oracle Database Vaultロールおよびその構成を管理できます。
- DV_ADMIN Database Vault構成管理者ロール
  DV_ADMINロールはOracle Database VaultのPL/SQLパッケージを制御します。
- DV_MONITOR Database Vault監視ロール
  DV_MONITORロールはOracle Database Vaultを監視するために使用します。
- DV_SECANALYST Database Vaultセキュリティ分析者ロール
  DV_SECANALYSTロールでは、アクティビティを分析できます。
- DV_AUDIT_CLEANUP監査証跡クリーンアップ・ロール
  DV_AUDIT_CLEANUPロールは、パージ操作のために使用します。
- DV_DATAPUMP_NETWORK_LINK Data Pumpネットワーク・リンク・ロール
  DV_DATAPUMP_NETWORK_LINKロールはData Pumpインポート操作のために使用します。
- DV_XSTREAM_ADMIN XStream管理ロール
  DV_XSTREAM_ADMINロールは、Oracle XStreamのために使用します。
- DV_GOLDENGATE_ADMIN GoldenGate管理ロール
  DV_GOLDENGATE_ADMINロールは、Oracle GoldenGateとともに使用します。
- DV_GOLDENGATE_REDO_ACCESS GoldenGate REDOログ・ロール
  DV_GOLDENGATE_REDO_ACCESSロールは、Oracle GoldenGateとともに使用します。
- DV_PATCH_ADMIN Database Vaultデータベース・パッチ・ロール
  DV_PATCH_ADMINロールは、パッチ操作のために使用します。
- DV_ACCTMGR Database Vaultアカウント・マネージャ・ロール
  DV_ACCTMGRロールは強力なロールであり、アカウント管理のために使用します。
- DV_REALM_OWNER Database VaultレルムDBAロール
  DV_REALM_OWNERロールは、レルム管理のために使用します。
- DV_REALM_RESOURCE Database Vaultアプリケーション・リソース所有者ロール
  DV_REALM_RESOURCEロールは、レルム・リソースの管理のために使用します。
- DV_POLICY_OWNER Database Vault所有者ロール
  DV_POLICY_OWNERロールでは、データベース・ユーザーが、制限されたレベルのOracle Database Vaultポリシーを管理できるようになります。
- DV_PUBLIC Database Vault PUBLICロール
  DV_PUBLICロールは使用されなくなりました。

親トピック: Oracle Database Vaultのスキーマ、ロールおよび アカウント

13.2.1 Oracle Database Vaultロールについて

Oracle Database Vaultには、Oracle Database Vaultの管理に必要な一連のロールが用意されています。

次の図は、これらのロールがデータベース内の職務分離の第1段階を実現するためにどのように設計されているかを示しています。これらのロールをどのように使用するかは、会社の要件によって異なります。

図13-1 Oracle Database Vaultロールの分類方法

ノート:

追加のオブジェクト権限およびロールを Oracle Database Vault ロールに付与して、権限の範囲を拡大できます。たとえば、SYSDBA 管理権限でログインしたユーザーは、オブジェクトが DVSYS スキーマまたはレルム

にないかぎり、オブジェクト権限を Oracle Database Vault ロールに付与できます。

関連トピック

- 職務分離のガイドライン
- Oracle Database管理アカウントの管理

親トピック: Oracle Database Vaultロール

13.2.2 Oracle Database Vaultロールの権限

Oracle Database Vaultのロールは、職務分離の利点が最大限に高まるよう設計されています。

DV_PATCH_ADMIN、DV_XSTREAM、DV_GOLDENGATE_ADMINおよびDV_GOLDENGATE_REDO_ACCESSロールは、システム権限がないため、次のセクションには含まれていません。

DVSYSスキーマ、EXECUTE権限

この権限を使用できるロール:

- DV_ADMIN (すべてのOracle Database Vault PL/SQLパッケージに対するEXECUTE権限が含まれます)
- DV_OWNER (すべてのOracle Database Vault PL/SQLパッケージに対するEXECUTE権限が含まれます)
- DV_POLICY_OWNER (一部のDBMS_MACADMプロシージャ)

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_AUDIT_CLEANUP
- DV_MONITOR
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE
- DV_SECANALYST

DVSYSスキーマ、SELECT権限

この権限を使用できるロール:

- DV_ADMIN
- DV_AUDIT_CLEANUP (一部のDatabase Vault表とビュー。AUDIT_TRAIL$表、DV$ENFORCEMENT_AUDITビューおよびDV$CONFIGURATION_AUDITビューに対してSELECT文を実行できます)
- DV_MONITOR
- DV_OWNER
- DV_POLICY_OWNER (一部のDBMS_MACADMプロシージャおよびPOLICY_OWNER*ビューのみ)
- DV_SECANALYST (一部のDatabase Vaultビュー。DV_SECANALYSTは、Oracle Database Vaultで提供されるビューを介して、DVSYSスキーマ・オブジェクトに問合せできます)

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE

DVSYSスキーマ、DELETE権限

この権限を使用できるロール:

- DV_AUDIT_CLEANUP (一部のDatabase Vault表とビュー、AUDIT_TRAIL$表、DV$ENFORCEMENT_AUDITビューおよびDV$CONFIGURATION_AUDITビューに対してDELETEを実行できます)
- DV_OWNER (一部のDatabase Vault表とビュー、AUDIT_TRAIL$表、DV$ENFORCEMENT_AUDITビューおよびDV$CONFIGURATION_AUDITビューに対してDELETEを実行できます)

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_ADMIN
- DV_MONITOR
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE
- DV_SECANALYST

DVSYSスキーマ、オブジェクトの権限の付与

この権限を使用できるロール: なし

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_ADMIN
- DV_AUDIT_CLEANUP
- DV_MONITOR
- DV_OWNER
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_SECANALYST
- DV_REALM_OWNER
- DV_REALM_RESOURCE

DVFスキーマ、EXECUTE権限

この権限を使用できるロール:

- DV_OWNER

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_ADMIN
- DV_AUDIT_CLEANUP
- DV_MONITOR
- DV_OWNER
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE
- DV_SECANALYST

DVFスキーマ、SELECT権限

この権限を使用できるロール:

- DV_OWNER
- DV_SECANALYST

この権限を拒否されるロール:

- DV_ACCTMGR

- DV_ADMIN
- DV_AUDIT_CLEANUP
- DV_MONITOR
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE

Database Vaultの監視権限

この権限を使用できるロール:

- DV_ADMIN
- DV_OWNER
- DV_MONITOR
- DV_SECANALYST

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_AUDIT_CLEANUP
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE

Database Vaultレポートの実行権限

この権限を使用できるロール:

- DV_ADMIN
- DV_OWNER
- DV_SECANALYST

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_AUDIT_CLEANUP
- DV_MONITOR
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE

SYSスキーマ、SELECT権限

この権限を使用できるロール:

- DV_MONITOR
- DV_OWNER
- DV_SECANALYST (DV_OWNERおよびDV_ADMINと同じシステム・ビュー)

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_ADMIN
- DV_AUDIT_CLEANUP
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE

SYSMANスキーマ、SELECT権限

この権限を使用できるロール:

- DV_OWNER (SYSMANの一部)
- DV_SECANALYST (SYSMANの一部)

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_ADMIN
- DV_AUDIT_CLEANUP
- DV_MONITOR
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE

ユーザー・アカウントおよびプロファイルのCREATE、ALTER、DROP権限

この権限には、DVSYSアカウントの削除や変更、およびDVSYSパスワードの変更のための権限は含まれません。

この権限を使用できるロール:

- DV_ACCTMGR

この権限を拒否されるロール:

- DV_ADMIN
- DV_AUDIT_CLEANUP
- DV_MONITOR
- DV_OWNER
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_OWNER
- DV_REALM_RESOURCE
- DV_SECANALYST

レルムを定義するスキーマのオブジェクトの管理

この権限には、CREATE ANY 、ALTER ANY およびDROP ANYなど、ANY権限が含まれます。

この権限を使用できるロール:

- DV_REALM_OWNER (このロールがあるユーザーは、システム権限を使用するには、レルム参加者またはレルム所有者である必要もあります。)

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_AUDIT_CLEANUP
- DV_ADMIN
- DV_MONITOR
- DV_OWNER (SYSMANの一部)
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_RESOURCE
- DV_SECANALYST (SYSMANの一部)

RESOURCEロール権限

RESOURCEロールは、CREATE CLUSTER 、CREATE INDEXTYPE 、CREATE OPERATOR 、CREATE PROCEDURE 、CREATE SEQUENCE 、CREATE TABLE 、CREATE TRIGGER、CREATE TYPEといったシステム権

限を提供します。

この権限を使用できるロール:

- DV_REALM_RESOURCE

この権限を拒否されるロール:

- DV_ACCTMGR
- DV_ADMIN
- DV_AUDIT_CLEANUP
- DV_MONITOR
- DV_OWNER (SYSMANの一部)
- DV_POLICY_OWNER
- DV_PUBLIC
- DV_REALM_OWNER
- DV_SECANALYST (SYSMANの一部)

親トピック: Oracle Database Vaultロール

13.2.3 ユーザーへのOracle Database Vaultのロールの付与

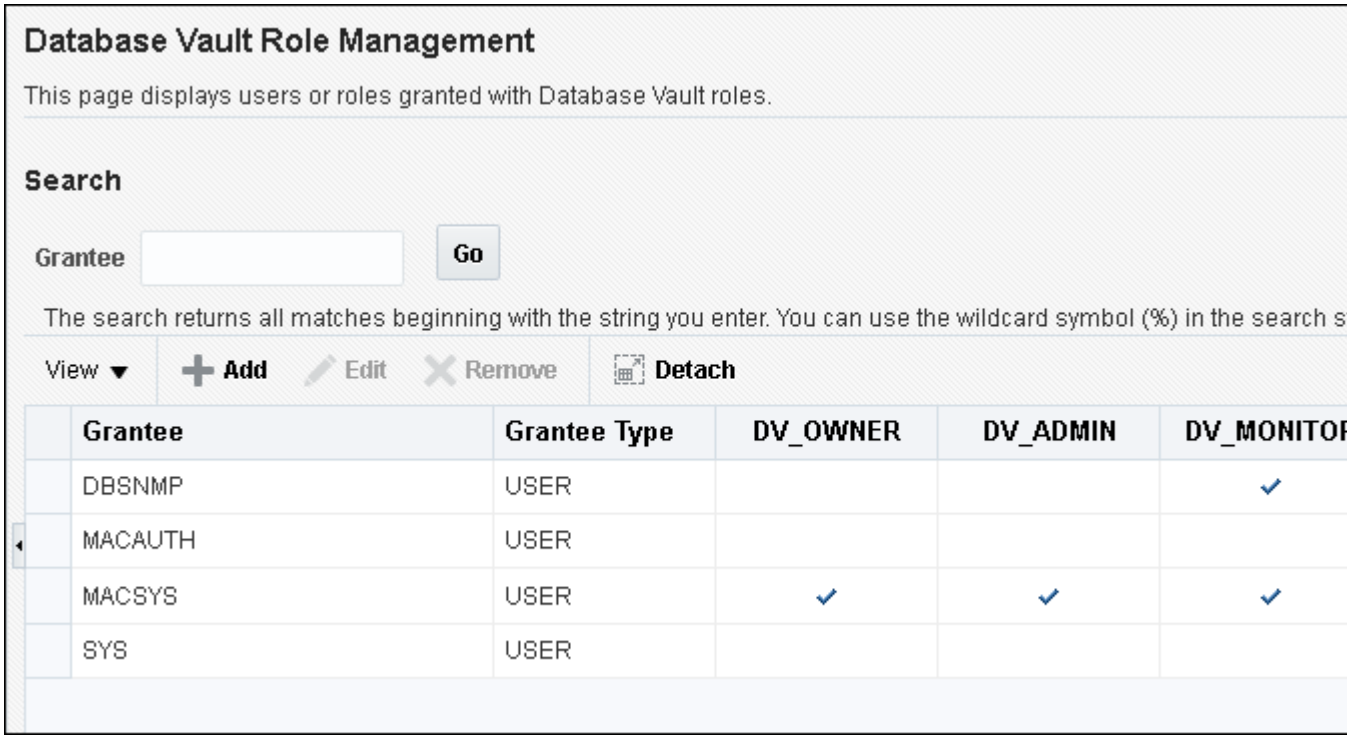
Enterprise Manager Cloud Controlを使用して、ユーザーにOracle Database Vaultのロールを付与できます。

1. DV_OWNERまたはSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud Controlから Oracle Database Vault Administratorにログインします。

   ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。

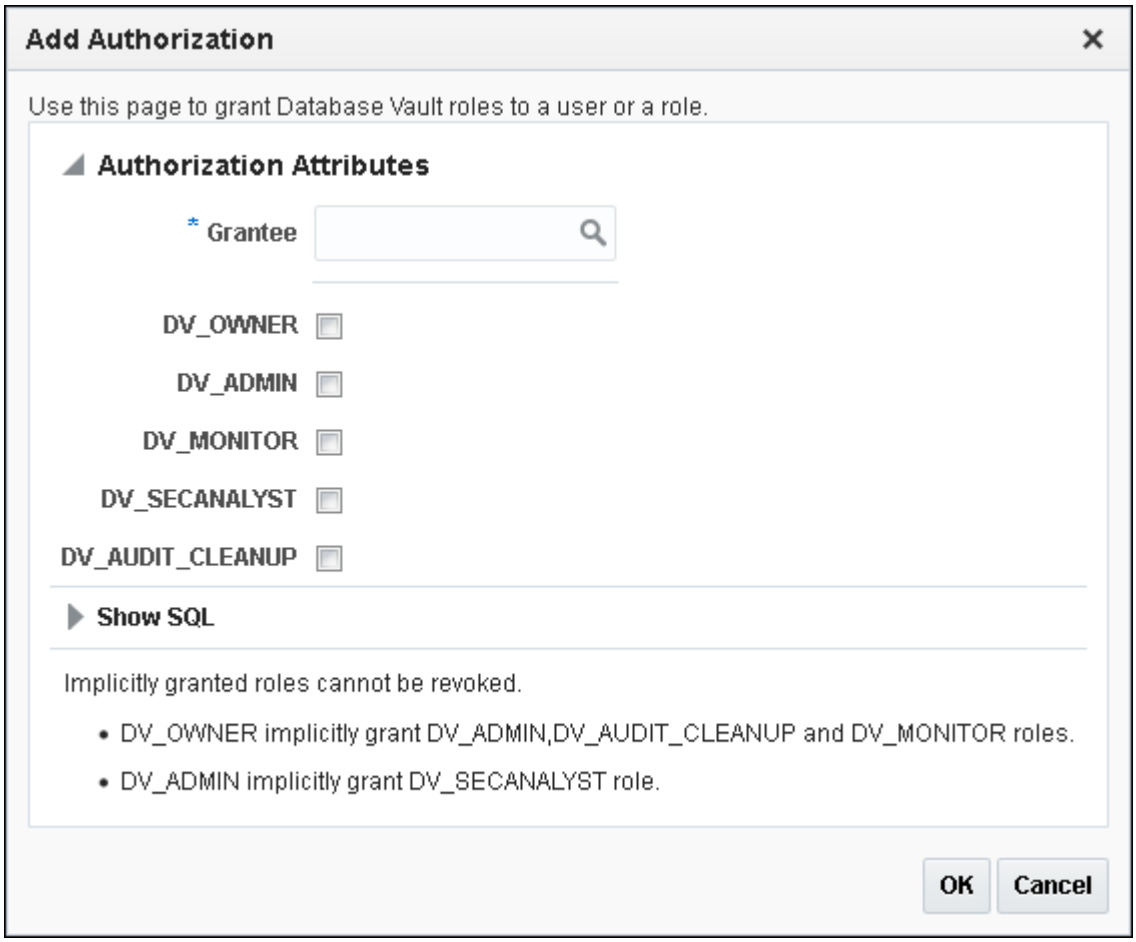
   他のユーザーにロールを付与できるユーザーの要件を知るには、ロールの説明を参照してください。

2. 「管理」ページの「Database Vaultコンポーネント」で、「Database Vaultロール管理」をクリックします。 「Database Vaultロール管理」ページが表示されます。



3. 次のいずれかを行います:
   - 付与のために新しいユーザーまたはロールを追加するには、「追加」ボタンをクリックして「認可の追加」ダイアロ

グ・ボックスを表示します。「権限受領者」フィールドに権限受領者を入力してから、付与するロールを選択します。次に「OK」をクリックします。



- 異なるロールを付与するか、「Database Vaultロール管理」ページに示されているユーザーまたはロールについてロール付与を変更するには、ユーザーまたはロールを選択し、「編集」をクリックしてから、必要に応じてロール付与を変更します。次に、「OK」をクリックします。

親トピック: Oracle Database Vaultロール

13.2.4 DV_OWNER Database Vault所有者ロール

DV_OWNERロールでは、Oracle Database Vaultロールおよびその構成を管理できます。

このマニュアルで、このロールを使用するサンプル・アカウントはsec_admin_owenです。

DV_OWNERロールに関連付けられた権限

DV_OWNERロールには、DV_ADMINロールによって提供される管理機能とDV_SECANALYSTロールによって提供されるレポート機能が含まれます。

このロールには、Oracle Database Vaultを監視する権限も備わっています。Oracle Database Vaultのインストール時に作成され、DVSYSスキーマに対するほとんどの権限を有します。DV_ADMINロールも有しています。

DV_OWNERロールに関連付けられたシステム権限とオブジェクト権限をすべて示すリストを確認するには、データベース・インスタンスにログインし、次の問合せを入力します。

```
SELECT TABLE_NAME, OWNER, PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE = 'DV_OWNER';
SELECT PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE = 'DV_OWNER';
```

Oracle Database Vaultを構成して有効化すると、DV_OWNERアカウントが作成されます。このロールを付与されたユーザー

は、ADMINオプションも付与され、任意のOracle Database Vaultロール(DV_ACCTMGRを除く)を任意のアカウントに付与できます。また、このロールを付与されたユーザーは、Oracle Database Vaultレポートの実行およびOracle Database Vaultの監視ができます。

ヒント:

DV_OWNER ユーザーの別個の名前付きアカウントを作成することをお薦めします。これにより、ユーザーが使用できなくなった場合(退社した場合など)、このユーザー・アカウントを再作成して、このユーザーにDV_OWNER ロールを付与できます。

DV_OWNERがGRANT操作とREVOKE操作に及ぼす影響

DV_OWNERロールを持つすべてのユーザーは、別のユーザーにDV_OWNERおよびDV_ADMINロールを付与できます。

このロールを付与されたアカウントは、付与されたDatabase Vaultロールを別のアカウントから取り消すことができます。SYSやSYSTEMなど、GRANT ANY ROLEシステム権限のみを持つ(直接またはロールを使用して間接的に付与された)アカウントには、他のデータベース・アカウントに対してDV_OWNERロールを付与する権限または取り消す権限がありません。また、DV_OWNERロールを持つユーザーは、DV_ACCTMGRロールの付与または取消しを実行できません。

DV_OWNERロールを持つユーザーのパスワード変更の管理

DV_OWNERロールを付与されている他のユーザーのパスワードを変更するには、あらかじめ、そのユーザー・アカウントからDV_OWNERロールを取り消しておく必要があります。

ただし、DV_OWNERロールの取消しには注意が必要です。サイトの少なくとも1人のユーザーに、このロールが付与されている必要があります。このロールが付与されている他のDV_OWNERユーザーのパスワードを変更する必要がある場合は、そのユーザーから一時的にDV_OWNERを取り消すことができます。また、DV_OWNERロールを付与されているユーザーは、自分自身からロールを取り消さなくても、自分のパスワードを変更できます。

DV_OWNERユーザー・パスワードを変更するには、次のようにします。

1. DV_OWNERロールを付与されているアカウントを使用して、データベース・インスタンスにログインします。
2. パスワード変更が必要なユーザー・アカウントからDV_OWNERロールを取り消します。
3. DV_ACCTMGRロールを付与されているユーザーとして接続し、このユーザーのパスワードを変更します。
4. DV_OWNERユーザーとして接続し、DV_OWNERロールをパスワードを変更したユーザーに再び付与します。

Oracle Database Vaultセキュリティを無効にした場合のDV_OWNERステータス

すべてのOracle Database Vaultロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.5 DV_ADMIN Database Vault構成管理者ロール

DV_ADMINロールはOracle Database VaultのPL/SQLパッケージを制御します。

これらのパッケージは、Oracle Enterprise Manager Cloud ControlでDatabase Vault Administratorユーザー・インタフェースの基礎となるインタフェースです。

DV_ADMINロールに関連付けられた権限

DV_ADMINロールは、DVSYSパッケージ(DBMS_MACADMおよびDBMS_MACUTL)に対するEXECUTE権限を保持します。

また、DV_ADMINにはDV_SECANALYSTロールによって提供された機能が含まれるため、これを使用するとOracle Database Vaultレポートの実行やOracle Database Vaultの監視が可能になります。インストール中、DV_ADMINロールがDV_OWNERロールにADMIN OPTION付きで付与されます。

また、DV_ADMINロールは、DBA_DV_POLICY、DBA_DV_POLICY_OWNERおよびDBA_DV_POLICY_OBJECTデータ・ディクショナリ・ビューに対するSELECT権限を提供します。

DV_ADMINロールに関連付けられたシステム権限とオブジェクト権限の完全なリストを検索するには、十分な権限でデータベース・インスタンスにログインし、次の問合せを入力します。

```
SELECT TABLE_NAME, OWNER, PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE = 'DV_ADMIN';
SELECT PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE = 'DV_ADMIN';
```

DV_ADMINがGRANT操作とREVOKE操作に及ぼす影響

SYSやSYSTEMなど、GRANT ANY ROLEシステム権限のみを持つアカウントには、他のデータベース・アカウントに対してDV_ADMINを付与する権限または取り消す権限がありません。

DV_OWNERロールを持つユーザーは、任意のデータベース・アカウントに対してこのロールを付与する、または取り消すことができます。

DV_ADMINロールを持つユーザーのパスワード変更の管理

DV_ADMINロールを付与されているユーザーのパスワードを変更するには、あらかじめ、このアカウントからDV_ADMINロールを取り消しておく必要があります。

DV_ADMINロールを付与されているユーザーは、自分自身からロールを取り消さなくても、自分のパスワードを変更できます。

DV_ADMINユーザー・パスワードを変更するには、次のようにします。

1. DV_OWNERロールを付与されているアカウントを使用して、データベース・インスタンスにログインします。
2. パスワード変更が必要なユーザー・アカウントからDV_ADMINロールを取り消します。
3. DV_ACCTMGRロールを付与されているユーザーとして接続し、このユーザーのパスワードを変更します。
4. DV_OWNERユーザーとして接続し、DV_ADMINロールをパスワードを変更したユーザーに再び付与します。

Oracle Database Vaultセキュリティを無効にした場合のDV_ADMINステータス

すべてのOracle Database Vaultロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.6 DV_MONITOR Database Vault監視ロール

DV_MONITORロールはOracle Database Vaultを監視するために使用します。

DV_MONITORロールにより、Oracle Enterprise Manager Cloud Controlエージェントは、Oracle Database Vaultでレルムまたはコマンド・ルール定義に関する違反未遂および構成の問題を監視できます。

このロールにより、Cloud Controlでは、レルム定義およびコマンド・ルール定義を読み取ってデータベース間で伝播できます。

DV_MONITORロールに関連付けられた権限

DV_MONITORロールに関連付けられているシステム権限はありませんが、SYSオブジェクトとDVSYSオブジェクトに対するSELECT権限を持ちます。

また、DV_MONITORロールは、DBA_DV_POLICY、DBA_DV_POLICY_OWNERおよびDBA_DV_POLICY_OBJECTデータ・ディクショナリ・ビューに対するSELECT権限を提供します。

DV_MONITORオブジェクト権限の完全なリストを検索するには、十分な権限(DV_OWNERなど)でデータベース・インスタンスにログインし、次の問合せを入力します。

```
SELECT TABLE_NAME, OWNER, PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE = 'DV_MONITOR';
```

DV_MONITORがGRANT操作とREVOKE操作に及ぼす影響

デフォルトでは、DV_MONITORロールはDV_OWNERロールおよびDBSNMPユーザーに付与されます。

DV_OWNERロールを付与されているユーザーのみ、別のユーザーに対してDV_MONITORロールを付与する、または取り消すことができます。

Oracle Database Vaultセキュリティを無効にした場合のDV_MONITORステータス

すべてのOracle Database Vaultロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの監視
- Oracle Database Vaultの監査
- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.7 DV_SECANALYST Database Vaultセキュリティ分析者ロール

DV_SECANALYSTロールにより、ユーザーはアクティビティを分析できます。

DV_SECANALYSTロールを使用して、Oracle Database Vaultレポートの実行およびOracle Database Vaultの監視を行います。

このロールは、データベース関連のレポートにも使用されます。また、「Oracle Database Vaultのデータ・ディクショナリ・ビュー」の説明にあるように、このロールを使用すると、DVSYSビューに問い合せることでDVSYS構成をチェックできます。

DV_SECANALYSTロールに関連付けられた権限

DV_SECANALYSTロールに関連付けられているシステム権限はありませんが、このロールには、DVSYSおよびDVFに関連するエンティティについてレポートするために、DVSYSスキーマ・オブジェクトと一部のSYSおよびSYSMANスキーマ・オブジェクトに対するSELECT権限があります。

また、DV_SECANALYSTロールは、DBA_DV_POLICY、DBA_DV_POLICY_OWNERおよびDBA_DV_POLICY_OBJECTデータ・ディクショナリ・ビューに対するSELECT権限を提供します。

DV_SECANALYSTオブジェクト権限の完全なリストを検索するには、十分な権限でデータベース・インスタンスにログインし、次の問合せを入力します。

```
SELECT TABLE_NAME, OWNER, PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE =
'DV_SECANALYST';
```

DV_SECANALYSTがGRANT操作とREVOKE操作に及ぼす影響

SYSやSYSTEMなど、GRANT ANY ROLEシステム権限のみを持つアカウントには、他のデータベース・アカウントに対してこのロールを付与する権限または取り消す権限がありません。

DV_OWNERロールを持つユーザーのみ、他のユーザーに対してこのロールを付与する、または取り消すことができます。

Oracle Database Vaultセキュリティを無効にした場合のDV_SECANALYSTステータス

すべてのOracle Database Vaultロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.8 DV_AUDIT_CLEANUP監査証跡クリーンアップ・ロール

DV_MONITORロールはパージ操作のために使用します。

非統合監査環境でDatabase Vault監査証跡のパージを担当するユーザーにDV_AUDIT_CLEANUPロールを付与します。

このロールを使用してパージ操作を完了する方法については、「Oracle Database Vault監査証跡のアーカイブおよびパージ」で説明します。

DV_AUDIT_CLEANUPロールに関連付けられた権限

DV_AUDIT_CLEANUPロールには、Database Vault関連の3つの監査ビューに対してSELECTおよびDELETE権限があります。

- DVSYS.AUDIT_TRAIL$表に対するSELECTおよびDELETE
- DVSYS.DV$ENFORCEMENT_AUDITビューに対するSELECTおよびDELETE
- DVSYS.DV$CONFIGURATION_AUDITビューに対するSELECTおよびDELETE

DV_AUDIT_CLEANUPがGRANT操作とREVOKE操作に及ぼす影響

デフォルトでは、このロールは、ADMIN OPTION付きのDV_OWNERロールに付与されます。

DV_OWNERロールを付与されているユーザーのみ、別のユーザーに対してDV_AUDIT_CLEANUPロールを付与する、または取り消すことができます。

Oracle Database Vaultセキュリティを無効にした場合のDV_AUDIT_CLEANUPステータス

すべてのOracle Database Vaultロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.9 DV_DATAPUMP_NETWORK_LINK Data Pumpネットワーク・リンク・ロール

DV_DATAPUMP_NETWORK_LINKロールはData Pumpインポート操作のために使用します。

Oracle Database Vault環境でNETWORK_LINKトランスポータブルData Pumpインポート操作の実行を担当するユーザーにDV_DATAPUMP_NETWORK_LINKロールを付与します。

このロールを使用すると、Oracle Data PumpのNETWORK_LINKトランスポータブル・インポート・プロセスをDatabase Vaultで厳格に制御できます。ただし、通常のOracle Data Pump操作を実行する方法は変更または制限されません。

DV_DATAPUMP_NETWORK_LINKロールに関連付けられた権限

DV_DATAPUMP_NETWORK_LINKロールに関連付けられているシステム権限はありませんが、DVSYSオブジェクトに対するEXECUTE権限を持ちます。

DV_DATAPUMP_NETWORK_LINKオブジェクト権限の完全なリストを検索するには、十分な権限でデータベース・インスタンスにログインし、次の問合せを入力します。

```
SELECT TABLE_NAME, OWNER, PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE =
'DV_DATAPUMP_NETWORK_LINK';
```

DV_DATAPUMP_NETWORK_LINKロールでは、NETWORK_LINKトランスポータブルData Pumpインポート操作を実行するための十分なデータベース権限が提供されないことに注意してください。DV_DATAPUMP_NETWORK_LINKロールは、データベース管理者がOracle Database Vault環境でNETWORK_LINK トランスポータブルData Pumpインポートを実行するための追加要件(Oracle Data Pumpで現在必要な権限を補うもの)です。

DV_DATAPUMP_NETWORK_LINKがGRANT操作とREVOKE操作に及ぼす影響

DV_OWNERロールを付与されているユーザーのみ、別のユーザーに対してDV_DATAPUMP_NETWORK_LINKロールを付与する、または取り消すことができます。

Oracle Database Vaultセキュリティを無効にした場合のDV_DATAPUMP_NETWORK_LINKステータス

すべてのOracle Databaseロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database VaultでのOracle Data Pumpの使用
- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.10 DV_XSTREAM_ADMIN XStream管理ロール

DV_XSTREAM_ADMINロールはOracle XStream用に使用します。

DV_XSTREAM_ADMINロールを、Oracle Database Vault環境でOracle XStreamの構成を担当する任意のユーザーに付与します。

これにより、XStreamプロセスの管理をDatabase Vaultで厳格に制御できます。ただし、管理者が通常XStreamを管理する方法は変更または制限されません。

DV_XSTREAM_ADMINロールに関連付けられた権限

DV_XSTREAM_ADMINロールに関連付けられた権限はありません。

DV_XSTREAM_ADMINロールは、XStreamを構成するための十分なデータベース権限は提供しません。DV_XSTREAM_ADMINロールは、データベース管理者がOracle Database Vault環境でXStreamを構成するための追加要件(XStreamで現在必要な権限を補うもの)です。

DV_XSTREAM_ADMINがGRANT操作とREVOKE操作に及ぼす影響

DV_OWNERロールを付与されているユーザーのみ、別のユーザーに対してDV_XSTREAM_ADMINロールを付与する、または取り消すことができます。

Oracle Database Vaultセキュリティを無効にした場合のDV_XSTREAM_ADMINステータス

すべてのOracle Databaseロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化
- Oracle Database VaultでXStreamを使用するための権限

親トピック: Oracle Database Vaultロール

13.2.11 DV_GOLDENGATE_ADMIN GoldenGate管理ロール

DV_GOLDENGATE_ADMINロールはOracle GoldenGateとともに使用します。

このロールを、Oracle Database Vault環境でOracle GoldenGateの構成を担当する任意のユーザーに付与します。

これにより、Oracle GoldenGateプロセスの管理をDatabase Vaultで厳格に制御できます。ただし、管理者が通常Oracle GoldenGateを管理する方法は変更または制限されません。

DV_GOLDENGATE_ADMINロールに関連付けられた権限

DV_GOLDENGATE_ADMINロールに関連付けられた権限はありません。

DV_GOLDENGATE_ADMINロールは、Oracle GoldenGateを構成するための十分なデータベース権限は提供しません。正確には、DV_GOLDENGATE_ADMINロールは、データベース管理者がOracle Database Vault環境でOracle GoldenGateを構成するための追加要件(Oracle GoldenGateで現在必要な権限を補うもの)です。

DV_GOLDENGATE_ADMINがGRANT操作とREVOKE操作に及ぼす影響

DV_OWNERロールを付与されているユーザーのみ、別のユーザーに対してDV_GOLDENGATE_ADMINロールを付与する、または取り消すことができます。

Oracle Database Vaultセキュリティを無効にした場合のDV_GOLDENGATE_ADMINステータス

すべてのOracle Databaseロールの保護は、Oracle Database Vaultが有効になっている場合にのみ適用されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化
- Oracle Database VaultでOracle GoldenGateを使用するための権限

親トピック: Oracle Database Vaultロール

13.2.12 DV_GOLDENGATE_REDO_ACCESS GoldenGate REDOログ・ロール

DV_GOLDENGATE_REDO_ACCESSロールはOracle GoldenGateとともに使用します。

DV_GOLDENGATE_REDO_ACCESSロールを、Oracle Database Vault環境でOracle GoldenGateのTRANLOGOPTIONS DBLOGREADERメソッドの使用によるREDOログへのアクセスを担当する任意のユーザーに付与します。

これにより、Oracle GoldenGateプロセスの管理をDatabase Vaultで厳格に制御できます。ただし、管理者が通常Oracle GoldenGateを管理する方法は変更または制限されません。

DV_GOLDENGATE_REDO_ACCESSロールに関連付けられた権限

DV_GOLDENGATE_REDO_ACCESSロールに関連付けられた権限はありません。

DV_GOLDENGATE_REDO_ACCESSロールは、Oracle GoldenGateを構成するための十分なデータベース権限は提供しません。DV_GOLDENGATE_REDO_ACCESSロールは、データベース管理者の追加要件(Oracle GoldenGateで現在必要な権限を補うもの)です。

DV_GOLDENGATE_REDO_ACCESSがGRANT操作とREVOKE操作に及ぼす影響

DV_GOLDENGATE_REDO_ACCESSロールをADMIN OPTION付きで付与することはできません。

DV_OWNERロールを付与されているユーザーのみ、別のユーザーに対してDV_GOLDENGATE_REDO_ACCESSロールを付与する、または取り消すことができます。

Oracle Database Vaultセキュリティを無効にした場合のDV_GOLDENGATE_REDO_ACCESSステータス

すべてのOracle Databaseロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化
- Oracle Database VaultでOracle GoldenGateを使用するための権限

親トピック: Oracle Database Vaultロール

13.2.13 DV_PATCH_ADMIN Database Vaultデータベース・パッチ・ロール

DV_PATCH_ADMINロールはパッチ操作のために使用します。

Database Vaultメタデータに指定された監査ポリシーやDatabase Vault統合監査ポリシーに従ってDatabase Vault関連のすべての監査レコードを生成するには、DV_PATCH_ADMINロールを使用する前にDV_ADMINロールが付与されたユーザーとしてDBMS_MACADM.ENABLE_DV_PATCH_ADMIN_AUDITプロシージャを実行します。

一時的に、DV_PATCH_ADMINロールを、データベースのパッチ適用担当の任意のデータベース管理者に付与します。この管理者がパッチ操作を行う前に、DBMS_MACADM.ENABLE_DV_PATCH_ADMIN_AUDITプロシージャを実行します。このプロシージャでは、既存の監査構成に従って、DV_PATCH_ADMINロールを付与されたユーザーによるアクションのレルム、コマンド・ルールおよびルール・セットの監査が可能になります。混合モード監査を使用する場合、このユーザーのアクションはAUDIT_TRAIL$表に書き込まれます。純粋な統合監査が有効な場合、このユーザーのアクションを取得する統合監査ポリシーを作成する必要があります。

パッチ操作の完了後、データベース・パッチ操作の実行を担当するユーザーの監査をすぐに無効にしないでください。このように、DV_PATCH_ADMINロール・ユーザーのアクションを追跡できます。下位互換性を確保するために、このタイプの監査はデフォルトで無効になっています。

DV_PATCH_ADMINロールに関連付けられた権限

DV_PATCH_ADMINロールでは、保護されたデータにアクセスできません。データベースのアップグレードには共通のDV_PATCH_ADMINの付与が必要であり、このロールを他のデータベース管理目的で使用しないことをお薦めします。

DV_PATCH_ADMINロールは、オブジェクト権限やシステム権限を持たない特殊なDatabase Vaultロールです。Database Vaultが有効なデータベースに、データベース管理者またはユーザーSYSがパッチを適用できるように(Database Vaultを無効にせずにデータベース・パッチを適用する場合など)設計されています。また、一部のパッチでは新しいスキーマを作成する必要があるため、データベース管理者はユーザーの作成も行うことができます。

DV_PATCH_ADMINロールを管理するには、次のガイドラインに従います。

- DV_PATCH_ADMINロールは、(たとえば、データベースをアップグレードする場合など)必要でないかぎり付与しないでください。
- ロールが不要になった場合は、DV_PATCH_ADMINロール付与を取り消してください。
- DV_PATCH_ADMINロールが付与されている間、監査レコードを確認してアクティビティをモニターしてください。

DV_PATCH_ADMINがGRANT操作とREVOKE操作に及ぼす影響

DV_OWNERロールを持つユーザーのみ、他のユーザーに対してDV_PATCH_ADMINロールを付与する、または取り消すことができます。

Oracle Database Vaultセキュリティを無効にした場合のDV_PATCH_ADMINステータス

すべてのOracle Databaseロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

マルチテナント環境でパッチを適用する際にDatabase Vaultを構成および有効化するためのガイダンス

DV_OWNERユーザーは、CDBルートの共通ユーザーに対してローカルに、または共通に構成できます。データベースにパッチを適用するためにDV_PATCH_ADMINを付与する必要がある場合、ローカルに付与されるDV_OWNERユーザーが行う必要がある処理に違いはありません。その構造により、DV_PATCH_ADMINは、CDBルートでローカルに付与されたDV_OWNER共通ユー

ザーによって付与されていても、パッチを完了するためにすべてのPDBでユーザーがDV_PATCH_ADMINを持っているように機能します。

関連トピック

- 監査の概要
- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.14 DV_ACCTMGR Database Vaultアカウント・マネージャ・ロール

DV_ACCTMGRロールは強力なロールであり、アカウント管理のために使用します。

DV_ACCTMGRロールを使用して、データベース・アカウントとデータベース・プロファイルの作成および管理を行います。このマニュアルでは、例のDV_ACCTMGRロールがbea_dvacctmgrというユーザーに割り当てられます。

DV_ACCTMGRロールに関連付けられた権限

このロールを付与されているユーザーは、DV_SECANALYST、DV_AUDIT_CLEANUPおよびDV_MONITORロールを付与されているユーザーを含め、ユーザー・アカウントまたはプロファイルに対してCREATE文、ALTER文、DROP文を使用できます。

このユーザーは、他のユーザーにCREATE SESSION権限を付与することもできます。ただし、DV_ACCTMGRロールを付与されているユーザーは、次の操作は実行できません。

- DVSYSアカウントに対するALTER文またはDROP文
- DV_ADMINまたはDV_OWNERロールが付与されているユーザーに対するALTER文またはDROP文
- DV_ADMINまたはDV_OWNERロールが付与されているユーザーのパスワード変更

CDBルート内のDV_ACCTMGRロールを付与された共通ユーザーは、共通DV_ACCTMGRユーザーにPDBでのSET CONTAINER権限またはDV_ACCTMGRロールがない場合でも、CDBルート内の共通ユーザーまたは共通プロファイルを変更できます。

DV_ACCTMGRロールに関連付けられたシステム権限とオブジェクト権限の完全なリストを検索するには、十分な権限でデータベース・インスタンスにログインし、次の問合せを入力します。

```
SELECT TABLE_NAME, OWNER, PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE = 'DV_ACCTMGR';
SELECT PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE = 'DV_ACCTMGR';
```

ヒント:

- DV_ACCTMGR ユーザーが他のユーザーに ANY 権限を付与または取り消せるようにする場合は、SYSDBA 権限を持つユーザーSYS としてログインし、このユーザーに GRANT ANY PRIVILEGE 権限および REVOKE ANY PRIVILEGE 権限を付与します。次に、このユーザーを所有者として Oracle システム権限およびロール管理レルムに追加します。
- DV_ACCTMGR ユーザーの別個の名前付きアカウントを作成することをお薦めします。これにより、このユーザーがパスワードを忘れた場合、元の DV_ACCTMGR アカウントとしてログインして、ユーザーのパスワードをリセットできます。それ以外の場合は、Oracle Database Vault を無効にし、SYS または SYSTEM としてログインしてパスワードを再作成し、Oracle Database Vault を再び有効に

する必要があります。

DV_ACCTMGRがGRANT操作とREVOKE操作に及ぼす影響

SYSやSYSTEMなど、GRANT ANY ROLEシステム権限のみを持つアカウントには、他のデータベース・アカウントに対してこのロールを付与する権限または取り消す権限がありません。

DV_ACCTMGRロールおよびADMIN OPTIONを持つアカウントは、このロールを指定されたデータベース・アカウントに付与することも、他のアカウントからこのロールを取り消すこともできます。

Oracle Database Vaultセキュリティを無効にした場合のDV_ACCTMGRステータス

すべてのOracle Databaseロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.15 DV_REALM_OWNER Database VaultレルムDBAロール

DV_REALM_OWNERロールはレルム管理のために使用します。

DV_REALM_OWNERロールを使用して、レルムを定義する複数のスキーマ内のデータベース・オブジェクトを管理します。

このロールを、レルムとそれに関連付けられたロール内の1つ以上のスキーマ・データベース・アカウントを管理するデータベース・アカウントに付与します。

DV_REALM_OWNERロールに関連付けられた権限

このロールを付与されているユーザーは、CREATE ANY、ALTER ANY、DROP ANYなどの強力なシステム権限をレルム内で使用できます。

ただし、これらの権限を使用するには、このユーザーをレルムの参加者または所有者にする必要があります。手順については、「レルム認可について」を参照してください。

DV_REALM_OWNERロールにはオブジェクト権限は付与されていませんが、いくつかのシステム権限が付与されています。DV_REALM_OWNERシステム権限の完全なリストを検索するには、十分な権限でデータベース・インスタンスにログインし、次の問合せを入力します。

```
SELECT PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE = 'DV_REALM_OWNER';
```

DV_REALM_OWNERがGRANT操作とREVOKE操作に及ぼす影響

Oracleシステム権限およびロール管理レルムのレルム所有者(SYSなど)は、このロールを、指定された任意のデータベース・アカウントまたはロールに付与できます。

このロールには強力なシステム権限がありますが、Oracle Database Vaultのロール(DV_OWNERまたはDV_ADMINロールなど)はないことに注意してください。

このロールを特定のレルムに関連付ける場合は、アカウントまたは業務関連のロールに割り当てた後、そのアカウントまたはロールをレルムで認可します。

Oracle Database Vaultセキュリティを無効にした場合のDV_REALM_OWNERステータス

すべてのOracle Databaseロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.16 DV_REALM_RESOURCE Database Vaultアプリケーション・リソース所有者ロール

DV_REALM_RESOURCEロールはレルム・リソースの管理のために使用します。

DV_REALM_RESOURCEロールは、レルムで一般に使用される表、ビュー、トリガー、シノニムおよびその他のオブジェクトの作成などの操作に使用します。

DV_REALM_RESOURCEロールに関連付けられた権限

DV_REALM_RESOURCEロールには、OracleのRESOURCEロールと同じシステム権限があります。さらに、CREATE SYNONYMとCREATE VIEWの両方がこのロールに付与されます。

DV_REALM_RESOURCEロールにはオブジェクト権限は付与されていませんが、いくつかのシステム権限が付与されています。DV_REALM_RESOURCEシステム権限の完全なリストを検索するには、十分な権限でデータベース・インスタンスにログインし、次の問合せを入力します。

```
SELECT PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE = 'DV_REALM_RESOURCE';
```

このロールには強力なシステム権限がありますが、Oracle Database Vaultのロール(DV_OWNERまたはDV_ADMINロールなど)はありません。

DV_REALM_RESOURCEがGRANT操作とREVOKE操作に及ぼす影響

DV_REALM_RESOURCEロールは、任意のデータベース・アプリケーションのサポートに使用されるデータベース表、オブジェクト、トリガー、ビュー、プロシージャなどを所有するデータベース・アカウントに付与できます。

これは、スキーマ・タイプのデータベース・アカウントを対象としたロールです。Oracleシステム権限およびロール管理レルムのレルム所有者(SYSなど)は、このロールを、任意のデータベース・アカウントまたはロールに付与できます。

Oracle Database Vaultセキュリティを無効にした場合のDV_REALM_RESOURCEステータス

すべてのOracle Databaseロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.17 DV_POLICY_OWNER Database Vault所有者ロール

DV_POLICY_OWNERロールでは、データベース・ユーザーが、制限されたレベルのOracle Database Vaultポリシーを管理できるようになります。

DV_POLICY_OWNERロールに関連付けられた権限

DV_POLICY_OWNERロールは、Database Vault管理ユーザー以外に、Database Vaultポリシーの有効化または無効化、レルムへの認可の追加またはレルムからの認可の削除、および次のデータベース・ビューに対するSELECT権限の使用のための十分な権限を提供します。

- DVSYS.POLICY_OWNER_COMMAND_RULE
- DVSYS.POLICY_OWNER_POLICY
- DVSYS.POLICY_OWNER_REALM
- DVSYS.POLICY_OWNER_REALM_AUTH
- DVSYS.POLICY_OWNER_REALM_OBJECT
- DVSYS.POLICY_OWNER_RULE_SET
- DVSYS.POLICY_OWNER_RULE
- DVSYS.POLICY_OWNER_RULE_SET_RULE

DV_POLICY_OWNERのみ、これらのビューを問合せできます。DV_OWNERロールとDV_ADMINロールがあるユーザーであっても、これらのビューを問い合せることはできません。

DV_POLICY_OWNERロールには、システム権限はありません。DV_POLICY_OWNERロールに関連付けられたオブジェクト権限をすべて示すリストを確認するには、データベース・インスタンスにログインし、次の問合せを入力します。

```
SELECT TABLE_NAME, OWNER, PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE =
'DV_POLICY_OWNER';
```

DV_POLICY_OWNERがGRANT操作とREVOKE操作に及ぼす影響

DV_POLICY_OWNERロールを付与されているユーザーは、このロールを他のユーザーに付与することや他のユーザーから取り消すことはできません。

Oracle Database Vaultセキュリティを無効にした場合のDV_POLICY_OWNERステータス

すべてのOracle Database Vaultロールの保護は、Oracle Database Vaultが有効な場合のみ実施されます。

Oracle Database Vaultが無効になっている場合、GRANT ANY ROLEシステム権限を持つアカウントはすべて、保護対象のDatabase Vaultロールに対してGRANT操作およびREVOKE操作を実行できます。

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: Oracle Database Vaultロール

13.2.18 DV_PUBLIC Database Vault PUBLICロール

DV_PUBLICロールは使用されなくなりました。

インストール時にDV_PUBLICロールは引き続き作成されますが、ロールや権限は付与されません。以前のリリースでDV_PUBLICに付与された権限はすべてPUBLICロールに直接付与されるようになりました。

親トピック: Oracle Database Vaultロール

13.3 登録中に作成されるOracle Database Vaultアカウント

登録プロセスの間に、Oracle Database Vault所有者およびOracle Database Vaultアカウント・マネージャのためのアカウントを作成する必要があります。

Oracle Database Vault所有者アカウントのアカウント名およびパスワードは、インストール時に指定する必要があります。Oracle Database Vaultアカウント・マネージャの作成はオプションですが、より適切な業務分離のためにお薦めします。

Oracle Database Vault所有者アカウントには、DV_OWNERロールが付与されます。このアカウントでは、Oracle Database Vaultのロールおよび構成を管理できます。

Oracle Database Vaultアカウント・マネージャ・アカウントには、DV_ACCTMGRロールが付与されます。このアカウントは、データベース・ユーザー・アカウントを管理して職務分離をしやすくするために使用されます。

インストール時にOracle Database Vaultアカウント・マネージャ・アカウントを作成しない場合、DV_OWNERとDV_ACCTMGRの両方のロールがOracle Database Vault所有者ユーザー・アカウントに付与されます。

表13-1に、インストール時に作成するアカウントの他に必要なOracle Database Vaultデータベース・アカウントを示します。

表13-1 Oracle Database Vaultで使用されるデータベース・アカウント

データベース・アカウント	ロールおよび権限	説明
DVSYS	いくつかのシステム権限およびオブジェクト権限が Oracle Database Vault をサポートするために用意されています。このアカウントでセッションを作成する権限は、インストールの最後に取り消され、アカウントはロックされます。	Oracle Database Vault のスキーマおよび関連オブジェクトの所有者
DVF	限定されたシステム権限が Oracle Database Vault をサポートするために用意されています。このアカウントでセッションを作成する権限は、インストールの最後に取り消され、アカウントはロックされます。	ファクタ・アイデンティティを取得するために作成される Oracle Database Vault のファンクションの所有者
LBACSYS	このアカウントは、Oracle Universal Installer のカスタム・インストール・オプションを使用して Oracle Label Security をインストールすると作成されます。(Oracle Database Vault のインストール時には作成されません。)このアカウントは削除または再作成しないでください。 ファクタと Oracle Label Security ポリシーを統合する場合、このファクタを使用するレルムの所有者としてこのユーザーを割り当てる必要があります。詳細は、「Oracle Label Security ポリシーでの Oracle Database Vault ファクタの使用方法」を参照してください。	Oracle Label Security のスキーマの所有者

Oracle Database Vaultの職務分離要件を満たすために別のデータベース・アカウントを作成できます。表13-2に、指針となるモデル・データベース・アカウントの一部を示します。(表13-2に示すアカウントは、Oracle Database Vaultロールの実装の指針となります。これらは、インストール時に作成される実際のアカウントではありません。)

表13-2 Oracle Database Vaultのモデル・データベース・アカウント

データベース・アカウント	ロールおよび権限	説明
EBROWN	DV_OWNER(DV_ADMIN および DV_SECANALYST を保持)	Oracle Database Vault レルムのレルム所有者になるアカウント。このアカウントには次のことが可能です。 ● DVSYS パッケージを実行する ● DVSYS スキーマ・オブジェクトに対する権限を付与する。 ● DVSYS スキーマ内のオブジェクトの選択 ● Oracle Database Vault アクティビティを監視する。 ● Oracle Database Vault 構成に関するレポートを実行する。
JGODFREY	DV_ACCTMGR	データベース・アカウントおよびプロファイルを管理するためのアカウント。このアカウントには次のことが可能です。 ● ユーザーの作成、変更および削除 ● プロファイルの作成、変更および削除 ● CREATE SESSION 権限の付与および取消し ● DV_ACCTMGR ロールの付与と取消しを行います。ただし、このアカウントが Database Vault のインストール時に作成された場合のみです(このアカウントは、ADMIN オプション付きで作成されました)。 ● CONNECT ロールの付与および取消し ノート: このアカウントでは、ロールの作成、および RESOURCE ロールまたは DBA ロールの付与は行えません。
RLAYTON	DV_ADMIN(DV_SECANALYST を保持)	アクセス制御管理者として機能するアカウント。このアカウントには次のことが可能です。 ● DVSYS パッケージを実行する ● Oracle Database Vault アクティビティを監視する。 ● Oracle Database Vault 構成に関するレポートを実行

データベース・アカウント	ロールおよび権限	説明
		する。 ノート: このアカウントでは、DVSYS 表を直接更新できません。
PSMYTHE	DV_SECANALYST	Oracle Database Vault レポートを実行するためのアカウント

関連トピック

- Oracle Database Vaultアカウントをエンタープライズ・ユーザー・アカウントとして構成
- バックアップOracle Database Vaultアカウント

親トピック: Oracle Database Vaultのスキーマ、ロールおよび アカウント

13.4 バックアップOracle Database Vaultアカウント

ベスト・プラクティスとして、DV_OWNERロールとDV_ACCTMGRロールのバックアップ・アカウントを保持することをお薦めします。

Oracle Database Vault登録プロセスでは、DV_OWNERロールとDV_ACCTMGRロールの日常用アカウントとバックアップ・アカウントの作成が必要となります。これらのロールのいずれかを付与されているユーザーが自分のパスワードを忘れた場合や組織を退職した場合に備えた安全対策として、これらのアカウントを保持し続ける必要があります。それにより、バックアップ・アカウントにログインして、パスワードの回復や新しいアカウントへのロールの付与が可能です。これらは、特権アカウント管理または組織のブレークグラス(または非常時パスワード回復)システムで安全を確保されているバックアップ・アカウントとしてのみ使用する必要があります。これらのロールのいずれかをユーザーに付与する場合は、GRANT文にWITH ADMIN OPTION句を含めます。

Oracle Database Vaultで実装されている強力な職務分離により、DV_OWNERアカウントにアクセスできなくなった場合はデータベースの再構築が必要になります。SYSアカウントは、DV_OWNERアカウントをオーバーライドできません。

関連トピック

- Oracle Database Vaultのアカウント・パスワードのリセット

親トピック: Oracle Database Vaultのスキーマ、ロールおよび アカウント

14 Oracle Database VaultレルムのAPI

DBMS_MACADM PL/SQLパッケージでは、Oracle Database Vaultレルムを構成できます。

DV_OWNERロールまたはDV_ADMINロールを付与されているユーザーのみがこれらのプロシージャを使用できます。これらのプロシージャで使用できる定数の詳細は、表20-1を参照してください。

- ADD_AUTH_TO_REALMプロシージャ
  ADD_AUTH_TO_REALMプロシージャは、所有者または参加者としてレルムにアクセスする権限をユーザーまたはロールに付与します。マルチテナント環境では、共通およびローカルの両方のレルムを認証できます。
- ADD_OBJECT_TO_REALMプロシージャ
  ADD_OBJECT_TO_REALMプロシージャは、レルム保護のために一連のオブジェクトを登録します。
- CREATE_REALMプロシージャ
  CREATE_REALMプロシージャはレルムを作成します。マルチテナント環境では、共通およびローカルの両方のレルムを作成できます。
- DELETE_AUTH_FROM_REALMプロシージャ
  DELETE_AUTH_FROM_REALMプロシージャは、レルムにアクセスするためのユーザーまたはロールの認可を削除します。
- DELETE_OBJECT_FROM_REALMプロシージャ
  DELETE_OBJECT_FROM_REALMプロシージャは、レルム保護から一連のオブジェクトを削除します。
- DELETE_REALMプロシージャ
  DELETE_REALMプロシージャは、レルム(認可されるユーザーと保護対象オブジェクトを指定するその関連構成情報を含む)を削除します。
- DELETE_REALM_CASCADEプロシージャ
  DELETE_REALM_CASCADEプロシージャは、レルム(認可されるユーザーと保護対象オブジェクトを指定するレルム関連Database Vault構成情報を含む)を削除します。
- RENAME_REALMプロシージャ
  RENAME_REALMプロシージャは、レルムの名前を変更します。名前の変更は、そのレルムが使用されているすべての箇所に反映されます。
- UPDATE_REALMプロシージャ
  UPDATE_REALMプロシージャはレルムを更新します。
- UPDATE_REALM_AUTHプロシージャ
  UPDATE_REALM_AUTHプロシージャは、レルムにアクセスするためのユーザーまたはロールの認可を更新します。

14.1 ADD_AUTH_TO_REALMプロシージャ

ADD_AUTH_TO_REALMプロシージャは、所有者または参加者としてレルムにアクセスする権限をユーザーまたはロールに付与します。マルチテナント環境では、共通およびローカルの両方のレルムを認証できます。

レルム認可の詳細は、「レルム認可について」を参照してください。

オプションで、認可を有効にする前に確認する必要のあるルール・セットを指定できます。

構文

```
DBMS_MACADM.ADD_AUTH_TO_REALM(
 realm_name    IN VARCHAR2,
 grantee       IN VARCHAR2,
```

```
rule_set_name IN VARCHAR2,
auth_options  IN NUMBER
auth_scope    IN NUMBER DEFAULT);
```

パラメータ

表14-1 ADD_AUTH_TO_REALMのパラメータ

パラメータ	説明
realm_name	レルム名。 現行のデータベース・インスタンスで既存のレルムを確認するには、「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューに問い合せます。
grantee	所有者または参加者として認可するユーザーまたはロール名。 現行のデータベース・インスタンスで既存のユーザーおよびロールを確認するには、『*Oracle Database* リファレンス』で説明されている DBA_USERS ビューおよび DBA_ROLES ビューに問い合せます。 特定のユーザーまたはロールの認可を検索するには、「DBA_DV_REALM_AUTH ビュー」で説明されている DVA_DV_REALM_AUTH ビューに問い合せます。 権限管理で使用されている既存のセキュア・アプリケーション・ロールを確認するには、DBA_DV_ROLE ビューに問い合せます。どちらも「Oracle Database Vault のデータ・ディクショナリ・ビュー」で説明されています。
rule_set_name	オプション。ランタイムでチェックするルール・セット。レルム認可は、ルール・セットの評価が TRUE の場合のみ有効です。 使用可能なルール・セットを確認するには、「DBA_DV_RULE_SET_RULE ビュー」で説明されている DBA_DV_RULE_SET ビューに問い合せます。
auth_options	オプション。レルムを認可する次のオプションのうち、1 つを指定します。 ● DBMS_MACUTL.G_REALM_AUTH_PARTICIPANT: 参加者。権限が標準の Oracle Database 権限付与プロセスを使用して付与されている場合、このアカウントまたはロールは、レルムで保護されているオブジェクトに対するアクセス、操作および作成を行うためのシステム権限または直接権限を付与できます。(デフォルト) ● DBMS_MACUTL.G_REALM_AUTH_OWNER: 所有者。このアカウントまたはロールには、レルムの参加者と同じ認可に加えて、レルム保護オブジェクトに対するレルム・セキュア・ロールおよび権限を付与または取り消す認可があります。 audit_options パラメータは、従来の監査にのみ適用されます。統合監査を有効にした場合

パラメータ	説明
	は、audit_options を使用するかわりに統合監査ポリシーを作成します。 参加者および所有者の詳細は、「レルム認可について」を参照してください。
auth_scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● 現在の PDB でローカルでレルムを認可するには、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● アプリケーション・ルートでレルムを認可するには、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

次の例では、ユーザーSYSADMをパフォーマンス統計レルムの参加者として認可します。デフォルトは参加者としてユーザーを認可することであるため、auth_optionsパラメータを省略できます。

```
BEGIN
 DBMS_MACADM.ADD_AUTH_TO_REALM(
realm_name  => 'Performance Statistics Realm',
grantee     => 'SYSADM');
END;
/
```

次の例では、ユーザーSYSADMをパフォーマンス統計レルムの所有者として設定します。

```
BEGIN
 DBMS_MACADM.ADD_AUTH_TO_REALM(
realm_name   => 'Performance Statistics Realm',
grantee      => 'SYSADM',
auth_options => DBMS_MACUTL.G_REALM_AUTH_OWNER);
END;
/
```

次の例では、ユーザーSYSADMがパフォーマンス統計レルムの所有者としての役割を果す前に、Check Conf Accessルール・セットをトリガーします。

```
BEGIN
 DBMS_MACADM.ADD_AUTH_TO_REALM(
realm_name    => 'Performance Statistics Realm',
grantee       => 'SYSADM',
rule_set_name => 'Check Conf Access',
auth_options  => DBMS_MACUTL.G_REALM_AUTH_OWNER);
END;
/
```

この例では、共通ユーザーC##HR_ADMINに共通レルムHR Statistics Realmへのアクセス権を共通で付与する方法を示します。このプロシージャを実行するユーザーはCDBルートにいる必要があり、ルール・セットはアプリケーション・ルートに存在する共通ルール・セットである必要があります。

```
BEGIN
 DBMS_MACADM.ADD_AUTH_TO_REALM(
realm_name    => 'HR Statistics Realm',
```

```
grantee       => 'C##HR_ADMIN',
rule_set_name => 'Check Access',
auth_options  => DBMS_MACUTL.G_REALM_AUTH_OWNER,
 auth_scope    => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

この例では、共通ユーザーC##HR_CLERKに共通レルムHR Statistics Realmへのアクセス権をローカルで付与する方法を示します。このプロシージャを実行するユーザーは、認可を適用するのと同じPDBにいる必要があります。既存のPDBを確認するには、DBA_PDBSデータ・ディクショナリ・ビューを問い合せます。ルール・セットはローカル・ルール・セットである必要があります。

```
BEGIN
 DBMS_MACADM.ADD_AUTH_TO_REALM(
  realm_name    => 'HR Statistics Realm',
  grantee       => 'C##HR_CLERK',
  rule_set_name => 'Check Access',
  auth_options  => DBMS_MACUTL.G_REALM_AUTH_OWNER,
  auth_scope    => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

親トピック: Oracle Database VaultレルムのAPI

14.2 ADD_OBJECT_TO_REALMプロシージャ

ADD_OBJECT_TO_REALMプロシージャは、レルム保護のために一連のオブジェクトを登録します。

構文

```
DBMS_MACADM.ADD_OBJECT_TO_REALM(
realm_name   IN VARCHAR2,
object_owner IN VARCHAR2,
object_name  IN VARCHAR2,
object_type  IN VARCHAR2);
```

パラメータ

表14-2 ADD_OBJECT_TO_REALMのパラメータ

パラメータ	説明
realm_name	レルム名。 現在のデータベース・インスタンスで既存のレルムを検索するには、DBA_DV_REALM ビューを問い合せます。
object_owner	レルムに追加するオブジェクトの所有者。ロールをレルムに追加する場合、ロールに所有者はいないため、ロールのオブジェクト所有者は%(すべて)として表示されます。 使用可能なユーザーを検索するには、DBA_USERS ビューを問い合せます。 特定のユーザーまたはロールの認可を検索するには、DVA_DV_REALM_AUTH ビューを問い合せます。

パラメータ	説明
object_name	オブジェクト名。(ワイルドカード%を使用できます)。DBMS_MACUTL.G_ALL_OBJECT 定数も使用できます。 使用可能なオブジェクトを検索するには、ALL_OBJECTS ビューを問い合せます。 既存のレルムで保護されるオブジェクトを検索するには、DBA_DV_REALM_OBJECT ビューを問い合せます。
object_type	TABLE、INDEX、ROLE などのオブジェクト・タイプ。(ワイルドカード%を使用できます)。 DBMS_MACUTL.G_ALL_OBJECT 定数も使用できます。

例

```
BEGIN
 DBMS_MACADM.ADD_OBJECT_TO_REALM(
  realm_name    => 'HR Apps',
  object_owner  => '%',
  object_name   => 'HR_SELECT_ROLE',
  object_type   => 'ROLE');
END;
/
```

関連トピック

- レルム・セキュア・オブジェクトについて

親トピック: Oracle Database VaultレルムのAPI

14.3 CREATE_REALMプロシージャ

CREATE_REALMプロシージャはレルムを作成します。マルチテナント環境では、共通およびローカルの両方のレルムを作成できます。

レルムを作成した後で、次のプロシージャを使用してレルム定義を完了します。

- ADD_OBJECT_TO_REALMプロシージャは、レルムに1つ以上のオブジェクトを登録します。
- ADD_AUTH_TO_REALMプロシージャは、レルムに対してユーザーまたはロールを認可します。

構文

```
DBMS_MACADM.CREATE_REALM(
 realm_name    IN VARCHAR2,
 description   IN VARCHAR2,
 enabled       IN VARCHAR2,
 audit_options IN NUMBER,
 realm_type    IN NUMBER DEFAULT,
 realm_scope   IN NUMBER DEFAULT
 pl_sql_stack  IN BOOLEAN DEFAULT);
```

パラメータ

表14-3 CREATE_REALMのパラメータ

パラメータ	説明
realm_name	レルム名(大/小文字混在で最大 128 文字)。 現行のデータベース・インスタンスで既存のレルムを確認するには、「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューに問い合せます。
description	レルムの目的の説明(大/小文字混在で最大 1024 文字)。
enabled	次のいずれかのオプションを指定してレルムのステータスを設定します。 ● レルム・チェックを有効にするには(デフォルト)、DBMS_MACUTL.G_YES または y ● シミュレーション・ログでの違反の取得など、すべてのレルム・チェックを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s
audit_options	レルムを監査する次のオプションのうち、1 つを指定します。 ● DBMS_MACUTL.G_REALM_AUDIT_OFF: レルムの監査を無効にします(デフォルト)。 ● DBMS_MACUTL.G_REALM_AUDIT_FAIL: 認可されていないユーザーがレルムによって保護されているオブジェクトの変更を試行するというようなレルム違反が発生した場合に、監査レコードが作成されます。 ● DBMS_MACUTL.G_REALM_AUDIT_SUCCESS: レルムで保護されているオブジェクトに対する認可されたアクティビティに関する監査レコードが作成されます。 ● DBMS_MACUTL.G_REALM_AUDIT_FAIL + DBMS_MACUTL.G_REALM_AUDIT_SUCCESS: レルムで保護されているオブジェクトに対する認可されたアクティビティおよび認可されていないアクティビティに関する監査レコードが作成されます。 audit_options パラメータは、従来の監査にのみ適用されます。統合監査を有効にした場合は、audit_options を使用するかわりに統合監査ポリシーを作成します。
realm_type	次のオプションのいずれかを指定します。 ● 0: 必須レルムのチェックを無効にします。 ● 1: レルムのオブジェクトに対する必須レルムのチェックを有効にします。レルム所有者またはレルム参加者のみが、レルム内のオブジェクトにアクセスできます。レルム所有者

パラメータ	説明
	や参加者ではないオブジェクト所有者およびオブジェクト権限ユーザーはアクセスできません。 必須レルムの詳細は、「必須レルムによるレルム内のオブジェクトへのユーザー・アクセスの制限」も参照してください。
realm_scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● レルムが現在の PDB でローカルである必要がある場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1)。 ● レルムがアプリケーション・ルート内にある必要がある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)。この設定では、関連付けられたすべての PDB 内のレルムが複製されます。 アプリケーション・ルートで共通レルムを作成し、関連付けられた PDB にそれを表示できるようにする場合は、アプリケーションを同期させる必要があります。たとえば: ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;
pl_sql_stack	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックを記録するかどうかを指定します。PL/SQL スタックを記録する場合は TRUE と入力し、記録しない場合は FALSE と入力します。デフォルトは FALSE です。

例

次の例では、失敗したアクセスと成功したアクセスの両方を追跡するよう監査を設定し、必須レルム・チェックを使用し、PL/SQLスタックを記録する、有効化されたレルムの作成方法を示します。

```
BEGIN
 DBMS_MACADM.CREATE_REALM(
  realm_name     => 'HR Apps',
  description    => 'Realm to protect the HR schema',
  enabled        => DBMS_MACUTL.G_YES,
  audit_options  => DBMS_MACUTL.G_REALM_AUDIT_OFF,
  realm_type     => 1,
  pl_sql_stack   => TRUE);
END;
/
```

この例では、前の例を少し変更したものを作成する方法を示しますが、アプリケーション・ルートにある共通レルムとして作成する方法となります。このレルムを作成するユーザーは、共通ユーザーである必要があり、アプリケーション・ルートでプロシージャを実行する必要があります。

```
BEGIN
 DBMS_MACADM.CREATE_REALM(
  realm_name     => 'HR Apps',
  description    => 'Realm to protect the HR schema',
  enabled        => DBMS_MACUTL.G_YES,
```

```
  audit_options  => DBMS_MACUTL.G_REALM_AUDIT_OFF,
  realm_type     => 1,
  realm_scope    => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

この例では、前の例のローカル・バージョンを作成する方法を示します。このレルムを作成するユーザーは、レルムがあるPDBにいる必要があります。既存のPDBを確認するには、DBA_PDBSデータ・ディクショナリ・ビューを問い合せます。

```
BEGIN
 DBMS_MACADM.CREATE_REALM(
  realm_name     => 'HR Apps',
  description    => 'Realm to protect the HR schema',
  enabled        => DBMS_MACUTL.G_YES,
  audit_options  => DBMS_MACUTL.G_REALM_AUDIT_OFF,
  realm_type     => 1,
  realm_scope    => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

関連項目:

例20-1

親トピック: Oracle Database VaultレルムのAPI

14.4 DELETE_AUTH_FROM_REALMプロシージャ

DELETE_AUTH_FROM_REALMプロシージャは、レルムにアクセスするためのユーザーまたはロールの認可を削除します。

構文

```
DBMS_MACADM.DELETE_AUTH_FROM_REALM(
 realm_name     IN VARCHAR2,
 grantee        IN VARCHAR2,
 auth_scope     IN NUMBER DEFAULT);
```

パラメータ

表14-4 DELETE_AUTH_FROM_REALMのパラメータ

パラメータ	説明
realm_name	レルム名。 現行のデータベース・インスタンスで既存のレルムを確認するには、「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューに問い合せます
grantee	ユーザーまたはロール名。 特定のユーザーまたはロールの認可を検索するには、「DBA_DV_REALM_AUTH ビュー」で説明されている DVA_DV_REALM_AUTH ビューに問い合せます。
auth_scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オ

パラメータ	説明
	プションは次のとおりです。 ● レルムが現在の PDB のローカルで認可されている場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● レルムがアプリケーション・ルートで認可されている場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2

例

```
BEGIN
DBMS_MACADM.DELETE_AUTH_FROM_REALM(
 realm_name     => 'HR Apps',
 grantee        => 'PSMITH',
 auth_scope     => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

親トピック: Oracle Database VaultレルムのAPI

14.5 DELETE_OBJECT_FROM_REALMプロシージャ

DELETE_OBJECT_FROM_REALMプロシージャは、レルム保護から一連のオブジェクトを削除します。

構文

```
DBMS_MACADM.DELETE_OBJECT_FROM_REALM(
realm_name   IN VARCHAR2,
object_owner IN VARCHAR2,
object_name  IN VARCHAR2,
object_type  IN VARCHAR2);
```

パラメータ

表14-5 DELETE_OBJECT_FROM_REALMのパラメータ

パラメータ	説明
realm_name	レルム名。 現行のデータベース・インスタンスで既存のレルムを確認するには、「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューに問い合せます。
object_owner	レルムに追加されたオブジェクトの所有者。 使用可能なユーザーを確認するには、『Oracle Database リファレンス』で説明されている DBA_USERS ビューを問い合せます。
object_name	オブジェクト名。(ワイルドカード%を使用できます。ワイルドカード%の例外については、「レルム・セキュア・オブジェクトについて」の「オブジェクト名」を参照)。

パラメータ	説明
	DBMS_MACUTL.G_ALL_OBJECT 定数も使用できます。 既存のレルムに保護されているオブジェクトを確認するには、「DBA_DV_REALM_OBJECT ビュー」で説明されている DBA_DV_REALM_OBJECT ビューに問い合せます。
object_type	TABLE、INDEX、ROLE などのオブジェクト・タイプ。(ワイルドカード%を使用できます。ワイルドカード%の例外については、「レルム・セキュア・オブジェクトについて」の「オブジェクト・タイプ」を参照)。 DBMS_MACUTL.G_ALL_OBJECT 定数も使用できます。

例

```
BEGIN
 DBMS_MACADM.DELETE_OBJECT_FROM_REALM(
realm_name   => 'Performance Statistics Realm',
object_owner => 'SYS',
object_name  => 'GATHER_SYSTEM_STATISTICS',
object_type  => 'ROLE');
END;
/
```

親トピック: Oracle Database VaultレルムのAPI

14.6 DELETE_REALMプロシージャ

DELETE_REALMプロシージャは、レルム(認可されるユーザーと保護対象オブジェクトを指定するその関連構成情報を含む)を削除します。

このプロシージャでは、実際のデータベース・オブジェクトまたはユーザーは削除されません。

レルムに対して認可されているユーザーを確認するには、DBA_DV_REALM_AUTHビューに問い合せます。レルムで保護されるオブジェクトを確認するには、DBA_DV_REALM_OBJECTビューに問い合せます。これらのビューについては、「Oracle Database Vaultのデータ・ディクショナリ・ビュー」で説明します。

構文

```
DBMS_MACADM.DELETE_REALM(
realm_name IN VARCHAR2);
```

パラメータ

表14-6 DELETE_REALMのパラメータ

パラメータ	説明
realm_name	レルム名。 現行のデータベース・インスタンスで既存のレルムを確認するには、「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューに問い合せます。

例

```
EXEC DBMS_MACADM.DELETE_REALM('Performance Statistics Realm');
```

親トピック: Oracle Database VaultレルムのAPI

14.7 DELETE_REALM_CASCADEプロシージャ

DELETE_REALM_CASCADEプロシージャは、レルム(認可されるユーザーと保護対象オブジェクトを指定するレルム関連Database Vault構成情報を含む)を削除します。

DBA_DV_REALM_AUTHビューは、レルムで認可されるユーザーを示し、DBA_DV_REALM_OBJECTビューは、保護対象オブジェクトを示します。

実際のデータベース・オブジェクトまたはユーザーは削除されません。このプロシージャは、DELETE_REALMプロシージャと同様に動作します。(以前のリリースでは同じではありませんでしたが、現在は同じです。どちらも下位互換性のために保持されています。)レルム関連のオブジェクトのリストを確認するには、DBA_DV_REALMビューに問い合せます。その認可を確認するには、DBA_DV_REALM_AUTHに問い合せます。どちらも「Oracle Database Vaultのデータ・ディクショナリ・ビュー」で説明されています。

構文

```
DBMS_MACADM.DELETE_REALM_CASCADE(
realm_name IN VARCHAR2);
```

パラメータ

表14-7 DELETE_REALM_CASCADEのパラメータ

パラメータ	説明
realm_name	レルム名。 現行のデータベース・インスタンスで既存のレルムを確認するには、「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューに問い合せます。

例

```
EXEC DBMS_MACADM.DELETE_REALM_CASCADE('Performance Statistics Realm');
```

親トピック: Oracle Database VaultレルムのAPI

14.8 RENAME_REALMプロシージャ

RENAME_REALMプロシージャは、レルムの名前を変更します。名前の変更は、そのレルムが使用されているすべての箇所に反映されます。

構文

```
DBMS_MACADM.RENAME_REALM(
 realm_name   IN VARCHAR2,
 new_name     IN VARCHAR2);
```

パラメータ

表14-8 RENAME_REALMのパラメータ

パラメータ	説明
realm_name	現在のレルム名。 現行のデータベース・インスタンスで既存のレルムを確認するには、「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューに問い合せます。
new_name	新しいレルム名(大/小文字混在で最大 128 文字)。

例

```
BEGIN
 DBMS_MACADM.RENAME_REALM(
realm_name => 'Performance Statistics Realm',
new_name   => 'Sector 2 Performance Statistics Realm');
END;
/
```

親トピック: Oracle Database VaultレルムのAPI

14.9 UPDATE_REALMプロシージャ

UPDATE_REALMプロシージャはレルムを更新します。

レルムの現在の設定について情報を確認するには、「DVSYS.DV$REALMビュー」で説明されている DVSYS.DV$REALMビューに問い合せます。

構文

```
DBMS_MACADM.UPDATE_REALM(
 realm_name    IN VARCHAR2,
 description   IN VARCHAR2,
 enabled       IN VARCHAR2,
 audit_options IN NUMBER DEFAULT NULL,
 realm_type    IN NUMBER DEFAULT NULL
 pl_sql_stack  IN BOOLEAN DEFAULT NULL);
```

パラメータ

表14-9 UPDATE_REALMのパラメータ

パラメータ	説明
realm_name	レルム名。 現行のデータベース・インスタンスで既存のレルムを確認するには、「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューに問い合せます。
description	レルムの目的の説明(大/小文字混在で最大 1024 文字)。

パラメータ	説明
enabled	次のいずれかのオプションを指定してレルムのステータスを設定します。 ● レルム・チェックを有効にするには、DBMS_MACUTL.G_YES または y ● シミュレーション・ログでの違反の取得など、すべてのレルム・チェックを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s enabled のデフォルトは設定済の値であり、DBA_DV_REALM データ・ディクショナリ・ビューに問い合せることで確認できます。
audit_options	レルムを監査する次のオプションのうち、1 つを指定します。 ● DBMS_MACUTL.G_REALM_AUDIT_OFF: レルムの監査を無効にします。 ● DBMS_MACUTL.G_REALM_AUDIT_FAIL: 認可されていないユーザーがレルムによって保護されているオブジェクトの変更を試行するというようなレルム違反が発生した場合に、監査レコードが作成されます。 ● DBMS_MACUTL.G_REALM_AUDIT_SUCCESS: レルムで保護されているオブジェクトに対する認可されたアクティビティに関する監査レコードが作成されます。 ● DBMS_MACUTL.G_REALM_AUDIT_FAIL + DBMS_MACUTL.G_REALM_AUDIT_SUCCESS: レルムで保護されているオブジェクトに対する認可されたアクティビティおよび認可されていないアクティビティに関する監査レコードが作成されます。 audit_options のデフォルトは設定済の値であり、DBA_DV_REALM データ・ディクショナリ・ビューに問い合せることで確認できます。 audit_options パラメータは、従来の監査にのみ適用されます。統合監査を有効にした場合は、audit_options を使用するかわりに統合監査ポリシーを作成します。
realm_type	realm_type パラメータを指定しない場合、Oracle Database Vault は現在の realm_type 設定を更新しません。 次のオプションのいずれかを指定します。 ● 0: 必須レルムのチェックのない通常のレルムになるようにレルムを設定します。 ● 1: レルムのオブジェクトに対する必須レルムのチェックを有効にします。レルム所有者またはレルム参加者のみが、レルム内のオブジェクトにアクセスできます。レルム所有者や参加

パラメータ	説明
	者ではないオブジェクト所有者およびオブジェクト権限ユーザーはアクセスできません。 必須レルムの詳細は、「必須レルムによるレルム内のオブジェクトへのユーザー・アクセスの制限」も参照してください。
pl_sql_stack	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックが記録されているかどうかを示します。TRUE は PL/SQL スタックが記録されていることを示し、FALSE は PL/SQL スタックが記録されていないことを示します。

例

```
BEGIN
 DBMS_MACADM.UPDATE_REALM(
realm_name    => 'Sector 2 Performance Statistics Realm',
description   => 'Realm to measure performance for Sector 2 applications',
enabled       => DBMS_MACUTL.G_YES,
audit_options => DBMS_MACUTL.G_REALM_AUDIT_FAIL + DBMS_MACUTL.G_REALM_AUDIT_SUCCESS),
realm_type    => 1);
END;
/
```

親トピック: Oracle Database VaultレルムのAPI

14.10 UPDATE_REALM_AUTHプロシージャ

UPDATE_REALM_AUTHプロシージャは、レルムにアクセスするためのユーザーまたはロールの認可を更新します。

構文

```
DBMS_MACADM.UPDATE_REALM_AUTH(
 realm_name    IN VARCHAR2,
 grantee       IN VARCHAR2,
 rule_set_name IN VARCHAR2,
 auth_options  IN NUMBER,
 auth_scope    IN NUMBER DEFAULT);
```

パラメータ

表14-10 UPDATE_REALM_AUTHのパラメータ

パラメータ	説明
realm_name	レルム名。 現行のデータベース・インスタンスで既存のレルムを確認するには、「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューに問い合せます。
grantee	ユーザーまたはロール名。 現行のデータベース・インスタンス内の使用可能なユーザーおよびロールを確認するには、『*Oracle Database* リファレンス』で説明されている DBA_USERS ビューおよび DBA_ROLES ビューに問

パラメータ	説明
	い合せます。 特定のユーザーまたはロールの認可を検索するには、「DBA_DV_REALM_AUTH ビュー」で説明されている DVA_DV_REALM_AUTH ビューに問い合せます。 権限管理で使用されている既存のセキュア・アプリケーション・ロールを確認するには、「DBA_DV_ROLE ビュー」で説明されている DBA_DV_ROLE ビューに問い合せます。
rule_set_name	オプション。ランタイムでチェックするルール・セット。レルム認可は、ルール・セットの評価が TRUE の場合のみ有効です。 使用可能なルール・セットを確認するには、DBA_DV_RULE_SET ビューに問い合せます。ルール・セットに関連付けられているルールを検索するには、DBA_DB_RULE_SET_RULE ビューに問い合せます。どちらも「Oracle Database Vault のデータ・ディクショナリ・ビュー」で説明されています。
auth_options	オプション。レルムを認可する次のオプションのうち、1 つを指定します。 ● DBMS_MACUTL.G_REALM_AUTH_PARTICIPANT: 参加者。権限が標準の Oracle Database 権限付与プロセスを使用して付与されている場合、このアカウントまたはロールは、レルムで保護されているオブジェクトに対するアクセス、操作および作成を行うためのシステム権限または直接権限を付与できます。 ● DBMS_MACUTL.G_REALM_AUTH_OWNER: 所有者。このアカウントまたはロールには、レルムの参加者と同じ認可に加えて、レルム保護オブジェクトに対するレルム・セキュア・ロールおよび権限を付与または取り消す認可があります。1 つのレルムに複数の所有者を設定できます。 auth_options 値のデフォルトは設定済の値であり、DBA_DV_REALM_AUTH データ・ディクショナリ・ビューに問い合せることで確認できます。 audit_options パラメータは、従来の監査にのみ適用されます。統合監査を有効にした場合は、audit_options を使用するかわりに統合監査ポリシーを作成します。
realm_auth	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● レルムが現在の PDB のローカルで認可されている場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● レルムがアプリケーション・ルートで認可されている場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

```
BEGIN
 DBMS_MACADM.UPDATE_REALM_AUTH(
realm_name    => 'Sector 2 Performance Statistics Realm',
grantee       => 'SYSADM',
rule_set_name => 'Check Conf Access',
auth_options  => DBMS_MACUTL.G_REALM_AUTH_OWNER);
END;
/
```

親トピック: Oracle Database VaultレルムのAPI

15 Oracle Database Vaultルール・セットのAPI

DBMS_MACADM PL/SQLパッケージをおよび一連のOracle Database Vaultルール・ファンクションを使用すると、ルール・セットを管理できます。

- DBMS_MACADMルール・セット・プロシージャ
  DBMS_MACADMルール・セット・プロシージャにより、ルール・セット、およびこれらのルール・セット内に入れられる個々のルールを構成できます。
- Oracle Database VaultのPL/SQLルール・セット・ファンクション
  Oracle Database Vaultには、ルール・セットで保護されるSQL文を検査するためにルール・セットで使用されるファンクションが用意されています。

15.1 DBMS_MACADMルール・セットのプロシージャ

DBMS_MACADMルール・セット・プロシージャにより、ルール・セット、およびこれらのルール・セット内に入れられる個々のルールを構成できます。

DV_OWNERロールまたはDV_ADMINロールを付与されているユーザーのみがこれらのプロシージャを使用できます。

- ADD_RULE_TO_RULE_SETプロシージャ
  ADD_RULE_TO_RULE_SETプロシージャは、ルールをルール・セットに追加し、ルール・セットの評価時にルールをチェックできるようにします。
- CREATE_RULEプロシージャ
  CREATE_RULEプロシージャは、後でルール・セットに追加できるコマンド・ルールを作成します。
- CREATE_RULE_SETプロシージャ
  CREATE_RULE_SETプロシージャはルール・セットを作成します。
- DELETE_RULEプロシージャ
  DELETE_RULEプロシージャはルールを削除します。
- DELETE_RULE_FROM_RULE_SETプロシージャ
  DELETE_RULE_FROM_RULE_SETプロシージャは、ルールをルール・セットから削除します。
- DELETE_RULE_SETプロシージャ
  DELETE_RULE_SETプロシージャはルール・セットを削除します。
- RENAME_RULEプロシージャ
  RENAME_RULEプロシージャは、ルールの名前を変更し、その名前変更は、そのルールが使用されているすべての箇所に反映されます
- RENAME_RULE_SETプロシージャ
  RENAME_RULE_SETプロシージャは、ルール・セットの名前を変更し、その名前変更は、そのルール・セットが使用されているすべての箇所に反映されます。
- UPDATE_RULEプロシージャ
  UPDATE_RULEプロシージャはルールを更新します。
- UPDATE_RULE_SETプロシージャ
  UPDATE_RULE_SETプロシージャはルール・セットを更新します。

関連トピック

- ルール・セットの構成

- Oracle Database Vaultユーティリティの API

親トピック: Oracle Database Vaultルール・セットの API

15.1.1 ADD_RULE_TO_RULE_SETプロシージャ

ADD_RULE_TO_RULE_SETプロシージャは、ルールをルール・セットに追加し、ルール・セットの評価時にルールをチェックするようにできます。

構文

```
DBMS_MACADM.ADD_RULE_TO_RULE_SET(
 rule_set_name  IN VARCHAR2,
 rule_name      IN VARCHAR2,
 rule_order     IN NUMBER,
 enabled        IN VARCHAR2,
 scope          IN NUMBER DEFAULT);
```

パラメータ

表15-1 ADD_RULE_TO_RULE_SETのパラメータ

パラメータ	説明
rule_set_name	ルール・セット名。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
rule_name	ルール・セットに追加するルール。 既存のルールを確認するには、「DBA_DV_RULE ビュー」で説明されている DBA_DV_RULE ビューに問い合せます。 ルール・セットに関連付けられているルールを確認するには、「DBA_DV_RULE ビュー」で説明されている DBA_DV_RULE_SET_RULE を使用します。
rule_order	このリリースには適用されませんが、ADD_RULE_TO_RULE_SET プロシージャを機能させるには値を指定する必要があります。1 を入力します。
enabled	オプション。ルール・セットの評価時にルールをチェックする必要があるかどうかを判断します。使用される値は、次のとおりです。 ● DBMS_MACUTL.G_YES (デフォルト)。ルール・セットの評価時にルールをチェックできるようにします。 ● DBMS_MACUTL.G_NO は、ルール・セットの評価時にルールをチェックできないようにします。

パラメータ	説明
	詳細は、表 20-1 を参照してください。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● ルールおよびルール・セットが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● ルールおよびルール・セットがアプリケーション・ルートにある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

次の例では、ルールをルール・セットに追加し、enabledパラメータを省略して、ルール・セットの評価時に自動的にルール・チェックが有効化されるようにします。

```
BEGIN
 DBMS_MACADM.ADD_RULE_TO_RULE_SET(
rule_set_name => 'Limit_DBA_Access',
rule_name     => 'Restrict DROP TABLE operations',
rule_order    => 1);
END;
/
```

次の例では、ルールをルール・セットに追加しますが、ルール・チェックを無効化します。

```
BEGIN
 DBMS_MACADM.ADD_RULE_TO_RULE_SET(
rule_set_name => 'Limit_DBA_Access',
rule_name     => 'Check UPDATE operations',
rule_order    => 1,
enabled       => DBMS_MACUTL.G_NO);
END;
/
```

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.1.2 CREATE_RULEプロシージャ

CREATE_RULEプロシージャは、後でルール・セットに追加できるコマンド・ルールを作成します。

マルチテナント環境では、共通およびローカルの両方のルールを作成できます。

構文

```
DBMS_MACADM.CREATE_RULE(
 rule_name  IN VARCHAR2,
 rule_expr  IN VARCHAR2
 scope      IN NUMBER DEFAULT);
```

パラメータ

表15-2 CREATE_RULEのパラメータ

パラメータ	説明
rule_name	ルール名(大/小文字混在で最大 128 文字)。空白を使用できます。 現行のデータベース・インスタンスで既存のルールを確認するには、「DBA_DV_RULE ビュー」で説明されている DBA_DV_RULE ビューに問い合せます。 ルール・セットに関連付けられているルールを確認するには、「DBA_DV_RULE_SET_RULE ビュー」で説明されている DBA_DV_RULE_SET_RULE に問い合せます。
rule_expr	PL/SQL BOOLEAN 式。 式に引用符が含まれる場合、二重引用符は使用しないでください。その場合は、2 つの一重引用符を使用します。式全体を一重引用符で囲んでください。たとえば: 'TO_CHAR(SYSDATE,''HH24'') = ''12''' ルール式の詳細は、「新規ルールの作成」を参照してください。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● ルールがアプリケーション・ルートにある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

次の例では、現行セッション・ユーザーがSYSADMであるかどうかをチェックするローカル・ルール式の作成方法を示します。このプロシージャを実行するユーザーは、ルールおよびそのルール・セットが存在するのと同じPDBにいる必要があります。既存のPDBを確認するには、show pdbsコマンドを実行します。ルールおよびルール・セットはローカルである必要があります。

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name  => 'Check UPDATE operations',
  rule_expr  =>'SYS_CONTEXT(''USERENV'',''SESSION_USER'') = ''SYSADM''',
  scope      => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

この例では、前の例のマルチテナント環境共通バージョンを示します。このプロシージャを実行するユーザーはCDBルートにいる必要があり、ルールおよびその関連付けられたルール・セットは共通である必要があります。ルールは、アプリケーション・ルートに存在することになります。

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name  => 'Check UPDATE operations',
  rule_expr  =>'SYS_CONTEXT(''USERENV'',''SESSION_USER'') = ''SYSADM''',
  scope      => DBMS_MACUTL.G_SCOPE_COMMON);
```

```
END;
/
```

この例では、パブリック・スタンドアロン・ファンクションOLS_LABEL_DOMINATESを使用してhr_ols_pol Oracle Label Securityポリシーのセッション・ラベルがhsラベルより優位にあるか同等であるかを確認するルール式の作成方法を示します。値0は、falseである場合を示します。(同等であるかどうかをチェックするには、1を示します。)

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name  => 'Check OLS Factor',
  rule_expr  => 'OLS_LABEL_DOMINATES(''hr_ols_pol'', ''hs'') = 1');
END;
/
```

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.1.3 CREATE_RULE_SETプロシージャ

CREATE_RULE_SETプロシージャはルール・セットを作成します。

ルール・セットを作成した後で、CREATE_RULEおよびADD_RULE_TO_RULE_SETプロシージャを使用してルールを作成し、ルール・セットに追加します。

構文

```
DBMS_MACADM.CREATE_RULE_SET(
 rule_set_name    IN VARCHAR2,
 description      IN VARCHAR2,
 enabled          IN VARCHAR2,
 eval_options     IN NUMBER,
 audit_options    IN NUMBER,
 fail_options     IN NUMBER,
 fail_message     IN VARCHAR2,
 fail_code        IN NUMBER,
 handler_options  IN NUMBER,
 handler          IN VARCHAR2,
 is_static        IN BOOLEAN DEFAULT,
 scope            IN NUMBER DEFAULT);
```

パラメータ

表15-3 CREATE_RULE_SETのパラメータ

パラメータ	説明
rule_set_name	ルール・セット名(大/小文字混在で最大 128 文字)。空白を使用できます。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
description	ルール・セットの目的の説明(大/小文字混在で最大 1024 文字)。
enabled	DBMS_MACUTL.G_YES(Yes)では、ルール・セットが有効になり、DBMS_MACUTL.G_NO(No)では無効になります。デフォルト値は DBMS_MACUTL.G_YES で

パラメータ	説明
	す。
eval_options	ルール・セットに複数のルールを割り当てる場合は、次の設定のいずれかを入力します。 ● DBMS_MACUTL.G_RULESET_EVAL_ALL: ルール・セット自体が True(デフォルト)と評価されるためには、ルール・セットのルールがすべて True と評価される必要があります。 ● DBMS_MACUTL.G_RULESET_EVAL_ANY: ルール・セット自体が True と評価されるためには、ルール・セットの少なくとも 1 つのルールが True と評価される必要があります。
audit_options	次の設定のいずれかを選択します。 ● DBMS_MACUTL.G_RULESET_AUDIT_OFF: ルール・セットの監査を無効にします(デフォルト)。 ● DBMS_MACUTL.G_RULESET_AUDIT_FAIL: ルール・セット違反が発生した場合に監査レコードを作成します。 ● DBMS_MACUTL.G_RULESET_AUDIT_SUCCESS: ルール・セット評価が合格の場合に監査レコードを作成します。 ● DBMS_MACUTL.G_RULESET_AUDIT_FAIL + DBMS_MACUTL.G_RULESET_AUDIT_SUCCESS: ルール・セット評価が合格と不合格のどちらの場合も監査レコードを作成します。 audit_options パラメータは、従来の監査にのみ適用されます。統合監査を有効にした場合は、audit_options を使用するかわりに統合監査ポリシーを作成します。
fail_options	エラーをレポートするオプション: ● DBMS_MACUTL.G_RULESET_FAIL_SHOW: エラー・メッセージを表示します(デフォルト)。 ● DBMS_MACUTL.G_RULESET_FAIL_SILENT: エラー・メッセージを表示しません。
fail_message	大/小文字混在の最大 80 文字で、失敗を示すエラー・メッセージを入力し、fail_code で指定した失敗コードに関連付けます。
fail_code	-20000 から-20999 または 20000 から 20999 の範囲の数値を入力し、fail_message パラメータに関連付けます。

パラメータ	説明
handler_options	次の設定のいずれかを選択します。 ● DBMS_MACUTL.G_RULESET_HANDLER_OFF: エラー処理を無効にします(デフォルト)。 ● DBMS_MACUTL.G_RULESET_HANDLER_FAIL: ルール・セット失敗時にハンドラをコールします。 ● DBMS_MACUTL.G_RULESET_HANDLER_SUCCESS: ルール・セット成功時にハンドラをコールします。
handler	カスタム・イベント・ハンドラ・ロジックを定義する PL/SQL ファンクションまたはプロシージャの名前。
is_static	オプション。アクセスされる際にルール・セットが評価される頻度を決定します。デフォルトは FALSE です。 ● TRUE: ルール・セットはユーザー・セッション中に 1 回、評価されます。その後、値は再利用されます。 ● FALSE: ルール・セットは毎回評価されます。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● ルール・セットを現在の PDB でローカルにする場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● ルール・セットをアプリケーション・ルートに存在させる場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

次の例では、ルール・セット自体でtrueに評価されるには少なくとも1つのルールでtrueに評価される必要があるように設定されている、失敗した試みと成功した試みの両方を監査する、有効化されたルール・セットを作成します。エラー・メッセージは表示しませんが、失敗の追跡に失敗コード20461を使用します。また、ルール・セットに対して違反がある場合は、ハンドラを使用して適切なユーザーに電子メール・アラートを送信します。

```
BEGIN
 DBMS_MACADM.CREATE_RULE_SET(
rule_set_name     => 'Limit_DBA_Access',
description       => 'DBA access through predefined processes',
enabled           => DBMS_MACUTL.G_YES,
eval_options      => DBMS_MACUTL.G_RULESET_EVAL_ANY,
audit_options     => DBMS_MACUTL.G_RULESET_AUDIT_FAIL +
DBMS_MACUTL.G_RULESET_AUDIT_SUCCESS,
fail_options      => DBMS_MACUTL.G_RULESET_FAIL_SILENT,
fail_message      => '',
```

```
fail_code        => 20461,
handler_options  => DBMS_MACUTL.G_RULESET_HANDLER_FAIL,
handler          => 'dbavowner.email_alert',
is_static        => TRUE);
END;
/
```

このルール・セットでは、失敗メッセージや失敗コードは使用されず、ハンドラも使用されません。このルール・セットはマルチテナント環境のアプリケーション・ルートに存在することになるため、このプロシージャを実行するユーザーは、アプリケーション・ルートにいる必要があります。このルール・セットに関連付けられているルールまたはコマンド・ルールは、共通である必要があります。

```
BEGIN
 DBMS_MACADM.CREATE_RULE_SET(
 rule_set_name    => 'Check_HR_Access',
 description      => 'Checks for failed access attempts to the HR schema',
 enabled          => DBMS_MACUTL.G_YES,
 eval_options     => DBMS_MACUTL.G_RULESET_EVAL_ANY,
 audit_options    => DBMS_MACUTL.G_RULESET_AUDIT_FAIL,
 fail_options     => DBMS_MACUTL.G_RULESET_FAIL_SILENT,
 fail_message     => '',
 fail_code        => '',
 handler_options  => DBMS_MACUTL.G_RULESET_HANDLER_OFF,
 handler          => '',
 is_static        => TRUE,
 scope            => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

このルール・セットは、前のルール・セットのローカル・バージョンです。このルール・セットを作成するユーザーは、このルール・セットが存在することになるPDBにいる必要があります。既存のPDBを確認するには、DBA_PDBSデータ・ディクショナリ・ビューを問い合せます。このルール・セットに関連付けられているルールまたはコマンド・ルールは、ローカルである必要があります。

```
BEGIN
 DBMS_MACADM.CREATE_RULE_SET(
 rule_set_name    => 'Check_HR_Access',
 description      => 'Checks for failed access attempts to the HR schema',
 enabled          => DBMS_MACUTL.G_YES,
 eval_options     => DBMS_MACUTL.G_RULESET_EVAL_ANY,
 audit_options    => DBMS_MACUTL.G_RULESET_AUDIT_FAIL,
 fail_options     => DBMS_MACUTL.G_RULESET_FAIL_SILENT,
 fail_message     => '',
 fail_code        => '',
 handler_options  => DBMS_MACUTL.G_RULESET_HANDLER_OFF,
 handler          => '',
 is_static        => TRUE,
 scope            => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

関連項目:

例20-2

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.1.4 DELETE_RULEプロシージャ

DELETE_RULEプロシージャはルールを削除します。

構文

```
DBMS_MACADM.DELETE_RULE(
 rule_name IN VARCHAR2);
```

パラメータ

表15-4 DELETE_RULEのパラメータ

パラメータ	説明
rule_name	ルール名。 現行のデータベース・インスタンスで既存のルールを確認するには、「DBA_DV_RULE ビュー」で説明されている DBA_DV_RULE ビューに問い合せます。 ルール・セットに関連付けられているルールを確認するには、「DBA_DV_RULE_SET_RULE ビュー」で説明されている DBA_DV_RULE_SET_RULE に問い合せます。

例

```
EXEC DBMS_MACADM.DELETE_RULE('Check UPDATE operations');
```

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.1.5 DELETE_RULE_FROM_RULE_SETプロシージャ

DELETE_RULE_FROM_RULE_SETプロシージャは、ルールをルール・セットから削除します。

構文

```
DBMS_MACADM.DELETE_RULE_FROM_RULE_SET(
 rule_set_name IN VARCHAR2,
 rule_name     IN VARCHAR2);
```

パラメータ

表15-5 DELETE_RULE_FROM_RULE_SETのパラメータ

パラメータ	説明
rule_set_name	ルール・セット名。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
rule_name	ルール・セットから削除するルール。 現行のデータベース・インスタンスで既存のルールを確認するには、「DBA_DV_RULE ビュー」で説明されている DBA_DV_RULE ビューに問い合せます。 ルール・セットに関連付けられているルールを確認するには、「DBA_DV_RULE_SET_RULE

パラメータ	説明
	ビュー」で説明されている DBA_DV_RULE_SET_RULE に問い合せます。

例

```
BEGIN
 DBMS_MACADM.DELETE_RULE_FROM_RULE_SET(
rule_set_name => 'Limit_DBA_Access',
rule_name     => 'Check UPDATE operations');
END;
/
```

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.1.6 DELETE_RULE_SETプロシージャ

DELETE_RULE_SETプロシージャはルール・セットを削除します。

構文

```
DBMS_MACADM.DELETE_RULE_SET(
 rule_set_name IN VARCHAR2);
```

パラメータ

表15-6 DELETE_RULE_SETのパラメータ

パラメータ	説明
rule_set_name	ルール・セット名。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。

例

```
EXEC DBMS_MACADM.DELETE_RULE_SET('Limit_DBA_Access');
```

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.1.7 RENAME_RULEプロシージャ

RENAME_RULEプロシージャは、ルールの名前を変更し、その名前変更は、そのルールが使用されているすべての箇所に反映されます

構文

```
DBMS_MACADM.RENAME_RULE(
 rule_name   IN VARCHAR2,
 new_name    IN VARCHAR2,
 scope       IN NUMBER DEFAULT);
```

パラメータ

表15-7 RENAME_RULEのパラメータ

パラメータ	説明
rule_name	現在のルール名。 現行のデータベース・インスタンスで既存のルールを確認するには、「DBA_DV_RULE ビュー」で説明されている DBA_DV_RULE ビューに問い合せます。 ルール・セットに関連付けられているルールを確認するには、「DBA_DV_RULE_SET_RULE ビュー」で説明されている DBA_DV_RULE_SET_RULE に問い合せます。
new_name	新しいルール名(大/小文字混在で最大 128 文字)。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● ルールがアプリケーション・ルートにある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

```
BEGIN
 DBMS_MACADM.RENAME_RULE(
rule_name  => 'Check UPDATE operations',
new_name   => 'Check Sector 2 Processes');
END;
/
```

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.1.8 RENAME_RULE_SETプロシージャ

RENAME_RULE_SETプロシージャは、ルール・セットの名前を変更し、その名前変更は、そのルール・セットが使用されているすべての箇所に反映されます。

構文

```
DBMS_MACADM.RENAME_RULE_SET(
 rule_set_name IN VARCHAR2,
 new_name      IN VARCHAR2,
 scope         IN NUMBER DEFAULT);
```

パラメータ

表15-8 RENAME_RULE_SETのパラメータ

パラメータ	説明

パラメータ	説明
rule_set_name	現在のルール・セット名。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
new_name	新しいルール・セット名(大/小文字混在で最大 128 文字)。空白を使用できます。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● ルール・セットが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● ルール・セットがアプリケーション・ルートにある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

```
BEGIN
 DBMS_MACADM.RENAME_RULE_SET(
rule_set_name => 'Limit_DBA_Access',
new_name      => 'Limit Sector 2 Access');
END;
/
```

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.1.9 UPDATE_RULEプロシージャ

UPDATE_RULEプロシージャはルールを更新します。

構文

```
DBMS_MACADM.UPDATE_RULE(
 rule_name  IN VARCHAR2,
 rule_expr  IN VARCHAR2);
```

パラメータ

表15-9 UPDATE_RULEのパラメータ

パラメータ	説明
rule_name	ルール名。 現行のデータベース・インスタンスで既存のルールを確認するには、「DBA_DV_RULE ビュー」で説明されている DBA_DV_RULE ビューに問い合せます。

パラメータ	説明
	ルール・セットに関連付けられているルールを確認するには、「DBA_DV_RULE_SET_RULE ビュー」で説明されている DBA_DV_RULE_SET_RULE に問い合せます。
rule_expr	PL/SQL BOOLEAN 式。 式に引用符が含まれる場合、二重引用符は使用しないでください。その場合は、2 つの一重引用符を使用します。式全体を一重引用符で囲んでください。たとえば: 'TO_CHAR(SYSDATE,''HH24'') = ''12''' ルール式の詳細は、「新規ルールの作成」を参照してください。 既存のルール式を確認するには、DBA_DV_RULE ビューに問い合せます。

例

```
BEGIN
 DBMS_MACADM.UPDATE_RULE(
rule_name  => 'Check UPDATE operations',
rule_expr  =>'SYS_CONTEXT(''USERENV'',''SESSION_USER'') = ''SYSADM'' AND
               (
                 UPPER(SYS_CONTEXT(''USERENV'',''MODULE'')) LIKE ''APPSRVR%'' OR
                 UPPER(SYS_CONTEXT(''USERENV'',''MODULE'')) LIKE ''DBAPP%'' )'
               );
END;
/
```

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.1.10 UPDATE_RULE_SETプロシージャ

UPDATE_RULE_SETプロシージャはルール・セットを更新します。

構文

```
DBMS_MACADM.UPDATE_RULE_SET(
 rule_set_name    IN VARCHAR2,
 description      IN VARCHAR2,
 enabled          IN VARCHAR2,
 eval_options     IN NUMBER,
 audit_options    IN NUMBER,
 fail_options     IN NUMBER,
 fail_message     IN VARCHAR2,
 fail_code        IN NUMBER,
 handler_options  IN NUMBER,
 handler          IN VARCHAR2,
 is_static        IN BOOLEAN DEFAULT);
```

パラメータ

表15-10 UPDATE_RULE_SETのパラメータ

パラメータ	説明

パラメータ	説明
rule_set_name	ルール・セット名。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
description	ルール・セットの目的の説明(大/小文字混在で最大 1024 文字)。
enabled	DBMS_MACUTL.G_YES(Yes)では、ルール・セット・チェックが有効になり、DBMS_MACUTL.G_NO(No)では無効になります。 enabled 設定のデフォルトは設定済の値であり、DBA_DV_RULE_SET データ・ディクショナリ・ビューに問い合せることで確認できます。
eval_options	ルール・セットに複数のルールを割り当てる場合は、次の設定のいずれかを入力します。 ● DBMS_MACUTL.G_RULESET_EVAL_ALL: ルール・セット自体が True と評価されるためには、ルール・セットのルールがすべて True と評価される必要があります。 ● DBMS_MACUTL.G_RULESET_EVAL_ANY: ルール・セット自体が True と評価されるためには、ルール・セットの少なくとも 1 つのルールが True と評価される必要があります。 eval_options のデフォルトは設定済の値であり、DBA_DV_RULE_SET データ・ディクショナリ・ビューに問い合せることで確認できます。
audit_options	次の設定のいずれかを選択します。 ● DBMS_MACUTL.G_RULESET_AUDIT_OFF: ルール・セットの監査を無効にします。 ● DBMS_MACUTL.G_RULESET_AUDIT_FAIL: ルール・セット違反が発生した場合に監査レコードを作成します。 ● DBMS_MACUTL.G_RULESET_AUDIT_SUCCESS: ルール・セット評価が合格の場合に監査レコードを作成します。 ● DBMS_MACUTL.G_RULESET_AUDIT_FAIL + DBMS_MACUTL.G_RULESET_AUDIT_SUCCESS: ルール・セット評価が合格と不合格のどちらの場合も監査レコードを作成します。 audit_options のデフォルトは設定済の値であり、DBA_DV_RULE_SET データ・ディクショ

パラメータ	説明
	ナリ・ビューに問い合せることで確認できます。
fail_options	エラーをレポートするオプション: ● DBMS_MACUTL.G_RULESET_FAIL_SHOW: エラー・メッセージを表示します。 ● DBMS_MACUTL.G_RULESET_FAIL_SILENT: エラー・メッセージを表示しません。 fail_options のデフォルトは設定済の値であり、DBA_DV_RULE_SET データ・ディクショナリ・ビューに問い合せることで確認できます。
fail_message	失敗を示すエラー・メッセージ。大/小文字混在の最大 80 文字で、fail_code で指定した失敗コードに関連付けます。
fail_code	-20000 から-20999 または 20000 から 20999 の範囲の数値を入力し、fail_message パラメータに関連付けます。
handler_options	次の設定のいずれかを選択します。 ● DBMS_MACUTL.G_RULESET_HANDLER_OFF: エラー処理を無効にします。 ● DBMS_MACUTL.G_RULESET_HANDLER_FAIL: ルール・セット失敗時にハンドラをコールします。 ● DBMS_MACUTL.G_RULESET_HANDLER_SUCCESS: ルール・セット成功時にハンドラをコールします。 handler_options のデフォルトは設定済の値であり、DBA_DV_RULE_SET データ・ディクショナリ・ビューに問い合せることで確認できます。
handler	カスタム・イベント・ハンドラ・ロジックを定義する PL/SQL ファンクションまたはプロシージャの名前。
is_static	オプション。SQL 文によってアクセスされる際に、ルール・セットが評価される頻度を決定します。デフォルトは FALSE です。 ● TRUE: ルール・セットはユーザー・セッション中に 1 回、評価されます。その後、値は再利用されます。 ● FALSE: ルール・セットは、SQL 文でアクセスされるたびに評価されます。

例

```
BEGIN
```

```
 DBMS_MACADM.UPDATE_RULE_SET(
rule_set_name    => 'Limit_DBA_Access',
description      => 'DBA access through predefined processes',
enabled          => DBMS_MACUTL.G_YES,
eval_options     => DBMS_MACUTL.G_RULESET_EVAL_ANY,
audit_options    => DBMS_MACUTL.G_RULESET_AUDIT_FAIL,
fail_options     => DBMS_MACUTL.G_RULESET_FAIL_SHOW,
fail_message     => 'Access denied!',
fail_code        => 20900,
handler_options  => DBMS_MACUTL.G_RULESET_HANDLER_OFF,
handler          => '',
is_static        =  TRUE);
END;
/
```

親トピック: DBMS_MACADMルール・セットのプロシージャ

15.2 Oracle Database VaultのPL/SQLルール・セット・ファンクション

Oracle Database Vaultには、ルール・セットで保護されるSQL文を検査するためにルール・セットで使用されるファンクションが用意されています。

- DV_SYSEVENTファンクション
  DV_SYSEVENTファンクションは、ルール・セットを起動するシステム・イベントを返します。
- DV_LOGIN_USERファンクション
  DV_LOGIN_USERファンクションは、セッション・ユーザー名をVARCHAR2データ型で返します。
- DV_INSTANCE_NUMファンクション
  DV_INSTANCE_NUMファンクションは、データベース・インスタンス番号をNUMBERデータ型で返します。
- DV_DATABASE_NAMEファンクション
  DV_DATABASE_NAMEファンクションは、データベース名をVARCHAR2データ型で返します。
- DV_DICT_OBJ_TYPEファンクション
  DV_DICT_OBJ_TYPEファンクションは、データベース操作が発生したディクショナリ・オブジェクトのタイプを返します。
- DV_DICT_OBJ_OWNERファンクション
  DV_DICT_OBJ_OWNERファンクションは、データベース操作が発生したディクショナリ・オブジェクトの所有者の名前を返します。
- DV_DICT_OBJ_NAMEファンクション
  DV_DICT_OBJ_NAMEファンクションは、データベース操作が発生したディクショナリ・オブジェクトの名前を返します。
- DV_SQL_TEXTファンクション
  DV_SQL_TEXTファンクションは、操作で使用されるデータベース文のSQLテキストの最初の4000文字を返します。

親トピック: Oracle Database Vaultルール・セットのAPI

15.2.1 DV_SYSEVENTファンクション

DV_SYSEVENTファンクションは、ルール・セットを起動するシステム・イベントを返します。

イベント名は、SQL文の構文のものと同じで、INSERTやCREATEなどです。戻り型はVARCHAR2です。

構文

```
DV_SYSEVENT ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Get System Event Firing the Maintenance Rule Set',
  rule_expr => 'DV_SYSEVENT = ''CREATE''');
END;
/
```

親トピック: Oracle Database VaultのPL/SQLルール・セット・ファンクション

15.2.2 DV_LOGIN_USERファンクション

DV_LOGIN_USERファンクションは、セッション・ユーザー名をVARCHAR2データ型で返します。

構文

```
DV_LOGIN_USER ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Session User Name',
  rule_expr => 'DV_LOGIN_USER = ''SEBASTIAN''');
END;
/
```

親トピック: Oracle Database VaultのPL/SQLルール・セット・ファンクション

15.2.3 DV_INSTANCE_NUMファンクション

DV_INSTANCE_NUMファンクションは、データベース・インスタンス番号をNUMBERデータ型で返します。

構文

```
DV_INSTANCE_NUM ()
RETURN NUMBER;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Database Instance Number',
  rule_expr => 'DV_INSTANCE_NUM BETWEEN 6 AND 9');
END;
/
```

親トピック: Oracle Database VaultのPL/SQLルール・セット・ファンクション

15.2.4 DV_DATABASE_NAMEファンクション

DV_DATABASE_NAMEファンクションは、データベース名をVARCHAR2データ型で返します。

構文

```
DV_DATABASE_NAME ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Database Name',
  rule_expr => 'DV_DATABASE_NAME = ''ORCL''');
END;
/
```

親トピック: Oracle Database VaultのPL/SQLルール・セット・ファンクション

15.2.5 DV_DICT_OBJ_TYPEファンクション

DV_DICT_OBJ_TYPEファンクションは、データベース操作が発生したディクショナリ・オブジェクトのタイプを返します。

たとえば、戻されるディクショナリ・オブジェクトは表、プロシージャまたはビューです。戻り型はVARCHAR2です。

構文

```
DV_DICT_OBJ_TYPE ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Dictionary Object Type',
  rule_expr => 'DV_DICT_OBJ_TYPE IN (''TABLE'', ''VIEW'')');
END;
/
```

親トピック: Oracle Database VaultのPL/SQLルール・セット・ファンクション

15.2.6 DV_DICT_OBJ_OWNERファンクション

DV_DICT_OBJ_OWNERファンクションは、データベース操作が発生したディクショナリ・オブジェクトの所有者の名前を返します。

戻り型はVARCHAR2です。

構文

```
DV_DICT_OBJ_OWNER ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Dictionary Object Owner',
  rule_expr => 'DV_DICT_OBJ_OWNER = ''JSMITH''');
END;
/
```

親トピック: Oracle Database VaultのPL/SQLルール・セット・ファンクション

15.2.7 DV_DICT_OBJ_NAMEファンクション

DV_DICT_OBJ_NAMEファンクションは、データベース操作が発生したディクショナリ・オブジェクトの名前を返します。

戻り型はVARCHAR2です。

構文

```
DV_DICT_OBJ_NAME ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Dictionary Object Name',
  rule_expr => 'DV_DICT_OBJ_NAME = ''SALES''');
END;
/
```

親トピック: Oracle Database VaultのPL/SQLルール・セット・ファンクション

15.2.8 DV_SQL_TEXTファンクション

DV_SQL_TEXTファンクションは、操作で使用されるデータベース文のSQLテキストの最初の4000文字を返します。

戻り型はVARCHAR2です。

構文

```
DV_SQL_TEXT ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check SQL Text',
  rule_expr => 'DV_SQL_TEXT = ''SELECT SALARY FROM HR.EMPLOYEES''');
```

```
END;
/
```

親トピック: Oracle Database VaultのPL/SQLルール・セット・ファンクション

16 Oracle Database Vaultコマンド・ルールのAPI

DBMS_MACADM PL/SQLパッケージは、コマンド・ルールを構成するためのプロシージャを提供します。

DV_OWNERロールまたはDV_ADMINロールを付与されているユーザーのみがこれらのプロシージャを使用できます。

- CREATE_COMMAND_RULEプロシージャ
  CREATE_COMMAND_RULEプロシージャは、コマンド・ルールを作成し、ルール・セットに関連付けます。
- CREATE_CONNECT_COMMAND_RULEプロシージャ
  CREATE_CONNECT_COMMAND_RULEプロシージャは、ユーザーおよびルール・セットに関連付けることができる、CONNECTコマンド・ルールを作成します。
- CREATE_SESSION_EVENT_CMD_RULEプロシージャ
  CREATE_SESSION_EVENT_CMD_RULEプロシージャは、ALTER SESSION文に基づいて、セッション・イベントに関連付けることができるコマンド・ルールを作成します。
- CREATE_SYSTEM_EVENT_CMD_RULEプロシージャ
  CREATE_SYSTEM_EVENT_CMD_RULEプロシージャは、ALTER SYSTEM文に基づいて、システム・イベントに関連付けることができるコマンド・ルールを作成します。
- DELETE_COMMAND_RULEプロシージャ
  DELETE_COMMAND_RULEプロシージャは、コマンド・ルールの宣言を削除します。
- DELETE_CONNECT_COMMAND_RULEプロシージャ
  DELETE_CONNECT_COMMAND_RULEプロシージャは、CREATE_CONNECT_COMMAND_RULEプロシージャで作成されたCONNECTコマンド・ルールを削除します。
- DELETE_SESSION_EVENT_CMD_RULEプロシージャ
  DELETE_SESSION_EVENT_CMD_RULEプロシージャは、イベントに関連付けられているセッション・コマンド・ルールを削除します。
- DELETE_SYSTEM_EVENT_CMD_RULEプロシージャ
  DELETE_SYSTEM_EVENT_CMD_RULEプロシージャは、イベントに関連付けられているシステム・コマンド・ルールを削除します。
- UPDATE_COMMAND_RULEプロシージャ
  UPDATE_COMMAND_RULEプロシージャは、コマンド・ルールの宣言を更新します。
- UPDATE_CONNECT_COMMAND_RULEプロシージャ
  UPDATE_CONNECT_COMMAND_RULEプロシージャは、CREATE_CONNECT_COMMAND_RULEプロシージャで作成されたCONNECTコマンド・ルールを更新します。
- UPDATE_SESSION_EVENT_CMD_RULEプロシージャ
  UPDATE_SESSION_EVENT_CMD_RULEプロシージャは、ALTER SESSION文に基づいて、セッション・イベント・コマンド・ルールを更新します。
- UPDATE_SYSTEM_EVENT_CMD_RULEプロシージャ
  UPDATE_SYSTEM_EVENT_CMD_RULEプロシージャは、ALTER SYSTEM文に基づいて、システム・イベント・コマンド・ルールを更新します。

関連トピック

- コマンド・ルールの構成
- Oracle Database VaultユーティリティのAPI

16.1 CREATE_COMMAND_RULEプロシージャ

CREATE_COMMAND_RULEプロシージャは、コマンド・ルールを作成し、ルール・セットに関連付けます。

オプションで、これを使用して、ルール・セットによるコマンド・ルールのルール・チェックを有効化できます。マルチテナント環境では、共通およびローカルの両方のコマンド・ルールを作成できます。

構文

```
DBMS_MACADM.CREATE_COMMAND_RULE(
 command          IN VARCHAR2,
 rule_set_name    IN VARCHAR2,
 object_owner     IN VARCHAR2,
 object_name      IN VARCHAR2,
 enabled          IN VARCHAR2,
 privilege_scope  IN NUMBER,
 clause_name      IN VARCHAR2,
 parameter_name   IN VARCHAR2,
 event_name       IN VARCHAR2,
 component_name   IN VARCHAR2,
 action_name      IN VARCHAR2,
 scope            IN NUMBER DEFAULT);
```

パラメータ

表16-1 CREATE_COMMAND_RULEのパラメータ

パラメータ	説明
command	保護する SQL 文。 次の項も参照してください。 ● デフォルト・コマンド・ルールについては、「デフォルトのコマンド・ルール」を参照してください ● 既存のコマンド・ルールのリストは、「DBA_DV_COMMAND_RULE ビュー」を参照してください ● 使用可能な SQL 文のリストは、「コマンド・ルールで保護できる SQL 文」を参照してください
rule_set_name	このコマンド・ルールに関連付けるルール・セットの名前。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
object_owner	このコマンド・ルールを適用するデータベース・スキーマ。SELECT、INSERT、UPDATE、DELETE および EXECUTE 文以外は、ワイルドカード%を使用できます。 使用可能なユーザーを確認するには、『Oracle Database リファレンス』で説明されている DBA_USERS ビューを問い合せます。

パラメータ	説明
	詳細は、「コマンド・ルールの作成」の「オブジェクト所有者」も参照してください。
object_name	コマンド・ルールで保護されるオブジェクト。(ワイルドカード%を使用できます。コマンド・ルールで保護されるオブジェクトの詳細は、「コマンド・ルールの作成」の「オブジェクト名」を参照)。 使用可能なオブジェクトを確認するには、『Oracle Database リファレンス』で説明されている ALL_OBJECTS ビューを問い合せます。
enabled	次のいずれかのオプションを指定してコマンド・ルールのステータスを設定します。 ● コマンド・ルールを有効にするには(デフォルト)、DBMS_MACUTL.G_YES または y (Yes) ● シミュレーション・ログでの違反の取得など、コマンド・ルールを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s
privilege_scope	廃止されたパラメータ
clause_name	コマンド・ルールの作成に使用された SQL 文の句。たとえば、ALTER SESSION SQL 文のコマンド・ルールに、SET 句を clause_name パラメータとして含めることができます。 ALTER SYSTEM および ALTER SESSION のコマンド・ルールにのみ適用されます。
parameter_name	clause_name パラメータからのパラメータ。たとえば、ALTER SESSION コマンド・ルールの場合は、clause_name が SET だと、parameter_name を EVENTS に設定できます。 ALTER SYSTEM および ALTER SESSION のコマンド・ルールにのみ適用されます。
event_name	コマンド・ルールで定義されているイベント。たとえば、ALTER SESSION コマンド・ルールで、clause_name に SET を、parameter_name として EVENTS を使用しているとします。トレース・イベントを追跡する場合、event_name を TRACE に設定できます。 parameter パラメータが EVENTS に設定されている ALTER SYSTEM および ALTER SESSION コマンド・ルールにのみ、適用されます。
component_name	event_name 設定のコンポーネントたとえば、TRACE イベントの場合は、component_name を GCS にできます。 parameter パラメータが EVENTS に設定されている ALTER SYSTEM および ALTER

パラメータ	説明
	SESSION コマンド・ルールにのみ、適用されます。
action_name	component_name 設定のアクション parameter パラメータが EVENTS に設定されている ALTER SYSTEM および ALTER SESSION コマンド・ルールにのみ、適用されます。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2) アプリケーション・ルートで共通コマンド・ルールを作成し、関連付けられた PDB にそれを表示できるようにする場合は、アプリケーションを同期させる必要があります。たとえば: ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;

ALTER SYSTEMコマンド・ルール設定

表16-2に、ALTER SYSTEMコマンド・ルール設定を示します。

表16-2 ALTER SYSTEMコマンド・ルール設定

clause_name	parameter_name — パラメータ値
ARCHIVE LOG	● ALL — sequence_number ● CHANGE — change_number ● CURRENT — 該当なし ● GROUP — group_number ● LOGFILE — log_file_name ● NEXT — 該当なし ● SEQUENCE — 該当なし
CHECK DATAFILES	該当なし — global または local

clause_name	parameter_name — パラメータ値
CHECKPOINT	該当なし — global または local
COPY LOGFILE	該当なし — 該当なし
DISTRIBUTED RECOVERY	該当なし — enable または disable
DUMP	• DATAFILE — 該当なし • FLASHBACK — 該当なし • LOGFILE — 該当なし • REDO — 該当なし • TEMPFILE — 該当なし • UNDO — 該当なし
END SESSION	DISCONNECT SESSION — 該当なし KILL SESSION — 該当なし
FLUSH	BUFFER_CACHE — 該当なし GLOBAL CONTEXT — 該当なし REDO — target_db_name SHARED_POOL — 該当なし
QUIESCE	QUIESCE RESTRICTED — 該当なし UNQUIESCE — 該当なし
REFRESH	LDAP_REGISTRATION — 該当なし
REGISTER	該当なし — 該当なし
RESET	initialization_parameter_name — 該当なし
RESUME	該当なし — 該当なし

clause_name	parameter_name — パラメータ値
SECURITY	RESTRICTED SESSION — enable または disable SET ENCRYPTION KEY — 該当なし SET ENCRYPTION WALLET — open または close
SET	EVENTS — event_string GLOBAL_TOPIC_ENABLED — true または false initialization_parameter_name — parameter_value LDAP_REGISTRATION_ENABLED — true または false LDAP_REG-SYNC_INTERVAL — 数値 SINGLETASK DEBUG — 該当なし USE_STORED_OUTLINES — true、false または category_name
SHUTDOWN DISPPATCHER	該当なし — dispatcher_name
SWITCH LOGFILE	該当なし — all or none
SUSPEND	該当なし — 該当なし
TX RECOVERY	該当なし — enable または disable

ALTER SESSIONコマンド・ルール設定

表16-3に、ALTER SESSIONコマンド・ルール設定を示します。

表16-3 ALTER SESSIONコマンド・ルール設定

clause_name	parameter_name — パラメータ値
ADVISE	該当なし — COMMIT、ROLLBACK または NOTHING
CLOSE DATABASE LINK	該当なし — database_link
COMMIT IN PROCEDURE	該当なし — ENABLE または DISABLE

clause_name	parameter_name — パラメータ値
GUARD	該当なし — ENABLE または DISABLE
ILM	ROW ACCESS TRACKING — 該当なし ROW MODIFICATION TRACKING — 該当なし
LOGICAL REPLICATION	該当なし — 該当なし
PARALLEL DML	該当なし — ENABLE、DISABLE または FORCE
PARALLEL DDL	該当なし — ENABLE、DISABLE または FORCE
PARALLEL QUERY	該当なし — ENABLE、DISABLE または FORCE
RESUMABLE	該当なし — ENABLE または DISABLE
SYNC WITH PRIMARY	該当なし — 該当なし
SET	APPLICATION ACTION — action_name APPLICATION MODULE — module_name CONSTRAINTS — IMMEDIATE、DEFERRED または DEFAULT CONTAINER — container_name CURRENT SCHEMA — schema_name EDITION — edition_name ERROR ON OVERLAP TIME — TRUE または FALSE EVENTS — event_string FLAGGER — OFF、FULL、INTERMEDIATE、ENTRY initialization_parameter_name — parameter_name INSTANCE — instance_number ISOLATION_LEVEL — SERIALIZABLE または READ

clause_name	parameter_name — パラメータ値
	COMMITTED ROW_ARCHIVAL_VISABILITY — ACTIVE または ALL SQL_TRANSFORMATION_PROFILE — profile_name STANDBY_MAX_DATA_DELAY — NONEnumber TIME_ZONE — LOCAL、DBTIMEZONE または other_value USE_PRIVATE_OUTLINES — TRUE、FALSE または category_name USE_STORED_OUTLINES — TRUE、FALSE または category_name

例

単純なコマンド・ルール

次の例では、OE.ORDERS表に対するSELECT文のために単純なコマンド・ルールを作成する方法を示します。このコマンド・ルールでは、コマンド・ルールは使用されません。

```
BEGIN
 DBMS_MACADM.CREATE_COMMAND_RULE(
command          => 'SELECT',
  rule_set_name   => 'Check User Role',
  object_owner    => 'OE',
object_name     => 'ORDERS',
enabled          => DBMS_MACUTL.G_YES);
END;
/
```

この例では、ユーザーがhr_audit_pol統合監査ポリシーを有効化または無効化するかどうかを確認するコマンド・ルールを作成する方法を示します。オブジェクトが統合監査ポリシーの場合は、commandパラメータに対して、AUDITでなくAUDIT POLICYが必要です。

```
BEGIN
DBMS_MACADM.CREATE_COMMAND_RULE(
  command          => 'AUDIT POLICY',
  rule_set_name    => 'Check ability to audit',
  object_owner     => '%',
  object_name      => 'hr_audit_pol',
  enabled          => DBMS_MACUTL.G_YES,
  scope            => DBMS_MACUTL.G.SCOPE_LOCAL);
END;
/
```

SET句を使用したALTER SESSIONコマンド・ルール

次の例では、ERROR_ON_OVERLAP_TIMEパラメータを指定してSET句を使用する、ALTER SESSIONコマンド・ルールを作成する方法を示します。

```
BEGIN
 DBMS_MACADM.CREATE_COMMAND_RULE(
command         => 'ALTER SESSION',
rule_set_name   => 'Test ERROR_ON_OVERLAP_TIME for FALSE',
object_owner    => '%',
object_name     => '%',
enabled         => DBMS_MACUTL.G_YES,
 clause_name     => 'SET',
 parameter_name  => 'ERROR_ON_OVERLAP_TIME',
 scope           => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

この例の説明は、次のとおりです。

- rule_set_name: ALTER SESSION SQL文のERROR_ON_OVERLAP_TIMEセッション・パラメータは、TRUEまたはFALSEのどちらかに設定する必要があります。この設定になっているかどうかを確認するルール・セットを作成できます。たとえば、次のルールがあるとします。

  ```
  EXEC DBMS_MACADM.CREATE_RULE('RULE_TRUE', 'UPPER(PARAMETER_VALUE) = ''TRUE''');
  ```

  このルールとともに使用されるルール・セットは、次のようにできます。

  ```
  BEGIN
   DBMS_MACADM.CREATE_RULE_SET(
    rule_set_name    => 'Test ERROR_ON_OVERLAP_TIME',
    description      => 'Checks if the ERROR_ON_OVERLAP_TIME setting is TRUE or
  FALSE',
    enabled          => DBMS_MACUTL.G_YES,
    eval_options     => DBMS_MACUTL.G_RULESET_EVAL_ALL,
    audit_options    => DBMS_MACUTL.G_RULESET_AUDIT_FAIL +
  DBMS_MACUTL.G_RULESET_AUDIT_SUCCESS,
    fail_options     => DBMS_MACUTL.G_RULESET_FAIL_SILENT,
    fail_message     => 'false error on overlaptime',
    fail_code        => 20461,
    handler_options  => DBMS_MACUTL.G_RULESET_HANDLER_FAIL,
    handler          => '',
    is_static        => false);
  END;
  /
  EXEC DBMS_MACADM.ADD_RULE_TO_RULE_SET('Test ERROR_ON_OVERLAP_TIME',
  'RULE_TRUE');
  ```

- ALTER SESSIONおよびALTER SYSTEMコマンド・ルールで、object_ownerおよびobject_nameを%に設定する必要があります。
- enabledでは、コマンド・ルールを作成時に有効にするために、DBMS_MACUTL.G_YES定数が使用されます。
- clause_nameでは、ALTER SESSION PL/SQL文のSET句を使用するようALTER SESSIONコマンド・ルールが設定されます。
- parameter_nameは、SET句のERROR_ON_OVERLAP_TIMEパラメータに設定されます。
- scopeでは、コマンド・ルールを共通コマンド・ルールにするよう設定するために、DBMS_MACUTL.G_SCOPE_COMMON定数が使用されます。このコマンド・ルールはマルチテナント環境のアプリケーション・ルート内にあるため、このプロシージャを実行するユーザーはCDBルート内にいる必要があります。このコマンド・ルールに関連付けられているルールまたはルール・セットは、共通である必要があります。

  コマンド・ルールをローカルで作成している場合は、scopeをDBMS_MACUTL.G_SCOPE_LOCALに設定します。その場合、このプロシージャを実行するユーザーは、コマンド・ルールが存在するPDB内にいる必要があります。既存のPDB

を確認するには、DBA_PDBSデータ・ディクショナリ・ビューを問い合せます。このコマンド・ルールに関連付けられているルールまたはルール・セットは、ローカルである必要があります。

CHECKPOINT句を使用したALTER SYSTEMコマンド・ルール

この例では、CHECKPOINT句を使用するALTER SYSTEMコマンド・ルールの作成方法を示します。CHECKPOINT設定のためにコマンド・ルールをテストするには、前の例のALTER SESSIONコマンド・ルールと同様に、ルール・セットおよびルールを作成する必要があります。この例では、parameter設定を指定しません。これは、CHECKPOINT設定にパラメータがないためです。

```
BEGIN
 DBMS_MACADM.CREATE_COMMAND_RULE(
command          => 'ALTER SYSTEM',
rule_set_name    => 'Test CHECKPOINT Setting',
object_owner     => '%',
object_name      => '%',
enabled          => DBMS_MACUTL.G_YES,
 clause_name      => 'CHECKPOINT',
 parameter_name   => '',
 scope            => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

SET句を使用したALTER SESSIONコマンド・ルール

次のALTER SESSIONコマンド・ルールでは、SET句を使用してevent_nameおよびcomponent_nameを指定します。clause_nameパラメータでSETが指定される場合は、event_name、component_nameおよびaction_nameパラメータのみを使用できます。

```
BEGIN
 DBMS_MACADM.CREATE_COMMAND_RULE(
  command          => 'ALTER SESSION',
  rule_set_name    => 'Check Trace Events',
  object_owner     => '%',
  object_name      => '%',
  enabled          => DBMS_MACUTL.G_YES,
  clause_name      => 'SET',
  parameter_name   => 'EVENTS',
  event_name       => 'TRACE',
  component_name   => 'GCS',
  scope            => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

このトピックの概念的情報については、ALTER SESSIONおよびALTER SYSTEMコマンド・ルールを参照してください。

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.2 CREATE_CONNECT_COMMAND_RULEプロシージャ

CREATE_CONNECT_COMMAND_RULEプロシージャは、ユーザーおよびルール・セットに関連付けることができる、CONNECTコマンド・ルールを作成します。

マルチテナント環境では、共通およびローカルの両方のコマンド・ルールを作成できます。

構文

```
DBMS_MACADM.CREATE_CONNECT_COMMAND_RULE(
 user_name        IN VARCHAR2,
 rule_set_name    IN VARCHAR2,
```

```
enabled          IN VARCHAR2,
scope            IN NUMBER DEFAULT);
```

パラメータ

表16-4 CREATE_CONNECT_COMMAND_RULEのパラメータ

パラメータ	説明
user_name	CONNECT コマンド・ルールの適用対象となるユーザー。%ワイルドカードを入力した場合は、CONNECT コマンド・ルールがすべてのデータベース・ユーザーに適用されます。 マルチテナント環境では、ルートでこのプロシージャを実行した場合、%を指定すると、すべての共通ユーザーに適用されます。PDB でプロシージャを実行すると、それは、この PDB にアクセスできるすべてのローカルおよび共通ユーザーに適用されます。一方は共通で一方はローカルという 2 つのコマンド・ルールがあり、それらが両方とも同一オブジェクトに適用される場合は、操作を成功させるために、両方とも正しく評価する必要があります。 マルチテナント環境では、このユーザーが、CONNECT コマンド・ルールが共通である場合は共通、CONNECT コマンド・ルールがローカルである場合はローカルまたは共通であることを確認します。 現在のインスタンス内の既存のデータベース・ユーザーを確認するには、Oracle Database リファレンスで説明されている、DBA_USERS ビューを問い合せます。
rule_set_name	このコマンド・ルールに関連付けるルール・セットの名前。マルチテナント環境では、このルール・セットが、CONNECT コマンド・ルールが共通である場合は共通、CONNECT コマンド・ルールがローカルである場合はローカルであることを確認してください。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
enabled	次のいずれかのオプションを指定してコマンド・ルールのステータスを設定します。 ● コマンド・ルールを有効にするには(デフォルト)、DBMS_MACUTL.G_YES または y (Yes) ● シミュレーション・ログでの違反の取得など、コマンド・ルールを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。

パラメータ	説明
	● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2) アプリケーション・ルートで共通 CONNECT コマンド・ルールを作成し、関連付けられた PDB にそれを表示できるようにする場合は、アプリケーションを同期させる必要があります。たとえば: ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;

例

次の例では、マルチテナント環境で共通CONNECTコマンド・ルールを作成する方法を示します。このコマンド・ルールはCDBルート内にあるため、このプロシージャを実行するユーザーはCDBルート内にいる必要があります。このコマンド・ルールに関連付けられているユーザー名またはルール・セットは、共通である必要があります。

```
BEGIN
 DBMS_MACADM.CREATE_CONNECT_COMMAND_RULE(
  rule_set_name   => 'Allow Sessions',
  user_name       => 'C##HR_ADMIN',
  enabled         => DBMS_MACUTL.G_SIMULATION,
  scope           => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

この例は、前の例のローカル・バージョンです。このプロシージャを実行するユーザーは、ローカルCONNECTコマンド・ルールが存在するPDB内にいる必要があります。使用可能なPDBを見つけるには、show pdbsコマンドを実行します。このコマンド・ルールに関連付けられているルール・セットは、ローカルである必要があります。ユーザーは、共通またはローカルのどちらかにできます。

```
BEGIN
 DBMS_MACADM.CREATE_CONNECT_COMMAND_RULE(
 rule_set_name   => 'Allow Sessions',
 user_name       => 'PSMITH',
 enabled         => DBMS_MACUTL.G_SIMULATION,
 scope           => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.3 CREATE_SESSION_EVENT_CMD_RULEプロシージャ

CREATE_SESSION_EVENT_CMD_RULEプロシージャは、ALTER SESSION文に基づいて、セッション・イベントに関連付けることができるコマンド・ルールを作成します。

マルチテナント環境では、セッション・イベントの共通およびローカルの両方のコマンド・ルールを作成できます。

構文

```
DBMS_MACADM.CREATE_SESSION_EVENT_CMD_RULE(
 rule_set_name   IN VARCHAR2,
 enabled         IN VARCHAR2,
 event_name      IN VARCHAR2 DEFAULT,
```

```
component_name  IN VARCHAR2 DEFAULT,
action_name     IN VARCHAR2 DEFAULT,
scope           IN NUMBER DEFAULT,
pl_sql_stack    IN BOOLEAN DEFAULT);
```

パラメータ

表16-5 CREATE_SESSION_EVENT_CMD_RULEのパラメータ

パラメータ	説明
rule_set_name	コマンド・ルールに関連付けるルール・セットの名前。マルチテナント環境では、このルール・セットが、セッション・イベント・コマンド・ルールが共通である場合は共通、そのコマンド・ルールがローカルである場合はローカルであることを確認してください。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
enabled	次のいずれかのオプションを指定してコマンド・ルールのステータスを設定します。 ● コマンド・ルールを有効にするには(デフォルト)、DBMS_MACUTL.G_YES または y (Yes) ● シミュレーション・ログでの違反の取得など、コマンド・ルールを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s
event_name	コマンド・ルールで定義されているイベント。この設定により、コマンド・ルールが ALTER SESSION SET EVENTS event_name 文に対応できるようになります。たとえば、トレース・イベントを追跡するには、event_name を TRACE に設定します。
component_name	event_name 設定のコンポーネント例の設定は、DV、OLS または GCS です。 ORADEBUG DOC COMPONENT RDBMS をユーザーSYS として発行することで、有効なコンポーネント名を確認できます。出力には、component_name 設定に使用できる親および子コンポーネントが表示されます。たとえば、XS (親)と XSSESSION (XS の子)の両方が有効なコンポーネント名です。親コンポーネントを選択すると、コマンド・ルールがそれとその子コンポーネントに適用されます。
action_name	component_name 設定のアクション
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカ

パラメータ	説明
	ルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2) アプリケーション・ルートで共通コマンド・ルールを作成し、関連付けられた PDB にそれを表示できるようにする場合は、アプリケーションを同期させる必要があります。たとえば: `ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;`
pl_sql_stack	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックを記録するかどうかを指定します。PL/SQL スタックを記録する場合は TRUE と入力し、記録しない場合は FALSE と入力します。デフォルトは FALSE です。

例

次の例では、マルチテナント環境で共通セッション・イベント・コマンド・ルールを作成する方法を示します。このコマンド・ルールはアプリケーション・ルート内にあるため、このプロシージャを実行するユーザーはCDBルート内にいる必要があります。このコマンド・ルールに関連付けられているユーザー名またはルール・セットは、共通である必要があります。

```
BEGIN
 DBMS_MACADM.CREATE_SESSION_EVENT_CMD_RULE(
  rule_set_name   => 'Allow Sessions',
  event_name      => 'TRACE',
  component_name  => 'DV',
  action_name     => 'CURSORTRACE',
  enabled         => DBMS_MACUTL.G_SIMULATION,
  scope           => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

この例では、47998トレース・イベントのためにセッション・イベントを作成する方法を示します。この例では、失敗した操作のPL/SQLスタックを記録します。

```
BEGIN
 DBMS_MACADM.CREATE_SESSION_EVENT_CMD_RULE(
  rule_set_name   => 'Allow Sessions',
  event_name      => '47998',
  enabled         => 'y',
  scope           => DBMS_MACUTL.G_SCOPE_LOCAL,
  pl_sql_stack    => TRUE);
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.4 CREATE_SYSTEM_EVENT_CMD_RULEプロシージャ

CREATE_SYSTEM_EVENT_CMD_RULEプロシージャは、ALTER SYSTEM文に基づいて、システム・イベントに関連付けることができるコマンド・ルールを作成します。

マルチテナント環境では、ALTER SYSTEMの共通およびローカルの両方のコマンド・ルールを作成できます。

構文

```
DBMS_MACADM.CREATE_SYSTEM_EVENT_CMD_RULE(
 rule_set_name   IN VARCHAR2,
 enabled         IN VARCHAR2,
 event_name      IN VARCHAR2 DEFAULT,
 component_name  IN VARCHAR2 DEFAULT,
 action_name     IN VARCHAR2 DEFAULT,
 scope           IN NUMBER DEFAULT
 pl_sql_stack    IN BOOLEAN DEFAULT);
```

パラメータ

表16-6 CREATE_SYSTEM_EVENT_CMD_RULEパラメータ

パラメータ	説明
rule_set_name	コマンド・ルールに関連付けるルール・セットの名前。マルチテナント環境では、このルール・セットが、システム・イベント・コマンド・ルールが共通である場合は共通、そのコマンド・ルールがローカルである場合はローカルであることを確認してください。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
event_name	コマンド・ルールで定義されているイベント。この設定により、コマンド・ルールが ALTER SYSTEM SET EVENTS event_name に対応できるようになります。たとえば、トレース・イベントを追跡するには、event_name を TRACE に設定します。
component_name	event_name 設定のコンポーネント例の設定は、DV、OLS または GCS です。 ORADEBUG DOC COMPONENT RDBMS をユーザーSYS として発行することで、有効なコンポーネント名を確認できます。出力には、component_name 設定に使用できる親および子コンポーネントが表示されます。たとえば、XS (親)と XSSESSION (XS の子)の両方が有効なコンポーネント名です。親コンポーネントを選択すると、コマンド・ルールがそれとその子コンポーネントに適用されます。
action_name	component_name 設定のアクション
enabled	次のいずれかのオプションを指定してコマンド・ルールのステータスを設定します。 ● コマンド・ルールを有効にするには(デフォルト)、DBMS_MACUTL.G_YES または y

パラメータ	説明
	● シミュレーション・ログでの違反の取得など、コマンド・ルールを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2) アプリケーション・ルートで共通コマンド・ルールを作成し、関連付けられた PDB にそれを表示できるようにする場合は、アプリケーションを同期させる必要があります。たとえば: `ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;`
pl_sql_stack	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックを記録するかどうかを指定します。PL/SQL スタックを記録する場合は TRUE と入力し、記録しない場合は FALSE と入力します。デフォルトは FALSE です。

例

次の例では、マルチテナント環境で共通システム・イベント・コマンド・ルールを作成する方法を示します。このコマンド・ルールはアプリケーション・ルート内にあるため、このプロシージャを実行するユーザーはCDBルート内にいる必要があります。このコマンド・ルールに関連付けられているユーザー名またはルール・セットは、共通である必要があります。

```
BEGIN
 DBMS_MACADM.CREATE_SYSTEM_EVENT_CMD_RULE(
  rule_set_name   => 'Enabled',
  event_name      => 'TRACE',
  component_name  => 'GSIPC',
  action_name     => 'HEAPDUMP',
  enabled         => DBMS_MACUTL.G_YES,
  scope           => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.5 DELETE_COMMAND_RULEプロシージャ

DELETE_COMMAND_RULEプロシージャは、コマンド・ルールの宣言を削除します。

構文

```
DBMS_MACADM.DELETE_COMMAND_RULE(
 command          IN VARCHAR2,
 object_owner     IN VARCHAR2,
 object_name      IN VARCHAR2,
 clause_name      IN VARCHAR2,
 parameter_name   IN VARCHAR2 DEFAULT,
 event_name       IN VARCHAR2 DEFAULT,
 component_name   IN VARCHAR2 DEFAULT,
 action_name      IN VARCHAR2 DEFAULT,
 scope            IN NUMBER DEFAULT);
```

パラメータ

表16-7 DELETE_COMMAND_RULEのパラメータ

パラメータ	説明
command	コマンド・ルールで保護される SQL 文。 使用可能なコマンド・ルールを確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
object_owner	このコマンド・ルールを適用するデータベース・スキーマ。 現行のデータベース・インスタンスで使用可能なユーザーを確認するには、『Oracle Database リファレンス』で説明されている DBA_USERS ビューを問い合せます。
object_name	オブジェクト名。ワイルドカード%を使用できます。 現行のデータベース・インスタンスで使用可能なオブジェクトを確認するには、『Oracle Database リファレンス』で説明されている ALL_OBJECTS ビューを問い合せます。
clause_name	コマンド・ルールの作成に使用された SQL 文の句。 ALTER SYSTEM および ALTER SESSION のコマンド・ルールにのみ適用されます。
parameter_name	clause_name パラメータからのパラメータ。 ALTER SYSTEM および ALTER SESSION のコマンド・ルールにのみ適用されます。
event_name	コマンド・ルールで定義されているイベント。 ALTER SYSTEM および ALTER SESSION のコマンド・ルールにのみ適用されます。
component_name	event_name 設定のコンポーネント ALTER SYSTEM および ALTER SESSION のコマンド・ルールにのみ適用されます。

パラメータ	説明
action_name	component_name 設定のアクション ALTER SYSTEM および ALTER SESSION のコマンド・ルールにのみ適用されます。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

コマンド・ルールを削除するときには、パラメータのrule_set_nameとenabledを省略して、その他のパラメータが前回のコマンド・ルールの更新時に使用した設定と一致していることを確認してください。DBA_DV_COMMAND_RULEデータ・ディクショナリ・ビューを問い合せることで、最新の設定を確認できます。

たとえば、次のコマンド・ルールを作成するとします。

```
BEGIN
 DBMS_MACADM.CREATE_COMMAND_RULE(
  command          => 'SELECT',
  rule_set_name    => 'Enabled',
  object_owner     => 'OE',
  object_name      => 'ORDERS',
  enabled          => DBMS_MACUTL.G_YES,
  scope            => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

このコマンド・ルールを削除するには、ここに示すパラメータと同じものを使用しますが、rule_set_nameとenabledは省略します。

```
BEGIN
 DBMS_MACADM.DELETE_COMMAND_RULE(
  command          => 'SELECT',
  object_owner     => 'OE',
  object_name      => 'ORDERS',
  scope            => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

次の例は、ALTER SESSIONコマンド・ルールを削除する方法を示しています。

```
BEGIN
 DBMS_MACADM.DELETE_COMMAND_RULE(
  command          => 'ALTER SESSION',
  object_owner     => '%',
  object_name      => '%',
  clause_name      => 'SET',
  parameter_name   => 'EVENTS',
  event_name       => 'TRACE',
  component_name   => 'GCS',
```

```
  scope           => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.6 DELETE_CONNECT_COMMAND_RULEプロシージャ

DELETE_CONNECT_COMMAND_RULEプロシージャは、CREATE_CONNECT_COMMAND_RULEプロシージャで作成されたCONNECTコマンド・ルールを削除します。

構文

```
DBMS_MACADM.DELETE_CONNECT_COMMAND_RULE(
 user_name        IN VARCHAR2,
 scope            IN NUMBER DEFAULT);
```

パラメータ

表16-8 DELETE_CONNECT_COMMAND_RULEのパラメータ

パラメータ	説明
user_name	CONNECT コマンド・ルールの適用対象となるユーザー。 このユーザーを確認するには、DBA_DV_COMMAND_RULE ビューの OBJECT_OWNER フィールドを問い合せます。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

```
BEGIN
 DBMS_MACADM.DELETE_CONNECT_COMMAND_RULE(
  user_name        => 'PSMITH',
  scope            => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.7 DELETE_SESSION_EVENT_CMD_RULEプロシージャ

DELETE_SESSION_EVENT_CMD_RULEプロシージャは、イベントに関連付けられているセッション・コマンド・ルールを削除します。

構文

```
DBMS_MACADM.DELETE_SESSION_EVENT_CMD_RULE(
 event_name       IN VARCHAR2 DEFAULT,
 component_name   IN VARCHAR2 DEFAULT,
 action_name      IN VARCHAR2 DEFAULT,
 scope            IN NUMBER DEFAULT);
```

パラメータ

表16-9 DELETE_SESSION_EVENT_CMD_RULEのパラメータ

パラメータ	説明
event_name	セッション・イベント・コマンド・ルールで定義されているイベント。 既存のコマンド・ルールについては、DBA_DV_COMMAND_RULE ビューを参照してください。
component_name	event_name 設定のコンポーネント
action_name	component_name 設定のアクション
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

次の例では、マルチテナント環境のアプリケーション・ルートで共通セッション・イベント・コマンド・ルールを削除する方法を示します。このプロシージャを実行するユーザーは、CDBルート内の共通ユーザーである必要があります。パラメータを指定するときは、コマンド・ルールを最後に更新したときに使用したパラメータとそれらが正確に一致することを確認します。コマンド・ルールの現在の設定を確認するには、「DBA_DV_COMMAND_RULEビュー」で説明されているDBA_DV_COMMAND_RULEビューを問い合せます。

```
BEGIN
DBMS_MACADM.DELETE_SESSION_EVENT_CMD_RULE(
 event_name       => '47999',
 scope            => DBMS_MACUTL.G_SCOPE_COMMON);
END;
 /
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.8 DELETE_SYSTEM_EVENT_CMD_RULEプロシージャ

DELETE_SYSTEM_EVENT_CMD_RULEプロシージャは、イベントに関連付けられているシステム・コマンド・ルールを削除しま

す。

構文

```
DBMS_MACADM.DELETE_SYSTEM_EVENT_CMD_RULE(
 event_name       IN VARCHAR2 DEFAULT,
 component_name   IN VARCHAR2 DEFAULT,
 action_name      IN VARCHAR2 DEFAULT,
 scope            IN NUMBER DEFAULT);
```

パラメータ

表16-10 DELETE_SYSTEM_EVENT_CMD_RULEのパラメータ

パラメータ	説明
event_name	システム・イベント・コマンド・ルールで定義されているイベント。 既存のコマンド・ルールについては、DBA_DV_COMMAND_RULE ビューを参照してください。
component_name	event_name 設定のコンポーネント
action_name	component_name 設定のアクション
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

次の例では、マルチテナント環境のアプリケーション・ルートで共通システム・イベント・コマンド・ルールを削除する方法を示します。このプロシージャを実行するユーザーは、CDBルート内の共通ユーザーである必要があります。パラメータを指定するときは、コマンド・ルールを最後に更新したときに使用したパラメータとそれらが正確に一致することを確認します。コマンド・ルールの現在の設定を確認するには、「DBA_DV_COMMAND_RULEビュー」で説明されているDBA_DV_COMMAND_RULEビューを問い合せます。

```
BEGIN
 DBMS_MACADM.DELETE_SYSTEM_EVENT_CMD_RULE(
 event_name       => 'TRACE',
 component_name   => 'DV',
 action_name      => '',
 scope            => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.9 UPDATE_COMMAND_RULEプロシージャ

UPDATE_COMMAND_RULEプロシージャは、コマンド・ルールの宣言を更新します。

マルチテナント環境では、共通およびローカルの両方のコマンド・ルールを更新できます。

構文

```
DBMS_MACADM.UPDATE_COMMAND_RULE(
 command          IN VARCHAR2,
 rule_set_name    IN VARCHAR2,
 object_owner     IN VARCHAR2,
 object_name      IN VARCHAR2,
 enabled          IN VARCHAR2,
 privilege_scope  IN NUMBER,
 clause_name      IN VARCHAR2,
 parameter_name   IN VARCHAR2 DEFAULT,
 event_name       IN VARCHAR2 DEFAULT,
 component_name   IN VARCHAR2 DEFAULT,
 action_name      IN VARCHAR2 DEFAULT,
 scope            IN NUMBER DEFAULT,
 pl_sql_stack     IN BOOLEAN DEFAULT);
```

パラメータ

表16-11 UPDATE_COMMAND_RULEのパラメータ

パラメータ	説明
command	更新するコマンド・ルール 次の項も参照してください。 ● 使用可能な SQL 文のリストは、「コマンド・ルールで保護できる SQL 文」を参照してください ● 既存のコマンド・ルールについては、DBA_DV_COMMAND_RULE ビューを参照してください。
rule_set_name	このコマンド・ルールに関連付けるルール・セットの名前。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「Oracle Database Vault のデータ・ディクショナリ・ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
object_owner	このコマンド・ルールを適用するデータベース・スキーマ。 使用可能なユーザーを確認するには、『Oracle Database リファレンス』で説明されている DBA_USERS ビューを問い合せます。詳細は、「コマンド・ルールの作成」の「オブジェクト所有者」も参照してください。
object_name	オブジェクト名。(ワイルドカード%を使用できます。コマンド・ルールで保護されるオブジェクトの詳細

パラメータ	説明
	は、「コマンド・ルールの作成」の「オブジェクト名」を参照)。 使用可能なオブジェクトを確認するには、『Oracle Database リファレンス』で説明されている ALL_OBJECTS ビューを問い合せます。
enabled	次のいずれかのオプションを指定してコマンド・ルールのステータスを設定します。 ● コマンド・ルールを有効にするには(デフォルト)、DBMS_MACUTL.G_YES または y ● シミュレーション・ログでの違反の取得など、コマンド・ルールを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s
privilege_scope	廃止されたパラメータ
clause_name	コマンド・ルールの作成に使用された SQL 文の句。たとえば、ALTER SESSION SQL 文のコマンド・ルールに、SET 句を clause_name パラメータとして含めることができます。 ALTER SYSTEM および ALTER SESSION のコマンド・ルールにのみ適用されます。
parameter_name	clause_name パラメータからのパラメータ。たとえば、ALTER SESSION コマンド・ルールの場合は、clause_name が SET だと、parameter_name を EVENTS に設定できます。 ALTER SYSTEM および ALTER SESSION のコマンド・ルールにのみ適用されます。
event_name	コマンド・ルールで定義されているイベント。たとえば、clause_name に SET を、parameter_name として EVENTS を使用する ALTER SESSION コマンド・ルールの場合に、event_name を TRACE に設定できます。 parameter パラメータが events に設定されている ALTER SYSTEM および ALTER SESSION コマンド・ルールにのみ、適用されます。
component_name	event_name 設定のコンポーネントたとえば、TRACE イベントの場合は、component_name を GCS にできます。 parameter パラメータが events に設定されている ALTER SYSTEM および ALTER SESSION コマンド・ルールにのみ、適用されます。
action_name	component_name 設定のアクションたとえば、component_name が GCS に設定されている

パラメータ	説明
	場合は、action_name 設定を DISK HIGH にできます。 parameter パラメータが events に設定されている ALTER SYSTEM および ALTER SESSION コマンド・ルールにのみ、適用されます。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2) アプリケーション・ルートで共通コマンド・ルールを更新し、関連付けられた PDB にそれを表示できるようにする場合は、アプリケーションを同期させる必要があります。たとえば: ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;
pl_sql_stack	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックを記録するかどうかを指定します。PL/SQL スタックを記録する場合は TRUE と入力し、記録しない場合は FALSE と入力します。

例

次の例では、HR.EMPLOYEESスキーマを保護する単純なコマンド・ルールを作成する方法を示します。

```
BEGIN
 DBMS_MACADM.UPDATE_COMMAND_RULE(
  command         => 'SELECT',
  rule_set_name   => 'Enabled',
  object_owner    => 'HR',
  object_name     => 'EMPLOYEES',
  enabled         => DBMS_MACUTL.G_SIMULATION,
  scope           => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

この例では、ALTER SESSION SQL文に基づく、より複雑なコマンド・ルールを更新する方法を示します。

```
BEGIN
 DBMS_MACADM.UPDATE_COMMAND_RULE(
  command         => 'ALTER SESSION',
  rule_set_name   => 'Enabled',
  object_owner    => '%',
  object_name     => '%',
  enabled         => 's',
  clause_name     => 'SET',
  parameter_name  => 'EVENTS',
  event_name      => 'TRACE',
  component_name  => 'GCS',
  scope           => DBMS_MACUTL.G_SCOPE_LOCAL);
```

```
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.10 UPDATE_CONNECT_COMMAND_RULEプロシージャ

UPDATE_CONNECT_COMMAND_RULEプロシージャは、CREATE_CONNECT_COMMAND_RULEプロシージャで作成されたCONNECTコマンド・ルールを更新します。

構文

```
DBMS_MACADM.UPDATE_CONNECT_COMMAND_RULE(
 user_name        IN VARCHAR2,
 rule_set_name    IN VARCHAR2,
 enabled          IN VARCHAR2,
 scope            IN NUMBER DEFAULT);
```

パラメータ

表16-12 UPDATE_CONNECT_COMMAND_RULEのパラメータ

パラメータ	説明
user_name	CONNECT コマンド・ルールの適用対象となるユーザー。%ワイルドカードを入力した場合は、CONNECT コマンド・ルールがすべてのデータベース・ユーザーに適用されます。 マルチテナント環境では、ルートでこのプロシージャを実行した場合、%を指定すると、すべての共通ユーザーに適用されます。PDB でプロシージャを実行すると、それは、この PDB にアクセスできるすべてのローカルおよび共通ユーザーに適用されます。一方は共通で一方はローカルという 2 つのコマンド・ルールがあり、それらが両方とも同一オブジェクトに適用される場合は、操作を成功させるために、両方とも正しく評価する必要があります。 マルチテナント環境では、このユーザーが、CONNECT コマンド・ルールが共通である場合は共通、CONNECT コマンド・ルールがローカルである場合はローカルまたは共通であることを確認します。 既存のコマンド・ルールを確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。 現在のインスタンス内の既存のデータベース・ユーザーを確認するには、Oracle Database リファレンスで説明されている、DBA_USERS ビューを問い合せます。
rule_set_name	このコマンド・ルールに関連付けるルール・セットの名前。マルチテナント環境では、このルール・セットが、CONNECT コマンド・ルールが共通である場合は共通、CONNECT コマンド・ルールがローカルである場合はローカルであることを確認してください。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。

パラメータ	説明
enabled	次のいずれかのオプションを指定してコマンド・ルールのステータスを設定します。 ● コマンド・ルールを有効にするには(デフォルト)、DBMS_MACUTL.G_YES または y ● シミュレーション・ログでの違反の取得など、コマンド・ルールを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2) アプリケーション・ルートで共通コマンド・ルールを更新し、関連付けられた PDB にそれを表示できるようにする場合は、アプリケーションを同期させる必要があります。たとえば: `ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;`

例

```
BEGIN
 DBMS_MACADM.UPDATE_CONNECT_COMMAND_RULE(
  rule_set_name   => 'Allow Sessions',
  user_name       => 'PSMITH',
  enabled         => 'DBMS_MACUTL.G_YES',
  scope           => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.11 UPDATE_SESSION_EVENT_CMD_RULEプロシージャ

UPDATE_SESSION_EVENT_CMD_RULEプロシージャは、ALTER SESSION文に基づいて、セッション・イベント・コマンド・ルールを更新します。

マルチテナント環境では、共通およびローカルの両方のセッション・イベント・コマンド・ルールを更新できます。

構文

```
DBMS_MACADM.UPDATE_SESSION_EVENT_CMD_RULE(
 rule_set_name   IN VARCHAR2,
 enabled         IN VARCHAR2,
 event_name      IN VARCHAR2 DEFAULT,
```

```
component_name  IN VARCHAR2 DEFAULT,
action_name     IN VARCHAR2 DEFAULT,
scope           IN NUMBER DEFAULT,
pl_sql_stack    IN BOOLEAN DEFAULT);
```

パラメータ

表16-13 UPDATE_SESSION_EVENT_CMD_RULEのパラメータ

パラメータ	説明
rule_set_name	コマンド・ルールに関連付けるルール・セットの名前。マルチテナント環境では、このルール・セットが、セッション・イベント・コマンド・ルールが共通である場合は共通、そのコマンド・ルールがローカルである場合はローカルであることを確認してください。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
enabled	次のいずれかのオプションを指定してコマンド・ルールのステータスを設定します。 ● コマンド・ルールを有効にするには(デフォルト)、DBMS_MACUTL.G_YES または y ● シミュレーション・ログでの違反の取得など、コマンド・ルールを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s
event_name	コマンド・ルールで定義されているイベント。この設定により、コマンド・ルールが ALTER SESSION SET EVENTS event_name 文に対応できるようになります。たとえば、トレース・イベントを追跡するには、event_name を TRACE に設定します。
component_name	event_name 設定のコンポーネント例の設定は、DV、OLS または GCS です。 ORADEBUG DOC COMPONENT RDBMS をユーザーSYS として発行することで、有効なコンポーネント名を確認できます。出力には、component_name 設定に使用できる親および子コンポーネントが表示されます。たとえば、XS (親)と XSSESSION (XS の子)の両方が有効なコンポーネント名です。親コンポーネントを選択すると、コマンド・ルールがそれとその子コンポーネントに適用されます。
action_name	component_name 設定のアクション
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。

パラメータ	説明
	● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2) アプリケーション・ルートで共通コマンド・ルールを更新し、関連付けられた PDB にそれを表示できるようにする場合は、アプリケーションを同期させる必要があります。たとえば: `ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;`
pl_sql_stack	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックを記録するかどうかを指定します。PL/SQL スタックを記録する場合は TRUE と入力し、記録しない場合は FALSE と入力します。

例

次の例では、マルチテナント環境で共通セッション・イベント・コマンド・ルールを更新する方法を示します。このコマンド・ルールはアプリケーション・ルート内にあるため、このプロシージャを実行するユーザーはCDBルート内にいる必要があります。このコマンド・ルールに関連付けられているユーザー名またはルール・セットは、共通である必要があります。

```
BEGIN
 DBMS_MACADM.UPDATE_SESSION_EVENT_CMD_RULE(
  rule_set_name => 'Allow Sessions',
  event_name    => '47999',
  enabled       => DBMS_MACUTL.G_NO,
  scope         => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

16.12 UPDATE_SYSTEM_EVENT_CMD_RULEプロシージャ

UPDATE_SYSTEM_EVENT_CMD_RULEプロシージャは、ALTER SYSTEM文に基づいて、システム・イベント・コマンド・ルールを更新します。

マルチテナント環境では、共通およびローカルの両方のセッション・イベント・コマンド・ルールを更新できます。

構文

```
DBMS_MACADM.UPDATE_SYSTEM_EVENT_CMD_RULE(
 rule_set_name   IN VARCHAR2,
 enabled         IN VARCHAR2,
 event_name      IN VARCHAR2 DEFAULT,
 component_name  IN VARCHAR2 DEFAULT,
 action_name     IN VARCHAR2 DEFAULT,
 scope           IN NUMBER DEFAULT,
 pl_sql_stack    IN BOOLEAN DEFAULT);
```

パラメータ

表16-14 UPDATE_SYSTEM_EVENT_CMD_RULEのパラメータ

パラメータ	説明
rule_set_name	コマンド・ルールに関連付けるルール・セットの名前。マルチテナント環境では、このルール・セットが、システム・イベント・コマンド・ルールが共通である場合は共通、そのコマンド・ルールがローカルである場合はローカルであることを確認してください。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
enabled	次のいずれかのオプションを指定してコマンド・ルールのステータスを設定します。 ● コマンド・ルールを有効にするには(デフォルト)、DBMS_MACUTL.G_YES または y ● シミュレーション・ログでの違反の取得など、コマンド・ルールを無効にするには、DBMS_MACUTL.G_NO または n ● SQL 文の実行を可能にするがシミュレーション・ログで違反を捕捉するには、DBMS_MACUTL.G_SIMULATION または s
event_name	コマンド・ルールで定義されているイベント。この設定により、コマンド・ルールが ALTER SYSTEM SET EVENTS event_name に対応できるようになります。たとえば、トレース・イベントを追跡するには、event_name を TRACE に設定します。
component_name	event_name 設定のコンポーネント例の設定は、DV、OLS または GCS です。 ORADEBUG DOC COMPONENT RDBMS をユーザーSYS として発行することで、有効なコンポーネント名を確認できます。出力には、component_name 設定に使用できる親および子コンポーネントが表示されます。たとえば、XS (親)と XSSESSION (XS の子)の両方が有効なコンポーネント名です。親コンポーネントを選択すると、コマンド・ルールがそれとその子コンポーネントに適用されます。
action_name	component_name 設定のアクション
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

パラメータ	説明
	アプリケーション・ルートで共通コマンド・ルールを更新し、関連付けられた PDB にそれを表示できるようにする場合は、アプリケーションを同期させる必要があります。たとえば: `ALTER PLUGGABLE DATABASE APPLICATION saas_sales_app SYNC;`
`pl_sql_stack`	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックを記録するかどうかを指定します。PL/SQL スタックを記録する場合は TRUE と入力し、記録しない場合は FALSE と入力します。

例

次の例では、マルチテナント環境で共通システム・イベント・コマンド・ルールを更新する方法を示します。このコマンド・ルールはアプリケーション・ルート内にあるため、このプロシージャを実行するユーザーはCDBルート内にいる必要があります。このコマンド・ルールに関連付けられているユーザー名またはルール・セットは、共通である必要があります。

```
BEGIN
 DBMS_MACADM.UPDATE_SYSTEM_EVENT_CMD_RULE(
 rule_set_name   => 'Disabled',
 event_name      => 'TRACE',
 component_name  => 'DV',
 enabled         => 'n',
 scope           => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

親トピック: Oracle Database Vaultコマンド・ルールのAPI

17 Oracle Database VaultファクタのAPI

DBMS_MACADM PL/SQLパッケージには、ファクタ関連のOracle Database Vaultルール・プロシージャおよび関数があり、DVFには、ファクタを管理する関数があります。

- DBMS_MACADMファクタ・プロシージャおよびファンクション
  DBMS_MACADM PL/SQLパッケージには、ファクタを構成するプロシージャおよびファンクションが用意されています。
- Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクション
  Oracle Database Vaultには手続き型インタフェースが用意されており、Database Vaultセキュリティ・オプションの管理およびDatabase Vaultセキュリティの実施の管理に使用できます。
- Oracle Database Vault DVF PL/SQLファクタ・ファンクション
  DBMS_MACADM PL/SQLパッケージを使用して様々なファクタを管理する際、Oracle Database VaultによってDVFスキーマ・ファンクションが維持されます。

17.1 DBMS_MACADMファクタのプロシージャおよびファンクション

DBMS_MACADM PL/SQLパッケージには、ファクタを構成するプロシージャおよびファンクションが用意されています。

DV_OWNERロールまたはDV_ADMINロールを付与されているユーザーのみがこれらのプロシージャとファンクションを使用できます。

- ADD_FACTOR_LINKプロシージャ
  ADD_FACTOR_LINKプロシージャは、2つのファクタの親子関係を指定します。
- ADD_POLICY_FACTORプロシージャ
  ADD_POLICY_FACTORプロシージャは、ファクタのラベルをポリシーのOracle Label Securityラベルに含めることを指定します。
- CHANGE_IDENTITY_FACTORプロシージャ
  CHANGE_IDENTITY_FACTORプロシージャは、アイデンティティを別ファクタと関連付けます。
- CHANGE_IDENTITY_VALUEプロシージャ
  CHANGE_IDENTITY_FACTORプロシージャは、アイデンティティの値を更新します。
- CREATE_DOMAIN_IDENTITYプロシージャ
  CREATE_DOMAIN_IDENTITYプロシージャは、Oracle Real Application Clusters (Oracle RAC)およびOracle Label Securityに使用されます。
- CREATE_FACTORプロシージャ
  CREATE_FACTORプロシージャはファクタを作成します。
- CREATE_FACTOR_TYPEプロシージャ
  CREATE_FACTOR_TYPEプロシージャは、ユーザー定義のファクタ・タイプを作成します。
- CREATE_IDENTITYプロシージャ
  CREATE_IDENTITYプロシージャは、指定されたファクタに対してアイデンティティおよび関連付けられた信頼レベルを割り当てます。
- CREATE_IDENTITY_MAPプロシージャ
  CREATE_IDENTITY_MAPプロシージャは、リンクされた子ファクタ(サブファクタ)の値からファクタのアイデンティティを導出できるテストを定義します。
- DELETE_FACTORプロシージャ
  DELETE_FACTORプロシージャはファクタを削除します。
- DELETE_FACTOR_LINKプロシージャ

DELETE_FACTOR_LINKプロシージャは、2つのファクタの親子関係を削除します。

- DELETE_FACTOR_TYPEプロシージャ
  DELETE_FACTOR_TYPEプロシージャはファクタ・タイプを削除します。
- DELETE_IDENTITYプロシージャ
  DELETE_IDENTITYプロシージャは、既存のファクタからアイデンティティを削除します。
- DELETE_IDENTITY_MAPプロシージャ
  DELETE_IDENTITY_MAPプロシージャは、ファクタのアイデンティティ・マップを削除します。
- DROP_DOMAIN_IDENTITYプロシージャ
  DROP_DOMAIN_IDENTITYプロシージャは、Oracle Real Application Clustersデータベース・ノードをドメインから削除します。
- GET_SESSION_INFOファンクション
  GET_SESSION_INFOファンクションは、現行セッションについてSYS.V_$SESSIONシステム表の情報を返します。
- GET_INSTANCE_INFOファンクション
  GET_INSTANCE_INFOファンクションは、現行のデータベース・インスタンスについてSYS.V_$INSTANCEシステム表の情報を返します。
- RENAME_FACTORプロシージャ
  RENAME_FACTORプロシージャは、ファクタの名前を変更します。名前の変更は、そのファクタが使用されているすべての箇所に反映されます。
- RENAME_FACTOR_TYPEプロシージャ
  RENAME_FACTORプロシージャは、ファクタ・タイプの名前を変更します。名前の変更は、そのファクタ・タイプが使用されているすべての箇所に反映されます。
- UPDATE_FACTORプロシージャ
  UPDATE_FACTORプロシージャは、ファクタ・タイプの説明を更新します。
- UPDATE_FACTOR_TYPEプロシージャ
  UPDATE_FACTOR_TYPEプロシージャはファクタ・タイプを更新します。
- UPDATE_IDENTITYプロシージャ
  UPDATE_IDENTITYプロシージャは、ファクタ・アイデンティティの信頼レベルを更新します。

関連トピック

- ファクタの構成
- Oracle Database VaultユーティリティのAPI

親トピック: Oracle Database VaultファクタのAPI

17.1.1 ADD_FACTOR_LINKプロシージャ

ADD_FACTOR_LINKプロシージャは、2つのファクタの親子関係を指定します。

構文

```
DBMS_MACADM.ADD_FACTOR_LINK(
parent_factor_name IN VARCHAR2,
child_factor_name  IN VARCHAR2,
label_indicator    IN VARCHAR2);
```

パラメータ

表17-1 ADD_FACTOR_LINKのパラメータ

パラメータ	説明
parent_factor_name	親ファクタ名。 現行のデータベース・インスタンスで既存の親子ファクタを確認するには、「DBA_DV_FACTOR_LINK ビュー」で説明されている DBA_DV_FACTOR_LINK ビューを問い合せます。
child_factor_name	子ファクタ名。
label_indicator	親ファクタにリンクされる子ファクタを、Oracle Label Security 統合での親ファクタのラベルに含めることを示します。DBMS_MACUTL.G_YES(Yes の場合)または DBMS_MACUTL.G_NO(No の場合)のいずれかを指定します。 ファクタに関連付けられている Oracle Label Security ポリシーおよびラベルを確認するには、「Oracle Database Vault のデータ・ディクショナリ・ビュー」で説明されている次のビューを問い合せます。 ● DBA_DV_MAC_POLICY: 現行のデータベース・インスタンスに定義されている Oracle Label Security ポリシーが表示されます。 ● DBA_DV_MAC_POLICY_FACTOR: 現行のデータベース・インスタンスの Oracle Label Security ポリシーに関連付けられているファクタが表示されます。 ● DBA_DV_POLICY_LABEL: 各ポリシーの DBA_DV_IDENTITY ビューの各ファクタ識別子に対する Oracle Label Security ラベルが表示されます。

例

```
BEGIN
 DBMS_MACADM.ADD_FACTOR_LINK(
parent_factor_name => 'HQ_ClientID',
child_factor_name  => 'Div1_ClientID',
label_indicator    => DBMS_MACUTL.G_YES);
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.2 ADD_POLICY_FACTORプロシージャ

ADD_POLICY_FACTORプロシージャは、ファクタのラベルをポリシーのOracle Label Securityラベルに含めることを指定します。

構文

```
DBMS_MACADM.ADD_POLICY_FACTOR(
policy_name  IN VARCHAR2,
```

```
factor_name  IN VARCHAR2);
```

パラメータ

表17-2 ADD_POLICY_FACTORのパラメータ

パラメータ	説明
policy_name	Oracle Label Security ポリシー名。 現行のデータベース・インスタンスに定義されているポリシーを確認するには、「DBA_DV_MAC_POLICY ビュー」で説明されている DBA_DV_MAC_POLICY ビューに問い合せます。 Oracle Label Security ポリシーに関連付けられているファクタを確認するには、「DBA_DV_MAC_POLICY ビュー」で説明されている DBA_DV_MAC_POLICY_FACTOR に問い合せます。
factor_name	ファクタ名。 既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。

例

```
BEGIN
 DBMS_MACADM.ADD_POLICY_FACTOR(
policy_name  => 'AccessData',
factor_name  => 'Sector2_ClientID');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.3 CHANGE_IDENTITY_FACTORプロシージャ

CHANGE_IDENTITY_FACTORプロシージャは、アイデンティティを別ファクタと関連付けます。

構文

```
DBMS_MACADM.CHANGE_IDENTITY_FACTOR(
factor_name     IN VARCHAR2,
value           IN VARCHAR2,
new_factor_name IN VARCHAR2);
```

パラメータ

表17-3 CHANGE_IDENTITY_FACTORのパラメータ

パラメータ	説明
factor_name	現在のファクタ名。 既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている

パラメータ	説明
	DBA_DV_FACTOR ビューに問い合せます。
value	更新するアイデンティティの値。 現行のデータベース・インスタンスで各ファクタの既存のアイデンティティを確認するには、「DBA_DV_IDENTITY ビュー」で説明されている DBA_DV_IDENTITY ビューに問い合せます。 現行のアイデンティティ・マップを確認するには、「DBA_DV_IDENTITY_MAP ビュー」で説明されている DBA_DV_IDENTITY_MAP ビューに問い合せます。
new_factor_name	アイデンティティと関連付けるファクタの名前。「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せることで確認できます。

例

```
BEGIN
 DBMS_MACADM.CHANGE_IDENTITY_FACTOR(
factor_name     => 'Sector2_ClientID',
value           => 'intranet',
new_factor_name => 'Sector4_ClientID');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.4 CHANGE_IDENTITY_VALUEプロシージャ

CHANGE_IDENTITY_FACTORプロシージャは、アイデンティティの値を更新します。

構文

```
DBMS_MACADM.CHANGE_IDENTITY_VALUE(
factor_name  IN VARCHAR2,
value        IN VARCHAR2,
new_value    IN VARCHAR2);
```

パラメータ

表17-4 CHANGE_IDENTITY_VALUEのパラメータ

パラメータ	説明
factor_name	ファクタ名。 既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
value	アイデンティティに関連付けられている現在の値。

パラメータ	説明
	現行のデータベース・インスタンスで各ファクタの既存のアイデンティティを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_IDENTITY ビューに問い合せます。 現行のアイデンティティ・マップを確認するには、「DBA_DV_IDENTITY_MAP ビュー」で説明されている DBA_DV_IDENTITY_MAP ビューに問い合せます。
new_value	新しいアイデンティティ値(大/小文字混在で最大 1024 文字)。

例

```
BEGIN
 DBMS_MACADM.CHANGE_IDENTITY_VALUE(
factor_name  => 'Sector2_ClientID',
value        => 'remote',
new_value    => 'intranet');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.5 CREATE_DOMAIN_IDENTITYプロシージャ

CREATE_DOMAIN_IDENTITYプロシージャは、Oracle Real Application Clusters (Oracle RAC)およびOracle Label Securityに使用されます。

Oracle RACデータベース・ノードをドメイン・ファクタ・アイデンティティに追加し、Oracle Label Securityポリシーに従ってラベルを付けます

構文

```
DBMS_MACADM.CREATE_DOMAIN_IDENTITY(
domain_name  IN VARCHAR2,
domain_host  IN VARCHAR2,
policy_name  IN VARCHAR2 DEFAULT NULL,
domain_label IN VARCHAR2 DEFAULT NULL);
```

パラメータ

表17-5 CREATE_DOMAIN_IDENTITYのパラメータ

パラメータ	説明
domain_name	ホストを追加するドメインの名前。 分散データベース・システムのネットワーク構造内でデータベースの論理的な場所を確認するには、「Oracle Database Vault の DVF PL/SQL ファクタ・ファンクション」で説明されている DVF.F$DATABASE_DOMAIN ファンクションを実行します。
domain_host	ドメインに追加される Oracle Real Application Clusters ホスト名。 データベースのホスト名を確認するには、「Oracle Database Vault の DVF PL/SQL ファクタ・

パラメータ	説明
	ファンクション」で説明されている DVF.F$DATABASE_HOSTNAME ファンクションを実行します。
policy_name	Oracle Label Security ポリシー名。ポリシー名を省略すると、ドメインはどのポリシーにも関連付けられません。 使用可能なポリシーを確認するには、「DBA_DV_MAC_POLICY ビュー」で説明されている DBA_DV_MAC_POLICY ビューに問い合せます。
domain_label	Oracle Label Security ポリシーを追加するドメインの名前。

例

```
BEGIN
 DBMS_MACADM.CREATE_DOMAIN_IDENTITY(
domain_name  => 'example',
domain_host  => 'mydom_host',
policy_name  => 'AccessData',
domain_label => 'sensitive');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.6 CREATE_FACTORプロシージャ

CREATE_FACTORプロシージャはファクタを作成します。

ファクタを作成した後で、「CREATE_IDENTITYプロシージャ」で説明されているCREATE_IDENTITYプロシージャを使用することで、アイデンティティを付与できます。

構文

```
DBMS_MACADM.CREATE_FACTOR(
factor_name       IN VARCHAR2,
factor_type_name  IN VARCHAR2,
description       IN VARCHAR2,
rule_set_name     IN VARCHAR2,
get_expr          IN VARCHAR2,
validate_expr     IN VARCHAR2,
identify_by       IN NUMBER,
labeled_by        IN NUMBER,
eval_options      IN NUMBER,
audit_options     IN NUMBER,
fail_options      IN NUMBER);
```

パラメータ

表17-6 CREATE_FACTORのパラメータ

パラメータ	説明
factor_name	ファクタ名(空白を使用せず、大/小文字混在で最大 128 文字)。

パラメータ	説明
	現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
factor_type_name	ファクタのタイプ(空白を使用せず、大/小文字混在で最大 128 文字)。 既存のファクタ・タイプを確認するには、「DBA_DV_FACTOR_TYPE ビュー」で説明されている DBA_DV_FACTOR_TYPE ビューに問い合せます。
description	ファクタの目的の説明(大/小文字混在で最大 1024 文字)。
rule_set_name	ファクタ・アイデンティティを設定する時期および方法の制御にルール・セットを使用する場合のルール・セット名。 既存のルール・セットを確認するには、「Oracle Database Vault のデータ・ディクショナリ・ビュー」で説明されている DBA_DV_RULE_SET ビューに問い合せます。ファクタへのルール・セットの割当ての詳細は、「ファクタへのルール・セットの割当て」も参照してください。
get_expr	ファクタのアイデンティティを取得する有効な PL/SQL 式。大/小文字混在で最大 255 文字まで使用できます。詳細は、「ファクタの取得メソッドの設定」を参照してください。audit_options パラメータも参照してください。
validate_expr	ファクタを検証するプロシージャの名前。これは、ブール値(TRUE または FALSE)を返してファクタのアイデンティティを検証する有効な PL/SQL 式です。詳細は、「ファクタの検証メソッドの設定」を参照してください。
identify_by	get_expr パラメータの式セットに基づいてファクタのアイデンティティを決定するオプション。 ● DBMS_MACUTL.G_IDENTIFY_BY_CONSTANT: 定数 ● DBMS_MACUTL.G_IDENTIFY_BY_METHOD: メソッド ● DBMS_MACUTL.G_IDENTIFY_BY_FACTOR: ファクタ ● DBMS_MACUTL.G_IDENTIFY_BY_CONTEXT: コンテキスト 詳細は、「ファクタの識別情報の設定」を参照してください。
labeled_by	ファクタのラベル付けのオプション。 ● DBMS_MACUTL.G_LABELED_BY_SELF: Oracle Label Security ポリシー(デフォルト)に関連付けられているラベルから直接ファクタのアイデンティティをラベル付

パラメータ	説明
	けします。 ● DBMS_MACUTL.G_LABELED_BY_FACTORS: 子ファクタ・アイデンティティのラベルからファクタ・アイデンティティ・ラベルを導出します。 詳細は、「ファクタの Oracle Label Security ラベル付け情報の設定」を参照してください。
eval_options	ユーザーがログインするときにファクタを評価するオプション。 ● DBMS_MACUTL.G_EVAL_ON_SESSION: データベース・セッションの作成時(デフォルト) ● DBMS_MACUTL.G_EVAL_ON_ACCESS: ファクタがアクセスされる際に毎回 ● DBMS_MACUTL.G_EVAL_ON_STARTUP: 起動時 詳細は、「ファクタの評価情報の設定」を参照してください。
audit_options	カスタムの Oracle Database Vault 監査レコードから生成する場合にファクタを監査するオプション。 ● DBMS_MACUTL.G_AUDIT_OFF: 監査を無効にします。 ● DBMS_MACUTL.G_AUDIT_ALWAYS: 常に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_GET_ERROR: get_expr がエラーを戻した場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_GET_NULL: get_expr が null の場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_VALIDATE_ERROR: 検証プロシージャがエラーを戻した場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_VALIDATE_FALSE: 検証プロシージャが false の場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_TRUST_LEVEL_NULL: 信頼レベルが設定されていない場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_TRUST_LEVEL_NEG: 信頼レベルが負の場合に監査します。 audit_options パラメータは、従来の監査にのみ適用されます。統合監査を有効にした

パラメータ	説明
	場合は、audit_options を使用するかわりに統合監査ポリシーを作成します。 詳細は、「ファクタの監査オプションの設定」を参照してください。
fail_options	ファクタ・エラーをレポートするオプション。 ● DBMS_MACUTL.G_FAIL_WITH_MESSAGE: エラー・メッセージを表示します(デフォルト)。 ● DBMS_MACUTL.G_FAIL_SILENTLY: エラー・メッセージを表示しません。 詳細は、「ファクタのエラー・オプションの設定」を参照してください。

例

```
BEGIN
 DBMS_MACADM.CREATE_FACTOR(
factor_name       => 'Sector2_DB',
factor_type_name  => 'Instance',
description       => ' ',
rule_set_name     => 'Limit_DBA_Access',
get_expr          => 'UPPER(SYS_CONTEXT(''USERENV'',''DB_NAME''))',
validate_expr     => 'dbavowner.check_db_access',
identify_by       => DBMS_MACUTL.G_IDENTIFY_BY_METHOD,
labeled_by        => DBMS_MACUTL.G_LABELED_BY_SELF,
eval_options      => DBMS_MACUTL.G_EVAL_ON_SESSION,
audit_options     => DBMS_MACUTL.G_AUDIT_OFF,
fail_options      => DBMS_MACUTL.G_FAIL_SILENTLY);
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.7 CREATE_FACTOR_TYPEプロシージャ

CREATE_FACTOR_TYPEプロシージャは、ユーザー定義のファクタ・タイプを作成します。

構文

```
DBMS_MACADM.CREATE_FACTOR_TYPE(
name        IN VARCHAR2,
description IN VARCHAR2);
```

パラメータ

表17-7 CREATE_FACTOR_TYPEのパラメータ

パラメータ	説明
name	ファクタ・タイプ名(空白を使用せず、大/小文字混在で最大 128 文字)。 既存のファクタ・タイプを確認するには、「DBA_DV_FACTOR_TYPE ビュー」で説明されている

パラメータ	説明
	DBA_DV_FACTOR_TYPE ビューに問い合せます。
description	ファクタ・タイプの目的の説明(大/小文字混在で最大 1024 文字)。

例

```
BEGIN
 DBMS_MACADM.CREATE_FACTOR_TYPE(
name        => 'Sector2Instance',
description => 'Checks DB instances used in Sector 2');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.8 CREATE_IDENTITYプロシージャ

CREATE_IDENTITYプロシージャは、指定されたファクタに対してアイデンティティおよび関連付けられた信頼レベルを割り当てます。

ファクタの作成後には、アイデンティティを割り当てる必要があります。

構文

```
DBMS_MACADM.CREATE_IDENTITY(
factor_name  IN VARCHAR2,
value        IN VARCHAR2,
trust_level  IN NUMBER);
```

パラメータ

表17-8 CREATE_IDENTITYのパラメータ

パラメータ	説明
factor_name	ファクタ名。 既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
value	ファクタの実際の値(大/小文字混在で最大 1024 文字)。たとえば、IP_Address ファクタのアイデンティティは、192.0.2.12 という IP アドレスになります。
trust_level	同じファクタの別のアイデンティティと比較した信頼の度合いを示す数値。一般に、信頼レベルの数値が高く設定されているほど信頼の度合いも高くなります。信頼レベル 10 は、非常に信頼度が高いことを表します。信頼レベルの数値が負の場合は信頼できません。 信頼レベルおよびラベル・セキュリティの詳細は、「ファクタ・アイデンティティの作成および構成」を参照してください。

例

```
BEGIN
 DBMS_MACADM.CREATE_IDENTITY(
factor_name  => 'Sector2_ClientID',
value        => 'intranet',
trust_level  => 5);
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.9 CREATE_IDENTITY_MAPプロシージャ

CREATE_IDENTITY_MAPプロシージャは、リンクされた子ファクタ(サブファクタ)の値からファクタのアイデンティティを導出できるテストを定義します。

構文

```
DBMS_MACADM.CREATE_IDENTITY_MAP(
identity_factor_name  IN VARCHAR2,
identity_factor_value IN VARCHAR2,
parent_factor_name    IN VARCHAR2,
child_factor_name     IN VARCHAR2,
operation             IN VARCHAR2,
operand1              IN VARCHAR2,
operand2              IN VARCHAR2);
```

パラメータ

表17-9 CREATE_IDENTITY_MAPのパラメータ

パラメータ	説明
identity_factor_name	アイデンティティ・マップの対象のファクタ。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
identity_factor_value	アイデンティティ・マップの評価が TRUE の場合は、ファクタで想定される値。 既存のファクタ・アイデンティティを確認するには、「DBA_DV_IDENTITY ビュー」で説明されている DBA_DV_IDENTITY ビューに問い合せます。 現行のファクタ・アイデンティティ・マップを確認するには、「DBA_DV_IDENTITY_MAP ビュー」で説明されている DBA_DV_IDENTITY_MAP を使用します。
parent_factor_name	マップが関連する親ファクタ・リンク。 既存の親子ファクタ・マップを確認するには、「DBA_DV_IDENTITY_MAP

パラメータ	説明
	ビュー」で説明されている DBA_DV_IDENTITY_MAP ビューに問い合せます。
child_factor_name	マップが関連する子ファクタ・リンク。
operation	アイデンティティ・マップの関係演算子(たとえば、<、>、=など)。
operand1	関係演算子の左オペランド。入力する下限値を表します。
operand2	関係演算子の右オペランド。入力する上限値を表します。

例

```
BEGIN
 DBMS_MACADM.CREATE_IDENTITY_MAP(
identity_factor_name  => 'Sector2_ClientID',
identity_factor_value => 'intranet',
parent_factor_name    => 'HQ_ClientID',
child_factor_name     => 'Div1_ClientID',
operation             => '<',
operand1              => '192.0.2.50',
operand2              => '192.0.2.100');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.10 DELETE_FACTORプロシージャ

DELETE_FACTORプロシージャはファクタを削除します。

構文

```
DBMS_MACADM.DELETE_FACTOR(
  factor_name IN VARCHAR2);
```

パラメータ

表17-10 DELETE_FACTORのパラメータ

パラメータ	説明
factor_name	ファクタ名。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。

例

```
EXEC DBMS_MACADM.DELETE_FACTOR('Sector2_ClientID');
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.11 DELETE_FACTOR_LINKプロシージャ

DELETE_FACTOR_LINKプロシージャは、2つのファクタの親子関係を削除します。

構文

```
DBMS_MACADM.DELETE_FACTOR_LINK(
parent_factor_name IN VARCHAR2,
child_factor_name  IN VARCHAR2);
```

パラメータ

表17-11 DELETE_FACTOR_LINKのパラメータ

パラメータ	説明
parent_factor_name	ファクタ名。 現行のデータベース・インスタンスで親子マッピングに使用されているファクタを確認するには、「DBA_DV_FACTOR_LINK ビュー」で説明されている DBA_DV_FACTOR_LINK ビューに問い合せます。
child_factor_name	ファクタ名

例

```
BEGIN
 DBMS_MACADM.DELETE_FACTOR_LINK(
parent_factor_name => 'HQ_ClientID',
child_factor_name  => 'Div1_ClientID');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.12 DELETE_FACTOR_TYPEプロシージャ

DELETE_FACTOR_TYPEプロシージャはファクタ・タイプを削除します。

構文

```
DBMS_MACADM.DELETE_FACTOR_TYPE(
name IN VARCHAR2);
```

パラメータ

表17-12 DELETE_FACTOR_TYPEのパラメータ

パラメータ	説明
name	ファクタ・タイプ名。 既存のファクタ・タイプを確認するには、「DBA_DV_FACTOR_TYPE ビュー」で説明されている

パラメータ	説明
	DBA_DV_FACTOR_TYPE ビューに問い合せます。

例

```
EXEC DBMS_MACADM.DELETE_FACTOR_TYPE('Sector2Instance');
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.13 DELETE_IDENTITYプロシージャ

DELETE_IDENTITYプロシージャは、既存のファクタからアイデンティティを削除します。

構文

```
DBMS_MACADM.DELETE_IDENTITY(
factor_name IN VARCHAR2,
value       IN VARCHAR2);
```

パラメータ

表17-13 DELETE_IDENTITYのパラメータ

パラメータ	説明
factor_name	ファクタ名。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
value	ファクタに関連付けられているアイデンティティ値。 現行のデータベース・インスタンス内の各ファクタのアイデンティティを確認するには、「DBA_DV_IDENTITY ビュー」で説明されている DBA_DV_IDENTITY ビューに問い合せます。

例

```
BEGIN
 DBMS_MACADM.DELETE_IDENTITY(
factor_name => 'Sector2_ClientID',
value       => 'intranet');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.14 DELETE_IDENTITY_MAPプロシージャ

DELETE_IDENTITY_MAPプロシージャは、ファクタのアイデンティティ・マップを削除します。

構文

```
DBMS_MACADM.DELETE_IDENTITY_MAP(
```

```
identity_factor_name  IN VARCHAR2,
identity_factor_value IN VARCHAR2,
parent_factor_name    IN VARCHAR2,
child_factor_name     IN VARCHAR2,
operation             IN VARCHAR2,
operand1              IN VARCHAR2,
operand2              IN VARCHAR2);
```

パラメータ

表17-14 DELETE_IDENTITY_MAPのパラメータ

パラメータ	説明
identity_factor_name	アイデンティティ・マップの対象のファクタ。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
identity_factor_value	アイデンティティ・マップの評価が TRUE の場合は、ファクタで想定される値。 既存のファクタ・アイデンティティを確認するには、「DBA_DV_IDENTITY ビュー」で説明されている DBA_DV_IDENTITY ビューに問い合せます。 現行のファクタ・アイデンティティ・マップを確認するには、「DBA_DV_IDENTITY_MAP ビュー」で説明されている DBA_DV_IDENTITY_MAP に問い合せます。
parent_factor_name	マップが関連する親ファクタ・リンク。 既存の親子ファクタを確認するには、「DBA_DV_FACTOR_LINK ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
child_factor_name	マップが関連する子ファクタ。
operation	アイデンティティ・マップの関係演算子(たとえば、<、>、=など)。
operand1	関係演算子の左(下限値)オペランド。
operand2	関係演算子の右(上限値)オペランド。

例

```
BEGIN
 DBMS_MACADM.DELETE_IDENTITY_MAP(
identity_factor_name  => 'Sector2_ClientID',
identity_factor_value => 'intranet',
parent_factor_name    => 'HQ_ClientID',
child_factor_name     => 'Div1_ClientID',
```

```
operation             => '<',
operand1              => '192.0.2.10',
operand2              => '192.0.2.15');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.15 DROP_DOMAIN_IDENTITYプロシージャ

DROP_DOMAIN_IDENTITYプロシージャは、Oracle Real Application Clustersデータベース・ノードをドメインから削除します。

構文

```
DBMS_MACADM.DROP_DOMAIN_IDENTITY(
domain_name  IN VARCHAR2,
domain_host  IN VARCHAR2);
```

パラメータ

表17-15 DROP_DOMAIN_IDENTITYのパラメータ

パラメータ	説明
domain_name	ホストが追加されたドメインの名前。 DB_DOMAIN 初期化パラメータで指定されたデータベースのドメインを確認するには、「F$DATABASE_DOMAIN ファンクション」で説明されている DVF.F$DATABASE_DOMAIN ファンクションを実行します。
domain_host	ドメインに追加された Oracle Real Application Clusters ホスト名。 指定されたデータベースのホスト名を確認するには、「F$DATABASE_NAME ファンクション」で説明されている DVF.F$DATABASE_HOSTNAME ファンクションを実行します。

例

```
BEGIN
 DBMS_MACADM.DROP_DOMAIN_IDENTITY(
domain_name  => 'example',
domain_host  => 'mydom_host');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.16 GET_SESSION_INFOファンクション

GET_SESSION_INFOファンクションは、現行セッションについてSYS.V_$SESSIONシステム表の情報を返します。

V$SESSIONデータ・ディクショナリ・ビューにも、この表のセッション情報が含まれます。詳細は、『Oracle Databaseリファレンス』を参照してください。

構文

```
DBMS_MACADM.GET_SESSION_INFO(
p_parameter IN VARCHAR2)
RETURN VARCHAR2;
```

パラメータ

表17-16 GET_SESSION_INFOのパラメータ

パラメータ	説明
p_parameter	SYS.V_$SESSION システム表の列名。

例

```
DECLARE
 session_var varchar2 := null;
BEGIN
 session_var = DBMS_MACADM.GET_SESSION_INFO('PROCESS');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.17 GET_INSTANCE_INFOファンクション

GET_INSTANCE_INFOファンクションは、現行のデータベース・インスタンスについてSYS.V_$INSTANCEシステム表の情報を返します。

V$INSTANCEデータ・ディクショナリ・ビューにも、この表のデータベース・インスタンス情報が含まれます。詳細は、『Oracle Databaseリファレンス』を参照してください。

構文

```
DBMS_MACADM.GET_INSTANCE_INFO(
p_parameter IN VARCHAR2)
RETURN VARCHAR2;
```

パラメータ

表17-17 GET_INSTANCE_INFOのパラメータ

パラメータ	説明
p_parameter	SYS.V_$INSTANCE システム表の列名。

例

```
DECLARE
 instance_var varchar2 := null;
BEGIN
 instance_var = DBMS_MACADM.GET_INSTANCE_INFO('INSTANCE_NAME');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.18 RENAME_FACTORプロシージャ

RENAME_FACTORプロシージャは、ファクタの名前を変更します。名前の変更は、そのファクタが使用されているすべての箇所に反映されます。

構文

```
DBMS_MACADM.RENAME_FACTOR(
factor_name     IN VARCHAR2,
new_factor_name IN VARCHAR2);
```

パラメータ

表17-18 RENAME_FACTORのパラメータ

パラメータ	説明
factor_name	現在のファクタ名。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
new_factor_name	新しいファクタ名(空白を使用せず、大/小文字混在で最大 128 文字)。

例

```
BEGIN
 DBMS_MACADM.RENAME_FACTOR(
factor_name     => 'Sector2_ClientID',
new_factor_name => 'Sector2_Clients');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.19 RENAME_FACTOR_TYPEプロシージャ

RENAME_FACTORプロシージャは、ファクタ・タイプの名前を変更します。名前の変更は、そのファクタ・タイプが使用されているすべての箇所に反映されます。

構文

```
DBMS_MACADM.RENAME_FACTOR_TYPE(
old_name  IN VARCHAR2,
new_name  IN VARCHAR2);
```

パラメータ

表17-19 RENAME_FACTOR_TYPEのパラメータ

パラメータ	説明
old_name	現在のファクタ・タイプ名。 現行のデータベース・インスタンスで既存のファクタ・タイプを確認するには、

パラメータ	説明
	「DBA_DV_FACTOR_TYPE ビュー」で説明されている DBA_DV_FACTOR_TYPE ビューに問い合せます。
new_name	新しいファクタ・タイプ名(空白を使用せず、大/小文字混在で最大 128 文字)。

例

```
BEGIN
 DBMS_MACADM.RENAME_FACTOR_TYPE(
old_name  => 'Sector2Instance',
new_name  => 'Sector2DBInstance');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.20 UPDATE_FACTORプロシージャ

UPDATE_FACTORプロシージャは、ファクタ・タイプの説明を更新します。

構文

```
DBMS_MACADM.UPDATE_FACTOR(
factor_name        IN VARCHAR2,
factor_type_name   IN VARCHAR2,
description        IN VARCHAR2,
rule_set_name      IN VARCHAR2,
get_expr           IN VARCHAR2,
validate_expr      IN VARCHAR2,
identify_by        IN NUMBER,
labeled_by         IN NUMBER,
eval_options       IN NUMBER,
audit_options      IN NUMBER,
fail_options       IN NUMBER);
```

パラメータ

表17-20 UPDATE_FACTOR

パラメータ	説明
factor_name	ファクタ名。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
factor_type_name	ファクタ・タイプ名。 既存のファクタ・タイプを確認するには、「DBA_DV_FACTOR_TYPE ビュー」で説明されている DBA_DV_FACTOR_TYPE ビューに問い合せます。

パラメータ	説明
description	ファクタの目的の説明(大/小文字混在で最大 1024 文字)。
rule_set_name	ファクタ・アイデンティティを設定する時期および方法の制御に使用されるルール・セットの名前。 既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューに問い合せます。 ファクタへのルール・セットの割当ての詳細は、「ファクタへのルール・セットの割当て」も参照してください。
get_expr	ファクタのアイデンティティを取得する有効な PL/SQL 式。大/小文字混在で最大 255 文字まで使用できます。詳細は、「ファクタの取得メソッドの設定」を参照してください。audit_options パラメータも参照してください。
validate_expr	ファクタを検証するプロシージャの名前。これは、ブール値(TRUE または FALSE)を返してファクタのアイデンティティを検証する有効な PL/SQL 式です。詳細は、「ファクタの検証メソッドの設定」を参照してください。
identify_by	get_expr パラメータの式セットに基づいてファクタのアイデンティティを決定するオプション。 ● DBMS_MACUTL.G_IDENTIFY_BY_CONSTANT: 定数 ● DBMS_MACUTL.G_IDENTIFY_BY_METHOD: メソッド ● DBMS_MACUTL.G_IDENTIFY_BY_FACTOR: ファクタ ● DBMS_MACUTL.G_IDENTIFY_BY_CONTEXT: コンテキスト 詳細は、「ファクタの識別情報の設定」を参照してください。
labeled_by	ファクタのラベル付けのオプション。 ● DBMS_MACUTL.G_LABELED_BY_SELF: Oracle Label Security ポリシーに関連付けられているラベルから直接ファクタのアイデンティティをラベル付けします。 ● DBMS_MACUTL.G_LABELED_BY_FACTORS: 子ファクタ・アイデンティティのラベルからファクタ・アイデンティティ・ラベルを導出します。 labeled_by のデフォルトは設定済の値であり、DBA_DV_FACTOR データ・ディクショナリ・ビューに問い合せることで確認できます。

パラメータ	説明
	詳細は、「ファクタの Oracle Label Security ラベル付け情報の設定」を参照してください。
eval_options	ユーザーがログインするときにファクタを評価するオプション。 ● DBMS_MACUTL.G_EVAL_ON_SESSION: データベース・セッションの作成時 ● DBMS_MACUTL.G_EVAL_ON_ACCESS: ファクタがアクセスされる際に毎回 ● DBMS_MACUTL.G_EVAL_ON_STARTUP: 起動時 eval_options のデフォルトは設定済の値であり、DBA_DV_FACTOR データ・ディクショナリ・ビューに問い合せることで確認できます。 詳細は、「ファクタの評価情報の設定」を参照してください。
audit_options	カスタムの Oracle Database Vault 監査レコードから生成する場合にファクタを監査するオプション。 ● DBMS_MACUTL.G_AUDIT_OFF: 監査を無効にします。 ● DBMS_MACUTL.G_AUDIT_ALWAYS: 常に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_GET_ERROR: get_expr がエラーを戻した場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_GET_NULL: get_expr が null の場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_VALIDATE_ERROR: 検証プロシージャがエラーを戻した場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_VALIDATE_FALSE: 検証プロシージャが false の場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_TRUST_LEVEL_NULL: 信頼レベルが設定されていない場合に監査します。 ● DBMS_MACUTL.G_AUDIT_ON_TRUST_LEVEL_NEG: 信頼レベルが負の場合に監査します。 audit_options のデフォルトは設定済の値であり、DBA_DV_FACTOR データ・ディクショナリ・ビューに問い合せることで確認できます。

パラメータ	説明
	audit_options パラメータは、従来の監査にのみ適用されます。統合監査を有効にした場合は、audit_options を使用するかわりに統合監査ポリシーを作成します。 詳細は、「ファクタの監査オプションの設定」を参照してください。
fail_options	ファクタ・エラーをレポートするオプション。 ● DBMS_MACUTL.G_FAIL_WITH_MESSAGE: エラー・メッセージを表示します。 ● DBMS_MACUTL.G_FAIL_SILENTLY: エラー・メッセージを表示しません。 fail_options のデフォルトは設定済の値であり、DBA_DV_FACTOR データ・ディクショナリ・ビューに問い合せることで確認できます。 詳細は、「ファクタのエラー・オプションの設定」を参照してください。

例

```
BEGIN
 DBMS_MACADM.UPDATE_FACTOR(
factor_name       => 'Sector2_DB',
factor_type_name  => 'Instance',
description       => ' ',
rule_set_name     => 'Limit_DBA_Access',
get_expr          => 'UPPER(SYS_CONTEXT(''USERENV'',''DB_NAME''))',
validate_expr     => 'dbavowner.check_db_access',
identify_by       => DBMS_MACUTL.G_IDENTIFY_BY_METHOD,
labeled_by        => DBMS_MACUTL.G_LABELED_BY_SELF,
eval_options      => DBMS_MACUTL.G_EVAL_ON_ACCESS,
audit_options     => DBMS_MACUTL.G_AUDIT_ALWAYS,
fail_options      => DBMS_MACUTL.G_FAIL_WITH_MESSAGE);
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.21 UPDATE_FACTOR_TYPEプロシージャ

UPDATE_FACTOR_TYPEプロシージャはファクタ・タイプを更新します。

構文

```
DBMS_MACADM.UPDATE_FACTOR_TYPE(
name          IN VARCHAR2,
description   IN VARCHAR2);
```

パラメータ

表17-21 UPDATE_FACTOR_TYPEのパラメータ

パラメータ	説明

パラメータ	説明
name	ファクタ・タイプ名。 現行のデータベース・インスタンスで既存のファクタ・タイプを確認するには、「DBA_DV_FACTOR_TYPE ビュー」で説明されている DBA_DV_FACTOR_TYPE ビューに問い合せます。
description	ファクタ・タイプの目的の説明(大/小文字混在で最大 1024 文字)。

例

```
BEGIN
 DBMS_MACADM.UPDATE_FACTOR_TYPE(
name        => 'Sector2DBInstance',
description => 'Checks DB instances used in Sector 2');
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.1.22 UPDATE_IDENTITYプロシージャ

UPDATE_IDENTITYプロシージャは、ファクタ・アイデンティティの信頼レベルを更新します。

構文

```
DBMS_MACADM.UPDATE_IDENTITY(
factor_name  IN VARCHAR2,
value        IN VARCHAR2,
trust_level  IN NUMBER);
```

パラメータ

表17-22 UPDATE_IDENTITYのパラメータ

パラメータ	説明
factor_name	ファクタ名。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。 アイデンティティがあるファクタを確認するには、「DBA_DV_IDENTITY ビュー」で説明されている DBA_DV_IDENTITY に問い合せます。
value	新しいファクタ・アイデンティティ(大/小文字混在で最大 1024 文字)。たとえば、IP_Address ファクタのアイデンティティは、192.0.2.12 という IP アドレスになります。
trust_level	同じファクタの別のアイデンティティと比較した信頼の度合いを示す数値。一般に、信頼レベルの数値が高く設定されているほど信頼の度合いも高くなります。信頼レベル 10 は、非常に信頼度が高いこ

パラメータ	説明
	とを表します。信頼レベルの数値が負の場合は信頼できません。 信頼レベルおよびラベル・セキュリティの詳細は、「ファクタ・アイデンティティの作成および構成」を参照してください。

例

```
BEGIN
 DBMS_MACADM.UPDATE_IDENTITY(
factor_name  => 'Sector2_ClientID',
value        => 'intranet',
trust_level  => 10);
END;
/
```

親トピック: DBMS_MACADMファクタのプロシージャおよびファンクション

17.2 Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクション

Oracle Database Vaultには手続き型インタフェースが用意されており、Database Vaultセキュリティ・オプションの管理およびDatabase Vaultセキュリティの実施の管理に使用できます。

- Oracle Database VaultランタイムPL/SQLプロシージャおよびファンクション
  Oracle Database Vaultには、ファクタ固有の一連のPL/SQLプロシージャおよびファンクションが用意されています。
- SET_FACTORプロシージャ
  SET_FACTORプロシージャは、ファクタ・アイデンティティを動的に設定する必要があるアプリケーションに公開できます。
- GET_FACTORファンクション
  GET_FACTORファンクションは、パブリック・ファクタ・ファンクションによるファクタのアイデンティティの解決を可能にするために、DVFスキーマに公開されます。戻り型はVARCHAR2です。
- GET_FACTOR_LABELファンクション
  GET_FACTOR_LABELファンクションは、指定されたファクタに指定されたOracle Label Securityポリシー用のラベルが割り当てられている場合、そのラベルを返します。戻り型はVARCHAR2です。
- GET_TRUST_LEVELファンクション
  GET_TRUST_LEVELファンクションは、リクエストされたファクタの現行セッションのアイデンティティの信頼レベルを返します。戻り型はVARCHAR2です。
- GET_TRUST_LEVEL_FOR_IDENTITYファンクション
  GET_TRUST_LEVEL_FOR_IDENTITYファンクションは、リクエストされたファクタおよびアイデンティティの信頼レベルを返します。戻り型はVARCHAR2です。
- ROLE_IS_ENABLEDファンクション
  ROLE_IS_ENABLEDファンクションは、データベース・ロールが有効であるかどうかを示すブール値を返します。戻り型はBOOLEANです。

親トピック: Oracle Database VaultファクタのAPI

17.2.1 Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクションについて

Oracle Database Vaultには、ファクタ固有の一連のPL/SQLプロシージャおよびファンクションが用意されています。

これは、レルム違反およびコマンド認可のDDLコマンドを検証するためのロジックを公開するプロシージャおよびファンクションです。その他に、ファクタの値をたとえばWebアプリケーションなどから設定する(関連付けられたルール・セットがTrueに評価される場合)、セッションまたは特定のファクタ・アイデンティティに対する信頼レベルを取得する、あるいはファクタ・アイデンティティのラベルを取得するためのプロシージャおよびファンクションが用意されています。これらのプロシージャおよびファンクションは、データベース管理者がすべてのDVSYSパッケージ・プロシージャに対するEXECUTE権限を一般のデータベース・アカウント群に付与しないように提供されています。プロシージャおよびファンクションは、必要な最小限のメソッドのみを公開します。これらのファンクションおよびプロシージャはすべて、必要とするアプリケーションでパブリックに使用できます。

親トピック: Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクション

17.2.2 SET_FACTORプロシージャ

SET_FACTORプロシージャは、ファクタ・アイデンティティを動的に設定する必要があるアプリケーションに公開できます。

パッケージ・プロシージャDBMS_MACADM.SET_FACTORをラップします。割当て用のルール・セットがファクタに関連付けられていて、ルール・セットがtrueを返した場合に、値が設定されます。通常のルール・セット処理が行われ、ファクタ値(アイデンティティ)の検証メソッドがコールされます。このプロシージャは、一般のデータベース・アカウント群で使用(実行)できます。

構文

```
SET_FACTOR(
p_factor IN VARCHAR2,
p_value  IN VARCHAR2);
```

パラメータ

表17-23 SET_FACTORのパラメータ

パラメータ	説明
p_factor	ファクタ名。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR データ・ディクショナリ・ビューに問い合せます。
p_value	アイデンティティ値。大/小文字混在で最大で 1024 文字まで入力できます。 現行のデータベース・インスタンス内の各ファクタのアイデンティティを確認するには、「DBA_DV_IDENTITY ビュー」で説明されている DBA_DV_IDENTITY データ・ディクショナリ・ビューに問い合せます。

例

```
EXECUTE SET_FACTOR(''Sector2_ClientID'', ''identity'');
```

親トピック: Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクション

17.2.3 GET_FACTORファンクション

GET_FACTORファンクションは、パブリック・ファクタ・ファンクションによるファクタのアイデンティティの解決を可能にするために、DVFスキーマに公開されます。戻り型はVARCHAR2です。

このファンクションにより、DVFスキーマのF$ファンクションが有効になります。このファンクションは、一般のデータベース・アカウント群で使用(実行)できます。

構文

```
GET_FACTOR(
p_factor IN VARCHAR2)
RETURN VARCHAR2;
```

パラメータ

表17-24 GET_FACTORのパラメータ

パラメータ	説明
p_factor	ファクタ名。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR データ・ディクショナリ・ビューに問い合せます。

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Get Client ID Factor Identity',
  rule_expr => 'GET_FACTOR(''Sector2_ClientID'')');
END;
/
```

親トピック: Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクション

17.2.4 GET_FACTOR_LABELファンクション

GET_FACTOR_LABELファンクションは、指定されたファクタに指定されたOracle Label Securityポリシー用のラベルが割り当てられている場合、そのラベルを返します。戻り型はVARCHAR2です。

ポリシーがOracle Label Securityを使用して構成されている場合は、ポリシーの最大セッション・ラベルとマージされるラベルを返します。このファンクションは、一般のデータベース・アカウント群で使用(実行)できます。

構文

```
GET_FACTOR_LABEL(
p_factor      IN VARCHAR2,
p_policy_name IN VARCHAR2)
RETURN VARCHAR2;
```

パラメータ

表17-25 GET_FACTOR_LABELのパラメータ

パラメータ	説明

パラメータ	説明
p_factor	ファクタ名。 現行のデータベース・インスタンス内の使用可能なファクタを確認するには、DBA_DV_FACTOR データ・ディクショナリ・ビューに問い合せます。Oracle Label Security ポリシーに関連付けられているファクタを確認するには、DBA_DV_MAC_POLICY_FACTOR を使用します。 「DBA_DV_FACTOR ビュー」および「DBA_DV_MAC_POLICY_FACTOR ビュー」を参照してください。
p_policy_name	Oracle Label Security ポリシー名。 現行のデータベース・インスタンスのポリシーおよびファクタに関する情報を検索する場合は、次のデータ・ディクショナリ・ビューを使用します。 ● DBA_DV_MAC_POLICY: 現行のデータベース・インスタンスに定義されている Oracle Label Security ポリシーが表示されます。「DBA_DV_MAC_POLICY ビュー」を参照してください。 ● DBA_DV_MAC_POLICY_FACTOR: 現行のデータベース・インスタンスの Oracle Label Security ポリシーに関連付けられているファクタが表示されます。「DBA_DV_MAC_POLICY_FACTOR ビュー」を参照してください。 ● DBA_DV_POLICY_LABEL: 各ポリシーの DBA_DV_IDENTITY ビューの各ファクタ識別子に対する Oracle Label Security ラベルが表示されます。「DBA_DV_POLICY_LABEL ビュー」を参照してください。

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Get the ClientID Factor Label',
  rule_expr => 'GET_FACTOR_LABEL(''Sector2_ClientID'', ''Access Locations'')');
END;
/
```

親トピック: Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクション

17.2.5 GET_TRUST_LEVELファンクション

GET_TRUST_LEVELファンクションは、リクエストされたファクタの現行セッションのアイデンティティの信頼レベルを返します。戻り型はVARCHAR2です。

このファンクションは、一般のデータベース・アカウント群で使用(実行)できます。使用可能な信頼レベルのリストは、「ファクタ・アイデンティティの作成および構成」を参照してください。

構文

```
GET_TRUST_LEVEL(
p_factor IN VARCHAR2)
```

```
RETURN VARCHAR2;
```

パラメータ

表17-26 GET_TRUST_LEVELのパラメータ

パラメータ	説明
p_factor	ファクタ名。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR データ・ディクショナリ・ビューに問い合せます。

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Get Client ID Trust Level',
  rule_expr => 'GET_TRUST_LEVEL(''Sector2_ClientID'')');
END;
/
```

親トピック: Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクション

17.2.6 GET_TRUST_LEVEL_FOR_IDENTITYファンクション

GET_TRUST_LEVEL_FOR_IDENTITYファンクションは、リクエストされたファクタおよびアイデンティティの信頼レベルを返します。戻り型はVARCHAR2です。

このファンクションは、一般のデータベース・アカウント群で使用(実行)できます。使用可能な信頼レベルのリストは、「ファクタ・アイデンティティの作成および構成」を参照してください。

構文

```
GET_TRUST_LEVEL_FOR_IDENTITY(
p_factor   IN VARCHAR2,
p_identity IN VARCHAR2)
RETURN VARCHAR2;
```

パラメータ

表17-27 GET_TRUST_LEVEL_FOR_IDENTITYのパラメータ

パラメータ	説明
p_factor	ファクタ名。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。
p_identity	アイデンティティ値。 現行のデータベース・インスタンス内の各ファクタのアイデンティティを確認するには、「DBA_DV_IDENTITY ビュー」で説明されている DBA_DV_IDENTITY データ・ディクショナリ・

パラメータ	説明
	ビューを使用します。

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Get Client ID Identity Trust Level',
  rule_expr => 'GET_TRUST_LEVEL_FOR_IDENTITY(''Sector2_ClientID'', ''identity'')');
END;
/
```

親トピック: Oracle Database VaultランタイムのPL/SQLプロシージャおよびファンクション

17.2.7 ROLE_IS_ENABLEDファンクション

ROLE_IS_ENABLEDファンクションは、データベース・ロールが有効であるかどうかを示すブール値を返します。戻り型はBOOLEANです。

このファンクションは、一般のデータベース・アカウント群で使用(実行)できます。

構文

```
ROLE_IS_ENABLED(
p_role IN VARCHAR2)
RETURN BOOLEAN;
```

パラメータ

表17-28 ROLE_IS_ENABLEDのパラメータ

パラメータ	説明
p_role	確認するデータベース・ロール名。 既存のロールを検索する場合は、次のデータ・ディクショナリ・ビューを使用します。 ● DBA_ROLES: 現行のデータベース・インスタンスで使用可能なロールを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のロールの認可を確認します。「DBA_DV_REALM ビュー」を参照してください。 ● DBA_DV_ROLE: 権限管理で使用されている既存のセキュア・アプリケーション・ロールを確認します。「DBA_DV_ROLE ビュー」を参照してください。

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check if SYSADM Role Is Enabled',
  rule_expr => 'ROLE_IS_ENABLED(''SYSADM'')');
END;
/
```

17.3 Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

DBMS_MACADM PL/SQLパッケージを使用して様々なファクタを管理する際、Oracle Database VaultによってDVFスキーマ・ファンクションが維持されます。

- Oracle Database Vault DVF PL/SQLファクタ・ファンクション
  Oracle Database Vaultには、頻繁に使用されるアクティビティ用にDVFファクタ固有のファンクションが用意されています。
- F$AUTHENTICATION_METHODファンクション
  F$AUTHENTICATION_METHODファンクションは認証の方式をVARCHAR2データ型で返します。
- F$CLIENT_IPファンクション
  F$CLIENT_IPファンクションは、クライアントが接続されているコンピュータのIPアドレスをVARCHAR2データ型で返します。
- F$DATABASE_DOMAINファンクション
  F$DATABASE_DOMAINファンクションは、DB_DOMAIN初期化パラメータに指定されているデータベースのドメインをVARCHAR2データ型で返します。
- F$DATABASE_HOSTNAMEファンクション
  F$DATABASE_HOSTNAMEファンクションは、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をVARCHAR2データ型で返します。
- F$DATABASE_INSTANCEファンクション
  F$DATABASE_INSTANCEファンクションは、現在のデータベース・インスタンスのインスタンス識別番号をVARCHAR2データ型で返します。
- F$DATABASE_IPファンクション
  F$DATABASE_IPファンクションは、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IPアドレスをVARCHAR2データ型で返します。
- F$DATABASE_NAMEファンクション
  F$DATABASE_NAMEファンクションは、DB_NAME初期化パラメータに指定されているデータベースの名前をVARCHAR2データ型で返します。
- F$DOMAINファンクション
  F$DOMAINファンクションは、ランタイム環境(ネットワークIT環境やそのサブセットなど)内の、特定の機密レベルで動作する物理的な構成または実装固有のファクタの名前付きコレクションを返します。戻り型はVARCHAR2です。
- FDV_CLIENT_IDENTIFIERファンクション
  FDV_CLIENT_IDENTIFIERファンクションは、Oracle Database Vaultクライアント識別子を返します。
- FDV_DBLINK_INFOファンクション
  FDV_DBLINK_INFOファンクションは、Oracle Database Vaultデータベース・リンクに関する情報を返します。
- FDV_MODULEファンクション
  FDV_MODULEファンクションは、Oracle Database Vaultモジュールに関する情報を返します。
- F$ENTERPRISE_IDENTITYファンクション
  F$ENTERPRISE_IDENTITYファンクションは、ユーザーのエンタープライズ全体のアイデンティティをVARCHAR2データ型で返します。
- F$IDENTIFICATION_TYPEファンクション
  F$IDENTIFICATION_TYPEファンクションは、データベースでのユーザーのスキーマの作成方法を返します。具体的

には、CREATE/ALTER USER構文のIDENTIFIED句が反映されます。戻り型はVARCHAR2です。

- F$LANGファンクション
  F$LANGファンクションは、既存のLANGUAGEパラメータより短い形式の言語名のISO略称を、ユーザーのセッションに対して返します。戻り型はVARCHAR2です。
- F$LANGUAGEファンクション
  F$LANGUAGEファンクションは、ユーザーのセッションで現在使用中の言語と地域、およびデータベース文字セットを返します。戻り型はVARCHAR2です。
- F$MACHINEファンクション
  F$MACHINEファンクションは、データベース・セッションを確立したデータベース・クライアントのコンピュータ(ホスト)名を返します。戻り型はVARCHAR2です。
- F$NETWORK_PROTOCOLファンクション
  F$NETWORK_PROTOCOLファンクションは、接続文字列のPROTOCOL=protocol部分に指定されている、通信に使用されるネットワーク・プロトコルを返します。戻り型はVARCHAR2です。
- F$PROXY_ENTERPRISE_IDENTITYファンクション
  F$PROXY_ENTERPRISE_IDENTITYファンクションは、プロキシ・ユーザーがエンタープライズ・ユーザーである場合、Oracle Internet Directoryの識別名(DN)を返します。戻り型はVARCHAR2です。
- F$PROXY_USERファンクション
  F$PROXY_USERファンクションは、プロキシ・ユーザーの名前を返します。
- F$SESSION_USERファンクション
  F$SESSION_USERファンクションは、現行ユーザーが認証されたデータベース・ユーザー名を返します。この値は、セッションを通して同じです。戻り型はVARCHAR2です。

親トピック: Oracle Database VaultファクタのAPI

17.3.1 Oracle Database Vault DVF PL/SQLファクタ・ファンクションについて

Oracle Database Vaultには、頻繁に使用されるアクティビティ用にDVFファクタ固有のファンクションが用意されています。

DVSYSスキーマから提供されるファンクションおよびプロシージャに加え、DVFスキーマにはシステムに定義されているファクタごとに1つのファンクションが含まれます。

その後、ファンクションは一般のデータベース・アカウント群でPL/SQLファンクションおよび標準SQLを介して使用できます。これにより、ファクタはOracle Label Security、Oracle Virtual Private Database(VPD)などで使用できるようになります。

通常、これらのファンクションをルール式に組み込むことができます。たとえば:

その後、ファンクションは一般のデータベース・アカウント群でPL/SQLファンクションおよび標準SQLを介して使用できます。これにより、ファクタはOracle Label Security、Oracle Virtual Private Database(VPD)などで使用できるようになります。

通常、これらのファンクションをルール式に組み込むことができます。たとえば:

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
 rule_name => 'Not Internal DBA',
 rule_expr => 'DVF.F$SESSION_USER NOT IN (''JSMTIH'', ''TBROWN'')');
END;
/
```

ファクタ・ファンクションの値を検索するには、DUALシステム表から選択します。たとえば:

```
SELECT DVF.F$SESSION_USER FROM DUAL;
F$SESSION_USER
------------------------------------------------
```

```
SEC_ADMIN_OWEN
```

ファクタ自体の名前は、大/小文字を区別しません。たとえば、次の文は同じ結果を返します。

```
select dvf.f$session_user from dual;
SELECT DVF.F$SESSION_USER FROM DUAL;
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.2 F$AUTHENTICATION_METHODファンクション

F$AUTHENTICATION_METHODファンクションは認証の方式をVARCHAR2データ型で返します。

次に、ユーザー・タイプの後に返される方式を続けて示します。

- パスワードで認証されるエンタープライズ・ユーザー、ローカル・データベース・ユーザー、またはパスワード・ファイルを使用するSYSDBA/SYSOPER(パスワードを使用するユーザー名によるプロキシ): PASSWORD
- Kerberosで認証されるエンタープライズ・ユーザーまたは外部ユーザー: KERBEROS
- Transport Layer Security (TLS)で認証されるエンタープライズ・ユーザーまたは外部ユーザー: SSL
- RADIUSで認証される外部ユーザー: RADIUS
- オペレーティング・システムで認証される外部ユーザーまたはSYSDBA/SYSOPER: OS
- DCEで認証される外部ユーザー: DCE
- 証明書、識別名(DN)またはパスワードを使用しないユーザー名によるプロキシ: NONE

IDENTIFICATION_TYPEを使用すると、認証方式がパスワード、KerberosまたはTLSの場合に、外部ユーザーとエンタープライズ・ユーザーを区別できます。

構文

```
DVF.F$AUTHENTICATION_METHOD ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check TLS Authentication Method',
  rule_expr => 'DVF.F$AUTHENTICATION_METHOD = ''SSL''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.3 F$CLIENT_IPファンクション

F$CLIENT_IPファンクションは、クライアントが接続されているコンピュータのIPアドレスをVARCHAR2データ型で返します。

構文

```
DVF.F$CLIENT_IP ()
RETURN VARCHAR2;
```

パラメータ

なし

例

次の例では、ルール作成文でDVF.F$CLIENT_IPを使用する方法を示します。入力できるのは、IPアドレスの範囲ではなく、1つのIPアドレスのみであることに注意してください。

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Client IP Address',
  rule_expr => 'DVF.F$CLIENT_IP = ''192.0.2.10''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.4 F$DATABASE_DOMAINファンクション

F$DATABASE_DOMAINファンクションは、DB_DOMAIN初期化パラメータに指定されているデータベースのドメインをVARCHAR2データ型で返します。

構文

```
DVF.F$DATABASE_DOMAIN ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Client Database Domain',
  rule_expr => 'DVF.F$DATABASE_DOMAIN NOT IN (''EXAMPLE'', ''YOURDOMAIN'')');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.5 F$DATABASE_HOSTNAMEファンクション

F$DATABASE_HOSTNAMEファンクションは、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をVARCHAR2データ型で返します。

構文

```
DVF.F$DATABASE_HOSTNAME ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
```

```
  rule_name => 'Check Host Name',
  rule_expr => 'DVF.F$DATABASE_HOSTNAME IN (''SHOBEEN'', ''MAU'')');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.6 F$DATABASE_INSTANCEファンクション

F$DATABASE_INSTANCEファンクションは、現在のデータベース・インスタンスのインスタンス識別番号をVARCHAR2データ型で返します。

構文

```
DVF.F$DATABASE_INSTANCE ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Database Instance ID',
  rule_expr => 'DVF.F$DATABASE_INSTANCE = ''SALES_DB''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.7 F$DATABASE_IPファンクション

F$DATABASE_IPファンクションは、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IPアドレスをVARCHAR2データ型で返します。

構文

```
DVF.F$DATABASE_IP ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Database IP address',
  rule_expr => 'DVF.F$DATABASE_IP = ''192.0.2.5''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.8 F$DATABASE_NAMEファンクション

F$DATABASE_NAMEファンクションは、DB_NAME初期化パラメータに指定されているデータベースの名前をVARCHAR2データ

型で返します。

構文

```
DVF.F$DATABASE_NAME ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Database DB_NAME Name',
  rule_expr => 'DVF.F$DATABASE_NAME = ''ORCL''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.9 F$DOMAINファンクション

F$DOMAINファンクションは、ランタイム環境(ネットワークIT環境やそのサブセットなど)内の、特定の機密レベルで動作する物理的な構成または実装固有のファクタの名前付きコレクションを返します。戻り型はVARCHAR2です。

データベースへのセキュア・アクセス・パス内にあるOracle Database Vaultノードのホスト名、IPアドレスおよびデータベース・インスタンス名などのファクタを使用してドメインを識別できます。ドメインを識別するファクタ識別子の組合せを使用して、各ドメインを一意に特定できます。これらの識別ファクタやその他のファクタを使用して、ドメイン内に最大セキュリティ・ラベルを定義できます。これにより、Oracle Database Vaultセッションに関する物理ファクタに応じて、データ・アクセスおよびコマンドが制限されます。必要なドメインの例として、企業機密、内部パブリック、パートナ、顧客があります。

構文

```
DVF.F$DOMAIN ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Domain',
  rule_expr => 'DVF.F$DOMAIN = ''EXAMPLE.COM''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.10 FDV_CLIENT_IDENTIFIERファンクション

FDV_CLIENT_IDENTIFIERファンクションは、Oracle Database Vaultクライアント識別子を返します。

構文

```
DVF.F$DV$_CLIENT_IDENTIFIER ()
```

```
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
  DBMS_MACADM.CREATE_RULE(
   rule_name => 'Check Database Vault Client Identifiers',
   rule_expr => 'DVF.F$DV$_CLIENT_IDENTIFIER = ''14903BUA765454'';
END;/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.11 FDV_DBLINK_INFOファンクション

FDV_DBLINK_INFOファンクションは、Oracle Database Vaultデータベース・リンクに関する情報を返します。

構文

```
DVF.F$DV$_DBLINK_INFO ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
  DBMS_MACADM.CREATE_RULE(
   rule_name => 'Check Database Vault database link info',
   rule_expr => 'DVF.F$DV$_DBLINK_INFO = ''SOURCE_GLOBAL_NAME=SALES.US.EXAMPLE.COM,
DBLINK_NAME=PDB2_LINK, SOURCE_AUDIT_SESSIONID=200057'';
END;/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.12 FDV_MODULEファンクション

FDV_MODULEファンクションは、Oracle Database Vaultモジュールに関する情報を返します。

構文

```
DVF.F$DV$_MODULE ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
  DBMS_MACADM.CREATE_RULE(
   rule_name => 'Check Database Vault modules',
   rule_expr => 'DVF.F$DV$_MODULE = ''SQL*Plus'';
END;/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.13 F$ENTERPRISE_IDENTITYファンクション

F$ENTERPRISE_IDENTITYファンクションは、ユーザーのエンタープライズ全体のアイデンティティをVARCHAR2データ型で返します。

- エンタープライズ・ユーザーの場合: Oracle Internet Directory DN。
- 外部ユーザーの場合: 外部アイデンティティ(Kerberosプリンシパル名、RADIUSおよびDCEスキーマ名、オペレーティング・システム・ユーザー名、証明書DN)。
- ローカル・ユーザーおよびSYSDBA/SYSOPERログインの場合: NULL

属性の値はプロキシ方式によって異なります。

- DNによるプロキシの場合: クライアントのOracle Internet Directory DN。
- 証明書によるプロキシの場合: 外部ユーザーではクライアントの証明書DN、グローバル・ユーザーではOracle Internet Directory DN。
- ユーザー名によるプロキシの場合: エンタープライズ・ユーザーであるクライアントではOracle Internet Directory DN、ローカル・データベース・ユーザーであるクライアントではNULL。

構文

```
DVF.F$ENTERPRISE_IDENTITY ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check User Enterprise Identity',
  rule_expr => 'DVF.F$ENTERPRISE_IDENTITY NOT IN (''JSMITH'', ''TSMITH'')');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.14 F$IDENTIFICATION_TYPEファンクション

F$IDENTIFICATION_TYPEファンクションは、データベースでのユーザーのスキーマの作成方法を返します。具体的には、CREATE/ALTER USER構文のIDENTIFIED句が反映されます。戻り型はVARCHAR2です。

次に、スキーマ作成時に使用される構文の後に返される識別タイプを続けて示します。

- IDENTIFIED BY password: LOCAL
- IDENTIFIED EXTERNALLY: EXTERNAL
- IDENTIFIED GLOBALLY: GLOBAL SHARED
- IDENTIFIED GLOBALLY AS DN: GLOBAL PRIVATE

構文

```
DVF.F$IDENTIFICATION_TYPE ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check User Schema Creation Type',
  rule_expr => 'DVF.F$IDENTIFICATION_TYPE = ''GLOBAL SHARED''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.15 F$LANGファンクション

F$LANGファンクションは、既存のLANGUAGEパラメータより短い形式の言語名のISO略称を、ユーザーのセッションに対して返します。戻り型はVARCHAR2です。

Oracle Databaseでサポートされている言語の一覧については、『Oracle Databaseグローバリゼーション・サポート・ガイド』を参照してください。

構文

```
DVF.F$LANG ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check ISO Abbreviated Language Name',
  rule_expr => 'DVF.F$LANG IN (''EN'', ''DE'', ''FR'')');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.16 F$LANGUAGEファンクション

F$LANGUAGEファンクションは、ユーザーのセッションで現在使用中の言語と地域、およびデータベース文字セットを返します。戻り型はVARCHAR2です。

戻り型は次の形式です。

`language_territory.characterset`

Oracle Databaseでサポートされている言語と地域の一覧については、『Oracle Databaseグローバリゼーション・サポート・ガイド』を参照してください。

構文

```
DVF.F$LANGUAGE ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Session Language and Territory',
  rule_expr => 'DVF.F$LANGUAGE = ''AMERICAN_AMERICA.WE8ISO8859P1''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.17 F$MACHINEファンクション

F$MACHINEファンクションは、データベース・セッションを確立したデータベース・クライアントのコンピュータ(ホスト)名を返します。戻り型はVARCHAR2です。

構文

```
DVF.F$MACHINE ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Client Computer Host Name',
  rule_expr => 'DVF.F$MACHINE NOT IN (''SHOBEEN'', ''SEBASTIAN'')');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.18 F$NETWORK_PROTOCOLファンクション

F$NETWORK_PROTOCOLファンクションは、接続文字列のPROTOCOL=protocol部分に指定されている、通信に使用されるネットワーク・プロトコルを返します。戻り型はVARCHAR2です。

構文

```
DVF.F$NETWORK_PROTOCOL ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Network Protocol',
  rule_expr => 'DVF.F$NETWORK_PROTOCOL = ''TCP''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.19 F$PROXY_ENTERPRISE_IDENTITYファンクション

F$PROXY_ENTERPRISE_IDENTITYファンクションは、プロキシ・ユーザーがエンタープライズ・ユーザーである場合、Oracle Internet Directoryの識別名(DN)を返します。戻り型はVARCHAR2です。

構文

```
DVF.F$PROXY_ENTERPRISE_IDENTITY ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Get OID DN of Enterprise User',
  rule_expr => 'DVF.F$PROXY_ENTERPRISE_IDENTITY = ''cn=Provisioning Admins''');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.20 F$PROXY_USERファンクション

F$PROXY_USERファンクションは、プロキシ・ユーザーの名前を返します。

構文

```
DVF.PROXY_USER ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
  DBMS_MACADM.CREATE_RULE(
   rule_name => 'Check Proxy Users',
   rule_expr => 'DVF.PROXY_USER NOT IN (''ECHICHESTER'', ''PFITCH'')');
END;/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

17.3.21 F$SESSION_USERファンクション

F$SESSION_USERファンクションは、現行ユーザーが認証されたデータベース・ユーザー名を返します。この値は、セッションを通して同じです。戻り型はVARCHAR2です。

構文

```
DVF.F$SESSION_USER ()
RETURN VARCHAR2;
```

パラメータ

なし

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Check Database User Name',
  rule_expr => 'DVF.F$SESSION_USER IN (''JSMITH'', ''TSMITH'')');
END;
/
```

親トピック: Oracle Database VaultのDVF PL/SQLファクタ・ファンクション

18 Oracle Database Vaultセキュア・アプリケーション・ロールのAPI

DBMS_MACADMおよびDBMS_MACSEC_ROLES PL/SQLパッケージは、Database Vaultセキュア・アプリケーション・ロールを管理します。

- DBMS_MACADMセキュア・アプリケーション・ロールのプロシージャ
  DBMS_MACADMパッケージでは、Oracle Database Vaultのセキュア・アプリケーション・ロールの作成、名前変更、割当て、割当て解除、更新および削除が可能です。
- DBMS_MACSEC_ROLESセキュア・アプリケーション・ロールのプロシージャおよびファンクション
  DBMS_MACSEC_ROLESパッケージは、ユーザーに対する認可をチェックし、Oracle Database Vaultセキュア・アプリケーション・ロールを設定します。

関連トピック

- Oracle Database Vaultのセキュア・アプリケーション・ロールの構成
- Oracle Database VaultユーティリティのAPI

18.1 DBMS_MACADMセキュア・アプリケーション・ロールのプロシージャ

DBMS_MACADMパッケージでは、Oracle Database Vaultのセキュア・アプリケーション・ロールの作成、名前変更、割当て、割当て解除、更新および削除が可能です。

- CREATE_ROLEプロシージャ
  CREATE_ROLEプロシージャは、Oracle Database Vaultセキュア・アプリケーション・ロールを作成します。
- DELETE_ROLEプロシージャ
  DELETE_ROLEプロシージャは、Oracle Database Vaultセキュア・アプリケーション・ロールを削除します。
- RENAME_ROLEプロシージャ
  RENAME_ROLEプロシージャは、Oracle Database Vaultセキュア・アプリケーション・ロールの名前を変更します。名前の変更は、そのロールが使用されているすべての箇所に反映されます。
- UPDATE_ROLEプロシージャ
  UPDATE_ROLEプロシージャは、Oracle Database Vaultセキュア・アプリケーション・ロールを更新します。

親トピック: Oracle Database Vaultセキュア・ アプリケーション・ ロール のAPI

18.1.1 CREATE_ROLEプロシージャ

CREATE_ROLEプロシージャは、Oracle Database Vaultセキュア・アプリケーション・ロールを作成します。

構文

```
DBMS_MACADM.CREATE_ROLE(
role_name      IN VARCHAR2,
enabled        IN VARCHAR2,
rule_set_name  IN VARCHAR2);
```

パラメータ

表18-1 CREATE_ROLEのパラメータ

パラメータ	説明
role_name	ロール名(空白を使用せず、最大 128 文字)。マルチテナント環境では、ロール名の先頭に c## または C##を付加します。 現行のデータベース・インスタンスで既存のセキュア・アプリケーション・ロールを確認するには、「DBA_DV_ROLE ビュー」で説明されている DBA_DV_ROLE ビューに問い合せます。
enabled	DBMS_MACUTL.G_YES(Yes)を選択するとロールが有効化できるようになり、DBMS_MACUTL.G_NO(No)を選択するとロールを有効化できなくなります。デフォルト値は DBMS_MACUTL.G_YES です。
rule_set_name	このセキュア・アプリケーションを有効にできるかどうかを判断するためのルール・セットの名前。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。

例

```
BEGIN
 DBMS_MACADM.CREATE_ROLE(
role_name      => 'Sector2_APP_MGR',
enabled        => DBMS_MACUTL.G_YES,
rule_set_name  => 'Check App2 Access');
END;
/
```

親トピック: DBMS_MACADMセキュア・アプリケーション・ロールのプロシージャ

18.1.2 DELETE_ROLEプロシージャ

DELETE_ROLEプロシージャは、Oracle Database Vaultセキュア・アプリケーション・ロールを削除します。

構文

```
DBMS_MACADM.DELETE_ROLE(
role_name IN VARCHAR2);
```

パラメータ

表18-2 DELETE_ROLEのパラメータ

パラメータ	説明
role_name	ロール名。 現行のデータベース・インスタンスで既存のセキュア・アプリケーション・ロールを確認するには、「DBA_DV_ROLE ビュー」で説明されている DBA_DV_ROLE ビューに問い合せます。

例

```
EXEC DBMS_MACADM.DELETE_ROLE('SECT2_APP_MGR');
```

親トピック: DBMS_MACADMセキュア・アプリケーション・ロールのプロシージャ

18.1.3 RENAME_ROLEプロシージャ

RENAME_ROLEプロシージャは、Oracle Database Vaultセキュア・アプリケーション・ロールの名前を変更します。名前の変更は、そのロールが使用されているすべての箇所に反映されます。

構文

```
DBMS_MACADM.RENAME_ROLE(
role_name       IN VARCHAR2,
new_role_name   IN VARCHAR2);
```

パラメータ

表18-3 RENAME_ROLEのパラメータ

パラメータ	説明
role_name	現在のロール名。 現行のデータベース・インスタンスで既存のセキュア・アプリケーション・ロールを確認するには、「DBA_DV_ROLE ビュー」で説明されている DBA_DV_ROLE ビューに問い合せます。
new_role_name	ロール名(空白を使用せず、最大 128 文字)。この名前が、『Oracle Database SQL 言語リファレンス』で説明されているロール作成の標準の Oracle ネーミング規則に準拠していることを確認してください。マルチテナント環境では、ロール名の先頭に c##または C##を付加します。

例

```
BEGIN
 DBMS_MACADM.RENAME_ROLE(
role_name       => 'SECT2_APP_MGR',
new_role_name   => 'SECT2_SYSADMIN');
END;
/
```

親トピック: DBMS_MACADMセキュア・アプリケーション・ロールのプロシージャ

18.1.4 UPDATE_ROLEプロシージャ

UPDATE_ROLEプロシージャは、Oracle Database Vaultセキュア・アプリケーション・ロールを更新します。

構文

```
DBMS_MACADM.UPDATE_ROLE(
role_name       IN VARCHAR2,
enabled         IN VARCHAR2,
rule_set_name   IN VARCHAR2);
```

パラメータ

表18-4 UPDATE_ROLEのパラメータ

パラメータ	説明
role_name	ロール名。 現行のデータベース・インスタンスで既存のセキュア・アプリケーション・ロールを確認するには、「DBA_DV_ROLE ビュー」で説明されている DBA_DV_ROLE ビューに問い合せます。
enabled	DBMS_MACUTL.G_YES(Yes)を選択するとロールが有効化できるようになり、DBMS_MACUTL.G_NO(No)を選択するとロールを有効化できなくなります。 enabled のデフォルトは設定済の値であり、DBA_DV_ROLE データ・ディクショナリ・ビューに問い合せることで確認できます。
rule_set_name	このセキュア・アプリケーションを有効にできるかどうかを判断するためのルール・セットの名前。 現行のデータベース・インスタンスで既存のルール・セットを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。

例

```
BEGIN
 DBMS_MACADM.UPDATE_ROLE(
role_name      => 'SECT2_SYSADMIN',
enabled        => DBMS_MACUTL.G_YES,
rule_set_name  => 'System Access Controls');
END;
/
```

親トピック: DBMS_MACADMセキュア・アプリケーション・ロールのプロシージャ

18.2 DBMS_MACSEC_ROLESセキュア・アプリケーション・ロールのプロシージャおよびファンクション

DBMS_MACSEC_ROLESパッケージは、ユーザーに対する認可をチェックし、Oracle Database Vaultセキュア・アプリケーション・ロールを設定します。

DBMS_MACSEC_ROLESパッケージは、すべてのユーザーが使用できます。

- CAN_SET_ROLEファンクション
  CAN_SET_ROLEファンクションは、メソッドを起動するユーザーに、Oracle Database Vaultセキュア・アプリケーション・ロールを使用する権限が付与されているかどうかをチェックします。
- SET_ROLEプロシージャ
  SET_ROLEプロシージャは、指定されたロールのSET ROLE PL/SQL文を発行します。

親トピック: Oracle Database Vaultセキュア・ アプリケーション・ ロール のAPI

18.2.1 CAN_SET_ROLEファンクション

CAN_SET_ROLEファンクションは、メソッドを起動するユーザーが、Oracle Database Vaultセキュア・アプリケーション・ロールを使用するように認可されているかどうかをチェックします。

認可は、ロールに関連付けられたルール・セットのチェックにより決定されます。戻り型はBOOLEANです。

構文

```
DBMS_MACSEC_ROLES.CAN_SET_ROLE(
p_role IN VARCHAR2)
RETURN BOOLEAN;
```

パラメータ

表18-5 CAN_SET_ROLEのパラメータ

パラメータ	説明
p_role	ロール名。 現行のデータベース・インスタンスで既存のセキュア・アプリケーション・ロールを確認するには、「DBA_DV_ROLE ビュー」で説明されている DBA_DV_ROLE ビューに問い合せます。

例

```
SET SERVEROUTPUT ON
BEGIN
 IF DBMS_MACSEC_ROLES.CAN_SET_ROLE('SECTOR2_APP_MGR')
  THEN DBMS_OUTPUT.PUT_LINE('''SECTOR2_APP_MGR'' can be enabled.');
 END IF;
END;
/
```

親トピック: DBMS_MACSEC_ROLESセキュア・アプリケーション・ロールのプロシージャおよびファンクション

18.2.2 SET_ROLEプロシージャ

SET_ROLEプロシージャは、指定されたロールのSET ROLE PL/SQL文を発行します。

このプロシージャのチェック・プロセスには、Oracle Database Vaultセキュア・アプリケーション・ロールと通常のOracle Databaseロールの両方が含まれます。

このプロシージャは、ロールに関連付けられているルール・セットがtrueと評価される場合にのみ、Oracle Database Vaultセキュア・アプリケーション・ロールを設定します。SET ROLEの発行前に、CAN_SET_ROLEメソッドがコールされ、ロールに関連付けられたルール・セットがチェックされます。監査などのランタイムのルール・セットの動作、障害処理およびイベント処理が、このプロセス中に発生します。

SET_ROLEプロシージャは、一般のデータベース・アカウント群で使用できます。

構文

```
DBMS_MACSEC_ROLES.SET_ROLE(
p_role IN VARCHAR2);
```

パラメータ

表18-6 SET_ROLEのパラメータ

パラメータ	説明
p_role	ロール名。セキュア・アプリケーション・ロールや標準ロールなど、複数のロールをカンマ(,)区切りで入力できます。 現行のデータベース・インスタンスで既存のセキュア・アプリケーション・ロールを確認するには、「DBA_DV_ROLE ビュー」で説明されている DBA_DV_ROLE ビューに問い合せます。 データベース内の既存のすべてのロールを確認するには、『Oracle Database リファレンス』で説明されている DBA_ROLES データ・ディクショナリ・ビューに問い合せます。

例

```
EXEC DBMS_MACSEC_ROLES.SET_ROLE('SECTOR2_APP_MGR, APPS_MGR');
```

ロール名前は大文字、小文字のいずれでも入力できます(Sector2_APP_MGRなど)。

親トピック: DBMS_MACSEC_ROLESセキュア・アプリケーション・ロールのプロシージャおよびファンクション

19 Oracle Database Vault Oracle Label SecurityのAPI

DBMS_MACADM PL/SQLパッケージを使用すると、Oracle Database VaultでOracle Label Securityラベルおよびポリシーを管理できます。

- CREATE_MAC_POLICYプロシージャ
  CREATE_MAC_POLICYプロシージャは、ファクタのラベルまたはOracle Label Securityセッション・ラベルを算出する際にラベルをマージするアルゴリズムを指定します。
- CREATE_POLICY_LABELプロシージャ
  CREATE_POLICY_LABELプロシージャは、Oracle Label Securityポリシーのアイデンティティにラベルを付けます。
- DELETE_MAC_POLICY_CASCADEプロシージャ
  DELETE_MAC_POLICY_CASCADEプロシージャは、Oracle Label Securityポリシーに関連するすべてのOracle Database Vaultオブジェクトを削除します。
- DELETE_POLICY_FACTORプロシージャ
  DELETE_POLICY_FACTORプロシージャは、Oracle Label Securityラベルの構成からファクタを削除します。
- DELETE_POLICY_LABELプロシージャ
  DELETE_POLICY_LABELプロシージャは、Oracle Label Securityポリシーのアイデンティティからラベルを削除します。
- UPDATE_MAC_POLICYプロシージャ
  UPDATE_MAC_POLICYプロシージャは、ファクタのラベルまたはOracle Label Securityセッション・ラベルを算出する際にラベルをマージするアルゴリズムを指定します。

関連トピック

- Oracle Database Vaultとその他のOracle製品の統合
- Oracle Database VaultユーティリティのAPI

19.1 CREATE_MAC_POLICYプロシージャ

CREATE_MAC_POLICYプロシージャは、ファクタのラベルまたはOracle Label Securityセッション・ラベルを算出する際にラベルをマージするアルゴリズムを指定します。

構文

```
DBMS_MACADM.CREATE_MAC_POLICY(
policy_name  IN VARCHAR2,
algorithm    IN VARCHAR2);
```

パラメータ

表19-1 CREATE_MAC_POLICYのパラメータ

パラメータ	説明
policy_name	既存のポリシーの名前。 現行のデータベース・インスタンスで既存のポリシーを確認するには、「DBA_DV_MAC_POLICY

パラメータ	説明
	ビュー」で説明されている DBA_DV_MAC_POLICY ビューに問い合せます。
algorithm	Oracle Label Security で 2 つのラベルをマージしている場合のマージ・アルゴリズム。表 19-2 に示されている、目的のマージ・アルゴリズムに対応するコードを入力します。たとえば、「最大レベル/論理和/論理和」マージ・アルゴリズムを選択する場合は、HUU と入力します。

表19-2 Oracle Label Securityマージ・アルゴリズム・コード

コード	値
HUU	最大レベル/論理和/論理和
HIU	最大レベル/論理積/論理和
HMU	最大レベル/減算/論理和
HNU	最大レベル/NULL/論理和
HUI	最大レベル/論理和/論理積
HII	最大レベル/論理積/論理積
HMI	最大レベル/減算/論理積
HNI	最大レベル/NULL/論理積
HUM	最大レベル/論理和/減算
HIM	最大レベル/論理積/減算
HMM	最大レベル/減算/減算
HNM	最大レベル/NULL/減算
HUN	最大レベル/論理和/NULL
HIN	最大レベル/論理積/NULL
HMN	最大レベル/減算/NULL

コード	値
HNN	最大レベル/NULL/NULL
LUU	最小レベル/論理和/論理和
LIU	最小レベル/論理積/論理和
LMU	最小レベル/減算/論理和
LNU	最小レベル/NULL/論理和
LUI	最小レベル/論理和/論理積
LII	最小レベル/論理積/論理積
LMI	最小レベル/減算/論理積
LNI	最小レベル/NULL/論理積
LUM	最小レベル/論理和/減算
LIM	最小レベル/論理積/減算
LMM	最小レベル/減算/減算
LNM	最小レベル/NULL/減算
LUN	最小レベル/論理和/NULL
LIN	最小レベル/論理積/NULL
LMN	最小レベル/減算/NULL
LNN	最小レベル/NULL/NULL

例

```
BEGIN
 DBMS_MACADM.CREATE_MAC_POLICY(
policy_name  => 'Access Locations',
algorithm    => 'HUU');
END;
/
```

19.2 CREATE_POLICY_LABELプロシージャ

CREATE_POLICY_LABELプロシージャは、Oracle Label Securityポリシーのアイデンティティにラベルを付けます。

構文

```
DBMS_MACADM.CREATE_POLICY_LABEL(
identity_factor_name   IN VARCHAR2,
identity_factor_value  IN VARCHAR2,
policy_name            IN VARCHAR2,
label                  IN VARCHAR2);
```

パラメータ

表19-3 CREATE_POLICY_LABELのパラメータ

パラメータ	説明
identity_factor_name	ラベルを付けるファクタの名前。 現行のデータベース・インスタンスで既存のファクタを確認するには、「DBA_DV_FACTOR ビュー」で説明されている DBA_DV_FACTOR ビューに問い合せます。 Oracle Label Security ポリシーに関連付けられているファクタを確認するには、「DBA_DV_MAC_POLICY_FACTOR ビュー」で説明されている DBA_DV_MAC_POLICY_FACTOR を使用します。
identity_factor_value	ラベルを付けるファクタのアイデンティティの値。 現行のデータベース・インスタンスで既存のファクタのアイデンティティを確認するには、「DBA_DV_IDENTITY ビュー」で説明されている DBA_DV_IDENTITY ビューに問い合せます。
policy_name	既存のポリシーの名前。 現行のデータベース・インスタンスで既存のポリシーを確認するには、「DBA_DV_MAC_POLICY ビュー」で説明されている DBA_DV_MAC_POLICY ビューに問い合せます。
label	Oracle Label Security ラベル名。 ファクタ・アイデンティティの既存のポリシー・ラベルを確認するには、「DBA_DV_POLICY_LABEL ビュー」で説明されている DBA_DV_POLICY_LABEL ビューに問い合せます。

例

```
BEGIN
 DBMS_MACADM.CREATE_POLICY_LABEL(
identity_factor_name   => 'App_Host_Name',
identity_factor_value  => 'Sect2_Fin_Apps',
policy_name            => 'Access Locations',
label                  => 'Sensitive');
END;
/
```

親トピック: Oracle Database Vault Oracle Label Security のAPI

19.3 DELETE_MAC_POLICY_CASCADEプロシージャ

DELETE_MAC_POLICY_CASCADEプロシージャは、Oracle Label Securityポリシーに関連するすべてのOracle Database Vaultオブジェクトを削除します。

構文

```
DBMS_MACADM.DELETE_MAC_POLICY_CASCADE(
policy_name  IN VARCHAR2);
```

パラメータ

表19-4 DELETE_MAC_POLICY_CASCADEのパラメータ

パラメータ	説明
policy_name	既存のポリシーの名前。 現行のデータベース・インスタンスで既存のポリシーを確認するには、「DBA_DV_MAC_POLICY ビュー」で説明されている DBA_DV_MAC_POLICY ビューに問い合せます。

例

```
EXEC DBMS_MACADM.DELETE_MAC_POLICY_CASCADE('Access Locations');
```

親トピック: Oracle Database Vault Oracle Label Security のAPI

19.4 DELETE_POLICY_FACTORプロシージャ

DELETE_POLICY_FACTORプロシージャは、Oracle Label Securityラベルの構成からファクタを削除します。

構文

```
DBMS_MACADM.DELETE_POLICY_FACTOR(
policy_name  IN VARCHAR2,
factor_name  IN VARCHAR2);
```

パラメータ

表19-5 DELETE_POLICY_FACTORのパラメータ

パラメータ	説明

パラメータ	説明
policy_name	既存のポリシーの名前。 現行のデータベース・インスタンスで既存のポリシーを確認するには、「DBA_DV_MAC_POLICY ビュー」で説明されている DBA_DV_MAC_POLICY ビューに問い合せます。
factor_name	Oracle Label Security ラベルに関連付けられているファクタの名前。 Oracle Label Security ポリシーに関連付けられているファクタを確認するには、「DBA_DV_MAC_POLICY_FACTOR ビュー」で説明されている DBA_DV_MAC_POLICY_FACTOR に問い合せます。

例

```
BEGIN
 DBMS_MACADM.DELETE_POLICY_FACTOR(
policy_name  => 'Access Locations',
factor_name  => 'App_Host_Name');
END;
/
```

親トピック: Oracle Database Vault Oracle Label Security のAPI

19.5 DELETE_POLICY_LABELプロシージャ

DELETE_POLICY_LABELプロシージャは、Oracle Label Securityポリシーのアイデンティティからラベルを削除します。

構文

```
DBMS_MACADM.DELETE_POLICY_LABEL(
identity_factor_name   IN VARCHAR2,
identity_factor_value  IN VARCHAR2,
policy_name            IN VARCHAR2,
label                  IN VARCHAR2);
```

パラメータ

表19-6 DELETE_POLICY_LABELのパラメータ

パラメータ	説明
identity_factor_name	ラベルを付けたファクタの名前。 現行のデータベース・インスタンスで Oracle Label Security ポリシーに関連付けられている既存のファクタを確認するには、「DBA_DV_MAC_POLICY_FACTOR ビュー」で説明されている DBA_DV_MAC_POLICY_FACTOR に問い合せます。
identity_factor_value	ラベルを付けたファクタのアイデンティティの値。 現行のデータベース・インスタンスで既存のファクタのアイデンティティを確認するには、

パラメータ	説明
	「DBA_DV_IDENTITY ビュー」で説明されている DBA_DV_IDENTITY ビューに問い合せます。
policy_name	既存のポリシーの名前。 現行のデータベース・インスタンスで既存のポリシーを確認するには、「DBA_DV_MAC_POLICY ビュー」で説明されている DBA_DV_MAC_POLICY ビューに問い合せます。
label	Oracle Label Security ラベル名。 ファクタ・アイデンティティの既存のポリシー・ラベルを確認するには、「DBA_DV_POLICY_LABEL ビュー」で説明されている DBA_DV_POLICY_LABEL ビューに問い合せます。

例

```
BEGIN
 DBMS_MACADM.DELETE_POLICY_LABEL(
identity_factor_name   => 'App_Host_Name',
identity_factor_value  => 'Sect2_Fin_Apps',
policy_name            => 'Access Locations',
label                  => 'Sensitive');
END;
/
```

親トピック: Oracle Database Vault Oracle Label Security のAPI

19.6 UPDATE_MAC_POLICYプロシージャ

UPDATE_MAC_POLICYプロシージャは、ファクタのラベルまたはOracle Label Securityセッション・ラベルを算出する際にラベルをマージするアルゴリズムを指定します。

構文

```
DBMS_MACADM.UPDATE_MAC_POLICY(
policy_name  IN VARCHAR2,
algorithm    IN VARCHAR2);
```

パラメータ

表19-7 UPDATE_MAC_POLICY

パラメータ	説明
policy_name	既存のポリシーの名前。 現行のデータベース・インスタンスで既存のポリシーを確認するには、「DBA_DV_MAC_POLICY ビュー」で説明されている DBA_DV_MAC_POLICY

パラメータ	説明
	ビューに問い合せます。
algorithm	Oracle Label Security で 2 つのラベルをマージしている場合のマージ・アルゴリズム。使用可能なアルゴリズムのリストは、表 19-2 を参照してください。

例

```
BEGIN
 DBMS_MACADM.UPDATE_MAC_POLICY(
policy_name  => 'Access Locations',
algorithm    => 'LUI');
END;
/
```

親トピック: Oracle Database Vault Oracle Label Security のAPI

20 Oracle Database VaultユーティリティのAPI

Oracle Database Vaultは、DBMS_MACUTL PL/SQLパッケージの一連のユーティリティのAPIを提供します。

- DBMS_MACUTLの定数
  DBMS_MACUTL PL/SQLパッケージで使用できる、一連の定数を使用できます。
- DBMS_MACUTLパッケージのプロシージャおよびファンクション
  DBMS_MACUTL PL/SQLパッケージでは、時間値、またはユーザーに適切な権限があるかどうかを知るなどのタスクを実行できます。

20.1 DBMS_MACUTLの定数

DBMS_MACUTL PL/SQLパッケージで使用できる、一連の定数を使用できます。

- DBMS_MACUTLの定数のリスト
  DBMS_MACUTL PL/SQLパッケージには、Oracle Database Vault PL/SQLパッケージと使用する定数(フィールド)が用意されています。
- 例: DBMS_MACUTLの定数を使用したレルムの作成
  定数を使用して、Oracle Database Vaultでのオブジェクトの作成時に単純なYesまたはNoの設定に回答できます。
- 例: DBMS_MACUTLの定数を使用したルール・セットの作成
  定数を使用して、使用される監査のタイプや失敗オプションなど、オプションを設定できます。
- 例: DBMS_MACUTLの定数を使用したファクタの作成
  定数を使用して、アイデンティティやラベリングなど、ファクタ固有の情報を設定できます。

親トピック: Oracle Database VaultユーティリティのAPI

20.1.1 DBMS_MACUTLの定数のリスト

DBMS_MACUTL PL/SQLパッケージには、Oracle Database Vault PL/SQLパッケージと使用する定数(フィールド)が用意されています。

表20-1に、DBMS_MACUTLパッケージの定数(つまり、フィールド)の説明をまとめています。

これらの定数の多くは、Oracle Database Vaultパッケージに同等のものがあります。たとえば、複数のプロシージャで使用可能なenabledパラメータでは、Y(Yes)または定数G_YESのいずれも使用できます。どちらを選択するかは個人の好みの問題です。どちらも結果は同じです。

表20-1 DBMS_MACUTLの定数のリスト

定数名	データ型	説明
G_ALL_OBJECT	VARCHAR2(1)	すべてのオブジェクト名またはオブジェクト・タイプを示すワイルドカードとして、レルム API object_name パラメータおよび object_type パラメータで使用します。

定数名	データ型	説明
G_AUDIT_ALWAYS	NUMBER	監査を有効にするために、ファクタ API audit_options パラメータで使用します。
G_AUDIT_OFF	NUMBER	監査を無効にするために、ファクタ API audit_options パラメータで使用します。
G_AUDIT_ON_GET_ERROR	NUMBER	get_expr パラメータで指定した式がエラーを返した場合に監査を行うために、ファクタ API audit_options パラメータで使用します。
G_AUDIT_ON_GET_NULL	NUMBER	get_expr フィールドの式が NULL の場合に監査を行うために、ファクタ API audit_options パラメータで使用します。
G_AUDIT_ON_TRUST_LEVEL_NEG	NUMBER	信頼レベルが負の場合に監査を行うために、ファクタ API audit_options パラメータで使用します。
G_AUDIT_ON_TRUST_LEVEL_NULL	NUMBER	信頼レベルが存在しない場合に監査を行うために、ファクタ API audit_options パラメータで使用します。
G_AUDIT_ON_VALIDATE_ERROR	NUMBER	検証ファンクションがエラーを返した場合に監査を行うために、ファクタ API audit_options パラメータで使用します。
G_AUDIT_ON_VALIDATE_FALSE	NUMBER	検証ファンクションが False の場合に監査を行うために、ファクタ API audit_options パラメータで使用します。
G_DISABLE	NUMBER	Oracle Database Vault のポリシーおよびコマンド・ルールを無効にするために使用します。
G_ENABLE	NUMBER	Oracle Database Vault のポリシーおよびコマンド・ルールを有効にするために使用します。
G_EVAL_ON_ACCESS	NUMBER	アクセスするたびにファクタを再評価するために、ファクタ API eval_options パラメータで使用します。

定数名	データ型	説明
G_EVAL_ON_SESSION	NUMBER	ユーザーがセッションにログインしたとき、1 回のみファクタを評価するために、ファクタ API eval_options パラメータで使用します。
G_FAIL_SILENTLY	NUMBER	失敗時にエラー・メッセージを表示しないように、fail_options パラメータで使用します。
G_FAIL_WITH_MESSAGE	NUMBER	失敗時にエラー・メッセージを表示するように、fail_options パラメータで使用します。
G_IDENTIFY_BY_CONSTANT	NUMBER	ファクタ API identify_by パラメータで使用します。get_expr パラメータで定義される PL/SQL 式では固定値です。
G_IDENTIFY_BY_CONTEXT	NUMBER	コンテキストを示すために、ファクタ API identify_by パラメータで使用します。
G_IDENTIFY_BY_FACTOR	NUMBER	factor_link$表からのサブファクタに対して、ファクタ API identify_by パラメータで使用します。
G_IDENTIFY_BY_METHOD	NUMBER	ファクタ API identify_by パラメータで使用します。get_expr フィールドの式。
G_IDENTIFY_BY_RULESET	NUMBER	ファクタ API identify_by パラメータで使用します。factor_expr$表での式およびルール・セット。
G_LABELED_BY_FACTORS	NUMBER	サブファクタおよびマージ・アルゴリズムからラベルを導出するために、ファクタ API labeled_by パラメータで使用します。
G_LABELED_BY_SELF	NUMBER	ファクタ・アイデンティティのラベル付けのために、ファクタ API labeled_by パラメータで使用します。
G_MAX_SESSION_LABEL	VARCHAR2(30)	これは、ファクタに基づいてユーザーが設定できる最高のラベルです。ユーザーに対するラベルは考慮されません。

定数名	データ型	説明
G_MIN_POLICY_LABEL	VARCHAR2(30)	NULL ラベルのファクタにデフォルト設定されるラベル。
G_NO	VARCHAR2(1)	次の API で使用します。 ● 親ファクタにリンクしている子ファクタが、Oracle Label Security 統合で親ファクタのラベルを構成しないことを示すために、ファクタ API label_indicator パラメータで使用します。 ● enabled パラメータを使用するすべての API。
G_OLS_SESSION_LABEL	VARCHAR2(30)	init_session の実行時のユーザーに対する Oracle Label Security セッション・ラベル。
G_PARTIAL	NUMBER	個別に変更するように Oracle Database Vault ポリシーの下のレルムおよびコマンド・ルールの強制状態を設定します。
G_REALM_AUDIT_FAIL	NUMBER	レルム違反が発生したときに監査を行うために、レルム API audit_options パラメータで使用します。
G_REALM_AUDIT_OFF	NUMBER	監査を無効にするために、レルム API audit_options パラメータで使用します。
G_REALM_AUDIT_SUCCESS	NUMBER	レルム API audit_options parameter パラメータで使用します。レルムへのアクセス成功時に監査を行います。
G_REALM_AUTH_OWNER	NUMBER	所有者に対するレルム認可を設定するために、レルム API auth_options パラメータで使用します。
G_REALM_AUTH_PARTICIPANT	NUMBER	参加者に対するレルム認可を設定するために、レルム API auth_options パラメータで使用します。

定数名	データ型	説明
G_RULESET_AUDIT_FAIL	NUMBER	ルール・セット失敗時に監査を行うために、ルール・セット API audit_options パラメータで使用します。
G_RULESET_AUDIT_OFF	NUMBER	監査を無効にするために、ルール・セット API audit_options パラメータで使用します。
G_RULESET_AUDIT_SUCCESS	NUMBER	ルール・セット成功時に監査を行うために、ルール・セット API audit_options パラメータで使用します。
G_RULESET_EVAL_ALL	NUMBER	すべてのルールが True と評価された場合にルール・セットが成功するように、ルール・セット API eval_options パラメータで使用します。
G_RULESET_EVAL_ANY	NUMBER	ルールのいずれかが True と評価された場合に成功するように、ルール・セット API eval_options パラメータで使用します。
G_RULESET_FAIL_SHOW	NUMBER	ルール・セットが失敗した場合にエラー・メッセージが表示されるように、ルール・セット API fail_options パラメータで使用します。
G_RULESET_FAIL_SILENT	NUMBER	ルール・セットが失敗した場合にエラー・メッセージが表示されないように、ルール・セット API fail_options パラメータで使用します。
G_RULESET_HANDLER_FAIL	NUMBER	ルール・セットが失敗した場合にハンドラ(handler パラメータで指定)をコールするように、ルール・セット API handler_options パラメータで使用します。
G_RULESET_HANDLER_OFF	NUMBER	ハンドラへのコールを無効にするか、ハンドラが使用されない場合に、ルール・セット API handler_options パラメータで使用します。
G_RULESET_HANDLER_SUCCESS	NUMBER	ルール・セットが成功した場合にハンドラをコールするように、ルール・セット API

定数名	データ型	説明
		handler_options パラメータで使用します。
G_SIMULATION	NUMBER	ポリシーの強制状態をシミュレーション・モードに設定するために使用します。このモードでは、レルムまたはコマンド・ルールの違反に対してエラーが発生しません。かわりに、エラーに関連する十分な情報(たとえば、ユーザーまたは SQL コマンド)とともに、指定されたログ表にエラーが記録されます。
G_USER_POLICY_LABEL	VARCHAR2(30)	前の値を考慮した上で、ユーザーのラベルに設定する必要があると Oracle Label Security で決められた値。
G_YES	VARCHAR2(1)	次の API で使用します。 ● 親ファクタにリンクしている子ファクタが、Oracle Label Security 統合で親ファクタのラベルを構成することを示すために、ファクタ API label_indicator パラメータで使用します。 ● enabled パラメータを使用するすべての API。

親トピック: DBMS_MACUTLの定数

20.1.2 例: DBMS_MACUTLの定数を使用したレルムの作成

定数を使用して、Oracle Database Vaultでのオブジェクトの作成時に単純なYesまたはNoの設定に回答できます。

例20-1に、レルム作成時にG_YESおよびG_REALM_AUDIT_FAIL DBMS_MACUTL定数を使用する方法を示します。

例20-1 DBMS_MACUTLの定数を使用したレルムの作成

```
BEGIN
 DBMS_MACADM.CREATE_REALM(
realm_name    => 'Performance Statistics Realm',
description   => 'Realm to measure performance',
enabled       =>  DBMS_MACUTL.G_YES,
audit_options =>  DBMS_MACUTL.G_REALM_AUDIT_FAIL);
END;
/
```

親トピック: DBMS_MACUTLの定数

20.1.3 例: DBMS_MACUTLの定数を使用したルール・セットの作成

定数を使用して、使用される監査のタイプや失敗オプションなど、オプションを設定できます。

例20-2は、ルール・セット作成時に複数のDBMS_MACUTLの定数を使用する方法を示しています。

例20-2 DBMS_MACUTLの定数を使用したルール・セットの作成

```
BEGIN
 DBMS_MACADM.CREATE_RULE_SET(
rule_set_name    => 'Limit_DBA_Access',
description      => 'DBA access through predefined processes',
enabled          => DBMS_MACUTL.G_YES,
eval_options     => DBMS_MACUTL.G_RULESET_EVAL_ALL,
audit_options    => DBMS_MACUTL.G_RULESET_AUDIT_FAIL,
fail_options     => DBMS_MACUTL.G_RULESET_FAIL_SHOW,
fail_message     => 'Rule Set Limit_DBA_Access has failed.',
fail_code        => 20000,
handler_options  => DBMS_MACUTL.G_RULESET_HANDLER_FAIL,
handler          => 'dbavowner.email_alert');
END;
/
```

親トピック: DBMS_MACUTLの定数

20.1.4 例: DBMS_MACUTLの定数を使用したファクタの作成

定数を使用して、アイデンティティやラベリングなど、ファクタ固有の情報を設定できます。

例20-3は、ファクタ作成時に、定数を使用する方法を示しています。

例20-3 DBMS_MACUTLの定数を使用したファクタの作成

```
BEGIN
 DBMS_MACADM.CREATE_FACTOR(
  factor_name       => 'Sector2_DB',
  factor_type_name  => 'Instance',
  description       => ' ',
  rule_set_name     => 'DB_access',
  get_expr          => 'UPPER(SYS_CONTEXT(''USERENV'',''DB_NAME''))',
  validate_expr     => 'dbavowner.check_db_access',
  identify_by       => DBMS_MACUTL.G_IDENTIFY_BY_FACTOR,
  labeled_by        => DBMS_MACUTL.G_LABELED_BY_SELF,
  eval_options      => DBMS_MACUTL.G_EVAL_ON_SESSION,
  audit_options     => DBMS_MACUTL.G_AUDIT_ALWAYS,
  fail_options      => DBMS_MACUTL.G_FAIL_SILENTLY);
END;
/
```

親トピック: DBMS_MACUTLの定数

20.2 DBMS_MACUTLパッケージのプロシージャおよびファンクション

DBMS_MACUTL PL/SQLパッケージでは、時間値、またはユーザーに適切な権限があるかどうかを知るなどのタスクを実行できます。

- CHECK_DVSYS_DML_ALLOWEDプロシージャ
  CHECK_DVSYS_DML_ALLOWEDプロシージャは、ユーザーがData Modification Language (DML)のコマンドを発行してDVSYSオブジェクトにアクセスできるかどうかを確認します。
- GET_CODE_VALUEファンクション

GET_CODE_VALUEファンクションは、コード・グループ内でコードの値を検索し、VARCHAR2値を返します。

- GET_SECONDファンクション
  GET_SECONDファンクションは、Oracle SS(秒)形式で秒(00から59)を返し、NUMBER値を返します。
- GET_MINUTEファンクション
  GET_MINUTEファンクションは、Oracle MI(分)形式の分(00から59)をNUMBER値で返します。
- GET_HOURファンクション
  GET_HOURファンクションは、Oracle HH24(時間)形式の時間(00から23)をNUMBER値で返します。
- GET_DAYファンクション
  GET_DAYファンクションは、Oracle DD(日)形式の日(01から31)をNUMBER値で返します。
- GET_MONTHファンクション
  GET_MONTHファンクションは、Oracle MM(月)形式の月(01から12)をNUMBER値で返します。
- GET_YEARファンクション
  GET_YEARファンクションは、Oracle YYYY(年)形式の年(0001から9999)をNUMBER値で返します。
- IS_ALPHAファンクション
  IS_ALPHAファンクションは、文字がアルファベットかどうかを示すBOOLEAN値を返します。
- IS_DIGITファンクション
  IS_DIGITファンクションは、文字が数値かどうかを示すBOOLEAN値を返します。
- IS_DVSYS_OWNERファンクション
  IS_DVSYS_OWNERファンクションは、Oracle Database Vault構成を管理する権限がユーザーに付与されているかどうかを示すBOOLEAN値を返します。
- IS_OLS_INSTALLEDファンクション
  IS_OLS_INSTALLEDファンクションは、Oracle Label Securityがインストールされているかどうかを示すBOOLEAN値を返します。
- IS_OLS_INSTALLED_VARCHARファンクション
  IS_OLS_INSTALLED_VARCHARファンクションは、Oracle Label Securityがインストールされているかどうかを示すBOOLEAN値を返します。
- ROLE_GRANTED_ENABLED_VARCHARファンクション
  ROLE_GRANTED_ENABLED_VARCHARファンクションは、ユーザーのロール付与と有効化のステータスを示すVARCHAR2値を返します。
- USER_HAS_OBJECT_PRIVILEGEファンクション
  USER_HAS_OBJECT_PRIVILEGEファンクションは、ユーザーまたはロールが、指定された1つのオブジェクト権限の付与によってオブジェクトにアクセスできるかどうかを示すBOOLEAN値を返します。
- USER_HAS_ROLEファンクション
  USER_HAS_ROLEファンクションは、ユーザーがロール権限を直接保持するのか間接的に(他のロールを介して)保持するのかを示すBOOLEAN値を返します。
- USER_HAS_ROLE_VARCHARファンクション
  USER_HAS_ROLE_VARCHARファンクションは、ユーザーがロール権限を直接保持するのか間接的に(他のロールを介して)保持するのかを示すVARCHAR2値を返します。
- USER_HAS_SYSTEM_PRIVILEGEファンクション
  USER_HAS_SYSTEM_PRIVILEGEファンクションは、ユーザーがシステム権限を直接保持するのか間接的に(ロールを介して)保持するのかを示すBOOLEAN値を返します。

親トピック: Oracle Database VaultユーティリティのAPI

20.2.1 CHECK_DVSYS_DML_ALLOWEDプロシージャ

CHECK_DVSYS_DML_ALLOWEDプロシージャは、ユーザーがData Modification Language (DML)のコマンドを発行してDVSYSオブジェクトにアクセスできるかどうかを確認します。

構文

```
DBMS_MACUTL.CHECK_DVSYS_DML_ALLOWED(
p_user IN VARCHAR2 DEFAULT USER);
```

パラメータ

表20-2 CHECK_DVSYS_DML_ALLOWEDのパラメータ

パラメータ	説明
p_user	チェックするユーザー。 現在のデータベース・インスタンスで既存のユーザーを検索するには、次のビューに問い合せます。 ● DBA_USERS: 現在のデータベース・インスタンスで使用可能なユーザーを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を確認します。「DBA_DV_REALM_AUTH ビュー」を参照してください。 ● DBA_DV_ROLE: 権限管理で使用されている既存のセキュア・アプリケーション・ロールを確認します。「DBA_DV_ROLE ビュー」を参照してください。

例

ユーザーSYSTEMがチェックに失格となります。

```
EXEC DBMS_MACUTL.CHECK_DVSYS_DML_ALLOWED('system');
ERROR at line 1:
ORA-47920: Authorization failed for user system to perform this operation
ORA-06512: at "DBMS_MACUTL", line 23
ORA-06512: at "DBMS_MACUTL", line 372
ORA-06512: at "DBMS_MACUTL", line 508
ORA-06512: at "DBMS_MACUTL", line 572
ORA-06512: at line 1
```

DV_OWNERロールを持つユーザーsec_admin_owenが、チェックに合格します。

```
EXEC DBMS_MACUTL.CHECK_DVSYS_DML_ALLOWED('sec_admin_owen');
PL/SQL procedure successfully completed.
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.2 GET_CODE_VALUEファンクション

GET_CODE_VALUEファンクションは、コード・グループ内でコードの値を検索し、VARCHAR2値を返します。

構文

```
DBMS_MACUTL.GET_CODE_VALUE(
```

```
p_code_group IN VARCHAR2,
p_code       IN VARCHAR2)
RETURN VARCHAR2;
```

パラメータ

表20-3 GET_CODE_VALUEのパラメータ

パラメータ	説明
p_code_group	コード・グループ(AUDIT_EVENTS、BOOLEAN など)。 現行のデータベース・インスタンス内の使用可能なコード・グループを確認するには、「DBA_DV_CODE ビュー」で説明されている DBA_DV_CODE ビューに問い合せます。
p_code	コードの ID。 この ID は、DBA_DV_CODE ビューを実行したときに表示されます。

例

```
BEGIN
 DBMS_MACADM.CREATE_RULE(
  rule_name => 'Get Label Algorithm for Maximum Level/Union/Null',
  rule_expr => 'DBMS_MACUTL.GET_CODE_VALUE(''LABEL_ALG'', ''HUN'') = ''Union''');
END;
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.3 GET_SECONDファンクション

GET_SECONDファンクションは、Oracle SS(秒)形式で秒(00から59)を返し、NUMBER値を返します。

時間データに基づいたルール式に有用です。

構文

```
DBMS_MACUTL.GET_SECOND(
p_date IN DATE DEFAULT SYSDATE)
RETURN NUMBER;
```

パラメータ

表20-4 GET_SECONDのパラメータ

パラメータ	説明
p_date	SS 形式の日付(たとえば 59)。 日付を指定しない場合、Oracle Database Vault は Oracle Database の SYSDATE ファンクションを使用して、データベースが存在するオペレーティング・システムに設定されている現在の日時を取得します。

例

```
SET SERVEROUTPUT ON
DECLARE
  seconds number;
BEGIN
  seconds := DBMS_MACUTL.GET_SECOND(TO_DATE('03-APR-2009 6:56 PM',
  'dd-mon-yyyy hh:mi PM'));
  DBMS_OUTPUT.PUT_LINE('Seconds: '||seconds);
END;
/
```

この例では固定日時を使用しており、次の値を返します。

```
Seconds: 56
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.4 GET_MINUTEファンクション

GET_MINUTEファンクションは、Oracle MI(分)形式の分(00から59)をNUMBER値で返します。

時間データに基づいたルール式に有用です。

構文

```
DBMS_MACUTL.GET_MINUTE(
p_date IN DATE DEFAULT SYSDATE)
RETURN NUMBER;
```

パラメータ

表20-5 GET_MINUTEのパラメータ

パラメータ	説明
p_date	MI 形式の日付(たとえば 2:30 の場合は 30)。 日付を指定しない場合、Oracle Database Vault は Oracle Database の SYSDATE ファンクションを使用して、データベースが存在するオペレーティング・システムに設定されている現在の日時を取得します。

例

```
SET SERVEROUTPUT ON
DECLARE
  minute number;
BEGIN
  minute := DBMS_MACUTL.GET_MINUTE(SYSDATE);
  DBMS_OUTPUT.PUT_LINE('Minute: '||minute);
END;
/
```

次のような出力が表示されます。

```
Minute: 17
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.5 GET_HOURファンクション

GET_HOURファンクションは、Oracle HH24(時間)形式の時間(00から23)をNUMBER値で返します。

時間データに基づいたルール式に有用です。

構文

```
DBMS_MACUTL.GET_HOUR(
p_date IN DATE DEFAULT SYSDATE)
RETURN NUMBER;
```

パラメータ

表20-6 GET_HOURのパラメータ

パラメータ	説明
p_date	HH24 形式の日付(たとえば午後 2:00 の場合は 14)。 日付を指定しない場合、Oracle Database Vault は Oracle Database の SYSDATE ファンクションを使用して、データベースが存在するオペレーティング・システムに設定されている現在の日時を取得します。

例

```
SET SERVEROUTPUT ON
DECLARE
  hours number;
BEGIN
  hours := DBMS_MACUTL.GET_HOUR(SYSDATE);
  DBMS_OUTPUT.PUT_LINE('Hour: '||hours);
END;
/
```

次のような出力が表示されます。

```
Hour: 12
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.6 GET_DAYファンクション

GET_DAYファンクションは、Oracle DD(日)形式の日(01から31)をNUMBER値で返します。

時間データに基づいたルール式に有用です。

構文

```
DBMS_MACUTL.GET_DAY(
p_date IN DATE DEFAULT SYSDATE)
RETURN NUMBER;
```

パラメータ

表20-7 GET_DAYのパラメータ

パラメータ	説明

パラメータ	説明
p_date	DD 形式の日付(たとえば月の初日の場合は 01)。 日付を指定しない場合、Oracle Database Vault は Oracle Database の SYSDATE ファンクションを使用して、データベースが存在するオペレーティング・システムに設定されている現在の日時を取得します。

例

```
SET SERVEROUTPUT ON
DECLARE
  day number;
BEGIN
  day := DBMS_MACUTL.GET_DAY(SYSDATE);
  DBMS_OUTPUT.PUT_LINE('Day: '||day);
END;
/
```

次のような出力が表示されます。

```
Day: 3
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.7 GET_MONTHファンクション

GET_MONTHファンクションは、Oracle MM(月)形式の月(01から12)をNUMBER値で返します。

時間データに基づいたルール式に有用です。

構文

```
DBMS_MACUTL.GET_MONTH(
p_date IN DATE DEFAULT SYSDATE)
RETURN NUMBER;
```

パラメータ

表20-8 GET_MONTHのパラメータ

パラメータ	説明
p_date	MM 形式の日付(たとえば 8 月の場合は 08)。 日付を指定しない場合、Oracle Database Vault は Oracle Database の SYSDATE ファンクションを使用して、データベースが存在するオペレーティング・システムに設定されている現在の日時を取得します。

例

```
SET SERVEROUTPUT ON
DECLARE
  month number;
BEGIN
  month := DBMS_MACUTL.GET_MONTH(SYSDATE);
  DBMS_OUTPUT.PUT_LINE('Month: '||month);
```

```
END;
/
```

次のような出力が表示されます。

```
Month: 4
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.8 GET_YEARファンクション

GET_YEARファンクションは、Oracle YYYY(年)形式の年(0001から9999)をNUMBER値で返します。

時間データに基づいたルール式に有用です。

構文

```
DBMS_MACUTL.GET_YEAR(
p_date IN DATE DEFAULT SYSDATE)
RETURN NUMBER;
```

パラメータ

表20-9 GET_YEARのパラメータ

パラメータ	説明
p_date	YYYY 形式の日付(たとえば 1984)。 日付を指定しない場合、Oracle Database Vault は SYSDATE ファンクションを使用して、データベースが存在するオペレーティング・システムに設定されている現在の日時を取得します。

例

```
SET SERVEROUTPUT ON
DECLARE
  year number;
BEGIN
  year := DBMS_MACUTL.GET_YEAR(SYSDATE);
  DBMS_OUTPUT.PUT_LINE('Year: '||year);
END;
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.9 IS_ALPHAファンクション

IS_ALPHAファンクションは、文字がアルファベットかどうかを示すBOOLEAN値を返します。

IS_ALPHAは、文字がアルファベットの場合にTRUEを返します。

構文

```
DBMS_MACUTL.IS_ALPHA(
c IN VARCHAR2)
RETURN BOOLEAN;
```

パラメータ

表20-10 IS_ALPHAのパラメータ

パラメータ	説明
c	1 文字の文字列

例

```
SET SERVEROUTPUT ON
BEGIN
 IF DBMS_MACUTL.IS_ALPHA('z')
  THEN DBMS_OUTPUT.PUT_LINE('The alphabetic character was found');
 ELSE
  DBMS_OUTPUT.PUT_LINE('No alphabetic characters today.');
 END IF;
END;
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.10 IS_DIGITファンクション

IS_DIGITファンクションは、文字が数値かどうかを示すBOOLEAN値を返します。

IS_DIGITは、文字が数値の場合にTRUEを返します。

構文

```
DBMS_MACUTL.IS_DIGIT(
c IN VARCHAR2)
RETURN BOOLEAN;
```

パラメータ

表20-11 IS_DIGITのパラメータ

パラメータ	説明
c	1 文字の文字列

例

```
SET SERVEROUTPUT ON
BEGIN
 IF DBMS_MACUTL.IS_DIGIT('7')
  THEN DBMS_OUTPUT.PUT_LINE('The numeric character was found');
 ELSE
  DBMS_OUTPUT.PUT_LINE('No numeric characters today.');
 END IF;
END;
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.11 IS_DVSYS_OWNERファンクション

IS_DVSYS_OWNERファンクションは、Oracle Database Vault構成を管理する権限がユーザーに付与されているかどうかを示すBOOLEAN値を返します。

IS_DVSYS_OWNERは、ユーザーに権限が付与されている場合にTRUEを返します。

構文

```
DBMS_MACUTL.IS_DVSYS_OWNER(
p_user IN VARCHAR2 DEFAULT USER)
RETURN BOOLEAN;
```

パラメータ

表20-12 IS_DVSYS_OWNERのパラメータ

パラメータ	説明
p_user	チェックするユーザー。 既存のユーザーを検索するには、次のビューに問い合せます。 ● DBA_USERS: 現在のデータベース・インスタンスで使用可能なユーザーを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を確認します。「DBA_DV_REALM_AUTH ビュー」を参照してください。 ● DBA_DV_ROLE: 権限管理で使用されている既存のセキュア・アプリケーション・ロールを確認します。「DBA_DV_ROLE ビュー」を参照してください。

例

```
SET SERVEROUTPUT ON
BEGIN
 IF DBMS_MACUTL.IS_DVSYS_OWNER('PSMITH')
  THEN DBMS_OUTPUT.PUT_LINE('PSMITH is authorized to manage Database Vault.');
 ELSE
  DBMS_OUTPUT.PUT_LINE('PSMITH is not authorized to manage Database Vault.');
 END IF;
END;
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.12 IS_OLS_INSTALLEDファンクション

IS_OLS_INSTALLEDファンクションは、Oracle Label Securityがインストールされているかどうかを示すBOOLEAN値を返します。

Oracle Label Securityがインストールされている場合、IS_OLS_INSTALLEDはTRUEを返します。

構文

```
DBMS_MACUTL.IS_OLS_INSTALLED()
RETURN BOOLEAN;
```

パラメータ

なし

例

```
SET SERVEROUTPUT ON
BEGIN
 IF DBMS_MACUTL.IS_OLS_INSTALLED()
  THEN DBMS_OUTPUT.PUT_LINE('OLS is installed');
 ELSE
  DBMS_OUTPUT.PUT_LINE('OLS is not installed');
 END IF;
END;
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.13 IS_OLS_INSTALLED_VARCHARファンクション

IS_OLS_INSTALLED_VARCHARファンクションは、Oracle Label Securityがインストールされているかどうかを示すBOOLEAN値を返します。

Oracle Label Securityがインストールされている場合、IS_OLS_INSTALLED_VARCHARはYを返します。

構文

```
DBMS_MACUTL.IS_OLS_INSTALLED_VARCHAR()
RETURN VARCHAR2;
```

パラメータ

なし

例

例については、「IS_OLS_INSTALLEDファンクション」を参照してください。

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.14 ROLE_GRANTED_ENABLED_VARCHARファンクション

ROLE_GRANTED_ENABLED_VARCHARファンクションでは、ユーザーのロール付与と有効化のステータスを示すVARCHAR2値が返されます。

ROLE_GRANTED_ENABLED_VARCHARファンクションは、十分な範囲で直接的または(別のロールを介して)間接的にユーザーにロールが付与されているか、そのロールが現在付与されていないがセッションにおいて有効になっているかを確認します。これらの条件のいずれかに該当する場合は、Yが返されます。

SYS_CONTEXTファンクションのSYS_SESSION_ROLESネームスペースは、DVSYSコマンド・ルールとして評価される場合はログイン・ユーザーのロールを表していないため、ROLE_GRANTED_ENABLED_VARCHARファンクションを使用して、ロールがログイン・ユーザーに対して有効になっているどうかを確認することをお薦めします。

構文

```
DBMS_MACUTL.ROLE_GRANTED_ENABLED_VARCHAR(
p_role IN VARCHAR2,
p_user IN VARCHAR2 DEFAULT USER,
  p_profile IN NUMBER(38) DEFAULT 1,
  p_scope IN VARCHAR2 DEFAULT LOCAL)
RETURN VARCHAR2;
```

パラメータ

表20-13 ROLE_GRANTED_ENABLED_VARCHARのパラメータ

パラメータ	説明
p_role	チェックするロール。 既存のロールを検索するには、次のビューに問い合せます。 ● DBA_ROLES: 現行のデータベース・インスタンスで使用可能なロールを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を確認します。「DBA_DV_REALM_AUTH ビュー」を参照してください。 ● DBA_DV_ROLE: 権限管理で使用されている既存のセキュア・アプリケーション・ロールを確認します。「DBA_DV_ROLE ビュー」を参照してください。
p_user	チェックするユーザー。ROLE_GRANTED_ENABLED_VARCHAR ファンクションをルール評価の一部として使用する場合は、ROLE_GRANTED_ENABLED_VARCHAR が Oracle Database Vault ルールとして評価されていると、p_user を CURRENT_USER に設定できません。かわりに、SYS_CONTEXT ファンクションの USERENV ネームスペースの SESSION_USER を使用してログイン・ユーザーを表すことができます。 既存のユーザーを検索するには、次のビューに問い合せます。 ● DBA_USERS: 現在のデータベース・インスタンスで使用可能なユーザーを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を確認します。「DBA_DV_REALM_AUTH ビュー」を参照してください。
p_profile	権限分析を使用していて、チェック対象のロールを使用している場合は、権限分析でそのロールの使用状況を取得できるように、1 を指定します。そうでない場合は、0 を入力します。
p_scope	共通に付与されるロールの場合は COMMON を、ローカルに付与されるロールの場合は LOCAL を指定します。

例

この例では、コマンド・ルールでDBMS_MACUTL.ROLE_GRANTED_ENABLED_VARCHARファンクションを使用して、ログイン・ユーザーにEMPLOYEEの有効化されたロールが付与されているかどうかを確認する方法を示します。

```
BEGIN
DBMS_MACADM.CREATE_RULE(
  rule_name  => 'does role exist',
  rule_expr  =>
'DVSYS.DBMS_MACUTL.ROLE_GRANTED_ENABLED_VARCHAR(''EMPLOYEE'','''''||dvsys.dv_login_us
er||''''') = ''Y''');
END;
```

```
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.15 USER_HAS_OBJECT_PRIVILEGEファンクション

USER_HAS_OBJECT_PRIVILEGEファンクションは、ユーザーまたはロールが、指定された1つのオブジェクト権限の付与によってオブジェクトにアクセスできるかどうかを示すBOOLEAN値を返します。

ユーザーやロールにオブジェクト権限がある場合、USER_HAS_OBJECT_PRIVILEGEはTRUEを返します。

構文

```
DBMS_MACUTL.USER_HAS_OBJECT_PRIVILEGE(
p_user          VARCHAR2,
p_object_owner  VARCHAR2,
p_object_name   VARCHAR2,
p_privilege     VARCHAR2)
RETURNS BOOLEAN;
```

パラメータ

表20-14 USER_HAS_OBJECT_PRIVILEGEのパラメータ

パラメータ	説明
p_user	チェックするユーザーまたはロール。 既存のユーザーを検索するには、次のビューに問い合せます。 ● DBA_USERS: 現在のデータベース・インスタンスで使用可能なユーザーを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_ROLES: 現行のデータベース・インスタンスで使用可能なロールを検索します。『Oracle Database リファレンス』を参照してください。 ● DVA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を検索します。「DBA_DV_REALM_AUTH ビュー」を参照してください。 ● DBA_DV_ROLE: 権限管理で使用されている既存のセキュア・アプリケーション・ロールを確認します。「DBA_DV_ROLE ビュー」を参照してください。
p_object_owner	スキーマなどのオブジェクト所有者。 使用可能なユーザーを確認するには、『Oracle Database リファレンス』で説明されているDBA_USERS ビューに問い合せます。 特定のユーザーの認可を検索するには、DVA_DV_REALM_AUTH ビューに問い合せます。
p_object_name	オブジェクト名。p_object_owner パラメータで指定されているスキーマ内の表など。

パラメータ	説明
	使用可能なオブジェクトを確認するには、『Oracle Database リファレンス』で説明されている ALL_OBJECTS ビューを問い合せます。 既存のレルムで保護されるオブジェクトを確認するには、DBA_DV_REALM_OBJECT ビューに問い合せます。
p_privilege	UPDATE などのオブジェクト権限。 PUBLIC 権限以外のデータベース・アカウントの権限を確認するには、DBA_DV_USER_PRIVS ビューに問い合せます。 データベース・アカウントのすべての権限を確認するには、DBA_DV_USER_PRIVS_ALL ビューに問い合せます。

例

```
SET SERVEROUTPUT ON
BEGIN
 IF DBMS_MACUTL.USER_HAS_OBJECT_PRIVILEGE(
   'SECTOR2_APP_MGR', 'OE', 'ORDERS', 'UPDATE')
  THEN DBMS_OUTPUT.PUT_LINE('SECTOR2_APP_MGR has the UPDATE privilege for the
OE.ORDERS table');
   ELSE
  DBMS_OUTPUT.PUT_LINE('SECTOR2_APP_MGR does not have the UPDATE privilege for the
OE.ORDERS table.');
 END IF;
END;
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.16 USER_HAS_ROLEファンクション

USER_HAS_ROLEファンクションは、ユーザーがロール権限を直接保持するのか間接的に(他のロールを介して)保持するのかを示すBOOLEAN値を返します。

ユーザーにロール権限がある場合、USER_HAS_ROLEはTRUEを返します。

構文

```
DBMS_MACUTL.USER_HAS_ROLE(
p_role IN VARCHAR2,
p_user IN VARCHAR2 DEFAULT USER)
RETURN BOOLEAN;
```

パラメータ

表20-15 USER_HAS_ROLEのパラメータ

パラメータ	説明

パラメータ	説明
p_role	チェックするロール権限。 既存のロールを検索するには、次のビューに問い合せます。 ● DBA_ROLES: 現行のデータベース・インスタンスで使用可能なロールを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を確認します。「DBA_DV_REALM_AUTH ビュー」を参照してください。 ● DBA_DV_ROLE: 権限管理で使用されている既存のセキュア・アプリケーション・ロールを確認します。「DBA_DV_ROLE ビュー」を参照してください。
p_user	チェックするユーザー。 既存のユーザーを検索するには、次のビューに問い合せます。 ● DBA_USERS: 現在のデータベース・インスタンスで使用可能なユーザーを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を確認します。「DBA_DV_REALM_AUTH ビュー」を参照してください。

例

```
SET SERVEROUTPUT ON
BEGIN
 IF DBMS_MACUTL.USER_HAS_ROLE('SECTOR2_APP_MGR', 'PSMITH')
  THEN DBMS_OUTPUT.PUT_LINE('User PSMITH has the SECTOR2_APP_MGR role');
   ELSE
  DBMS_OUTPUT.PUT_LINE('User PSMITH does not have the SECTOR2_APP_MGR role.');
 END IF;
END;
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.17 USER_HAS_ROLE_VARCHARファンクション

USER_HAS_ROLE_VARCHARファンクションは、ユーザーがロール権限を直接保持するのか間接的に(他のロールを介して)保持するのかを示すVARCHAR2値を返します。

ユーザーにロール権限が指定されている場合、USER_HAS_ROLE_VARCHARはYを返します。

構文

```
DBMS_MACUTL.USER_HAS_ROLE_VARCHAR(
p_role IN VARCHAR2,
p_user IN VARCHAR2 DEFAULT USER)
RETURN VARCHAR2;
```

パラメータ

表20-16 USER_HAS_ROLE_VARCHARのパラメータ

パラメータ	説明
p_role	チェックするロール。 既存のロールを検索するには、次のビューに問い合せます。 ● DBA_ROLES: 現行のデータベース・インスタンスで使用可能なロールを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を確認します。「DBA_DV_REALM_AUTH ビュー」を参照してください。 ● DBA_DV_ROLE: 権限管理で使用されている既存のセキュア・アプリケーション・ロールを確認します。「DBA_DV_ROLE ビュー」を参照してください。
p_user	チェックするユーザー。 既存のユーザーを検索するには、次のビューに問い合せます。 ● DBA_USERS: 現在のデータベース・インスタンスで使用可能なユーザーを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を確認します。「DBA_DV_REALM_AUTH ビュー」を参照してください。

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

20.2.18 USER_HAS_SYSTEM_PRIVILEGEファンクション

USER_HAS_SYSTEM_PRIVILEGEファンクションは、ユーザーがシステム権限を直接保持するのか間接的に(ロールを介して)保持するのかを示すBOOLEAN値を返します。

ユーザーにシステム権限が指定されている場合、USER_HAS_SYSTEM_PRIVILEGEはTRUEを返します。

構文

```
DBMS_MACUTL.USER_HAS_SYSTEM_PRIVILEGE(
p_privilege  IN VARCHAR2,
p_user       IN VARCHAR2 DEFAULT USER)
RETURN BOOLEAN;
```

パラメータ

表20-17 USER_HAS_SYSTEM_PRIVILEGEのパラメータ

パラメータ	説明

パラメータ	説明
p_privilege	チェックするシステム権限。 PUBLIC 権限以外のデータベース・アカウントの権限を確認するには、「DBA_DV_USER_PRIVS ビュー」で説明されている DBA_DV_USER_PRIVS ビューに問い合せます。 データベース・アカウントのすべての権限を確認するには、「DBA_DV_USER_PRIVS_ALL ビュー」で説明されている DBA_DV_USER_PRIVS_ALL を使用します。
p_user	チェックするユーザー。 既存のユーザーを検索するには、次のビューに問い合せます。 ● DBA_USERS: 現在のデータベース・インスタンスで使用可能なユーザーを検索します。『Oracle Database リファレンス』を参照してください。 ● DBA_DV_REALM_AUTH: 特定のユーザーまたはロールの認可を確認します。「DBA_DV_REALM_AUTH ビュー」を参照してください。

例

```
SET SERVEROUTPUT ON
BEGIN
 IF DBMS_MACUTL.USER_HAS_SYSTEM_PRIVILEGE('EXECUTE', 'PSMITH')
  THEN DBMS_OUTPUT.PUT_LINE('User PSMITH has the EXECUTE ANY PRIVILEGE privilege.');
   ELSE
  DBMS_OUTPUT.PUT_LINE('User PSMITH does not have the EXECUTE ANY PRIVILEGE
privilege.');
 END IF;
END;
/
```

親トピック: DBMS_MACUTLパッケージのプロシージャおよびファンクション

21 Oracle Database Vaultの一般管理API

DBMS_MACADM PL/SQLパッケージ・プロシージャおよびCONFIGURE_DVスタンドアロン・プロシージャを使用すると、一般的なメンテナンス・タスクを実行できます。

- DBMS_MACADM一般システム・メンテナンス・プロシージャ
  DBMS_MACADM PL/SQLパッケージの一般システム・メンテナンスのプロシージャは、ユーザーの認可またはOracle Database Vaultへの新しい言語の追加などのタスクを実行します。
- CONFIGURE_DV一般システム・メンテナンス・プロシージャ
  CONFIGURE_DVプロシージャは、最初の2つのOracle Databaseユーザー・アカウントを構成します。このアカウントには、DV_OWNERロールとDV_ACCTMGRロールがそれぞれ付与されます。

21.1 DBMS_MACADM一般システム・メンテナンスのプロシージャ

DBMS_MACADM PL/SQLパッケージの一般システム・メンテナンスのプロシージャは、ユーザーの認可またはOracle Database Vaultへの新しい言語の追加などのタスクを実行します。

- ADD_APP_EXCEPTIONプロシージャ
  ADD_APP_EXCEPTIONプロシージャにより、共通ユーザーまたはパッケージはローカル・スキーマにアクセスできるようになります。
- ADD_NLS_DATAプロシージャ
  ADD_NLS_DATAプロシージャは、Oracle Database Vaultに新しい言語を追加します。
- AUTH_DATAPUMP_CREATE_USERプロシージャ
  AUTH_DATAPUMP_CREATE_USERプロシージャは、Oracle Data Pumpユーザーに対して、Oracle Data Pumpのインポート操作中のユーザーの作成を認可します。
- AUTH_DATAPUMP_GRANTプロシージャ
  AUTH_DATAPUMP_GRANTプロシージャは、Oracle Data Pumpユーザーに対して、Oracle Data Pumpのインポート操作中のOracle Database Vaultで保護されたロールおよびシステム権限の付与を認可します。
- AUTH_DATAPUMP_GRANT_ROLEプロシージャ
  AUTH_DATAPUMP_GRANT_ROLEプロシージャは、Oracle Data Pumpユーザーに対して、Oracle Data Pumpのインポート操作中の特定のロールの付与を認可します。
- AUTH_DATAPUMP_GRANT_SYSPRIVプロシージャ
  AUTH_DATAPUMP_GRANT_SYSPRIVプロシージャは、Oracle Data Pumpユーザーに対して、Oracle Data Pumpのインポート操作中のシステム権限の付与を認可します。
- AUTHORIZE_DATAPUMP_USERプロシージャ
  AUTHORIZE_DATAPUMP_USERプロシージャは、Oracle Database Vaultが有効な場合に、ユーザーがOracle Data Pump操作を実行することを認可します。
- AUTHORIZE_DBCAPTUREプロシージャ
  AUTHORIZE_DBCAPTUREプロシージャは、Oracle Database Replayのワークロード取得操作を実行する認可をユーザーに付与します。
- AUTHORIZE_DBREPLAYプロシージャ
  AUTHORIZE_DBREPLAYプロシージャは、Oracle Database Replayのワークロード・リプレイ操作を実行する認可をユーザーに付与します。
- AUTHORIZE_DDLプロシージャ

AUTHORIZE_DDLプロシージャは、指定したスキーマに対してデータ定義言語(DDL)文を実行する認可をユーザーに付与します。

- AUTHORIZE_DIAGNOSTIC_ADMINプロシージャ
  AUTHORIZE_DIAGNOSTIC_ADMINプロシージャは、診断ビューおよび表を問い合せる認可をユーザーに付与します。
- AUTHORIZE_MAINTENANCE_USERプロシージャ
  AUTHORIZE_MAINTENANCE_USERプロシージャは、Oracle Database Vault環境で情報ライフサイクル管理(ILM)操作を実行するための認可をユーザーに与えます。
- AUTHORIZE_PREPROCESSORプロシージャ
  AUTHORIZE_PREPROCESSORプロシージャは、外部表からプリプロセッサ・プログラムを実行する認可をユーザーに付与します。
- AUTHORIZE_PROXY_USERプロシージャ
  AUTHORIZE_PROXY_USERプロシージャは、プロキシ・ユーザーにデータベース認可がある場合、他のユーザー・アカウントをプロキシする認可をプロキシ・ユーザーに付与します。
- AUTHORIZE_SCHEDULER_USERプロシージャ
  AUTHORIZE_SCHEDULER_USERプロシージャは、Oracle Database Vaultが有効な場合に、データベース・ジョブをスケジュールする権限をユーザーに付与します。
- AUTHORIZE_TTS_USERプロシージャ
  AUTHORIZE_TTS_USERプロシージャは、Oracle Database Vaultが有効な場合に、表領域に対してOracle Data Pumpトランスポータブル表領域操作を実行するようにユーザーを認可します。
- DELETE_APP_EXCEPTIONプロシージャ
  DELETE_APP_EXCEPTIONプロシージャは、Database Vault操作の制御の例外リストから共通ユーザーまたは共通ユーザーのパッケージを削除します。
- DISABLE_APP_PROTECTIONプロシージャ
  DISABLE_APP_PROTECTIONプロシージャは、Database Vault操作の制御を無効にします。
- DISABLE_DVプロシージャ
  DISABLE_DVプロシージャは、Oracle Database Vaultを無効にします。
- DISABLE_DV_DICTIONARY_ACCTSプロシージャ
  DISABLE_DV_DICTIONARY_ACCTSプロシージャでは、どのユーザーもDVSYSまたはDVFスキーマ・ユーザーとしてデータベースにログインできません。
- DISABLE_DV_PATCH_ADMIN_AUDITプロシージャ
  DISABLE_DV_PATCH_ADMIN_AUDITプロシージャは、DV_PATCH_ADMINロールを持つユーザーによるアクションのレルム、コマンド・ルールおよびルール・セットの監査を無効にします。
- DISABLE_ORADEBUGプロシージャ
  DISABLE_ORADEBUGプロシージャは、Oracle Database Vault環境でORADEBUGユーティリティの使用を無効にします。
- ENABLE_APP_PROTECTIONプロシージャ
  ENABLE_APP_PROTECTIONプロシージャは、Database Vault操作の制御を有効にします。
- ENABLE_DVプロシージャ
  ENABLE_DVプロシージャは、Oracle Database VaultおよびOracle Label Securityを有効にします。
- ENABLE_DV_DICTIONARY_ACCTSプロシージャ
  ENABLE_DV_DICTIONARY_ACCTSプロシージャにより、ユーザーはDVSYSまたはDVFユーザーとしてデータベースにログインできます。

- ENABLE_DV_PATCH_ADMIN_AUDITプロシージャ
  ENABLE_DV_PATCH_ADMIN_AUDITプロシージャは、DV_PATCH_ADMINロールを持つユーザーによるアクションのレルム、コマンド・ルールおよびルール・セットの監査を有効にします。
- ENABLE_ORADEBUGプロシージャ
  ENABLE_ORADEBUGプロシージャは、Oracle Database Vault環境でORADEBUGユーティリティの使用を有効にします。
- UNAUTH_DATAPUMP_CREATE_USERプロシージャ
  UNAUTH_DATAPUMP_CREATE_USERプロシージャは、Oracle Data Pumpユーザーから、Oracle Data Pumpのインポート操作中にユーザーを作成する認可を削除します。
- UNAUTH_DATAPUMP_GRANTプロシージャ
  UNAUTH_DATAPUMP_GRANTプロシージャは、Oracle Data Pumpユーザーから、Oracle Data Pumpのインポート操作中にOracle Database Vaultで保護されたロールおよびシステム権限を付与する認可を削除します。
- UNAUTH_DATAPUMP_GRANT_ROLEプロシージャ
  UNAUTH_DATAPUMP_GRANT_ROLEプロシージャは、Oracle Data Pumpユーザーから、Oracle Data Pumpのインポート操作中に特定のロールを付与する認可を削除します。
- UNAUTH_DATAPUMP_GRANT_SYSPRIVプロシージャ
  UNAUTH_DATAPUMP_GRANT_SYSPRIVプロシージャは、Oracle Data Pumpユーザーから、Oracle Data Pumpのインポート操作中にシステム権限を付与する認可を削除します。
- UNAUTHORIZE_DATAPUMP_USERプロシージャ
  UNAUTHORIZE_DATAPUMP_USERプロシージャは、AUTHORIZE_DATAPUMP_USERプロシージャによって付与された権限を取り消します。
- UNAUTHORIZE_DBCAPTUREプロシージャ
  UNAUTHORIZE_DBCAPTUREプロシージャは、Oracle Database Replayのワークロード取得操作を実行する認可をユーザーから取り消します。
- UNAUTHORIZE_DBREPLAYプロシージャ
  UNAUTHORIZE_DBREPLAYプロシージャは、Oracle Database Replayのワークロード・リプレイ操作を実行する認可をユーザーから取り消します。
- UNAUTHORIZE_DDLプロシージャ
  UNAUTHORIZE_DDLプロシージャは、DBMS_MACDM.AUTHORIZE_DDLプロシージャを使用してDDL文を実行する認可を付与されたユーザーから認可を取り消します。
- UNAUTHORIZE_DIAGNOSTIC_ADMINプロシージャ
  UNAUTHORIZE_DIAGNOSTIC_ADMINプロシージャは、DBMS_MACADM.AUTHORIZE_DIAGNOSTIC_ADMINプロシージャを使用して診断ビューおよび表の問合せの認可を付与されたユーザーから認可を取り消します。
- UNAUTHORIZE_MAINTENANCE_USERプロシージャ
  UNAUTHORIZE_MAINTENANCE_USERプロシージャは、Oracle Database Vault環境で情報ライフサイクル管理(ILM)操作を実行するための認可を与えられたユーザーから権限を取り消します。
- UNAUTHORIZE_PREPROCESSORプロシージャ
  UNAUTHORIZE_PREPROCESSORプロシージャは、外部表からプリプロセッサ・プログラムを実行する認可をユーザーから取り消します。
- UNAUTHORIZE_PROXY_USERプロシージャ
  UNAUTHORIZE_PROXY_USERプロシージャは、DBMS_MACADM.AUTHORIZE_PROXY_USERプロシージャによってプロキシの認可を付与されたユーザーから認可を取り消します。

- UNAUTHORIZE_SCHEDULER_USERプロシージャ
  UNAUTHORIZE_SCHEDULER_USERプロシージャは、AUTHORIZE_SCHEDULER_USERプロシージャによって付与された権限を取り消します。
- UNAUTHORIZE_TTS_USERプロシージャ
  UNAUTHORIZE_TTS_USERプロシージャは、Oracle Data Pumpトランスポータブル表領域操作を実行する認可を以前に付与されたユーザーを認可から削除します。

親トピック: Oracle Database Vaultの一般 管理 API

21.1.1 ADD_APP_EXCEPTIONプロシージャ

ADD_APP_EXCEPTIONプロシージャにより、共通ユーザーまたはパッケージはローカル・スキーマにアクセスできるようになります。

共通ユーザーに対してプラガブル・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するようにDatabase Vault操作の制御を構成する場合は、この手順を使用します。この手順はコンテナ全体に適用されるため、CDBルートから実行する必要があります。例外がパッケージに対するものである場合、指定したパッケージの所有者の文はローカル・スキーマにアクセスできます。

構文

```
DBMS_MACADM.ADD_APP_EXCEPTION(
owner          IN VARCHAR2,
  package_name   IN VARCHAR2);
```

パラメータ

表21-1 ADD_APP_EXCEPTION

パラメータ	説明
owner	例外として追加するユーザーの名前 使用可能な共通ユーザーのリストを検索するには、DBA_USERS データ・ディクショナリ・ビューのUSERNAME および COMMON 列を問い合せます。
package_name	ユーザー・アカウント全体ではなくパッケージを指定する場合に例外として追加するパッケージの名前。このパッケージは、owner パラメータで指定したユーザーが所有する必要があります。特定のパッケージではなく、スキーマ全体の例外を作成する場合、package_name パラメータに'%'を指定します。

例

```
EXEC DBMS_MACADM.ADD_APP_EXCEPTION ('C##HR_ADMIN', '%'); --Applies to the user
c##hr_admin
EXEC DBMS_MACADM.ADD_APP_EXCEPTION('C##HR_ADMIN', 'validateHRdata'); --Applies to the
package validateHRdata
```

関連トピック

- 例外リストへの共通ユーザーおよびパッケージの追加
- ENABLE_APP_PROTECTIONプロシージャ
- DISABLE_APP_PROTECTIONプロシージャ

- DELETE_APP_EXCEPTIONプロシージャ

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.2 ADD_NLS_DATAプロシージャ

ADD_NLS_DATAプロシージャは、Oracle Database Vaultに新しい言語を追加します。

構文

```
DBMS_MACADM.ADD_NLS_DATA(
language       IN VARCHAR );
```

パラメータ

表21-2 ADD_NLS_DATA

パラメータ	説明
language	次のいずれかの設定を入力します。(このパラメータでは大/小文字は区別されません。) ● ENGLISH ● GERMAN ● SPANISH ● FRENCH ● ITALIAN ● JAPANESE ● KOREAN ● BRAZILIAN PORTUGUESE ● SIMPLIFIED CHINESE ● TRADITIONAL CHINESE

例

```
EXEC DBMS_MACADM.ADD_NLS_DATA('french');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.3 AUTH_DATAPUMP_CREATE_USERプロシージャ

AUTH_DATAPUMP_CREATE_USERプロシージャは、Oracle Data Pumpユーザーに対して、Oracle Data Pumpのインポート操作中のユーザーの作成を認可します。

このプロシージャは、impdpユーティリティにのみ適用されます。

構文

```
DBMS_MACADM.AUTH_DATAPUMP_CREATE_USER(
```

```
uname       IN VARCHAR2);
```

パラメータ

表21-3 AUTH_DATAPUMP_CREATE_USER

パラメータ	説明
uname	インポート操作中にユーザーを作成する必要がある Oracle Data Pump ユーザーの名前。 ユーザーの現在のステータスを検索するには、DBA_DV_DATAPUMP_AUTH データ・ディクショナリ・ビューを問い合せます。

例

```
EXEC DBMS_MACADM.AUTH_DATAPUMP_CREATE_USER('DP_MGR');
```

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.4 AUTH_DATAPUMP_GRANTプロシージャ

AUTH_DATAPUMP_GRANTプロシージャは、Oracle Data Pumpユーザーに対して、Oracle Data Pumpのインポート操作中のOracle Database Vaultで保護されたロールおよびシステム権限の付与を認可します。

このプロシージャは、impdpユーティリティにのみ適用されます。この認可では、DV_OWNER、DV_ADMIN、DV_MONITORなど、Oracle Database Vaultのロールはカバーされないことに注意してください。

構文

```
DBMS_MACADM.AUTH_DATAPUMP_GRANT(
uname       IN VARCHAR2);
```

パラメータ

表21-4 AUTH_DATAPUMP_GRANT

パラメータ	説明
user_name	インポート操作中にロールおよび権限をユーザーに付与する必要がある Oracle Data Pump ユーザーの名前。 ユーザーの現在のステータスを検索するには、DBA_DV_DATAPUMP_AUTH データ・ディクショナリ・ビューを問い合せます。

例

```
EXEC DBMS_MACADM.AUTH_DATAPUMP_GRANT('DP_MGR');
```

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.5 AUTH_DATAPUMP_GRANT_ROLEプロシージャ

AUTH_DATAPUMP_GRANT_ROLEプロシージャは、Oracle Data Pumpユーザーに対して、Oracle Data Pumpのインポート操作中の特定のロールの付与を認可します。

このプロシージャは、impdpユーティリティにのみ適用されます。

構文

```
DBMS_MACADM.AUTH_DATAPUMP_GRANT_ROLE(
uname       IN VARCHAR2,
role        IN VARCHAR2 DEFAULT %);
```

パラメータ

表21-5 AUTH_DATAPUMP_GRANT_ROLE

パラメータ	説明
uname	インポート操作中に特定のロールをユーザーに付与する必要がある Oracle Data Pump ユーザーの名前。 ユーザーの現在のステータスを検索するには、DBA_DV_DATAPUMP_AUTH データ・ディクショナリ・ビューを問い合せます。
role	ユーザーに付与するロール。DV_OWNER、DV_ADMIN、DV_MONITOR など、Oracle Database Vault のロールは指定しないでください。この値を省略するか、%を指定すると、インポート操作中にユーザーには(Oracle Database Vault のロール以外の)ロールの付与が認可されます。ユーザーが DBMS_MACADM.AUTH_DATAPUMP_GRANT プロシージャで認可されている場合、またはユーザーに特定のロールを付与する権限がある場合、ユーザーは引き続きこれらのロールを付与できます。

例

```
EXEC DBMS_MACADM.AUTH_DATAPUMP_GRANT_ROLE('DP_MGR', 'DBA');
```

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.6 AUTH_DATAPUMP_GRANT_SYSPRIVプロシージャ

AUTH_DATAPUMP_GRANT_SYSPRIVプロシージャは、Oracle Data Pumpユーザーに対して、Oracle Data Pumpのインポート操作中のシステム権限の付与を認可します。

プロシージャは、IMPDPユーティリティにのみ適用されます。

構文

```
DBMS_MACADM.AUTH_DATAPUMP_GRANT_SYSPRIV(
```

```
uname       IN VARCHAR2);
```

パラメータ

表21-6 AUTH_DATAPUMP_GRANT_SYSPRIV

パラメータ	説明
uname	IMPDP 操作中にユーザーにシステム権限を付与する必要がある Oracle Data Pump ユーザーの名前。 ユーザーの現在のステータスを検索するには、DBA_DV_DATAPUMP_AUTH データ・ディクショナリ・ビューを問い合せます。

例

```
EXEC DBMS_MACADM.AUTH_DATAPUMP_GRANT_SYSPRIV('DP_MGR');
```

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.7 AUTHORIZE_DATAPUMP_USERプロシージャ

AUTHORIZE_DATAPUMP_USERプロシージャは、Oracle Database Vaultが有効な場合に、ユーザーがOracle Data Pump操作を実行することを認可します。

expdpとimpdpの両方のユーティリティに適用されます。

構文

```
DBMS_MACADM.AUTHORIZE_DATAPUMP_USER(
user_name       IN VARCHAR2,
schema_name     IN VARCHAR2 DEFAULT NULL,
table_name      IN VARCHAR2 DEFAULT NULL);
```

パラメータ

表21-7 AUTHORIZE_DATAPUMP_USER

パラメータ	説明
user_name	権限を付与する Oracle Data Pump ユーザーの名前。 Oracle Data Pump の使用権限(EXP_FULL_DATABASE および IMP_FULL_DATABASE ロール)を持つユーザーのリストを検索するには、次のように DBA_ROLE_PRIVS データ・ディクショナリ・ビューに問い合せます。 SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS WHERE GRANTED_ROLE LIKE '%FULL%'
schema_name	Oracle Data Pump ユーザーがエクスポートまたはインポートする必要のあるデータベース・スキー

パラメータ	説明
	マ名。このパラメータを省略すると、データベース内のすべてのスキーマをエクスポートおよびインポートするグローバル権限がユーザーに付与されます。この場合、ユーザーに DV_OWNER ロールが付与されていることを確認します。
table_name	schema_name パラメータで指定されたスキーマ内の表の名前。このパラメータを省略した場合、指定したユーザーは、schema_name パラメータで指定されたスキーマ内のすべての表をエクスポートおよびインポートできます。

例

```
EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('DP_MGR');
EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('DP_MGR', 'HR');
EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('DP_MGR', 'HR', 'EMPLOYEES');
```

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.8 AUTHORIZE_DBCAPTUREプロシージャ

AUTHORIZE_DBCAPTUREプロシージャは、Oracle Database Replayのワークロード取得操作を実行する認可をユーザーに付与します。

この認可を与えられているユーザーについて情報を確認するには、DBA_DV_DBCAPTURE_AUTHデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.AUTHORIZE_DBCAPTURE(
uname       IN VARCHAR2);
```

パラメータ

表21-8 AUTHORIZE_DBCAPTURE

パラメータ	説明
uname	Database Replay のワークロード取得認可を付与するユーザーの名前

例21-1 例

```
EXEC DBMS_MACADM.AUTHORIZE_DBCAPTURE('PFITCH');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.9 AUTHORIZE_DBREPLAYプロシージャ

AUTHORIZE_DBREPLAYプロシージャは、Oracle Database Replayのワークロード・リプレイ操作を実行する認可をユーザーに付与します。

この認可を与えられているユーザーについて情報を確認するには、DBA_DV_DBREPLAY_AUTHデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.AUTHORIZE_DBREPLAY(
uname       IN VARCHAR2);
```

パラメータ

表21-9 AUTHORIZE_DBREPLAY

パラメータ	説明
uname	Database Replay のワークロード・リプレイ認可を付与するユーザーの名前

例21-2 例

```
EXEC DBMS_MACADM.AUTHORIZE_DBREPLAY('PFITCH');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.10 AUTHORIZE_DDLプロシージャ

AUTHORIZE_DDLプロシージャは、指定したスキーマに対してデータ定義言語(DDL)文を実行する認可をユーザーに付与します。

DDLの認可により、権限受領者は、レルムを認可されたユーザー、またはOracle Database Vaultロールを付与されたユーザーに対してDDL操作を実行できます。ただし、DDL認可では、権限受領者がレルムで保護されたスキーマに対してDDL操作を実行することはできません。このような操作を有効化するには、レルムに対してユーザーを認可する必要があります。

この認可を与えられているユーザーについて情報を確認するには、DBA_DV_DDL_AUTHデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.AUTHORIZE_DDL(
user_name       IN VARCHAR2,
schema_name     IN VARCHAR2);
```

パラメータ

表21-10 AUTHORIZE_DDL

パラメータ	説明
user_name	DDL 認可を付与するユーザーの名前。
schema_name	ユーザーが DDL 文を実行するデータベース・スキーマの名前。すべてのスキーマを指定するには%を入力します。

例

次の例では、ユーザーpsmithはすべてのスキーマでDDL文を実行できます。

```
EXEC DBMS_MACADM.AUTHORIZE_DDL('psmith', '%');
```

この例では、ユーザーpsmithはHRスキーマでのみDDL文を実行できます。

```
EXEC DBMS_MACADM.AUTHORIZE_DDL('psmith', 'HR');
```

関連トピック

- Oracle Database VaultでのDDL操作の実行

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.11 AUTHORIZE_DIAGNOSTIC_ADMINプロシージャ

AUTHORIZE_DIAGNOSTIC_ADMINプロシージャは、診断ビューおよび表を問い合せる認可をユーザーに付与します。

これらのビューおよび表を次に示します。

ビューおよび表V$	ビューおよび表X$
V$DIAG_OPT_TRACE_RECORDS	X$DBGTFOPTT
V$DIAG_SESS_OPT_TRACE_RECORDS	X$DBGTFSOPTT
V$DIAG_TRACE_FILE_CONTENTS	X$DBGTFVIEW

この認可がないと、ユーザーがこれらの表およびビューを問い合せた場合、値は返されません。

構文

```
DBMS_MACADM.AUTHORIZE_DIAGNOSTIC_ADMIN(
uname       IN VARCHAR2);
```

パラメータ

表21-11 AUTHORIZE_DIAGNOSTIC_ADMIN

パラメータ	説明
uname	権限を付与するユーザーの名前。

例

```
EXEC DBMS_MACADM.AUTHORIZE_DIAGNOSTIC_ADMIN('PFITCH');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.12 AUTHORIZE_MAINTENANCE_USERプロシージャ

AUTHORIZE_MAINTENANCE_USERプロシージャは、Oracle Database Vault環境で情報ライフサイクル管理(ILM)操作を実行するための認可をユーザーに与えます。

この認可を与えられているユーザーについて情報を確認するには、DBA_DV_MAINTENANCE_AUTHビューに問い合せます。

構文

```
DBMS_MACADM.AUTHORIZE_MAINTENANCE_USER(
```

```
 uname         IN VARCHAR2,
 sname         IN VARCHAR2 DEFAULT NULL,
 objname       IN VARCHAR2 DEFAULT NULL,
 objtype       IN VARCHAR2 DEFAULT NULL,
 action        IN VARCHAR2 DEFAULT NULL);
```

パラメータ

表21-12 AUTHORIZE_MAINTENANCE_USER

パラメータ	説明
uname	権限を付与するユーザーの名前
sname	メンテナンス操作を実行するデータベース・スキーマの名前。すべてのスキーマを指定するには%を入力します。
objname	メンテナンス操作を実行する、sname パラメータで指定されているスキーマ内のオブジェクトの名前(表の名前など)。
objtype	table、index、tablespace など、objname オブジェクトのタイプ。
action	メンテナンス・アクション。情報ライフサイクル管理の場合は ilm と入力します。

例

次の例では、ユーザーpsmithに、HR.EMPLOYEES表のILM機能を管理するためのDatabase Vault認可を与えることができます。

```
BEGIN
 DBMS_MACADM.AUTHORIZE_MAINTENANCE_USER (
  uname        => 'psmith',
  sname        => 'HR',
  objname      => 'EMPLOYEES',
  objtype      => 'TABLE',
  action       => 'ILM');
END;
/
```

関連トピック

- Oracle Database Vaultでの情報ライフサイクル管理の使用

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.13 AUTHORIZE_PREPROCESSORプロシージャ

AUTHORIZE_PREPROCESSORプロシージャは、外部表からプリプロセッサ・プログラムを実行する認可をユーザーに付与します。

この認可を与えられているユーザーについて情報を確認するには、DBA_DV_PREPROCESSOR_AUTHデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.AUTHORIZE_PREPROCESSOR(
uname       IN VARCHAR2);
```

パラメータ

表21-13 AUTHORIZE_PREPROCESSOR

パラメータ	説明
uname	外部表からプリプロセッサ・プログラムを実行する認可を付与するユーザーの名前

例21-3 例

```
EXEC DBMS_MACADM.AUTHORIZE_PREPROCESSOR('PFITCH');
```

関連トピック

- Oracle Database Vaultでのプリプロセッサ・プログラムの実行
- DBA_DV_PREPROCESSOR_AUTHビュー

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.14 AUTHORIZE_PROXY_USERプロシージャ

AUTHORIZE_PROXY_USERプロシージャは、プロキシ・ユーザーにデータベース認可がある場合、他のユーザー・アカウントをプロキシする認可をプロキシ・ユーザーに付与します。

たとえば、CREATE SESSION権限は、有効なデータベース認可です。

AUTHORIZE_PROXY_USERは、特定のユーザーが別のユーザーのプロキシとして接続できるかどうかを制御しません。その部分はGRANT CONNECT THROUGHによって制御されます。これは、DV_ACCTMGRロールを持つユーザーのみが発行できます。そのかわり、AUTHORIZE_PROXY_USERは、ターゲット・ユーザーが持つすべてのDatabase Vault認可をプロキシ・ユーザーが継承できるかどうかを制御します。たとえば、プロキシ・ユーザーhr_proxy_userがユーザーHRとして正常に接続しているとします。この場合、hr_proxy_userはHRとして、HRがアクセス権を持つすべてのオブジェクトにアクセスできます。ただし、ターゲット・オブジェクトがDatabase Vaultによって保護され、かつ、それに対するアクセスがHRに認可されている場合は、hr_proxy_userにHRのプロキシ認可が付与されている場合にのみ、hr_proxy_userはそのオブジェクトにアクセスできます。hr_proxy_userにHRのプロキシ認可が付与されていない場合は、HRとして接続した後でも、hr_proxy_userはHRがアクセスを認可されているDatabase Vaultで保護されたオブジェクトにはアクセスできません。

AUTHORIZE_PROXY_USERを使用して認可を与えられているユーザーについて情報を確認するには、DBA_DV_PROXY_AUTHビューに問い合せます。

構文

```
DBMS_MACADM.AUTHORIZE_PROXY_USER(
proxy_user    IN VARCHAR2,
user_name     IN VARCHAR2);
```

パラメータ

表21-14 AUTHORIZE_PROXY_USER

パラメータ	説明

パラメータ	説明
proxy_user	プロキシ・ユーザーの名前。
user_name	proxy_user ユーザーによってプロキシされるデータベース・ユーザーの名前。すべてのユーザーを指定するには%を入力します。

例

次の例では、プロキシ・ユーザーprestonはすべてのユーザーをプロキシできます。

```
EXEC DBMS_MACADM.AUTHORIZE_PROXY_USER('preston', '%');
```

この例では、プロキシ・ユーザーprestonはデータベース・ユーザーdkentのみをプロキシできます。

```
EXEC DBMS_MACADM.AUTHORIZE_PROXY_USER('preston', 'dkent');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.15 AUTHORIZE_SCHEDULER_USERプロシージャ

AUTHORIZE_SCHEDULER_USERプロシージャは、Oracle Database Vaultが有効な場合に、データベース・ジョブをスケジュールする権限をユーザーに付与します。

この権限は、データベース・ジョブのスケジュール権限を持つすべてのユーザーに適用されます。この権限には、CREATE JOB、CREATE ANY JOB、CREATE EXTERNAL JOB、EXECUTE ANY PROGRAM、EXECUTE ANY CLASS、MANAGE SCHEDULERが含まれます。

構文

```
DBMS_MACADM.AUTHORIZE_SCHEDULER_USER(
user_name       IN VARCHAR2,
schema_name     IN VARCHAR2 DEFAULT NULL);
```

パラメータ

表21-15 AUTHORIZE_SCHEDULER_USER

パラメータ	説明
user_name	権限を付与するユーザーの名前。 ジョブをスケジュールする権限(CREATE JOB および CREATE ANY JOB など)を持つユーザーのリストを検索するには、DBA_SYS_PRIVS データ・ディクショナリ・ビューの GRANTEE 列および PRIVILEGE 列を問い合せます。
schema_name	ジョブをスケジュールするデータベース・スキーマの名前。このパラメータを省略すると、データベース内のすべてのスキーマをスケジュールするグローバル権限がユーザーに付与されます。

例

次の例では、ユーザーJOB_MGRにすべてのスキーマでジョブを実行する権限を付与します。

```
EXEC DBMS_MACADM.AUTHORIZE_SCHEDULER_USER('JOB_MGR');
```

次の例では、ユーザーJOB_MGRにHRスキーマでのみジョブを実行する権限を付与します。

```
EXEC DBMS_MACADM.AUTHORIZE_SCHEDULER_USER('JOB_MGR', 'HR');
```

関連トピック

- Oracle Database VaultでのOracle Schedulerの使用

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.16 AUTHORIZE_TTS_USERプロシージャ

AUTHORIZE_TTS_USERプロシージャは、Oracle Database Vaultが有効な場合に、表領域に対してOracle Data Pumpトランスポータブル表領域操作を実行するようにユーザーを認可します。

EXPDPとIMPDPの両方のユーティリティに適用されます。

構文

```
DBMS_MACADM.AUTHORIZE_TTS_USER(
uname       IN VARCHAR2,
tsname      IN VARCHAR2);
```

パラメータ

表21-16 AUTHORIZE_TTS_USER

パラメータ	説明
uname	Oracle Data Pump トランスポータブル表領域操作を実行するために認可するユーザーの名前。 ユーザーとその現在の権限のリストを検索するには、DBA_SYS_PRIVS データ・ディクショナリ・ビューに問い合せます。
tsname	uname ユーザーがトランスポータブル表領域操作を実行する表領域の名前。 表領域のリストを検索するには、DBA_TABLESPACES データ・ディクショナリ・ビューに問い合せます。

例

```
EXEC DBMS_MACADM.AUTHORIZE_TTS_USER('PSMITH', 'HR_TS');
```

関連トピック

- Database VaultにおけるOracle Data Pumpの通常操作をユーザーまたはロールに認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.17 DELETE_APP_EXCEPTIONプロシージャ

DELETE_APP_EXCEPTIONプロシージャは、Database Vault操作の制御の例外リストから共通ユーザーまたは共通ユー

ザーのパッケージを削除します。

例外リストを使用すると、ユーザーまたはパッケージがローカルPDBデータにアクセスできます。例外リストからユーザーまたはパッケージを削除すると、ユーザーまたはパッケージがPDBローカル・データからブロックされます。

構文

```
DBMS_MACADM.DELETE_APP_EXCEPTION(
 owner          IN VARCHAR2,
  package_name  IN VARCHAR2);
```

パラメータ

表21-17 DELETE_APP_EXCEPTION

パラメータ	説明
owner	例外になることから削除するユーザーの名前
package_name	例外になることから削除するパッケージの名前

例

```
EXEC DBMS_MACADM.DELETE_APP_EXCEPTION ('C##HR_ADMIN'); --Applies to the user
c##hr_admin
EXEC DBMS_MACADM.DELETE_APP_EXCEPTION('C##HR_ADMIN', 'validateHRdata'); --Applies to
the package validateHRdata
```

関連トピック

- 例外リストへの共通ユーザーおよびパッケージの追加
- ADD_APP_EXCEPTIONプロシージャ
- ENABLE_APP_PROTECTIONプロシージャ
- DISABLE_APP_PROTECTIONプロシージャ

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.18 DISABLE_APP_PROTECTIONプロシージャ

DISABLE_APP_PROTECTIONプロシージャは、Database Vault操作の制御を無効にします。

構文

```
DBMS_MACADM.DISABLE_APP_PROTECTION(
pdb_name      IN VARCHAR2 DEFAULT NULL);
```

パラメータ

表21-18 DISABLE_APP_PROTECTION

パラメータ	説明
pdb_name	Database Vault 操作制御を無効にするプラガブル・データベース(PDB)の名前。この設定を省略すると、CDB 環境内のすべての PDB に適用されます。 使用可能な PDB のリストを検索するには、DBA_PDBS データ・ディクショナリ・ビューを問い合せま

パラメータ	説明
	す。

例

```
EXEC DBMS_MACADM.DISABLE_APP_PROTECTION; --Applies to all PDBs
EXEC DBMS_MACADM.DISABLE_APP_PROTECTION('hr_pdb'); --Applies to a specific PDB
```

関連トピック

- Database Vault操作の制御の無効化

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.19 DISABLE_DVプロシージャ

DISABLE_DVプロシージャは、Oracle Database Vaultを無効にします。

このプロシージャの実行後、データベースを再起動する必要があります。

構文

```
DBMS_MACADM.DISABLE_DV;
```

パラメータ

なし

例

```
EXEC DBMS_MACADM.DISABLE_DV;
```

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.20 DISABLE_DV_DICTIONARY_ACCTSプロシージャ

DISABLE_DV_DICTIONARY_ACCTSプロシージャでは、どのユーザーもDVSYSまたはDVFスキーマ・ユーザーとしてデータベースにログインできません。

これらの2つのアカウントはデフォルトでロックされます。DV_OWNERロールを付与されているユーザーのみがこのプロシージャを実行できます。ユーザーがDVSYSおよびDVFにログインできるかどうかのステータスを確認するには、DBA_DV_DICTIONARY_ACCTSデータ・ディクショナリ・ビューに問い合せます。セキュリティをさらに強化するには、このプロシージャを実行して、DVSYSおよびDVFスキーマの保護を改善します。無効化はすぐに行われるため、このプロシージャの実行後にデータベースを再起動する必要はありません。

構文

```
DBMS_MACADM.DISABLE_DV_DICTIONARY_ACCTS;
```

パラメータ

なし

例

```
EXEC DBMS_MACADM.DISABLE_DV_DICTIONARY_ACCTS;
```

関連トピック

- Oracle Database Vault監査証跡のアーカイブおよびパージ

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.21 DISABLE_DV_PATCH_ADMIN_AUDITプロシージャ

DISABLE_DV_PATCH_ADMIN_AUDITプロシージャは、DV_PATCH_ADMINロールを持つユーザーによるアクションのレルム、コマンド・ルールおよびルール・セットの監査を無効にします。

このプロシージャは、失敗したアクションではなく、このユーザーの成功したアクションを無効にします。DV_PATCH_ADMINユーザーでデータベース・パッチ操作が完了した後、このプロシージャを実行する必要があります。監査が有効になっているかどうかを判断するには、DBA_DV_PATCH_AUDITデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.DISABLE_DV_PATCH_ADMIN_AUDIT;
```

パラメータ

なし

例

```
EXEC DBMS_MACADM.DISABLE_DV_PATCH_ADMIN_AUDIT;
```

関連トピック

- DV_PATCH_ADMIN Database Vaultデータベース・パッチ・ロール
- ENABLE_DV_PATCH_ADMIN_AUDITプロシージャ

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.22 DISABLE_ORADEBUGプロシージャ

DISABLE_ORADEBUGプロシージャは、Oracle Database Vault環境でORADEBUGユーティリティの使用を無効にします。

無効化はすぐに行われるため、このプロシージャの実行後にデータベースを再起動する必要はありません。ORADEBUGユーティリティをDatabase Vaultで利用できるかどうかのステータスを確認するには、DVYS.DBA_DV_ORADEBUGデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.DISABLE_ORADEBUG;
```

パラメータ

なし

例

```
EXEC DBMS_MACADM.DISABLE_ORADEBUG;
```

関連トピック

- Oracle Database Vault環境でのORADEBUGユーティリティの使用

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.23 ENABLE_APP_PROTECTIONプロシージャ

ENABLE_APP_PROTECTIONプロシージャは、Database Vault操作の制御を有効にします。

構文

```
DBMS_MACADM.ENABLE_APP_PROTECTION(
pdb_name      IN VARCHAR2 DEFAULT NULL);
```

パラメータ

表21-19 ENABLE_APP_PROTECTION

パラメータ	説明
pdb_name	無効になった後、1 つの PDB で Database Vault 操作の制御を再度有効にできるようになります。デフォルトでは、pdb_name 設定を省略してから、すべての PDB で操作の制御を有効にします。 使用可能な PDB のリストを検索するには、DBA_PDBS データ・ディクショナリ・ビューを問い合せます。

例

```
EXEC DBMS_MACADM.ENABLE_APP_PROTECTION; --Applies to all PDBs
EXEC DBMS_MACADM.ENABLE_APP_PROTECTION('hr_pdb'); --Applies to a specific PDB
```

関連トピック

- Database Vault操作の制御の有効化

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.24 ENABLE_DVプロシージャ

ENABLE_DVプロシージャは、Oracle Database VaultおよびOracle Label Securityを有効にします。

アプリケーション・コンテナでDBMS_MACADM.ENABLE_DVを実行する場合は、アプリケーション・アクションの外部のアプリケーション・コンテナで実行する必要があります。

このプロシージャの実行後、データベースを再起動する必要があります。

構文

```
DBMS_MACADM.ENABLE_DV(
 strict_mode  IN VARCHAR2 DEFAULT);
```

パラメータ

表21-20 ENABLE_DV

パラメータ	説明
strict_mode	マルチテナント環境では、次のいずれかのモードを指定します。 ● n は、通常モードを指定します。このモードでは、PDB で Database Vault を有効または無効にできます。(デフォルト) ● y は、厳密モードを指定します。このモードでは、Database Vault が有効になっていない PDB が、それらの Database Vault を有効にしてそれらの PDB を再起動するまで制限モードになります。 マルチテナント環境ですべての PDB にこの設定を適用するには、CDB ルートで DBMS_MACADM.ENABLE_DV プロシージャを実行します。アプリケーション・コンテナ内のすべての PDB にそれを適用するには、アプリケーション・ルートでプロシージャを実行します。 非マルチテナント環境では、このパラメータは省略します。

例

次の例では、Oracle Database Vaultを通常モードで有効にします。

```
EXEC DBMS_MACADM.ENABLE_DV;
```

この例では、マルチテナント環境で、厳密モードでOracle Database Vaultを有効にします。

```
EXEC DBMS_MACADM.ENABLE_DV (strict_mode => 'y');
```

関連トピック

- Oracle Database Vaultの無効化および有効化

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.25 ENABLE_DV_DICTIONARY_ACCTSプロシージャ

ENABLE_DV_DICTIONARY_ACCTSプロシージャにより、ユーザーはDVSYSまたはDVFユーザーとしてデータベースにログインできます。

デフォルトで、DVSYSおよびDVFアカウントはロックされています。

DV_OWNERロールを付与されているユーザーのみがこのプロシージャを実行できます。ユーザーがDVSYSおよびDVFにログインできるかどうかのステータスを確認するには、DBA_DV_DICTIONARY_ACCTSデータ・ディクショナリ・ビューに問い合せます。セキュリティをさらに強化するには、DVSYSおよびDVFスキーマの保護を改善する必要がある場合にこのプロシージャを実行するのみです。有効化はすぐに行われるため、このプロシージャの実行後にデータベースを再起動する必要はありません。

構文

```
DBMS_MACADM.ENABLE_DV_DICTIONARY_ACCTS;
```

パラメータ

なし

例

```
EXEC DBMS_MACADM.ENABLE_DV_DICTIONARY_ACCTS;
```

関連トピック

- Oracle Database Vault監査証跡のアーカイブおよびパージ

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.26 ENABLE_DV_PATCH_ADMIN_AUDITプロシージャ

ENABLE_DV_PATCH_ADMIN_AUDITプロシージャは、DV_PATCH_ADMINロールを持つユーザーによるアクションのレルム、コマンド・ルールおよびルール・セットの監査を有効にします。

このプロシージャはパッチのアップグレード時にこれらのユーザーのアクションを監査するように設計されています。この監査が有効になっているかどうかを確認するには、DBA_DV_PATCH_AUDITデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.ENABLE_DV_PATCH_ADMIN_AUDIT;
```

パラメータ

なし

例

```
EXEC DBMS_MACADM.ENABLE_DV_PATCH_ADMIN_AUDIT;
```

関連トピック

- DV_PATCH_ADMIN Database Vaultデータベース・パッチ・ロール
- DISABLE_DV_PATCH_ADMIN_AUDITプロシージャ

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.27 ENABLE_ORADEBUGプロシージャ

ENABLE_ORADEBUGプロシージャは、Oracle Database Vault環境でORADEBUGユーティリティの使用を有効にします。

有効化はすぐに行われるため、このプロシージャの実行後にデータベースを再起動する必要はありません。ORADEBUGユーティリティをDatabase Vaultで利用できるかどうかのステータスを確認するには、DVYS.DBA_DV_ORADEBUGデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.ENABLE_ORADEBUG;
```

パラメータ

なし

例

```
EXEC DBMS_MACADM.ENABLE_ORADEBUG;
```

関連トピック

- Oracle Database Vault環境でのORADEBUGユーティリティの使用

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.28 UNAUTH_DATAPUMP_CREATE_USERプロシージャ

UNAUTH_DATAPUMP_CREATE_USERプロシージャは、Oracle Data Pumpユーザーから、Oracle Data Pumpのインポート操作中にユーザーを作成する認可を削除します。

このプロシージャは、impdpユーティリティにのみ適用されます。

構文

```
DBMS_MACADM.UNAUTH_DATAPUMP_CREATE_USER(
uname       IN VARCHAR2);
```

パラメータ

表21-21 UNAUTH_DATAPUMP_CREATE_USER

パラメータ	説明
uname	認可を削除する必要がある Oracle Data Pump ユーザーの名前。 ユーザーの現在のステータスを検索するには、DBA_DV_DATAPUMP_AUTH データ・ディクショナリ・ビューを問い合せます。

例

```
EXEC DBMS_MACADM.UNAUTH_DATAPUMP_CREATE_USER('DP_MGR');
```

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.29 UNAUTH_DATAPUMP_GRANTプロシージャ

UNAUTH_DATAPUMP_GRANTプロシージャは、Oracle Data Pumpユーザーから、Oracle Data Pumpのインポート操作中にOracle Database Vaultで保護されたロールおよびシステム権限を付与する認可を削除します。

このプロシージャは、impdpユーティリティにのみ適用されます。

構文

```
DBMS_MACADM.UNAUTH_DATAPUMP_GRANT(
uname       IN VARCHAR2);
```

パラメータ

表21-22 UNAUTH_DATAPUMP_GRANT

パラメータ	説明

パラメータ	説明
user_name	認可を削除する必要がある Oracle Data Pump ユーザーの名前。 ユーザーの現在のステータスを検索するには、DBA_DV_DATAPUMP_AUTH データ・ディクショナリ・ビューを問い合せます。

例

```
EXEC DBMS_MACADM.UNAUTH_DATAPUMP_GRANT('DP_MGR');
```

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.30 UNAUTH_DATAPUMP_GRANT_ROLEプロシージャ

UNAUTH_DATAPUMP_GRANT_ROLEプロシージャは、Oracle Data Pumpユーザーから、Oracle Data Pumpのインポート操作中に特定のロールを付与する認可を削除します。

このプロシージャは、impdpユーティリティにのみ適用されます。

構文

```
DBMS_MACADM.UNAUTH_DATAPUMP_GRANT_ROLE(
uname       IN VARCHAR2,
role        IN VARCHAR2 DEFAULT %);
```

パラメータ

表21-23 UNAUTH_DATAPUMP_GRANT_ROLE

パラメータ	説明
uname	認可を削除する必要がある Oracle Data Pump ユーザーの名前。 ユーザーの現在のステータスを検索するには、DBA_DV_DATAPUMP_AUTH データ・ディクショナリ・ビューを問い合せます。
role	ユーザーにインポート操作中の付与が認可されているロール。DV_OWNER、DV_ADMIN、DV_MONITOR など、Oracle Database Vault のロールは指定しないでください。この値を省略すると、ユーザーにはインポート中のロールの付与は認可されません。

例

```
EXEC DBMS_MACADM.UNAUTH_DATAPUMP_GRANT_ROLE('DP_MGR', 'DBA');
```

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.31 UNAUTH_DATAPUMP_GRANT_SYSPRIVプロシージャ

UNAUTH_DATAPUMP_GRANT_SYSPRIVプロシージャは、Oracle Data Pumpユーザーから、Oracle Data Pumpのインポート操作中にシステム権限を付与する認可を削除します。

このプロシージャは、impdpユーティリティのみに適用されます。

構文

```
DBMS_MACADM.UNAUTH_DATAPUMP_GRANT_SYSPRIV(
uname       IN VARCHAR2);
```

パラメータ

表21-24 UNAUTH_DATAPUMP_GRANT_SYSPRIV

パラメータ	説明
uname	認可を削除する必要がある Oracle Data Pump ユーザーの名前。 ユーザーの現在のステータスを検索するには、DBA_DV_DATAPUMP_AUTH データ・ディクショナリ・ビューを問い合せます。

例

```
EXEC DBMS_MACADM.UNAUTH_DATAPUMP_GRANT_SYSPRIV('DP_MGR');
```

関連トピック

- ユーザーまたはロールへのData Pumpの通常エクスポート操作および通常インポート操作の認可

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.32 UNAUTHORIZE_DATAPUMP_USERプロシージャ

UNAUTHORIZE_DATAPUMP_USERプロシージャは、AUTHORIZE_DATAPUMP_USERプロシージャによって付与された権限を取り消します。

このプロシージャを実行する場合は、同等のAUTHORIZE_DATAPUMP_USERプロシージャと設定が完全に一致することを確認してください。

たとえば、次の2つのプロシージャはパラメータが一致するため、機能します。

```
EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('DP_MGR');
EXEC DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER('DP_MGR');
```

しかし、次のプロシージャはパラメータが一致しないため、UNAUTHORIZE_DATAPUMP_USERプロシージャは機能しません。

```
EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('JSMITH');
EXEC DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER('JSMITH', 'HR');
```

構文

```
DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER(
user_name       IN VARCHAR2,
schema_name     IN VARCHAR2 DEFAULT NULL,
table_name      IN VARCHAR2 DEFAULT NULL);
```

パラメータ

表21-25 UNAUTHORIZE_DATAPUMP_USER

パラメータ	説明
user_name	権限を取り消す Oracle Data Pump ユーザーの名前。 AUTHORIZE_DATAPUMP_USER プロシージャのユーザーと認可のリストを確認するには、次のように DBA_DV_DATAPUMP_AUTH データ・ディクショナリ・ビューに問い合せます。 SELECT * FROM DBA_DV_DATAPUMP_AUTH;
schema_name	Oracle Data Pump ユーザーがエクスポートまたはインポート権限を付与されているデータベース・スキーマ名。
table_name	schema name パラメータで指定されたスキーマ内の表の名前。

例

```
EXEC DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER('JSMITH');
EXEC DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER('JSMITH', 'HR');
EXEC DBMS_MACADM.UNAUTHORIZE_DATAPUMP_USER('JSMITH', 'HR', 'SALARY');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.33 UNAUTHORIZE_DBCAPTUREプロシージャ

UNAUTHORIZE_DBCAPTUREプロシージャは、Oracle Database Replayのワークロード取得操作を実行する認可をユーザーから取り消します。

この認可を与えられているユーザーについて情報を確認するには、DBA_DV_DBCAPTURE_AUTHデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.UNAUTHORIZE_DBCAPTURE(
uname       IN VARCHAR2);
```

パラメータ

表21-26 UNAUTHORIZE_DBCAPTURE

パラメータ	説明
uname	Database Replay のワークロード取得認可を取り消すユーザーの名前

例21-4 例

```
EXEC DBMS_MACADM.UNAUTHORIZE_DBCAPTURE('PFITCH');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.34 UNAUTHORIZE_DBREPLAYプロシージャ

UNAUTHORIZE_DBREPLAYプロシージャは、Oracle Database Replayのワークロード・リプレイ操作を実行する認可をユーザーから取り消します。

この認可を与えられているユーザーについて情報を確認するには、DBA_DV_DBREPLAY_AUTHデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.UNAUTHORIZE_DBREPLAY(
uname       IN VARCHAR2);
```

パラメータ

表21-27 UNAUTHORIZE_DBREPLAY

パラメータ	説明
uname	Database Replay のワークロード・リプレイ認可を取り消すユーザーの名前

例21-5 例

```
EXEC DBMS_MACADM.UNAUTHORIZE_DBREPLAY('PFITCH');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.35 UNAUTHORIZE_DDLプロシージャ

UNAUTHORIZE_DDLプロシージャは、DBMS_MACDM.AUTHORIZE_DDLプロシージャを使用してDDL文を実行する認可を付与されたユーザーから認可を取り消します。

この認可を与えられているユーザーについて情報を確認するには、DBA_DV_DDL_AUTHデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.UNAUTHORIZE_DDL(
user_name       IN VARCHAR2,
schema_name     IN VARCHAR2);
```

パラメータ

表21-28 UNAUTHORIZE_DDL

パラメータ	説明
user_name	DDL 認可を取り消すユーザーの名前。
schema_name	ユーザーが DDL 文を実行するデータベース・スキーマの名前。すべてのスキーマを指定するには%を入力します。

例

次の例では、すべてのスキーマについて、DDL文実行の認可をpsmithユーザーから取り消します。

```
EXEC DBMS_MACADM.UNAUTHORIZE_DDL('psmith', '%');
```

この例では、HRスキーマについてのみ、DDL文実行の認可をpsmithユーザーから取り消します。

```
EXEC DBMS_MACADM.UNAUTHORIZE_DDL('psmith', 'HR');
```

関連トピック

- Oracle Database VaultでのDDL操作の実行

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.36 UNAUTHORIZE_DIAGNOSTIC_ADMINプロシージャ

UNAUTHORIZE_DIAGNOSTIC_ADMINプロシージャは、DBMS_MACADM.AUTHORIZE_DIAGNOSTIC_ADMINプロシージャを使用して診断ビューおよび表の問合せの認可を付与されたユーザーから認可を取り消します。

これらのビューおよび表を次に示します。

ビューおよび表V$	ビューおよび表X$
V$DIAG_OPT_TRACE_RECORDS	X$DBGTFOPTT
V$DIAG_SESS_OPT_TRACE_RECORDS	X$DBGTFSOPTT
V$DIAG_TRACE_FILE_CONTENTS	X$DBGTFVIEW

この認可がないと、ユーザーがこれらの表およびビューを問い合せた場合、値は返されません。

構文

```
DBMS_MACADM.UNAUTHORIZE_DIAGNOSTIC_ADMIN(
uname       IN VARCHAR2);
```

パラメータ

表 21-29 UNAUTHORIZE_DIAGNOSTIC_ADMIN

パラメータ	説明
uname	認可を取り消すユーザーの名前。

例

```
EXEC DBMS_MACADM.UNAUTHORIZE_DIAGNOSTIC_ADMIN('PFITCH');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.37 UNAUTHORIZE_MAINTENANCE_USERプロシージャ

UNAUTHORIZE_MAINTENANCE_USERプロシージャは、Oracle Database Vault環境で情報ライフサイクル管理(ILM)操作を実行するための認可を与えられたユーザーから権限を取り消します。

ILM認可の設定について情報を確認するには、DBA_DV_MAINTENANCE_AUTHビューに問い合せます。

このプロシージャを実行する場合は、同等のAUTHORIZE_MAINTENANCE_USERプロシージャと設定が完全に一致すること

を確認してください。

たとえば、次の2つのプロシージャは、パラメータ設定が一致するため機能します。

```
EXEC DBMS_MACADM.AUTHORIZE_MAINTENANCE_USER('psmith', 'OE', 'ORDERS', 'TABLE',
'ILM');
EXEC DBMS_MACADM.UNAUTHORIZE_MAINTENANCE_USER('psmith', 'OE', 'ORDERS', 'TABLE',
'ILM');
```

しかしながら、次の2つの文は、設定が一致しないため失敗します。

```
EXEC DBMS_MACADM.AUTHORIZE_MAINTENANCE_USER('psmith', 'OE', 'ORDERS', 'TABLE',
'ILM');
EXEC DBMS_MACADM.UNAUTHORIZE_MAINTENANCE_USER('psmith', '%', '%', '%', 'ILM');
```

構文

```
DBMS_MACADM.UNAUTHORIZE_MAINTENANCE_USER(
 uname         IN VARCHAR2,
 sname         IN VARCHAR2 DEFAULT NULL,
 objname       IN VARCHAR2 DEFAULT NULL,
 objtype       IN VARCHAR2 DEFAULT NULL,
 action        IN VARCHAR2 DEFAULT NULL);
```

パラメータ

表21-30 UNAUTHORIZE_MAINTENANCE_USER

パラメータ	説明
uname	認可を取り消すユーザーの名前
sname	メンテナンス操作を実行するデータベース・スキーマの名前。すべてのスキーマを指定するには%を入力します。
objname	メンテナンス操作を実行する、sname パラメータで指定されているスキーマ内のオブジェクトの名前(表の名前など)。
objtype	table、index、tablespace など、objname オブジェクトのタイプ。
action	メンテナンス・アクション。情報ライフサイクル管理の場合は ilm と入力します。

例

次の例では、Database Vaultユーザーpsmithが、どのHRスキーマ・オブジェクトでもILM操作を実行できなくなるよう、権限を取り消されます。

```
BEGIN
 DBMS_MACADM.UNAUTHORIZE_MAINTENANCE_USER (
  uname       => 'psmith',
  sname       => 'HR',
  objname     => 'EMPLOYEES',
  objtype     => 'TABLE',
  action      => 'ILM');
END;
/
```

関連トピック

- Oracle Database Vaultでの情報ライフサイクル管理の使用

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.38 UNAUTHORIZE_PREPROCESSORプロシージャ

UNAUTHORIZE_PREPROCESSORプロシージャは、外部表からプリプロセッサ・プログラムを実行する認可をユーザーから取り消します。

この認可を与えられているユーザーについて情報を確認するには、DBA_DV_PREPROCESSOR_AUTHデータ・ディクショナリ・ビューに問い合せます。

構文

```
DBMS_MACADM.UNAUTHORIZE_PREPROCESSOR(
uname       IN VARCHAR2);
```

パラメータ

表21-31 UNAUTHORIZE_PREPROCESSOR

パラメータ	説明
uname	外部表からプリプロセッサ・プログラムを実行する認可を取り消すユーザーの名前

例21-6 例

```
EXEC DBMS_MACADM.UNAUTHORIZE_PREPROCESSOR('PFITCH');
```

関連トピック

- Oracle Database Vaultでのプリプロセッサ・プログラムの実行
- DBA_DV_PREPROCESSOR_AUTHビュー

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.39 UNAUTHORIZE_PROXY_USERプロシージャ

UNAUTHORIZE_PROXY_USERプロシージャは、DBMS_MACADM.AUTHORIZE_PROXY_USERプロシージャによってプロキシの認可を付与されたユーザーから認可を取り消します。

構文

```
DBMS_MACADM.UNAUTHORIZE_PROXY_USER(
proxy_user    IN VARCHAR2,
user_name     IN VARCHAR2);
```

パラメータ

表21-32 UNAUTHORIZE_PROXY_USER

パラメータ	説明
proxy_user	認可を取り消すプロキシ・ユーザーの名前。

パラメータ	説明
user_name	proxy_user ユーザーによってプロキシされたデータベース・ユーザーの名前。すべてのユーザーを指定するには%を入力します。

例

次の例では、すべてのユーザーをプロキシするプロキシの認可をユーザーprestonから取り消します。

```
DBMS_MACADM.UNAUTHORIZE_PROXY_USER('preston', '%');
```

この例では、データベース・ユーザーpsmithのみをプロキシするプロキシ認可をユーザーprestonから取り消します。

```
EXEC DBMS_MACADM.UNAUTHORIZE_PROXY_USER('preston', 'psmith');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.40 UNAUTHORIZE_SCHEDULER_USERプロシージャ

UNAUTHORIZE_SCHEDULER_USERプロシージャは、AUTHORIZE_SCHEDULER_USERプロシージャによって付与された権限を取り消します。

このプロシージャを実行する場合は、同等のAUTHORIZE_SCHEDULER_USERプロシージャと設定が完全に一致することを確認してください。たとえば、次の2つのプロシージャはパラメータが一致するため、機能します。

```
EXEC DBMS_MACADM.AUTHORIZE_SCHEDULER_USER('JOB_MGR');
EXEC DBMS_MACADM.UNAUTHORIZE_SCHEDULER_USER('JOB_MGR');
```

しかし、次のプロシージャはパラメータが一致しないため、UNAUTHORIZE_SCHEDULER_USERプロシージャは機能しません。

```
EXEC DBMS_MACADM.AUTHORIZE_SCHEDULER_USER('JOB_MGR');
EXEC DBMS_MACADM.UNAUTHORIZE_SCHEDULER_USER('JOB_MGR', 'HR');
```

構文

```
DBMS_MACADM.UNAUTHORIZE_SCHEDULER_USER
user_name      IN VARCHAR2,
schema_name    IN VARCHAR2 DEFAULT NULL);
```

パラメータ

表21-33 UNAUTHORIZE_SCHEDULER_USER

パラメータ	説明
user_name	権限を取り消す、ジョブ・スケジュール・ユーザーの名前。 AUTHORIZE_SCHEDULER_USER プロシージャのユーザーと認可のリストを確認するには、次のように DBA_DV_JOB_AUTH データ・ディクショナリ・ビューに問い合せます。 SELECT * FROM DBA_DV_JOB_AUTH;
schema_name	ユーザーにジョブ・スケジュール権限が付与されているデータベース・スキーマの名前。

例

```
EXEC DBMS_MACADM.UNAUTHORIZE_SCHEDULER_USER('JOB_MGR');
EXEC DBMS_MACADM.UNAUTHORIZE_SCHEDULER_USER('JOB_MGR', 'HR');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.1.41 UNAUTHORIZE_TTS_USERプロシージャ

UNAUTHORIZE_TTS_USERプロシージャは、Oracle Data Pumpトランスポータブル表領域操作を実行する認可を以前に付与されたユーザーを認可から削除します。

構文

```
DBMS_MACADM.UNAUTHORIZE_TTS_USER
uname       IN VARCHAR2,
tsname      IN VARCHAR2);
```

パラメータ

表21-34 UNAUTHORIZE_TTS_USER

パラメータ	説明
uname	Oracle Data Pump トランスポータブル表領域操作を実行する認可対象から削除するユーザーの名前。 ユーザーとその現在の権限のリストを検索するには、DBA_SYS_PRIVS データ・ディクショナリ・ビューに問い合せます。
tsname	トランスポータブル表領域操作で使用される表領域の名前。 表領域のリストを検索するには、DBA_TABLESPACES データ・ディクショナリ・ビューに問い合せます。

例

```
EXEC DBMS_MACADM.UNAUTHORIZE_TTS_USER('PSMITH', 'HR_TS');
```

親トピック: DBMS_MACADM一般システム・メンテナンスのプロシージャ

21.2 CONFIGURE_DVの一般システム・メンテナンス・プロシージャ

CONFIGURE_DVプロシージャは、最初の2つのOracle Databaseユーザー・アカウントを構成します。このアカウントには、DV_OWNERロールとDV_ACCTMGRロールがそれぞれ付与されます。

この構成のステータスは、DBA_DV_STATUSデータ・ディクショナリ・ビューを問い合せることで確認できます。CONFIGURE_DVプロシージャを実行する前に、2つのユーザー・アカウントを作成し、そのアカウントにCREATE SESSION権限を付与します。アカウントはローカルまたは共通のいずれかになります。共通のユーザー・アカウントを作成する場合、これらのユーザーに付与されるDatabase Vaultロールは、現在のプラガブル・データベース(PDB)にのみ適用されます。CONFIGURE_DVプロシージャに対してこれらのユーザー・アカウントを参照します。

CONFIGURE_DVプロシージャは、SYSスキーマにあります。Oracleでは、前のリリースで作成した既存のOracle Database

Vault構成スクリプトがこのリリースでも機能し続けるよう、シノニムDVSYS.CONFIGURE_DVが用意されています。

OracleデータベースでOracle Database Vaultを構成して有効化する準備が整ったら、CONFIGURE_DVプロシージャを1度実行するのみで構いません。このプロシージャの実行後、utlrp.sqlスクリプトとDBMS_MACADM.ENABLE_DVを順に実行して、登録プロセスを完了する必要があります。セキュリティを強化するために、ここで作成する2つのアカウントをバックアップ・アカウントとして使用し、日常使用するアカウントを追加で作成することをお薦めします。詳細は、「バックアップOracle Database Vaultアカウント」を参照してください。

CONFIGURE_DVの実行後に、入力した設定を変更する場合、ユーザーまたはDV_OWNERロールを持つ他のユーザーはDatabase Vaultを無効にしてから、SYSDBAまたはSYSOPER管理権限を持つ管理者にデータベースの再起動を依頼する必要があります。マルチテナント環境で作業している場合、通常、ユーザーSYSとして、CONTAINER句をALLに設定してDV_OWNERユーザーにDV_OWNERロールを付与します。

CONFIGURE_DVプロシージャを実行すると、DVSYSスキーマで表やパッケージの欠落などの問題が確認されます。問題が見つかった場合、ORA-47500「Database Vaultを構成できません。」エラーが生じます。この場合、CDB$ROOTおよび関連するPDBでcatmac.sqlを実行して、Oracle Database Vaultを再インストールできます。

同時に、CONFIGURE_DVプロシージャ、DBMS_MACADM.ENABLE_DVプロシージャおよびutlrp.sqlスクリプトは、Oracle Database Configuration Assistant(DBCA)を使用してOracle DatabaseでOracle Database Vaultを構成して有効化する場合の代替となるコマンドラインになるように設計されています。

OracleデータベースでOracle Database Vaultを構成して有効化する場合は、CONFIGURE_DVプロシージャをユーザーSYSとして実行する必要があります。

構文

```
CONFIGURE_DV
dvowner_uname          IN VARCHAR2,
dvacctmgr_uname        IN VARCHAR2,
  force_local_dvowner   IN BOOLEAN;
```

パラメータ

表21-35 CONFIGURE_DV

パラメータ	説明
dvowner_uname	Database Vault 所有者になるユーザーの名前。このユーザーには DV_OWNER ロールが付与されます。
dvacctmgr_uname	Database Vault アカウント・マネージャになるユーザーの名前。このユーザーには DV_ACCTMGR ロールが付与されます。この設定を省略した場合、dvowner_uname パラメータで指定されるユーザーが Database Vault アカウント・マネージャになり、このユーザーに DV_ACCTMGR ロールが付与されます。
force_local_dvowner	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 DV_OWNER (dvowner_unname ユーザー)にのみ適用されます。PDB で作成されたユーザーには適用されません。 ● TRUE を指定すると、dvowner_unname ユーザーの DV_OWNER ロール権限がルートに対してローカルになるように制限されます。

パラメータ	説明
	● デフォルト設定である FALSE によって、dvowner_unname ユーザーはルートに関連付けられているすべてのコンテナに対して DV_OWNER 権限を持つことができるようになります。

例

```
CREATE USER c##dbv_owner_root_backup IDENTIFIED BY password CONTAINER = CURRENT;
CREATE USER c##dbv_acctmgr_root_backup IDENTIFIED BY password CONTAINER = CURRENT;
GRANT CREATE SESSION TO c##dbv_owner_root_backup, c##dbv_acctmgr_root_backup;
BEGIN
 CONFIGURE_DV (
 dvowner_uname         => 'c##dbv_owner_root_backup',
 dvacctmgr_uname       => 'c##adbv_acctmgr_root_backup',
   force_local_dvowner   => TRUE);
 END;
/
```

関連トピック

- Oracle Database Vaultのアンインストール
- Oracle Database Vaultの再インストール
- Oracle Database Vaultの開始

親トピック: Oracle Database Vaultの一般 管理 API

22 Oracle Database VaultポリシーのAPI

DBMS_MACADM PL/SQLパッケージを使用してOracle Database Vaultポリシーを管理できます。

DV_OWNERロールまたはDV_ADMINロールを付与されているユーザーのみがこれらのプロシージャを使用できます。

- ADD_CMD_RULE_TO_POLICYプロシージャ
  ADD_COMMAND_RULE_TO_POLICYプロシージャでは、既存のコマンド・ルールをOracle Database Vaultポリシーに追加できます。
- ADD_OWNER_TO_POLICYプロシージャ
  ADD_OWNER_TO_POLICYプロシージャでは、既存のデータベース・ユーザーを所有者としてOracle Database Vaultポリシーに追加できます。
- ADD_REALM_TO_POLICYプロシージャ
  ADD_REALM_TO_POLICYプロシージャでは、既存のレルムをOracle Database Vaultポリシーに追加できます。
- CREATE_POLICYプロシージャ
  CREATE_POLICYプロシージャでは、Oracle Database Vaultポリシーを作成できます。
- DELETE_CMD_RULE_FROM_POLICYプロシージャ
  DELETE_CMD_RULE_FROM_POLICYプロシージャでは、既存のコマンド・ルールをOracle Database Vaultポリシーから削除できます。
- DELETE_OWNER_FROM_POLICYプロシージャ
  DELETE_OWNER_FROM_POLICYプロシージャでは、Oracle Database Vaultポリシーから所有者を削除できます。
- DELETE_REALM_FROM_POLICYプロシージャ
  DELETE_REALM_FROM_POLICYプロシージャでは、既存のレルムをOracle Database Vaultポリシーから削除できます。
- DROP_POLICYプロシージャ
  DROP_POLICYプロシージャでは、既存のOracle Database Vaultポリシーを削除できます。
- RENAME_POLICYプロシージャ
  UPDATE_POLICY_DESCRIPTIONプロシージャでは、既存のOracle Database Vaultポリシーの名前を変更できます。
- UPDATE_POLICY_DESCRIPTIONプロシージャ
  UPDATE_POLICY_DESCRIPTIONプロシージャでは、Oracle Database Vaultポリシー内のdescriptionフィールドを更新できます。
- UPDATE_POLICY_STATEプロシージャ
  UPDATE_POLICY_STATEプロシージャでは、Oracle Database Vaultポリシー内のpolicy_stateフィールドを更新できます。

関連トピック

- Oracle Database Vaultポリシーの構成
- Oracle Database VaultユーティリティのAPI

22.1 ADD_CMD_RULE_TO_POLICYプロシージャ

ADD_COMMAND_RULE_TO_POLICYプロシージャでは、既存のコマンド・ルールをOracle Database Vaultポリシーに追加できます。

コマンド・ルールがどの状態であっても、コマンド・ルールをポリシーに追加できます。たとえば、無効になっているコマンド・ルールを有効になっているポリシーに追加できます。この場合、無効になっているコマンド・ルールは、ポリシーに追加されると自動的に有効になります。コマンド・ルールは、1つのポリシーのみに追加できます。つまり、同じコマンド・ルールを複数のポリシーに割り当てることはできません。

構文

```
DBMS_MACADM.ADD_CMD_RULE_TO_POLICY(
 policy_name    IN VARCHAR2,
 command        IN VARCHAR2,
 object_owner   IN VARCHAR2,
 object_name    IN VARCHAR2,
 clause_name    IN VARCHAR2 DEFAULT,
 parameter_name IN VARCHAR2 DEFAULT,
 event_name     IN VARCHAR2 DEFAULT,
 component_name IN VARCHAR2 DEFAULT,
 action_name    IN VARCHAR2 DEFAULT,
 scope          IN NUMBER DEFAULT);
```

パラメータ

表22-1 ADD_CMD_RULE_TO_POLICYパラメータ

パラメータ	説明
policy_name	ポリシー名。現行のデータベース・インスタンスで既存の Database Vault ポリシーを確認するには、「DBA_DV_POLICY ビュー」で説明されている DBA_DV_POLICY ビューに問い合せます。
command	コマンド・ルール名 現行のデータベース・インスタンスで既存の Database Vault コマンド・ルールを確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューに問い合せます。
object_owner	コマンド・ルールを適用するデータベース・スキーマ このコマンド・ルールの既存のオブジェクト所有者を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
object_name	コマンド・ルールで保護されるオブジェクト このコマンド・ルールの既存のオブジェクトを確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
clause_name	ALTER SYSTEM および ALTER SESSION コマンド・ルールの場合は、コマンド・ルールの作成に使用された SQL 文の句 このコマンド・ルールの既存の句を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説

パラメータ	説明
	明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
parameter_name	ALTER SYSTEM および ALTER SESSION コマンド・ルールの場合は、clause_name パラメータのパラメータ このコマンド・ルールの既存のパラメータを確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
event_name	ALTER SYSTEM および ALTER SESSION コマンド・ルールの場合は、コマンド・ルールで定義されているイベント このコマンド・ルールの既存のイベント名を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
component_name	event_name 設定のコンポーネント このコマンド・ルールの既存のコンポーネント名を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
action_name	component_name 設定のアクション。 このコマンド・ルールの既存のアクション名を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがすべての PDB に適用される場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

次の例では、共通コマンド・ルールをDatabase Vaultポリシーに追加する方法を示します。このコマンド・ルールはマルチテナント環境のアプリケーション・ルート内にあるため、このプロシージャを実行するユーザーはアプリケーション・ルート内またはCDBルート内にいる必要があります。このコマンド・ルールに関連付けられているルールまたはルール・セットは、共通である必要があります。

```
BEGIN
 DBMS_MACADM.ADD_CMD_RULE_TO_POLICY(
  policy_name     => 'HR_DV_Policy',
  command         => 'ALTER SESSION',
  object_owner    => '%',
  object_name     => '%',
```

```
  clause_name    => 'PARALLEL DDL',
  parameter_name => '',
  event_name     => '',
  action_name    => '',
  scope          => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.2 ADD_OWNER_TO_POLICYプロシージャ

ADD_OWNER_TO_POLICYプロシージャでは、既存のデータベース・ユーザーを所有者としてOracle Database Vaultポリシーに追加できます。

有効になっているポリシーに所有者を追加すると、その変更内容がすぐに反映されます。ポリシーに追加するユーザーの数に制限はありません。

構文

```
DBMS_MACADM.ADD_OWNER_TO_POLICY(
policy_name   IN VARCHAR2,
owner_name    IN VARCHAR2);
```

パラメータ

表22-2 ADD_OWNER_TO_POLICYのパラメータ

パラメータ	説明
policy_name	ポリシー名。現行のデータベース・インスタンスで既存の Database Vault ポリシーを確認するには、「DBA_DV_POLICY ビュー」で説明されている DBA_DV_POLICY ビューに問い合せます。
owner_name	ユーザー名。現在のインスタンス内の既存のデータベース・ユーザー(ロールではない)を確認するには、Oracle Database リファレンスで説明されている、DBA_USERS ビューを問い合せます。既存のポリシー所有者を確認するには、「DBA_DV_POLICY_OWNER ビュー」で説明されている DBA_DV_POLICY_OWNER ビューを問い合せます。

例

```
BEGIN
 DBMS_MACADM.ADD_OWNER_TO_POLICY(
policy_name    => 'HR_DV_Policy',
owner_name     => 'PSMITH');
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.3 ADD_REALM_TO_POLICYプロシージャ

ADD_REALM_TO_POLICYプロシージャでは、既存のレルムをOracle Database Vaultポリシーに追加できます。

無効になっているレルムを有効になっているポリシーに追加できます。この場合、レルムは、追加されると自動的に有効になります。レルムは、1つのポリシーのみに追加できます。つまり、同じレルムを複数のポリシーに割り当てることはできません。

構文

```
DBMS_MACADM.ADD_REALM_TO_POLICY(
policy_name    IN VARCHAR2,
realm_name     IN VARCHAR2);
```

パラメータ

表22-3 ADD_REALM_TO_POLICYのパラメータ

パラメータ	説明
policy_name	ポリシー名。現在のデータベース・インスタンスで既存の Database Vault ポリシーを検索するには、DBA_DV_POLICY ビューを問い合せます。
realm_name	レルム名。現在のデータベース・インスタンスで既存の Database Vault レルムを検索します。

例

```
BEGIN
 DBMS_MACADM.ADD_REALM_TO_POLICY(
policy_name     => 'HR_DV_Policy',
realm_name      => 'HR Realm');
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.4 CREATE_POLICYプロシージャ

CREATE_POLICYプロシージャでは、Oracle Database Vaultポリシーを作成できます。

ポリシーの作成後、少なくとも1つのレルムと1つのコマンド・ルールをポリシーに追加する必要があります。必要に応じて、これらのレルムおよびコマンド・ルールを個別に強制するよう設定するか、ポリシーで使用されている強制を使用できます。

ポリシーの所有者は必要ありませんが、ポリシーに所有者を割り当てない場合は、DV_OWNERまたはDV_ADMINロールを付与されているユーザーがポリシーを管理する必要があります。

ポリシーを作成した後で、次のプロシージャを使用してポリシー定義を完了します。

- ADD_REALM_TO_POLICYは、レルムをポリシーに追加します。
- ADD_CMD_RULE_TO_POLICYは、コマンド・ルールをポリシーに追加します。
- ADD_OWNER_TO_POLICYは、指定したデータベース・ユーザーがポリシーを管理できるようにします。

構文

```
DBMS_MACADM.CREATE_POLICY(
 policy_name  IN VARCHAR2,
 description  IN VARCHAR2 DEFAULT,
 policy_state IN NUMBER,
 pl_sql_stack IN BOOLEAN DEFAULT);
```

パラメータ

表22-4 CREATE_POLICYのパラメータ

パラメータ	説明
policy_name	ポリシー名(大/小文字混在で最大 128 文字) 現行のデータベース・インスタンスで既存のポリシーを確認するには、「DBA_DV_POLICY ビュー」で説明されている DBA_DV_POLICY ビューに問い合せます。
description	ポリシーの目的の説明(大/小文字混在で最大 4000 文字)。
policy_state	ポリシーを有効にする方法を指定します。使用される値は、次のとおりです。 ● DBMS_MACADM.G_ENABLED (1)。これは、ポリシーをその作成後に有効にします。 ● DBMS_MACADM.G_DISABLED (0)。これは、ポリシーをその作成後に無効にします。 ● DBMS_MACADM.G_SIMULATION (2)。これは、ポリシーをシミュレーション・モードに設定します。シミュレーション・モードでは、ポリシー内で使用されるレルムまたはコマンド・ルールに対する違反が、ユーザー名や使用された SQL 文などエラーを説明する十分な情報とともに、指定されたログ表に記録されます。 ● DBMS_MACADM.G_PARTIAL (3)。これは、ポリシーを部分モードに設定します。部分モードでは、ポリシーに関連付けられているレルムまたはコマンド・ルールの強制状態を個別に変更できます。 シミュレーション・モードの詳細は、シミュレーション・モードについてを参照してください。
pl_sql_stack	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックを記録するかどうかを指定します。PL/SQL スタックを記録する場合は TRUE と入力し、記録しない場合は FALSE と入力します。

例

次の例では、部分状態を使用するポリシーを作成し、PL/SQLスタックの取得を有効にします。後で、レルムまたはコマンド・ルールをこのポリシーに追加するときに、それらの強制状態を個別に変更できます。

```
BEGIN
 DBMS_MACADM.CREATE_POLICY(
  policy_name  => 'HR_DV_Policy',
  description  => 'Policy to protect the HR schema',
  policy_state => DBMS_MACADM.G_ENABLED,
  pl_sql_stack => TRUE);
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.5 DELETE_CMD_RULE_FROM_POLICYプロシージャ

DELETE_CMD_RULE_FROM_POLICYプロシージャでは、既存のコマンド・ルールをOracle Database Vaultポリシーから

削除できます。

ポリシーの状態に関係なく、いつでもポリシーからコマンド・ルールを削除できます。ポリシーからコマンド・ルールを削除しても、コマンド・ルールの状態は同じままになります。つまり、ポリシーが有効になっており、ポリシーからコマンド・ルールを削除した場合、コマンド・ルールは、ポリシーから削除した後も有効なままとなります。

構文

```
DBMS_MACADM.DELETE_CMD_RULE_FROM_POLICY(
 policy_name    IN VARCHAR2,
 command        IN VARCHAR2,
 object_owner   IN VARCHAR2,
 object_name    IN VARCHAR2,
 clause_name    IN VARCHAR2 DEFAULT,
 parameter_name IN VARCHAR2 DEFAULT,
 event_name     IN VARCHAR2 DEFAULT,
 component_name IN VARCHAR2 DEFAULT,
 action_name    IN VARCHAR2 DEFAULT,
 scope          IN NUMBER DEFAULT);
```

パラメータ

表22-5 DELETE_CMD_RULE_FROM_POLICYのパラメータ

パラメータ	説明
policy_name	ポリシー名。現行のデータベース・インスタンスで既存の Database Vault ポリシーを確認するには、「DBA_DV_POLICY ビュー」で説明されている DBA_DV_POLICY ビューに問い合せます。
command	コマンド・ルール名 現行のデータベース・インスタンスで既存の Database Vault コマンド・ルールを確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューに問い合せます。
object_owner	コマンド・ルールを適用するデータベース・スキーマ このコマンド・ルールの既存のオブジェクト所有者を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
object_name	コマンド・ルールで保護されるオブジェクト このコマンド・ルールの既存のオブジェクトを確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
clause_name	ALTER SYSTEM および ALTER SESSION コマンド・ルールの場合は、コマンド・ルールの作成に使用された SQL 文の句 このコマンド・ルールの既存の句を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説

パラメータ	説明
	明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
parameter_name	ALTER SYSTEM および ALTER SESSION コマンド・ルールの場合は、clause_name パラメータのパラメータ このコマンド・ルールの既存のパラメータを確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
event_name	ALTER SYSTEM および ALTER SESSION コマンド・ルールの場合は、コマンド・ルールで定義されているイベント このコマンド・ルールの既存のイベント名を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
component_name	event_name 設定のコンポーネント このコマンド・ルールの既存のコンポーネント名を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
action_name	component_name 設定のアクション。 このコマンド・ルールの既存のアクション名を確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
scope	マルチテナント環境の場合は、このプロシージャの実行方法を決定します。デフォルトはローカルです。オプションは次のとおりです。 ● コマンド・ルールが現在の PDB でローカルである場合は、DBMS_MACUTL.G_SCOPE_LOCAL (または 1) ● コマンド・ルールがアプリケーション・ルート内にある場合は、DBMS_MACUTL.G_SCOPE_COMMON (または 2)

例

次の例では、Database Vaultポリシーから共通コマンド・ルールを削除する方法を示します。このコマンド・ルールはマルチテナント環境のアプリケーション・ルート内にあるため、このプロシージャを実行するユーザーはCDBルート内にいる必要があります。

```
BEGIN
 DBMS_MACADM.DELETE_CMD_RULE_FROM_POLICY(
  policy_name    => 'HR_DV_Policy',
  command        => 'ALTER SESSION',
  object_owner   => '%',
  object_name    => '%',
  clause_name    => 'END SESSION',
  parameter_name => 'KILL SESSION',
```

```
  event_name     => '',
  action_name    => '',
  scope          => DBMS_MACUTL.G_SCOPE_COMMON);
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.6 DELETE_OWNER_FROM_POLICYプロシージャ

DELETE_OWNER_FROM_POLICYプロシージャでは、Oracle Database Vaultポリシーから所有者を削除できます。

ポリシーの状態(有効または無効)に関係なく、いつでもポリシーから所有者を削除できます。変更は即座に反映されます。

構文

```
DBMS_MACADM.DELETE_OWNER_FROM_POLICY(
policy_name    IN VARCHAR2,
owner_name     IN VARCHAR2);
```

パラメータ

表22-6 DELETE_OWNER_FROM_POLICYのパラメータ

パラメータ	説明
policy_name	ポリシー名。現行のデータベース・インスタンスで既存の Database Vault ポリシーを確認するには、「DBA_DV_POLICY ビュー」で説明されている DBA_DV_POLICY ビューに問い合せます。
owner_name	ユーザー名。現在のインスタンスで既存のポリシー所有者を確認するには、「DBA_DV_POLICY_OWNER ビュー」で説明されている DBA_DV_POLICY_OWNER ビューを問い合せます。

例

```
BEGIN
 DBMS_MACADM.DELETE_OWNER_FROM_POLICY(
policy_name     => 'HR_DV_Policy',
owner_name      => 'PSMITH');
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.7 DELETE_REALM_FROM_POLICYプロシージャ

DELETE_REALM_FROM_POLICYプロシージャでは、既存のレルムをOracle Database Vaultポリシーから削除できます。

ポリシーの状態(有効または無効)に関係なく、いつでもポリシーからレルムを削除できます。変更は即座に反映されます。

構文

```
DBMS_MACADM.DELETE_REALM_FROM_POLICY(
policy_name    IN VARCHAR2,
realm_name     IN VARCHAR2);
```

パラメータ

表22-7 DELETE_REALM_FROM_POLICYのパラメータ

パラメータ	説明
policy_name	ポリシー名。現在のデータベース・インスタンスで既存の Database Vault ポリシーを検索するには、DBA_DV_POLICY ビューを問い合せます。
realm_name	レルム名。現在のデータベース・インスタンスで既存の Database Vault レルムを検索するには、DV_REALM ビューを問い合せます。

例

```
BEGIN
 DBMS_MACADM.DELETE_REALM_FROM_POLICY(
policy_name    => 'HR_DV_Policy',
realm_name     => 'HR Realm');
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.8 DROP_POLICYプロシージャ

DROP_POLICYプロシージャでは、既存のOracle Database Vaultポリシーを削除できます。

ポリシーの状態(有効または無効)に関係なく、いつでもポリシーを削除できます。

構文

```
DBMS_MACADM.DROP_POLICY(
policy_name       IN VARCHAR2);
```

パラメータ

表22-8 DROP_POLICYのパラメータ

パラメータ	説明
policy_name	ポリシー名。現行のデータベース・インスタンスで既存の Database Vault ポリシーを確認するには、「DBA_DV_POLICY ビュー」で説明されている DBA_DV_POLICY ビューに問い合せます。

例

```
BEGIN
 DBMS_MACADM.DROP_POLICY(
policy_name       => 'HR_DV_Policy');
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.9 RENAME_POLICYプロシージャ

UPDATE_POLICY_DESCRIPTIONプロシージャでは、既存のOracle Database Vaultポリシーの名前を変更できます。

ポリシーの状態(有効または無効)に関係なく、いつでもポリシーの名前を変更できます。変更は即座に反映されます。

構文

```
DBMS_MACADM.RENAME_POLICY(
policy_name      IN VARCHAR2,
new_policy_name  IN VARCHAR2);
```

パラメータ

表22-9 RENAME_POLICYのパラメータ

パラメータ	説明
policy_name	ポリシー名。現行のデータベース・インスタンスで既存の Database Vault ポリシーを確認するには、「DBA_DV_POLICY ビュー」で説明されている DBA_DV_POLICY ビューに問い合せます。
new_policy_name	新しいポリシー名(大/小文字混在で最大 128 文字)

例

```
BEGIN
 DBMS_MACADM.RENAME_POLICY(
policy_name      => 'HR_DV_Policy',
new_policy_name  => 'HR_WEST_COAST_DV_Policy');
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.10 UPDATE_POLICY_DESCRIPTIONプロシージャ

UPDATE_POLICY_DESCRIPTIONプロシージャでは、Oracle Database Vaultポリシー内のdescriptionフィールドを更新できます。

構文

```
DBMS_MACADM.UPDATE_POLICY_DESCRIPTION(
policy_name  IN VARCHAR2,
description  IN VARCHAR2 DEFAULT);
```

パラメータ

表22-10 UPDATE_POLICY_DESCRIPTIONのパラメータ

パラメータ	説明
policy_name	ポリシー名。現行のデータベース・インスタンスで既存の Database Vault ポリシーを確認するには、「DBA_DV_POLICY ビュー」で説明されている DBA_DV_POLICY ビューに問い合せます。
description	ポリシーの目的の新しい説明(大/小文字混在で最大 4000 文字)

例

```
BEGIN
 DBMS_MACADM.UPDATE_POLICY_DESCRIPTION(
```

```
policy_name  => 'HR_DV_Policy',
description  => 'HR schema protection policy');
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

22.11 UPDATE_POLICY_STATEプロシージャ

UPDATE_POLICY_STATEプロシージャでは、Oracle Database Vaultポリシー内のpolicy_stateフィールドを更新できます。

構文

```
DBMS_MACADM.UPDATE_POLICY_STATE(
policy_name    IN VARCHAR2,
policy_state   IN NUMBER,
 pl_sql_stack   IN BOOLEAN DEFAULT);
```

パラメータ

表22-11 UPDATE_POLICY_STATEのパラメータ

パラメータ	説明
policy_name	ポリシー名。現行のデータベース・インスタンスで既存の Database Vault ポリシーを確認するには、「DBA_DV_POLICY ビュー」で説明されている DBA_DV_POLICY ビューに問い合せます。
policy_state	ポリシーを有効にする方法を指定します。使用される値は、次のとおりです。 ● DBMS_MACADM.G_ENABLED (1)。これは、ポリシーをその作成後に有効にします。 ● DBMS_MACADM.G_DISABLED (0)。これは、ポリシーをその作成後に無効にします。 ● DBMS_MACADM.G_SIMULATION (2)。これは、ポリシーをシミュレーション・モードに設定します。シミュレーション・モードでは、ポリシー内で使用されるレルムまたはコマンド・ルールに対する違反が、ユーザー名や使用された SQL 文などエラーを説明する十分な情報とともに、指定されたログ表に記録されます。 ● DBMS_MACADM.G_PARTIAL (3)。これは、ポリシーを部分モードに設定します。部分モードでは、ポリシーに関連付けられているレルムまたはコマンド・ルールの強制状態を個別に変更できます。 シミュレーション・モードの詳細は、シミュレーション・モードについてを参照してください。
pl_sql_stack	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックを記録するかどうかを指定します。PL/SQL スタックを記録する場合は TRUE と入力し、記録しない場合は FALSE と入力します。

例

```
BEGIN
 DBMS_MACADM.UPDATE_POLICY_STATE(
  policy_name   => 'HR_DV_Policy',
  policy_state  => DBMS_MACADM.G_DISABLED,
  pl_sql_stack  => TRUE);
END;
/
```

親トピック: Oracle Database VaultポリシーのAPI

23 Oracle Database VaultのAPIリファレンス

Oracle Database Vaultには、PL/SQLパッケージとスタンドアロン・プロシージャの両方に豊富なAPIセットが用意されています。

- DBMS_MACADM PL/SQLパッケージの内容
  DBMS_MACADMパッケージを使用すると、レルム、ファクタ、ルール・セット、コマンド・ルール、セキュア・アプリケーション・ロールおよびOracle Label Securityポリシーを構成できます。
- DBMS_MACSEC_ROLES PL/SQLパッケージの内容
  DBMS_MACSEC_ROLESパッケージを使用すると、Oracle Database Vaultセキュア・アプリケーション・ロールを確認し設定できます。
- DBMS_MACUTL PL/SQLパッケージの内容
  DBMS_MACUTL PL/SQLパッケージは、エラー処理など、他のOracle Database Vaultのパッケージで共通に使用される定数およびユーティリティ・メソッドを定義します。
- CONFIGURE_DV PL/SQLプロシージャ
  CONFIGURE_DVは、最初の2つのOracle Databaseユーザー・アカウントを構成します。このアカウントには、DV_OWNERロールとDV_ACCTMGRロールがそれぞれ付与されます。
- DVF PL/SQLインタフェースの内容
  DVFスキーマにより、一連のファクタ関連PL/SQLファンクションが提供されます。

23.1 DBMS_MACADM PL/SQLパッケージの内容

DBMS_MACADMパッケージを使用すると、レルム、ファクタ、ルール・セット、コマンド・ルール、セキュア・アプリケーション・ロールおよびOracle Label Securityポリシーを構成できます。

DBMS_MACADMパッケージは、DV_ADMINロールまたはDV_OWNERロールを付与されているユーザーのみが使用できます。

DBMS_MACADMのレルム・プロシージャ

表23-1に、DBMS_MACADMパッケージ内のレルム・プロシージャを示します。

表23-1 DBMS_MACADMのレルム・プロシージャ

プロシージャ	説明
ADD_AUTH_TO_REALM プロシージャ	所有者または参加者としてレルムにアクセスする権限をユーザーまたはロールに付与します。
ADD_OBJECT_TO_REALM プロシージャ	レルム保護に一連のオブジェクトを登録します。
CREATE_REALM プロシージャ	レルムを作成します。
DELETE_AUTH_FROM_REALM プロシージャ	レルムにアクセスするためのユーザーまたはロールの認可を削除します。
DELETE_OBJECT_FROM_REALM プロシージャ	レルム保護から一連のオブジェクトを削除します。

プロシージャ	説明
DELETE_REALM プロシージャ	レルム(認可されるユーザーと保護対象オブジェクトを指定するレルム関連 Database Vault 構成情報を含む)を削除します。
DELETE_REALM_CASCADE プロシージャ	レルム(認可されるユーザーと保護対象オブジェクトを指定するレルム関連 Database Vault 構成情報を含む)を削除します。
RENAME_REALM プロシージャ	レルムの名前を変更します。名前の変更は、そのレルムが使用されているすべての箇所に反映されます。
UPDATE_REALM プロシージャ	レルムを更新します。
UPDATE_REALM_AUTH プロシージャ	レルムにアクセスするためのユーザーまたはロールの認可を更新します。

DBMS_MACADMのルール・セット・プロシージャとルール・プロシージャ

表23-2に、DBMS_MACADMパッケージ内のルール・セット・プロシージャおよびルール・プロシージャを示します。

表23-2 DBMS_MACADMのルール・セット・プロシージャとルール・プロシージャ

プロシージャ	説明
CREATE_RULE_SET プロシージャ	ルール・セットを作成します。
RENAME_RULE_SET プロシージャ	ルール・セットの名前を変更します。名前の変更は、そのルール・セットが使用されているすべての箇所に反映されます。
DELETE_RULE_FROM_RULE_SET プロシージャ	ルールをルール・セットから削除します。
DELETE_RULE_SET プロシージャ	ルール・セットを削除します。
UPDATE_RULE_SET プロシージャ	ルール・セットを更新します。
CREATE_RULE プロシージャ	ルールを作成します。
ADD_RULE_TO_RULE_SET プロシージャ	ルールをルール・セットに追加します。
DELETE_RULE プロシージャ	ルールを削除します。
RENAME_RULE プロシージャ	ルールの名前を変更します。名前の変更は、そのルールが使用されているすべての箇所に反映されます。

プロシージャ	説明
UPDATE_RULE プロシージャ	ルールを更新します。

DBMS_MACADMのコマンド・ルール・プロシージャ

表23-3に、DBMS_MACADMパッケージ内のコマンド・ルール・プロシージャを示します。

表23-3 DBMS_MACADMのコマンド・ルール・プロシージャ

プロシージャ	説明
CREATE_COMMAND_RULE プロシージャ	コマンド・ルールを作成し、ルール・セットに関連付けて、ルール・セットによるコマンド・ルールのルール・チェックが有効化されるようにします。
CREATE_CONNECT_COMMAND_RULE プロシージャ	CONNECT コマンド・ルールを作成します。
CREATE_SESSION_EVENT_CMD_RULE プロシージャ	ALTER SESSION SQL 文を使用して、セッション・イベント・コマンド・ルールを作成します。
CREATE_SYSTEM_EVENT_CMD_RULE プロシージャ	ALTER SYSTEM SQL 文を使用して、システム・イベント・コマンド・ルールを作成します。
DELETE_COMMAND_RULE プロシージャ	コマンド・ルールの宣言を削除します。
DELETE_CONNECT_COMMAND_RULE プロシージャ	CONNECT コマンド・ルールの宣言を削除します。
DELETE_SESSION_EVENT_CMD_RULE プロシージャ	SESSION_EVENT_CMD コマンド・ルールの宣言を削除します。
DELETE_SYSTEM_EVENT_CMD_RULE プロシージャ	SYSTEM_EVENT_CMD コマンド・ルールの宣言を削除します。
UPDATE_COMMAND_RULE プロシージャ	コマンド・ルールの宣言を更新します。
UPDATE_CONNECT_COMMAND_RULE プロシージャ	CONNECT コマンド・ルールの宣言を更新します。
UPDATE_SESSION_EVENT_CMD_RULE プロシージャ	SESSION_EVENT_CMD コマンド・ルールの宣言を更新します。

プロシージャ	説明
UPDATE_SYSTEM_EVENT_CMD_RULE プロシージャ	SYSTEM_EVENT_CMD コマンド・ルールの宣言を更新します。

DBMS_MACADMファクタのプロシージャおよびファンクション

DBMS_MACADMパッケージ内のファクタ・プロシージャおよびファンクションを示します。

表23-4 DBMS_MACADMファクタのプロシージャおよびファンクション

プロシージャまたはファンクション	説明
ADD_FACTOR_LINK プロシージャ	2 つのファクタの親子関係を指定します。
ADD_POLICY_FACTOR プロシージャ	ファクタのラベルをポリシーの Oracle Label Security ラベルに含めることを指定します。
CHANGE_IDENTITY_FACTOR プロシージャ	アイデンティティを別ファクタと関連付けます。
CHANGE_IDENTITY_VALUE プロシージャ	アイデンティティの値を更新します。
CREATE_DOMAIN_IDENTITY プロシージャ	Oracle Real Application Clusters(Oracle RAC)データベース・ノードをドメイン・ファクタ・アイデンティティに追加し、Oracle Label Security ポリシーに従ってラベルを付けます。
CREATE_FACTOR プロシージャ	ファクタを作成します。
CREATE_FACTOR_TYPE プロシージャ	ファクタ・タイプを作成します。
CREATE_IDENTITY プロシージャ	アイデンティティを作成します。
CREATE_IDENTITY_MAP プロシージャ	リンクされた子ファクタ(サブファクタ)の値からファクタのアイデンティティを導出するために使用される一連のテストを定義します。
DELETE_FACTOR プロシージャ	ファクタを削除します。
DELETE_FACTOR_LINK プロシージャ	2 つのファクタの親子関係を削除します。
DELETE_FACTOR_TYPE プロシージャ	ファクタ・タイプを削除します。
DELETE_IDENTITY プロシージャ	アイデンティティを削除します。

プロシージャまたはファンクション	説明
DELETE_IDENTITY_MAP プロシージャ	ファクタからアイデンティティ・マップを削除します。
DROP_DOMAIN_IDENTITY プロシージャ	Oracle RAC データベース・ノードをドメインから削除します。
GET_INSTANCE_INFO ファンクション	現行データベース・インスタンスについて SYS.V_$INSTANCE システム表の情報を返します。VARCHAR2 値を返します。
GET_SESSION_INFO ファンクション	現行セッションについて SYS.V_$SESSION システム表の情報を返します。VARCHAR2 値を返します。
RENAME_FACTOR プロシージャ	ファクタの名前を変更します。名前の変更は、そのファクタが使用されているすべての箇所に反映されます。
RENAME_FACTOR_TYPE プロシージャ	ファクタ・タイプの名前を変更します。名前の変更は、そのファクタ・タイプが使用されているすべての箇所に反映されます。
UPDATE_FACTOR プロシージャ	ファクタを更新します。
UPDATE_FACTOR_TYPE プロシージャ	ファクタ・タイプの説明を更新します。
UPDATE_IDENTITY プロシージャ	ファクタ・アイデンティティの信頼レベルを更新します。

DBMS_MACADMセキュア・アプリケーション・ロールのプロシージャ

表23-5に、DBMS_MACADMパッケージ内のセキュア・アプリケーション・ロール・プロシージャを示します。

表23-5 DBMS_MACADMセキュア・アプリケーション・ロールのプロシージャ

プロシージャ	説明
CREATE_ROLE プロシージャ	Oracle Database Vault セキュア・アプリケーション・ロールを作成します。
DELETE_ROLE プロシージャ	Oracle Database Vault セキュア・アプリケーション・ロールを削除します。
RENAME_ROLE プロシージャ	Oracle Database Vault セキュア・アプリケーション・ロールの名前を変更します。名前の変更は、そのロールが使用されているすべての箇所に反映されます。
UNASSIGN_ROLE プロシージャ	ユーザーから Oracle Database Vault セキュア・アプリケーション・ロー

プロシージャ	説明
	ルを割当て解除します。
UPDATE_ROLE プロシージャ	Oracle Database Vault セキュア・アプリケーション・ロールを更新します。

DBMS_MACADMのOracle Label Securityプロシージャ

表23-6に、DBMS_MACADMパッケージ内のOracle Label Securityプロシージャを示します。

表23-6 DBMS_MACADMのOracle Label Securityプロシージャ

プロシージャ	説明
CREATE_MAC_POLICY プロシージャ	ファクタのラベルまたは Oracle Label Security セッション・ラベルを算出する際にラベルのマージに使用されるアルゴリズムを指定します。
CREATE_POLICY_LABEL プロシージャ	Oracle Label Security ポリシー内のアイデンティティにラベルを付けます。
DELETE_MAC_POLICY_CASCADE プロシージャ	Oracle Label Security ポリシーに関連するすべての Oracle Database Vault オブジェクトを削除します。
DELETE_POLICY_FACTOR プロシージャ	Oracle Label Security ラベルの構成からファクタを削除します。
DELETE_POLICY_LABEL プロシージャ	Oracle Label Security ポリシーのアイデンティティからラベルを削除します。
UPDATE_MAC_POLICY プロシージャ	ファクタのラベルまたは Oracle Label Security セッション・ラベルを算出する際にラベルのマージに使用されるアルゴリズムを指定します。

DBMS_MACADMのDatabase Vaultポリシー・プロシージャ

表23-7に、DBMS_MACADMパッケージ内のDatabase Vaultポリシー・プロシージャを示します。

表23-7 DBMS_MACADMのDatabase Vaultポリシー・プロシージャ

プロシージャ	説明
ADD_CMD_RULE_TO_POLICY プロシージャ	Database Vault ポリシーにコマンド・ルールを追加します。
ADD_OWNER_TO_POLICY プロシージャ	Database Vault ポリシーに所有者を追加します。
ADD_REALM_TO_POLICY プロシージャ	Database Vault ポリシーにレルムを追加します。

プロシージャ	説明
CREATE_POLICY プロシージャ	Creates a Database Vault ポリシーを作成します。
DELETE_CMD_RULE_FROM_POLICY プロシージャ	Database Vault ポリシーからコマンド・ルールを削除します。
DELETE_OWNER_FROM_POLICY プロシージャ	Database Vault ポリシーから所有者を削除します。
DELETE_REALM_FROM_POLICY プロシージャ	Database Vault ポリシーからレルムを削除します。
DROP_POLICY プロシージャ	Database Vault ポリシーを削除します。
RENAME_POLICY プロシージャ	Database Vault ポリシーの名前を変更します。
UPDATE_POLICY_DESCRIPTION プロシージャ	Database Vault ポリシーの説明を更新します。
UPDATE_POLICY_STATE プロシージャ	Database Vault ポリシーの有効化ステータスを更新します。

DBMS_MACADMの一般管理プロシージャ

表23-8に、DBMS_MACADMパッケージ内の一般管理プロシージャを示します。

表23-8 DBMS_MACADMの一般管理プロシージャ

プロシージャ	説明
ADD_NLS_DATA プロシージャ	Oracle Database Vault に新しい言語を追加します。
ADD_APP_EXCEPTION プロシージャ	共通スキーマまたはパッケージがローカル・スキーマにアクセスできるようにします
AUTHORIZE_DATAPUMP_USER プロシージャ	Oracle Database Vault が有効な場合に、Oracle Data Pump 操作を実行する権限をユーザーに付与します。
AUTHORIZE_DDL プロシージャ	データ定義言語(DDL)文を実行する認可をユーザーに付与します。
AUTHORIZE_MAINTENANCE_USER プロシージャ	情報ライフサイクル管理(ILM)操作を実行するためのユーザー認可を付与します。
AUTHORIZE_PROXY_USER プロシージャ	他のユーザー・アカウントをプロキシする認可をプロキシ・ユーザーに付与します。

プロシージャ	説明
AUTHORIZE_SCHEDULER_USER プロシージャ	Oracle Database Vault が有効な場合に、データベース・ジョブをスケジュールする権限をユーザーに付与します。
AUTHORIZE_TTS_USER プロシージャ	Oracle Database Vault が有効な場合に、表領域に対して Oracle Data Pump トランスポータブル表領域操作を実行するようにユーザーを認可します。
DELETE_APP_EXCEPTION プロシージャ	ローカル・スキーマにアクセスするための共通ユーザーまたはパッケージの例外を削除します
DISABLE_DV_DICTIONARY_ACCTS プロシージャ	ユーザーが DVSYS や DFV スキーマ・アカウントにログインできないようにします。
DISABLE_DV_PATCH_ADMIN	DV_PATCH_ADMIN ユーザーの監査を無効にします。
DISABLE_DV プロシージャ	Oracle Database Vault を無効にします。
DISABLE_APP_PROTECTION プロシージャ	Database Vault 操作の制御を無効にします
DISABLE_ORADEBUG プロシージャ	Oracle Database Vault 環境で ORADEBUG ユーティリティの使用を無効にします。
ENABLE_DV_DICTIONARY_ACCTS プロシージャ	ユーザーが DVSYS や DFV スキーマ・アカウントにログインできるようにします。
ENABLE_DV_PATCH_ADMIN	DV_PATCH_ADMIN ユーザーの監査を有効にします。
ENABLE_DV プロシージャ	Oracle Database Vault を有効にします。
ENABLE_APP_PROTECTION プロシージャ	Database Vault 操作の制御を有効にします
ENABLE_ORADEBUG プロシージャ	Oracle Database Vault 環境で ORADEBUG ユーティリティの使用を有効にします。
UNAUTHORIZE_DATAPUMP_USER プロシージャ	DBMS_MACADM.AUTHORIZE_DATAPUMP_USER プロシージャによって付与された権限を取り消します。
UNAUTHORIZE_DDL プロシージャ	DBMS_MACDM.AUTHORIZE_DDL プロシージャを使用して DDL 文を

プロシージャ	説明
	実行する認可を付与されたユーザーから認可を取り消します。
UNAUTHORIZE_MAINTENANCE_USER プロシージャ	ILM 操作を実行するための認可を取り消します。
UNAUTHORIZE_PROXY_USER プロシージャ	DBMS_MACADM.AUTHORIZE_PROXY_USER プロシージャによってプロキシの認可を付与されたユーザーから認可を取り消します。
UNAUTHORIZE_SCHEDULER_USER プロシージャ	DBMS_MACADM.AUTHORIZE_SCHEDULER_USER プロシージャによって付与された権限を取り消します。
UNAUTHORIZE_TTS_USER プロシージャ	Oracle Database Vault が有効な場合に、表領域に対して Oracle Data Pump トランスポータブル表領域操作を実行する認可を付与されたユーザーから認可を取り消します。

親トピック: Oracle Database VaultのAPIリファレンス

23.2 DBMS_MACSEC_ROLES PL/SQLパッケージの内容

DBMS_MACSEC_ROLESパッケージでは、Oracle Database Vaultセキュア・アプリケーション・ロールを確認および設定できます。

このパッケージは、一般のデータベース・アカウント群で使用できます。

表23-9に、DBMS_MACSEC_ROLESパッケージの内容を示します。

表23-9 DBMS_MACSEC_ROLES PL/SQLパッケージの内容

プロシージャまたはファンクション	説明
CAN_SET_ROLE ファンクション	メソッドを起動するユーザーに、指定された Oracle Database Vault セキュア・アプリケーション・ロールを使用する権限が付与されているかどうかをチェックします。BOOLEAN 値を返します。
SET_ROLE プロシージャ	Oracle Database Vault セキュア・アプリケーション・ロールに対して SET ROLE 文を発行します。

親トピック: Oracle Database VaultのAPIリファレンス

23.3 DBMS_MACUTL PL/SQLパッケージの内容

DBMS_MACUTL PL/SQLパッケージは、エラー処理など、他のOracle Database Vaultのパッケージで共通に使用される定数およびユーティリティ・メソッドを定義します。

このパッケージは、一般のデータベース・アカウント群で実行できます。このパッケージを使用すると、セキュリティ開発者は、スクリ

プト化された構成ファイルで定数を利用できます。また、USER_HAS_ROLEなどのユーティリティ・メソッドは、Oracle Database Vaultルールで使用することも可能です。

表23-10に、DBMS_MACUTLパッケージの内容を示します。

表23-10 DBMS_MACUTL PL/SQLパッケージの内容

プロシージャまたはファンクション	説明
CHECK_DVSYS_DML_ALLOWED プロシージャ	Oracle Database Vault 構成を更新するユーザーによってパブリック・パッケージが無視されていないことを検証します。
GET_CODE_VALUE ファンクション	コード・グループ内でコードの値を検索します。
GET_SECOND ファンクション	Oracle SS 形式(00 から 59)で秒を返します。時間データに基づいたルール式に有用です。
GET_MINUTE ファンクション	Oracle MI 形式(00 から 59)で分を返します。時間データに基づいたルール式に有用です。
GET_HOUR ファンクション	Oracle HH24 形式(00 から 23)で月を返します。時間データに基づいたルール式に有用です。
GET_DAY ファンクション	Oracle DD 形式(01 から 31)で日を返します。時間データに基づいたルール式に有用です。
GET_MONTH ファンクション	Oracle MM 形式(01 から 12)で月を返します。時間データに基づいたルール式に有用です。
GET_YEAR ファンクション	Oracle YYYY 形式(0001 から 9999)で年を返します。時間データに基づいたルール式に有用です。
IS_ALPHA ファンクション	文字がアルファベットかどうかをチェックします。
IS_DIGIT ファンクション	文字が数値かどうかをチェックします。
IS_DVSYS_OWNER ファンクション	Oracle Database Vault 構成を管理する権限がユーザーに付与されているかどうかを判断します。
IS_OLS_INSTALLED ファンクション	Oracle Label Security がインストールされているかどうかについてインジケータを返します。
IS_OLS_INSTALLED_VARCHAR ファ	Oracle Label Security がインストールされているかどうかについてインジケータを

プロシージャまたはファンクション	説明
ンクション	返します。
USER_HAS_ROLE ファンクション	ユーザーがロール権限を直接保持するのか間接的に(他のロールを介して)保持するのかをチェックします。
USER_HAS_ROLE_VARCHAR ファンクション	ユーザーがロール権限を直接保持するのか間接的に(他のロールを介して)保持するのかをチェックします。
USER_HAS_SYSTEM_PRIVILEGE ファンクション	ユーザーがシステム権限を直接保持するのか間接的に(ロールを介して)保持するのかをチェックします。

親トピック: Oracle Database VaultのAPIリファレンス

23.4 CONFIGURE_DV PL/SQLプロシージャ

CONFIGURE_DVは、最初の2つのOracle Databaseユーザー・アカウントを構成します。このアカウントには、DV_OWNERロールとDV_ACCTMGRロールがそれぞれ付与されます。

このプロシージャは、Oracle Database VaultのOracle Databaseへの登録プロセスの一部として使用されます。これはデータベース・インスタンスに対して1回のみ実行する必要があります。

親トピック: Oracle Database VaultのAPIリファレンス

23.5 DVF PL/SQLインタフェースの内容

DVFスキーマにより、一連のファクタ関連PL/SQLファンクションが提供されます。

その後、ファンクションは一般のデータベース・アカウント群でPL/SQLファンクションおよび標準SQLを介して使用できます。

表23-11に、DVFファクタのファンクションを示します。

表23-11 DVF PL/SQLインタフェースの内容

ファンクション	説明
F$CLIENT_IP	クライアントが接続されているコンピュータの IP アドレスを返します。
F$DATABASE_DOMAIN	DB_DOMAIN 初期化パラメータに指定されているようにデータベースのドメインを返します。
F$DATABASE_HOSTNAME	データベース・インスタンスが実行されているコンピュータのホスト名を返します。
F$DATABASE_INSTANCE	現在のデータベース・インスタンスのデータベース・インスタンス識別番号を返します。

ファンクション	説明
F$DATABASE_IP	データベース・インスタンスが実行されているコンピュータの IP アドレスを返します。
F$DATABASE_NAME	DB_NAME 初期化パラメータに指定されているようにデータベースの名前を返します。
F$DOMAIN	ランタイム環境(ネットワーク IT 環境やそのサブセットなど)内の、特定の機密レベルで動作する物理的な構成または実装固有のファクタの名前付きコレクションを返します。
F$ENTERPRISE_IDENTITY	ユーザーのエンタープライズ全体のアイデンティティを返します。
F$IDENTIFICATION_TYPE	データベースでのユーザーのスキーマの作成方法を返します。具体的には、CREATE USER または ALTER USER 構文の IDENTIFIED 句が反映されます。
F$LANG	既存の LANGUAGE パラメータより短い形式の、言語名の ISO 略称を返します。
F$LANGUAGE	セッションで現在使用中の言語と地域、およびデータベース文字セットを VARCHAR2 データ型で返します。
F$MACHINE	データベース・セッションを確立したデータベース・クライアントのコンピュータ(ホスト)名を返します。
F$NETWORK_PROTOCOL	接続文字列の PROTOCOL=protocol 部分に指定されている、通信に使用されるネットワーク・プロトコルを返します。
F$PROXY_ENTERPRISE_IDENTITY	プロキシ・ユーザーがエンタープライズ・ユーザーである場合、Oracle Internet Directory の識別名(DN)を返します。
F$SESSION_USER	現行ユーザーが認証されたデータベース・ユーザー名を返します。

親トピック: Oracle Database VaultのAPIリファレンス

24 Oracle Database Vaultのデータ・ディクショナリ・ビュー

Database Vault固有のデータ・ディクショナリ・ビューに問い合せることで、Oracle Database Vault構成設定に関する情報を検索できます。

- Oracle Database Vaultのデータ・ディクショナリ・ビューについて
  Oracle Database Vaultには、DV_SECANALYSTロールまたはDV_ADMINロールを介してアクセスできる一連のDBAスタイルのデータ・ディクショナリ・ビューが用意されています。
- CDB_DV_STATUSビュー
  CDB_DV_STATUSデータ・ディクショナリ・ビューには、すべてのPDBのDatabase Vault操作の制御、構成および有効化のステータスが表示されます。
- DBA_DV_APP_EXCEPTIONビュー
  DBA_DV_APP_EXCEPTIONデータ・ディクショナリ・ビューには、Database Vault操作の制御の例外リストにある共通スキーマとパッケージ名がリストされます。
- DBA_DV_CODEビュー
  DBA_DV_CODEデータ・ディクショナリ・ビューには、ユーザー・インタフェース、エラー・メッセージおよび制約チェックの一般的な参照コードが示されます。
- DBA_DV_COMMAND_RULEビュー
  DBA_DV_COMMAND_RULEデータ・ディクショナリ・ビューには、コマンド・ルールにより保護されるSQL文が表示されます。
- DBA_DV_DATAPUMP_AUTHビュー
  DBA_DV_DATAPUMP_AUTHデータ・ディクショナリ・ビューでは、Oracle Database Vault環境でOracle Data Pumpを使用するための認可が示されます。
- DBA_DV_DBCAPTURE_AUTHビュー
  DBA_DV_DBCAPTURE_AUTHデータ・ディクショナリ・ビューには、Oracle Database Replayのワークロード取得操作を実行する認可が付与されているユーザーが表示されます。
- DBA_DV_DBREPLAYビュー
  DBA_DV_DBREPLAY_AUTHデータ・ディクショナリ・ビューには、Oracle Database Replayのワークロード・リプレイ操作を実行する認可が付与されているユーザーが表示されます。
- DBA_DV_DDL_AUTHビュー
  DBA_DV_DDLデータ・ディクショナリ・ビューには、DBMS_MACADM.AUTHORIZE_DDLプロシージャで指定されたユーザーとスキーマが示されます。
- DBA_DV_DICTIONARY_ACCTSビュー
  DBA_DV_DICTIONARY_ACCTSデータ・ディクショナリ・ビューには、ユーザーがDVSYSスキーマ・アカウントとDVFスキーマ・アカウントに直接ログインできるかどうかが示されます。
- DBA_DV_FACTORビュー
  DBA_DV_FACTORデータ・ディクショナリ・ビューには、現行のデータベース・インスタンス内の既存のファクタが表示されます。
- DBA_DV_FACTOR_TYPEビュー
  DBA_DV_FACTOR_TYPEデータ・ディクショナリ・ビューには、システムで使用されているファクタ・タイプの名前と説明が表示されます。

- DBA_DV_FACTOR_LINKビュー
  DBA_DV_FACTOR_LINKデータ・ディクショナリ・ビューには、子ファクタの関連によりアイデンティティが決まる各ファクタの関係が表示されます。
- DBA_DV_IDENTITYビュー
  DBA_DV_IDENTITYデータ・ディクショナリ・ビューには、各ファクタのアイデンティティが表示されます。
- DBA_DV_IDENTITY_MAPビュー
  DBA_DV_IDENTITY_MAPデータ・ディクショナリ・ビューには、各ファクタ・アイデンティティのマップが表示されます。
- DBA_DV_JOB_AUTHビュー
  DBA_DV_JOB_AUTHデータ・ディクショナリ・ビューでは、Oracle Database Vault環境でOracle Schedulerを使用するための認可が示されます。
- DBA_DV_MAC_POLICYビュー
  DBA_DV_MAC_POLICYデータ・ディクショナリ・ビューには、Oracle Database Vaultで使用するために定義されたOracle Label Securityポリシーが表示されます。
- DBA_DV_MAC_POLICY_FACTORビュー
  DBA_DV_MAC_POLICYデータ・ディクショナリ・ビューには、Oracle Label Securityポリシーに関連付けられているファクタが表示されます。
- DBA_DV_MAINTENANCE_AUTHビュー
  DBA_DV_MAINTENANCE_AUTHデータ・ディクショナリ・ビューでは、情報ライフサイクル管理(ILM)機能を使用するためのOracle Database Vault認可の構成について情報が示されます。
- DBA_DV_ORADEBUGビュー
  DBA_DV_ORADEBUGデータ・ディクショナリ・ビューには、ユーザーがOracle Database Vault環境でORADEBUGユーティリティを使用できるかどうかが示されます。
- DBA_DV_PATCH_ADMIN_AUDIT View
  DBA_DV_PATCH_ADMIN_AUDITデータ・ディクショナリ・ビューには、DV_ADMIN_PATCHロールを付与されているユーザーに対して監査が有効か無効かが示されます。
- DBA_DV_POLICYビュー
  DBA_DV_POLICYデータ・ディクショナリ・ビューでは、現在のデータベース・インスタンスで作成されたOracle Database Vaultポリシーが示されます。
- DBA_DV_POLICY_LABELビュー
  DBA_DV_POLICY_LABELデータ・ディクショナリ・ビューには、各ポリシーのDBA_DV_IDENTITYビューの各ファクタ識別子に対するOracle Label Securityラベルが表示されます。
- DBA_DV_POLICY_OBJECTビュー
  DBA_DV_POLICY_OBJECTデータ・ディクショナリ・ビューでは、現在のデータベース・インスタンス内のOracle Database Vaultポリシーによって保護されるオブジェクトに関する情報が示されます。
- DBA_DV_POLICY_OWNERビュー
  DBA_DV_POLICY_OWNERデータ・ディクショナリ・ビューでは、現在のデータベース・インスタンスで作成されたOracle Database Vaultポリシーの所有者が示されます。
- DBA_DV_PREPROCESSOR_AUTHビュー
  DBA_DV_PREPROCESSOR_AUTHデータ・ディクショナリ・ビューには、外部表からプリプロセッサ・プログラムを実行する認可を付与されているユーザーが表示されます。
- DBA_DV_PROXY_AUTHビュー
  DBA_DV_PROXY_AUTHデータ・ディクショナリ・ビューには、DBMS_MACADM.AUTHORIZE_PROXY_USERプロシージャで指定されたプロキシ・ユーザーとスキーマが示されます。

- DBA_DV_PUB_PRIVSビュー
  DBA_DV_PUB_PRIVSデータ・ディクショナリ・ビューは、Oracle Database Vault Administratorで使用されるOracle Database Vault権限管理レポートに反映されるデータが表示されます。
- DBA_DV_REALMビュー
  DBA_DV_REALMデータ・ディクショナリ・ビューには、現行のデータベース・インスタンスで作成されたレルムが表示されます。
- DBA_DV_REALM_AUTHビュー
  DBA_DV_REALM_AUTHデータ・ディクショナリ・ビューには、レルム・オブジェクトにアクセスできる、データベース・ユーザー・アカウントまたはロール認可(GRANTEE)が表示されます。
- DBA_DV_REALM_OBJECTビュー
  DBA_DV_REALM_OBJECTデータ・ディクショナリ・ビューには、データベース・スキーマ、またはレルムによって保護されているスキーマのサブセットが表示されます。
- DBA_DV_ROLEビュー
  DBA_DV_ROLEデータ・ディクショナリ・ビューには、権限管理で使用されるOracle Database Vaultセキュア・アプリケーション・ロールが表示されます。
- DBA_DV_RULEビュー
  DBA_DV_RULEデータ・ディクショナリ・ビューには、定義済のルールが表示されます。
- DBA_DV_RULE_SETビュー
  DBA_DV_RULE_SETデータ・ディクショナリ・ビューには、作成済のルール・セットが表示されます。
- DBA_DV_RULE_SET_RULEビュー
  DBA_DV_RULE_SET_RULEデータ・ディクショナリ・ビューには、既存のルール・セットに関連付けられているルールが表示されます。
- DBA_DV_SIMULATION_LOGビュー
  DBA_DV_SIMULATION_LOGデータ・ディクショナリ・ビューでは、シミュレーション・モードが有効になっているレルムおよびコマンド・ルールのシミュレーション・ログ情報が取得されます。
- DBA_DV_STATUSまたはSYS.DBA_DV_STATUSビュー
  DBA_DV_STATUSまたはSYS.DBA_DV_STATUSデータ・ディクショナリ・ビューには、有効化され構成されているOracle Database Vaultのステータスが示されます。
- DBA_DV_TTS_AUTHビュー
  DBA_DV_TTS_AUTHデータ・ディクショナリ・ビューには、Oracle Data Pumpのトランスポータブル操作を実行するための認可をDBMS_MACADM.AUTHORIZE_TTS_USERプロシージャによって与えられたユーザーが示されます。
- DBA_DV_USER_PRIVSビュー
  DBA_DV_USER_PRIVSデータ・ディクショナリ・ビューには、PUBLICロールによって付与された権限を除くデータベース・ユーザー・アカウントの権限が表示されます。
- DBA_DV_USER_PRIVS_ALLビュー
  DBA_DV_USER_PRIVS_ALLデータ・ディクショナリ・ビューには、PUBLICロールによって付与された権限などデータベース・ユーザー・アカウントの権限が表示されます。
- DVSYS.DV$CONFIGURATION_AUDITビュー
  DVSYS.DV$CONFIGURATION_AUDITデータ・ディクショナリ・ビューは、DVSYS.AUDIT_TRAIL$表の監査証跡レコードを取得します。
- DVSYS.DV$ENFORCEMENT_AUDITビュー
  DVSYS.DV$ENFORCEMENT_AUDITデータ・ディクショナリ・ビューでは、DVSYS.AUDIT_TRAIL$表の強制関連監査の詳細が提供されます。

- DVSYS.DV$REALMビュー
  DVSYS.DV$REALMデータ・ディクショナリ・ビューでは、Oracle Database Vaultのレルムの作成に使用された設定(割り当てられた監査オプションや、レルムが必須レルムかどうかなど)が示されます。
- DVSYS.POLICY_OWNER_COMMAND_RULEビュー
  DVSYS.POLICY_OWNER_COMMAND_RULEデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーに関連付けられているコマンド・ルールについて情報を確認できます。
- DVSYS.POLICY_OWNER_POLICYビュー
  DVSYS.POLICY_OWNER_POLICYデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、他のポリシー所有者によって作成されたポリシーを含め、現在のデータベース・インスタンス内の既存のポリシーの名前、説明および状態などの情報を確認できます。
- DVSYS.POLICY_OWNER_REALMビュー
  POLICY_OWNER_REALMデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーに関連付けられているレルムについて情報を確認できます。
- DVSYS.POLICY_OWNER_REALM_AUTHビュー
  DVSYS.POLICY_OWNER_REALM_AUTHデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーに関連付けられているレルムに与えられた認可についての情報を確認できます。
- DVSYS.POLICY_OWNER_REALM_OBJECTビュー
  DVSYS.POLICY_OWNER_REALM_OBJECTデータ・ディクショナリ・ビューでは、ユーザーが、Database Vaultポリシーに関連付けられているレルムに追加されたオブジェクトについて情報を確認できます。DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。
- DVSYS.POLICY_OWNER_RULEビュー
  DVSYS.POLICY_OWNER_RULEデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、ルール名とその式など、Database Vaultポリシー内のルール・セットに関連付けられているルールについて情報を確認できます。DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。
- DVSYS.POLICY_OWNER_RULE_SETビュー
  DVSYS.POLICY_OWNER_RULE_SETデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーに関連付けられているルール・セットについて情報を確認できます。
- DVSYS.POLICY_OWNER_RULE_SET_RULEビュー
  DVSYS.POLICY_OWNER_RULE_SET_RULEデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーで使用されるルールを含むルール・セットについて情報を確認できます。
- AUDSYS.DV$CONFIGURATION_AUDITビュー
  AUDSYS.DV$CONFIGURATION_AUDITビューは、統合監査証跡のDatabase Vault監査レコードを取得する点を除き、DVSYS.DV$CONFIGURATION_AUDITビューとほぼ同じです。
- AUDSYS.DV$ENFORCEMENT_AUDITビュー
  AUDSYS.DV$ENFORCEMENT_AUDITビューは、統合監査証跡のDatabase Vault監査レコードを取得する点を除き、DVSYS.DV$ENFORCEMENT_AUDITビューとほぼ同じです。

24.1 Oracle Database Vaultのデータ・ディクショナリ・ビューについて

Oracle Database Vaultには、DV_SECANALYSTロールまたはDV_ADMINロールを介してアクセスできる一連のDBAスタイルのデータ・ディクショナリ・ビューが用意されています。

これらのビューは、DVSYSスキーマおよびLBACSYSスキーマの様々な基礎となるOracle Database Vaultの表に、存在する

可能性がある主キーおよび外部キーの列を公開せずにアクセスできます。これらのビューは、コア表に格納されているコードまたは関連表のコードに対するラベルを取得しなければならない結合を実行する必要なしに、データベース管理ユーザーがOracle Database Vault構成の状態についてレポートを作成するためのものです。

関連項目:

Oracle Database Vaultでレポートを実行する場合は、「Oracle Database Vaultレポート」を参照してください。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.2 CDB_DV_STATUSビュー

CDB_DV_STATUSデータ・ディクショナリ・ビューには、すべてのPDBのDatabase Vault操作の制御、構成および有効化のステータスが表示されます。

DBAロールを付与されたユーザーなど、Oracle Database管理ユーザーのみ、このビューを問合せできます。Database Vault管理者は、このビューにアクセスできません。

たとえば:

```
SELECT * FROM CDB_DV_STATUS;
```

次のような出力が表示されます。

```
NAME                 STATUS  CON_ID
-------------------- ------- ------
DV_APP_PROTECTION    ENABLED      5
DV_CONFIGURE_STATUS  TRUE         5
DV_ENABLE_STATUS     TRUE         5
```

関連するビュー

. DBA_DV_STATUSまたはSYS.DBA_DV_STATUSビュー

列	データ型	Null	説明
NAME	VARCHAR2(19)	NOT NULL	次の設定のいずれかを示します。 ● DV_APP_PROTECTION は、Database Vault 操作の制御が有効か無効かを示します。 ● DV_CONFIGURE_STATUS は、Oracle Database Vault が構成されているかどうか(つまり、CONFIGURE_DV プロシージャが使用されたかどうか)を示します。 ● DV_ENABLE_STATUS は、Oracle Database Vault が有効になっているかどうか(つまり、DBMS_MACADM.ENABLE_DV プロ

列	データ型	Null	説明
			シージャが使用されたかどうか)を示します。
STATUS	VARCHAR2(64)	NOT NULL	DV_CONFIGURE_STATUS および DV_ENABLE_STATUS の場合、TRUE は Oracle Database Vault が構成され有効になっていることを意味し、FALSE はそうでないことを意味します。 DV_APP_PROTECTION の場合、出力は ENABLED または DISABLED です。
CON_ID	NUMBER	NOT NULL	Oracle Database Vault が使用されている PDB コンテナの識別番号

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.3 DBA_DV_APP_EXCEPTIONビュー

DBA_DV_APP_EXCEPTIONデータ・ディクショナリ・ビューには、Database Vault操作の制御の例外リストにある共通スキーマとパッケージ名がリストされます。

このビューはCDBルートからのみ問い合せる必要があります。プラガブル・データベース(PDB)からこのビューを問い合せようとすると、出力は表示されません。

たとえば:

```
SELECT * FROM DBA_DV_APP_EXCEPTION WHERE GRANTEE = 'C##HR_ADMIN';
```

次のような出力が表示されます。

```
GRANTEE        PACKAGE_NAME
-------------- ------------
C##HR_ADMIN     PATCH_APP
```

列	データ型	Null	説明
GRANTEE	VARCHAR(128)	NOT NULL	権限受領者の名前 共通ユーザーの名前を検索するには、DBA_USERS データ・ディクショナリ・ビューの USERNAME および COMMON 列を問い合せます。
PACKAGE_NAME	VARCHAR(128)	NOT NULL	パッケージの名前

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.4 DBA_DV_CODEビュー

DBA_DV_CODEデータ・ディクショナリ・ビューには、ユーザー・インタフェース、エラー・メッセージおよび制約チェックの一般的な参照コードが示されます。

これらのコードは、ユーザー・インタフェースやビュー、変換可能な形式の入力の検証に使用されます。

たとえば:

```
SELECT CODE, VALUE FROM DBA_DV_CODE WHERE CODE_GROUP = 'BOOLEAN';
```

次のような出力が表示されます。

```
CODE    VALUE
------- --------
Y       True
N       False
```

列	データ型	Null	説明
CODE_GROUP	VARCHAR(128)	NOT NULL	表 24-1 に示されているコード・グループのいずれかを表示します
CODE	VARCHAR(128)	NOT NULL	ブール・コードが使用されます。Y(YES)または N(NO)のいずれかです。
VALUE	VARCHAR(4000)	NULL	ブール値が使用されます。ブール・コードが Y の場合は True で、ブール・コードが N の場合は False です。
LANGUAGE	VARCHAR(3)	NOT NULL	この Oracle Database Vault インストールの言語。 サポートされる言語は次のとおりです。 ● en: 英語 ● de: ドイツ語 ● es: スペイン語 ● fr: フランス語 ● it: イタリア語 ● ja: 日本語 ● ko: 韓国語 ● pt_BR: ポルトガル語(ブラジル) ● zh_CN: 簡体字中国語

列	データ型	Null	説明
			● zh_TW: 繁体字中国語
DESCRIPTION	VARCHAR(1024)	NULL	コード・グループの簡単な説明。

表24-1に、DBA_DV_CODEデータ・ディクショナリ・ビューのCODE_GROUP列で使用可能な値を示します。

表24-1 DBA_DV_CODEビューのCODE_GROUPの値

CODE_GROUP名	説明
AUDIT_EVENTS	カスタム・イベント監査証跡レコードに使用されるアクション番号およびアクション名を含む
BOOLEAN	単純な Yes/No または True/False 参照
DB_OBJECT_TYPE	レルム・オブジェクトおよびコマンド認可に使用できるデータベース・オブジェクト・タイプ
SQL_CMDS	コマンド・ルールによって保護される DDL コマンド
FACTOR_AUDIT	ファクタ取得処理の監査オプション
FACTOR_EVALUATE	ファクタ取得用の評価オプション(セッションごとまたはアクセスごと)
FACTOR_FAIL	ファクタ取得メソッドが失敗した場合にエラーを伝播するためのオプション
FACTOR_IDENTIFY	ファクタ識別子の解決方法(メソッドごとまたはファクタごとなど)を決定するためのオプション
FACTOR_LABEL	セッション確立フェーズでファクタ識別子をラベル付けする方法を決定するためのオプション
LABEL_ALG	ポリシーごとにデータベース・セッションの最大セッション・ラベルを決定するために使用されるアルゴリズム。Oracle Label Security マージ・アルゴリズム・コードのリストは、表19-2 を参照してください。
OPERATORS	アイデンティティ・マップに使用されるブール演算子
REALM_AUDIT	レルム・アクセスまたはレルム違反を監査するためのオプション
REALM_OPTION	レルムの所有権のオプション

CODE_GROUP名	説明
RULESET_AUDIT	ルール・セットの実行またはルール・セット・エラーを監査するためのオプション
RULESET_EVALUATE	関連付けられたルールのすべてが true か、関連付けられたルールのいずれかが true かによって、ルール・セットの成功または失敗を決定するためのオプション
RULESET_EVENT	ルール・セットが成功または失敗と評価される場合にカスタム・イベント・ハンドラを起動するためのオプション
RULESET_FAIL	ルール・セットの失敗のランタイム表示を決定するためのオプション

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.5 DBA_DV_COMMAND_RULEビュー

DBA_DV_COMMAND_RULEデータ・ディクショナリ・ビューには、コマンド・ルールにより保護されるSQL文が表示されます。

コマンド・ルールの詳細は、「コマンド・ルールの構成」を参照してください。

たとえば:

```
SELECT COMMAND, RULE_SET_NAME FROM DBA_DV_COMMAND_RULE;
```

次のような出力が表示されます。

```
COMMAND          RULE_SET_NAME
---------------- ------------------------------
GRANT            Can Grant VPD Administration
REVOKE           Can Grant VPD Administration
ALTER SYSTEM     Allow System Parameters
ALTER USER       Can Maintain Own Account
CREATE USER      Can Maintain Account/Profiles
DROP USER        Can Maintain Account/Profiles
CREATE PROFILE   Can Maintain Account/Profiles
DROP PROFILE     Can Maintain Account/Profiles
ALTER PROFILE    Can Maintain Account/Profiles
```

列	データ型	Null	説明
COMMAND	VARCHAR(128)	NOT NULL	コマンド・ルールの名前。デフォルトのコマンド・ルールのリストは、「デフォルトのコマンド・ルール」を参照してください。
CLAUSE_NAME	VARCHAR(100)	NOT NULL	コマンド・ルールの作成に使用された、ALTER SYSTEM または ALTER SESSION SQL 文のどちらかの句。たとえば、ALTER SESSION 文の SET 句をリストできます。 使用可能な句の値をすべて示すリストについては、次のトピックを参照してください。

列	データ型	Null	説明
			● 表 16-2 ● 表 16-3
PARAMETER_NAME	VARCHAR(128)	NOT NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールの CLAUSE_NAME 設定からのパラメータ。
EVENT_NAME	VARCHAR(128)	NOT NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールで定義されているイベント。
COMPONENT_NAME	VARCHAR(128)	NOT NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールの EVENT_NAME 設定のコンポーネント。
ACTION_NAME	VARCHAR(128)	NOT NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールの EVENT_NAME 設定のアクション。
RULE_SET_NAME	VARCHAR(128)	NOT NULL	このコマンド・ルールに関連付けられたルール・セットの名前。
OBJECT_OWNER	VARCHAR(128)	NOT NULL	コマンド・ルールが影響するオブジェクトの所有者。
OBJECT_NAME	VARCHAR(128)	NOT NULL	コマンド・ルールが影響するデータベース・オブジェクト(データベース表など)の名前。
ENABLED	VARCHAR(1)	NOT NULL	有効な値は次のとおりです。 ● Y はコマンド・ルールが有効になっていることを示します ● N はそれが無効になっていることを示します ● S はそれがシミュレーション・モードになっていることを示します
PRIVILEGE_SCOPE	NUMBER	NOT NULL	廃止された列
COMMON	VARCHAR(3)	NOT NULL	マルチテナント環境の場合は、コマンド・ルールがローカルか共通かを示します。使用される値は、次のとおりです。 ● コマンド・ルールが共通の場合は YES ● コマンド・ルールがローカルの場合は NO

列	データ型	Null	説明
INHERITED	VARCHAR(3)	NOT NULL	COMMON 列の出力が YES の場合は、コマンド・ルールの継承ステータスを示します。値は次のとおりです。 ● YES は、コマンド・ルールが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、コマンド・ルールがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
ID#	NUMBER	NOT NULL	コマンド・ルールの ID 番号。これは、コマンド・ルール作成時に自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR(3)	NULL	コマンド・ルールがデフォルト(つまり、Oracle によって提供されている)コマンド・ルールであるかユーザーが作成したコマンド・ルールであるかを示します。使用される値は、次のとおりです。 ● コマンド・ルールがデフォルト・コマンド・ルールである場合は YES ● コマンド・ルールがユーザーが作成したコマンド・ルールである場合は NO
PL_SQL_STACK	VARCHAR(3)	NULL	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックが記録されているかどうかを示します。TRUE は PL/SQL スタックが記録されていることを示し、FALSE は PL/SQL スタックが記録されていないことを示します。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.6 DBA_DV_DATAPUMP_AUTHビュー

DBA_DV_DATAPUMP_AUTHデータ・ディクショナリ・ビューでは、Oracle Database Vault環境でOracle Data Pumpを使用するための認可が示されます。

詳細は、「Oracle Database VaultでのOracle Data Pumpの使用」を参照してください。

たとえば:

```
SELECT * FROM DBA_DV_DATAPUMP_AUTH WHERE GRANTEE = 'PRESTON';
```

次のような出力が表示されます。

```
GRANTEE SCHEMA OBJECT  TYPE  ACTION
------- ------ ------- ----- -----------
PRESTON OE     ORDERS  %     CREATE_USER
```

列	データ型	Null	説明
GRANTEE	VARCHAR2(128)	NOT NULL	データ・ポンプの認可を付与されたユーザーの名前。
SCHEMA	VARCHAR2(128)	NOT NULL	ユーザーGRANTEE がデータ・ポンプ操作の実行を認可されるスキーマの名前。
OBJECT	VARCHAR2(128)	NOT NULL	GRANTEE ユーザーがデータ・ポンプの認可(表など)を持つ SCHEMA パラメータで指定されるスキーマ内のオブジェクトの名前。
TYPE	VARCHAR2(32)	NOT NULL	Oracle Data Pump インポート操作の場合、付与のタイプ(ROLE など)を示します。
ACTION	VARCHAR2(30)	NOT NULL	Oracle Data Pump インポート操作の場合、TYPE に関連付けられているアクション(%、TABLE、CREATE_USER または GRANT)を示します。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.7 DBA_DV_DBCAPTURE_AUTHビュー

DBA_DV_DBCAPTURE_AUTHデータ・ディクショナリ・ビューには、Oracle Database Replayのワークロード取得操作を実行する認可が付与されているユーザーが表示されます。

詳細は、Oracle Database VaultでのOracle Database Replayの使用を参照してください。

たとえば:

```
SELECT * FROM DBA_DV_DBCAPTURE_AUTH WHERE GRANTEE = 'PFITCH';
```

次のような出力が表示されます。

```
GRANTEE
-------
PFITCH
```

列	データ型	Null	説明
GRANTEE	VARCHAR2(128)	NOT NULL	Database Replay のワークロード取得認可を付与

列	データ型	Null	説明
			されているユーザーの名前

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.8 DBA_DV_DBREPLAYビュー

DBA_DV_DBREPLAY_AUTHデータ・ディクショナリ・ビューには、Oracle Database Replayのワークロード・リプレイ操作を実行する認可が付与されているユーザーが表示されます。

詳細は、Oracle Database VaultでのOracle Database Replayの使用を参照してください。

たとえば:

```
SELECT * FROM DBA_DV_DBREPLAY_AUTH WHERE GRANTEE = 'PFITCH';
```

次のような出力が表示されます。

```
GRANTEE
-------
PFITCH
```

列	データ型	Null	説明
GRANTEE	VARCHAR2(128)	NOT NULL	Database Replay のワークロード・リプレイ認可を付与されているユーザーの名前

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.9 DBA_DV_DDL_AUTHビュー

DBA_DV_DDLデータ・ディクショナリ・ビューには、DBMS_MACADM.AUTHORIZE_DDLプロシージャで指定されたユーザーとスキーマが示されます。

このプロシージャにより、データ定義言語(DDL)文を実行する認可がユーザーに付与されます。

たとえば:

```
SELECT * FROM DBA_DV_DDL_AUTH WHERE GRANTEE = 'psmith';
```

次のような出力が表示されます。

```
GRANTEE SCHEMA
------- ------
PSMITH  HR
```

列	データ型	Null	説明
GRANTEE	VARCHAR2(128)	NOT NULL	DDL 認可を付与されたユーザーの名前。
SCHEMA	VARCHAR2(128)	NOT NULL	ユーザーGRANTEE が DDL 操作の実行を認可され

列	データ型	Null	説明
			るスキーマの名前。

関連項目:

- AUTHORIZE_DDLプロシージャ
- UNAUTHORIZE_DDLプロシージャ

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.10 DBA_DV_DICTIONARY_ACCTSビュー

DBA_DV_DICTIONARY_ACCTSデータ・ディクショナリ・ビューには、ユーザーがDVSYSスキーマ・アカウントとDVFスキーマ・アカウントに直接ログインできるかどうかが示されます。

たとえば:

```
SELECT * FROM DBA_DV_DICTIONARY_ACCTS;
```

次のような出力が表示されます。

```
STATE
-------
ENABLED
```

列	データ型	Null	説明
STATE	VARCHAR2(8)	NOT NULL	ユーザーが DVSYS スキーマと DVF スキーマに直接ログインできるかどうかについて説明します。使用される値は、次のとおりです。 ● ENABLED は、ユーザーが DVSYS スキーマと DVF スキーマに直接ログインできることを示します。 ● DISABLED は、ユーザーが DVSYS スキーマと DVF スキーマに直接ログインできないことを示します。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.11 DBA_DV_FACTORビュー

DBA_DV_FACTORデータ・ディクショナリ・ビューには、現行のデータベース・インスタンス内の既存のファクタが表示されます。

たとえば:

```
SELECT NAME, GET_EXPR FROM DBA_DV_FACTOR WHERE NAME = 'Session_User';
```

次のような出力が表示されます。

```
NAME           GET_EXPR
-------------- ------------------------------------------
Session_User   UPPER(SYS_CONTEXT('USERENV', 'SESSION_USER'))
```

関連するビュー

- DBA_DV_FACTOR_LINKビュー
- DBA_DV_FACTOR_TYPEビュー

列	データ型	Null	説明
NAME	VARCHAR2(128)	NOT NULL	ファクタの名前。デフォルトのファクタのリストは、「デフォルトのファクタ」を参照してください。
DESCRIPTION	VARCHAR2(4000)	NULL	ファクタの説明。
FACTOR_TYPE_NAME	VARCHAR2(128)	NOT NULL	ファクタの目的を分類するために使用されるファクタのカテゴリ。
ASSIGN_RULE_SET_NAME	VARCHAR2(128)	NULL	ファクタのアイデンティティを制御するために使用されるルール・セット。
GET_EXPR	VARCHAR2(1024)	NULL	ファクタのアイデンティティを取得する PL/SQL 式。
VALIDATE_EXPR	VARCHAR2(1024)	NULL	ファクタのアイデンティティの検証に使用される PL/SQL 式。ブール値を返します。
IDENTIFIED_BY	NUMBER	NOT NULL	GET_EXPR 列に示された式に基づいてファクタのアイデンティティが決まります。使用される値は、次のとおりです。 ● 0: 定数 ● 1: メソッド ● 2: ファクタ
IDENTIFIED_BY_MEANING	VARCHAR2(4000)	NULL	IDENTIFIED_BY 列の対応する値にテキスト説明が提供されます。使用される値は、次のとおりです。 ● 定数: IDENTIFIED_COLUMN が 0 の

列	データ型	Null	説明
			場合 ● メソッド: IDENTIFIED_COLUMN が 1 の場合 ● ファクタ: IDENTIFIED_COLUMN が 2 の場合
LABELED_BY	NUMBER	NOT NULL	ファクタのラベル付けが決まります。 ● 0: Oracle Label Security ポリシーに関連付けられているラベルから直接ファクタのアイデンティティをラベル付けします。 ● 1: 子ファクタ・アイデンティティのラベルからファクタ・アイデンティティ・ラベルを導出します。
LABELED_BY_MEANING	VARCHAR2(4000)	NULL	LABELED_BY 列の対応する値にテキスト説明が提供されます。使用される値は、次のとおりです。 ● 自己: LABELED_BY 列が 0 の場合 ● ファクタ: LABELED_BY 列が 1 の場合
EVAL_OPTIONS	NUMBER	NOT NULL	ユーザーのログイン時にファクタを評価する方法が決定されます。 ● 0: データベース・セッションの作成時 ● 1: ファクタがアクセスされる際に毎回 ● 2: 起動時
EVAL_OPTIONS_MEANING	VARCHAR2(4000)	NULL	EVAL_OPTIONS 列の対応する値にテキスト説明が提供されます。使用される値は、次のとおりです。 ● セッション: EVAL_OPTIONS が 0 の場合 ● アクセス: EVAL_OPTIONS が 1 の場

列	データ型	Null	説明
			合 ● 起動時: EVAL_OPTIONS が 2 の場合
AUDIT_OPTIONS	NUMBER	NOT NULL	カスタムの Oracle Database Vault 監査レコードから生成する場合にファクタを監査するオプション。使用される値は、次のとおりです。 ● 0: 監査は設定されていません ● 1: 常に監査します ● 2: get_expr がエラーを返した場合に監査します。 ● 4: get_expr が null の場合に監査します。 ● 8: 検証プロシージャがエラーを返した場合に監査します。 ● 16: 検証プロシージャが false の場合に監査します。 ● 32: 信頼レベルが設定されていない場合に監査します。 ● 64: 信頼レベルが負の場合に監査します。
FAIL_OPTIONS	NUMBER	NOT NULL	ファクタ・エラーをレポートするオプション。 ● 1: エラー・メッセージを表示します。 ● 2: エラー・メッセージを表示しません。
FAIL_OPTIONS_MEANING	VARCHAR2(4000)	NULL	FAIL_OPTIONS 列の対応する値にテキスト説明が提供されます。使用される値は、次のとおりです。 ● エラー・メッセージを表示 ● エラー・メッセージを表示しない:

列	データ型	Null	説明
ID#	NUMBER	NOT NULL	ファクタの ID 番号。これは、ファクタ作成時に自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR(3)	NOT NULL	ファクタがデフォルト(つまり、Oracle によって提供されている)ファクタであるかユーザーが作成したファクタであるかを示します。使用される値は、次のとおりです。 ● ファクタがデフォルト・ファクタである場合は YES ● ファクタがユーザーが作成したファクタである場合は NO

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.12 DBA_DV_FACTOR_TYPEビュー

DBA_DV_FACTOR_TYPEデータ・ディクショナリ・ビューには、システムで使用されているファクタ・タイプの名前と説明が表示されます。

たとえば:

```
SELECT * FROM DBA_DV_FACTOR_TYPE WHERE NAME = 'Time';
```

次のような出力が表示されます。

```
NAME       DESCRIPTION
---------- --------------------------------------------------------------------
Time       Time-based factor
```

関連するビュー

- DBA_DV_FACTORビュー
- DBA_DV_FACTOR_LINKビュー

列	データ型	Null	説明
NAME	VARCHAR(128)	NOT NULL	ファクタ・タイプの名前。
DESCRIPTION	VARCHAR(1024)	NULL	ファクタ・タイプの説明。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.13 DBA_DV_FACTOR_LINKビュー

DBA_DV_FACTOR_LINKデータ・ディクショナリ・ビューには、子ファクタの関連によりアイデンティティが決まる各ファクタの関係が

表示されます。

このビューでは、各親ファクタおよび子ファクタに1つのエントリが含まれます。このビューを使用して、ファクタ・リンクからアイデンティティ・マップに関係を解決できます。

たとえば:

```
SELECT PARENT_FACTOR_NAME, CHILD_FACTOR_NAME FROM DBA_DV_FACTOR_LINK;
```

次のような出力が表示されます。

```
PARENT_FACTOR_NAME              CHILD_FACTOR_NAME
------------------------------ ------------------------------
Domain                          Database_Instance
Domain                          Database_IP
Domain                          Database_Hostname
```

関連するビュー

- DBA_DV_FACTORビュー
- DBA_DV_FACTOR_TYPEビュー

列	データ型	Null	説明
PARENT_FACTOR_NAME	VARCHAR(128)	NOT NULL	親ファクタの名前
CHILD_FACTOR_NAME	VARCHAR(128)	NOT NULL	親ファクタの子ファクタの名前
LABEL_IND	VARCHAR(1)	NOT NULL	親ファクタにリンクされた子ファクタが、Oracle Label Security 統合での親ファクタのラベルに含まれるかどうかを示します。使用される値は、次のとおりです。 ● Y(はい) ● N(いいえ)

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.14 DBA_DV_IDENTITYビュー

DBA_DV_IDENTITYデータ・ディクショナリ・ビューには、各ファクタのアイデンティティが表示されます。

たとえば:

```
SELECT * FROM DBA_DV_IDENTITY WHERE VALUE = 'GLOBAL SHARED';
```

1つのファクタ・アイデンティティのみ作成されている仮定して、次のような出力結果が表示されます。

```
FACTOR_NAME          VALUE          TRUST_LEVEL
----------------     -------------- ------------
Identification_Type  GLOBAL SHARED  1
```

関連するビュー

- DBA_DV_FACTORビュー
- DBA_DV_IDENTITY_MAPビュー

列	データ型	Null	説明
FACTOR_NAME	VARCHAR(128)	NOT NULL	ファクタの名前。
VALUE	VARCHAR(1024)	NOT NULL	ファクタの値。
TRUST_LEVEL	NUMBER	NOT NULL	同じファクタの別のアイデンティティと比較した信頼の度合いを示す数値。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.15 DBA_DV_IDENTITY_MAPビュー

DBA_DV_IDENTITY_MAPデータ・ディクショナリ・ビューには、各ファクタ・アイデンティティのマップが表示されます。

このビューには、別のファクタによって識別されるファクタの、親と子のファクタの組合せリンクへのマッピングが含まれます。各ファクタでは、マップはOR演算子で結合され、異なるファクタでは、AND演算子で結合されます。

このビューを使用して、他のファクタ(ドメインなど)によって識別されるファクタ、または連続するドメイン(年齢や温度など)のあるファクタのアイデンティティを解決できます。

たとえば:

```
SELECT FACTOR_NAME, IDENTITY_VALUE FROM DBA_DV_IDENTITY_MAP;
```

次のような出力が表示されます。

```
FACTOR_NAME      IDENTITY_VALUE
---------------- --------------------
Sector2_Program  Accounting-Sensitive
```

関連するビュー

- DBA_DV_FACTORビュー
- DBA_DV_IDENTITYビュー

列	データ型	Null	説明
FACTOR_NAME	VARCHAR(128)	NOT NULL	アイデンティティ・マップの対象のファクタ。
IDENTITY_VALUE	VARCHAR(1024)	NOT NULL	アイデンティティ・マップの評価が TRUE の場合は、ファクタで想定される値。
OPERATION_CODE	VARCHAR(128)	NOT NULL	OPERATION_VALUE 列における操作の説明的な名前。

列	データ型	Null	説明
OPERATION_VALUE	VARCHAR(4000)	NULL	アイデンティティ・マップの関係演算子(たとえば、<、>、=など)。
OPERAND1	VARCHAR(1024)	NULL	関係演算子の左オペランド。入力する下限値を表します。
OPERAND2	VARCHAR(1024)	NULL	関係演算子の右オペランド。入力する上限値を表します。
PARENT_FACTOR_NAME	VARCHAR(128)	NULL	マップが関連する親ファクタ・リンク。
CHILD_FACTOR_NAME	VARCHAR(128)	NULL	マップが関連する子ファクタ・リンク。
LABEL_IND	VARCHAR(1)	NULL	親ファクタにリンクされた子ファクタが、Oracle Label Security 統合での親ファクタのラベルに含まれるかどうかを示します。使用される値は、次のとおりです。 ● Y(はい) ● N(いいえ)

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.16 DBA_DV_JOB_AUTHビュー

DBA_DV_JOB_AUTHデータ・ディクショナリ・ビューでは、Oracle Database Vault環境でOracle Schedulerを使用するための認可が示されます。

たとえば:

```
SELECT * FROM DBA_DV_JOB_AUTH WHERE GRANTEE = 'PRESTON';
```

次のような出力が表示されます。

```
GRANTEE SCHEMA
------- ------
PRESTON OE
```

列	データ型	Null	説明
GRANTEE	VARCHAR2(128)	NOT NULL	Oracle Scheduler 認可を付与されたユーザーの名前。
SCHEMA	VARCHAR2(128)	NOT NULL	ユーザーGRANTEE が Oracle Scheduler 操作の実

列	データ型	Null	説明
			行を認可されるスキーマの名前。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.17 DBA_DV_MAC_POLICYビュー

DBA_DV_MAC_POLICYデータ・ディクショナリ・ビューには、Oracle Database Vaultで使用するために定義されたOracle Label Securityポリシーが表示されます。

たとえば:

```
SELECT POLICY_NAME, ALGORITHM_CODE, ALGORITHM_MEANING
 FROM DBA_DV_MAC_POLICY;
```

次のような出力が表示されます。

```
POLICY_NAME     ALGORITHM_CODE    ALGORITHM_MEANING
--------------- ----------------- --------------------------------
ACCESS_DATA     LUI               Minimum Level/Union/Intersection
```

関連するビュー

- DBA_DV_MAC_POLICY_FACTORビュー
- DBA_DV_POLICY_LABELビュー

列	データ型	Null	説明
POLICY_NAME	VARCHAR(128)	NOT NULL	ポリシーの名前。
ALGORITHM_CODE	VARCHAR(128)	NOT NULL	ポリシーに使用されるマージ・アルゴリズム・コード。アルゴリズム・コードのリストは、表 19-2 を参照してください。
ALGORITHM_MEANING	VARCHAR(4000)	NULL	ALGORITHM_CODE 列の対応する値にテキスト説明が提供されます。アルゴリズム・コードの説明のリストは、表 19-2 を参照してください。
ERROR_LABEL	VARCHAR(4000)	NULL	初期化エラーに対して指定されたラベル。セッションの初期化中に構成エラーまたはランタイム・エラーが発生すると設定されます。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.18 DBA_DV_MAC_POLICY_FACTORビュー

DBA_DV_MAC_POLICYデータ・ディクショナリ・ビューには、Oracle Label Securityポリシーに関連付けられているファクタが表示されます。

このビューを使用して、DBA_DV_MAC_POLICYビューを使用する各ポリシーの最大セッション・ラベルを構成しているのがどのファクタであるかを特定できます。

たとえば:

```
SELECT * FROM DBA_DV_MAC_POLICY_FACTOR;
```

次のような出力が表示されます。

```
FACTOR_NAME    MAC_POLICY_NAME
-------------- ------------------
App_Host_Name  Access Locations
```

関連するビュー

- DBA_DV_MAC_POLICYビュー
- DBA_DV_POLICY_LABELビュー

列	データ型	Null	説明
FACTOR_NAME	VARCHAR(128)	NOT NULL	ファクタの名前
MAC_POLICY_NAME	VARCHAR(128)	NOT NULL	このファクタに関連付けられている Oracle Label Security ポリシーの名前

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.19 DBA_DV_MAINTENANCE_AUTHビュー

DBA_DV_MAINTENANCE_AUTHデータ・ディクショナリ・ビューでは、情報ライフサイクル管理(ILM)機能を使用するためのOracle Database Vault認可の構成について情報が示されます。

たとえば:

```
SELECT GRANTEE, ACTION STATE FROM DBA_DV_MAINTENANCE_AUTH;
```

次のような出力が表示されます。

```
GRANTEE                   ACTION
------------------------- --------
PSMITH                    ILM
```

列	データ型	Null	説明
GRANTEE	VARCHAR(128)	NOT NULL	権限受領者の名前
SCHEMA	VARCHAR(128)	NOT NULL	スキーマ名または% (すべてのスキーマ)
OBJECT	VARCHAR(128)	NOT NULL	オブジェクト名または% (スキーマ内のすべてのオブジェクト)

列	データ型	Null	説明
OBJECT_TYPE	VARCHAR(30)	NOT NULL	オブジェクト・タイプ
ACTION	VARCHAR(30)	NOT NULL	ILM 操作の場合は、メンテナンス・アクション ILM

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.20 DBA_DV_ORADEBUGビュー

DBA_DV_ORADEBUGデータ・ディクショナリ・ビューには、ユーザーがOracle Database Vault環境でORADEBUGユーティリティを使用できるかどうかが示されます。

たとえば:

```
SELECT * FROM DBA_DV_ORADEBUG;
```

次のような出力が表示されます。

```
STATE
--------
DISABLED
```

列	データ型	Null	説明
STATE	VARCHAR2(8)	NOT NULL	Database Vault が有効な環境で ORADEBUG ユーティリティを使用できるかどうかについて説明します。使用される値は、次のとおりです。 ● ENABLED は、ユーザーが ORADEBUG ユーティリティを実行できることを示します。 ● DISABLED は、ユーザーが ORADEBUG ユーティリティを実行できないことを示します。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.21 DBA_DV_PATCH_ADMIN_AUDITビュー

DBA_DV_PATCH_ADMIN_AUDITデータ・ディクショナリ・ビューには、DV_ADMIN_PATCHロールを付与されているユーザーに対して監査が有効か無効かが示されます。

DBMS_MACADM.ENABLE_DV_PATCH_ADMIN_AUDITプロシージャは、このタイプの監査を有効にします。

たとえば:

```
SELECT * FROM DBA_DV_PATCH_ADMIN_AUDIT;
```

次のような出力が表示されます。

```
STATE
--------
```

```
DISABLED
```

列	データ型	Null	説明
STATE	VARCHAR2(8)	NOT NULL	DV_ADMIN_PATCH ロールのユーザーに対して監査が有効か無効かについて説明します。使用される値は、次のとおりです。 ● ENABLED は、監査が有効になっていることを示します。 ● DISABLED は、監査が無効になっていることを示します。

関連項目:

- ENABLE_DV_PATCH_ADMIN_AUDITプロシージャ
- DISABLE_DV_PATCH_ADMIN_AUDITプロシージャ

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.22 DBA_DV_POLICYビュー

DBA_DV_POLICYデータ・ディクショナリ・ビューでは、現在のデータベース・インスタンスで作成されたOracle Database Vaultポリシーが示されます。

たとえば:

```
SELECT POLICY_NAME, STATE FROM DBA_DV_POLICY
  WHERE STATE = 'ENABLED';
```

次のような出力が表示されます。

```
POLICY_NAME                       STATE
--------------------------------- -------
Oracle Account Management Controls ENABLED
Oracle System Protection Controls  ENABLED
```

関連するビュー

- DBA_DV_POLICY_OWNERビュー
- DBA_DV_POLICY_OBJECTビュー
- DBA_DV_SIMULATION_LOGビュー
- DVSYS.POLICY_OWNER_POLICYビュー

列	データ型	Null	説明
POLICY_NAME	VARCHAR(128)	NOT NULL	作成された Oracle Database Vault ポリシーの名前。デフォルト・ポリシーのリストは、デフォルトの Oracle Database

列	データ型	Null	説明
			Vault ポリシーを参照してください。
DESCRIPTION	VARCHAR(1024)	NULL	作成されたポリシーの説明
STATE	VARCHAR(8)	NULL	ポリシーが有効かどうかを指定します。使用される値は、次のとおりです。 ● ENABLED ● DISABLED ● SIMULATION
ID#	VARCHAR(1)	NOT NULL	ポリシー作成時にポリシーに割り当てられた、システムによって生成された ID。
ORACLE_SUPPLIED	VARCHAR(3)	NULL	ポリシーがデフォルトの Oracle Database Vault ポリシーであるかどうかを示します。
PL_SQL_STACK	VARCHAR(3)	NULL	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックが記録されているかどうかを示します。TRUE は PL/SQL スタックが記録されていることを示し、FALSE は PL/SQL スタックが記録されていないことを示します。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.23 DBA_DV_POLICY_LABELビュー

DBA_DV_POLICY_LABELデータ・ディクショナリ・ビューには、各ポリシーのDBA_DV_IDENTITYビューの各ファクタ識別子に対するOracle Label Securityラベルが表示されます。

たとえば:

```
SELECT * FROM DBA_DV_POLICY_LABEL;
```

次のような出力が表示されます。

```
IDENTITY_VALUE    FACTOR_NAME      POLICY_NAME        LABEL
----------------- --------------   -----------------  ---------
App_Host_Name     Sect2_Fin_Apps   Access Locations   Sensitive
```

関連するビュー

- DBA_DV_MAC_POLICYビュー
- DBA_DV_MAC_POLICY_FACTORビュー

列	データ型	Null	説明
IDENTITY_VALUE	VARCHAR(1024)	NOT NULL	ファクタ識別子の名前。
FACTOR_NAME	VARCHAR(128)	NOT NULL	ファクタ識別子に関連付けられているファクタの名前。
POLICY_NAME	VARCHAR(128)	NOT NULL	このファクタに関連付けられている Oracle Label Security ポリシーの名前。
LABEL	VARCHAR(4000)	NOT NULL	ポリシーに関連付けられている Oracle Label Security ラベルの名前。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.24 DBA_DV_POLICY_OBJECTビュー

DBA_DV_POLICY_OBJECTデータ・ディクショナリ・ビューでは、現在のデータベース・インスタンス内のOracle Database Vaultポリシーによって保護されるオブジェクトに関する情報が示されます。

たとえば:

```
SELECT POLICY_NAME, OBJECT_TYPE FROM DBA_DV_POLICY_OBJECT WHERE POLICY_NAME LIKE
'%Protection Controls';
```

次のような出力が表示されます。

```
POLICY_NAME                       OBJECT_TYPE
--------------------------------- ------------
Oracle System Protection Controls  REALM
```

関連するビュー

- DBA_DV_POLICYビュー
- DBA_DV_POLICY_OWNERビュー

列	データ型	Null	説明
POLICY_NAME	VARCHAR(128)	NOT NULL	作成された Oracle Database Vault ポリシーの名前。 デフォルト・ポリシーのリストは、デフォルトの Oracle Database Vault ポリシーを参照してください。
OBJECT_TYPE	VARCHAR(12)	NULL	REALM など、保護されているオブジェクトのタイプ
COMMAND	VARCHAR(128)	NULL	Database Vault ポリシーによって保護されるコマンド・ルールの名前

列	データ型	Null	説明
COMMAND_OBJ_OWNER	VARCHAR(128)	NULL	Database Vault ポリシーに関連付けられているオブジェクト所有者の名前
COMMAND_OBJ_NAME	VARCHAR(128)	NULL	Database Vault ポリシーに関連付けられているオブジェクトの名前
COMMAND_CLAUSE	VARCHAR(100)	NULL	コマンド・ルールの作成に使用された、ALTER SYSTEM または ALTER SESSION SQL 文のどちらかの句。たとえば、ALTER SESSION 文の SET 句をリストできます。 使用可能な句の値をすべて示すリストについては、次のトピックを参照してください。 ● 表 16-2 ● 表 16-3
COMMAND_PARAMETER	VARCHAR(128)	NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールの CLAUSE_NAME 設定からのパラメータ
COMMAND_EVENT	VARCHAR(128)	NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールで定義されているイベント。
COMMAND_COMPONENT	VARCHAR(128)	NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールの EVENT_NAME 設定のコンポーネント
COMMAND_ACTION	VARCHAR(128)	NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールの EVENT_NAME 設定のアクション。
COMMON	VARCHAR(3)	NULL	マルチテナント環境の場合は、ポリシー・オブジェクトがローカルか共通かを示します。使用される値は、次のとおりです。 ● ポリシー・オブジェクトが共通の場合は YES ● ポリシー・オブジェクトがローカルの場合は NO
INHERITED	VARCHAR(3)	NULL	COMMON 列の出力が YES の場合は、ポリシー・オブジェクトの継承ステータスを示します。値は次のとおりです。 ● YES は、ポリシー・オブジェクトが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケー

列	データ型	Null	説明
			ション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、ポリシー・オブジェクトがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.25 DBA_DV_POLICY_OWNERビュー

DBA_DV_POLICY_OWNERデータ・ディクショナリ・ビューでは、現在のデータベース・インスタンスで作成されたOracle Database Vaultポリシーの所有者が示されます。

たとえば:

```
SELECT * FROM DBA_DV_POLICY_OWNER;
```

次のような出力が表示されます。

```
POLICY_OWNER                       POLICY_OWNER
---------------------------------- ------------
Oracle System Protection Controls  PSMITH
```

関連するビュー

- DBA_DV_POLICYビュー
- DBA_DV_POLICY_OBJECTビュー

列	データ型	Null	説明
POLICY_NAME	VARCHAR(128)	NOT NULL	作成された Oracle Database Vault ポリシーの名前。 デフォルト・ポリシーのリストは、デフォルトの Oracle Database Vault ポリシーを参照してください。
POLICY_OWNER	VARCHAR(128)	NOT NULL	独自の Database Vault ポリシーがあるユーザーの名前

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.26 DBA_DV_PREPROCESSOR_AUTHビュー

DBA_DV_PREPROCESSOR_AUTHデータ・ディクショナリ・ビューには、外部表からプリプロセッサ・プログラムを実行する認可を

付与されているユーザーが表示されます。

詳細は、Oracle Database VaultでのOracle Database Replayの使用を参照してください。

たとえば:

```
SELECT * FROM DBA_DV_PREPROCESSOR_AUTH WHERE GRANTEE = 'PFITCH';
```

次のような出力が表示されます。

```
GRANTEE
-------
PFITCH
```

列	データ型	Null	説明
GRANTEE	VARCHAR2(128)	NOT NULL	プリプロセッサ・プログラムを実行する認可を付与されているユーザーの名前

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.27 DBA_DV_PROXY_AUTHビュー

DBA_DV_PROXY_AUTHデータ・ディクショナリ・ビューには、DBMS_MACADM.AUTHORIZE_PROXY_USERプロシージャで指定されたプロキシ・ユーザーとスキーマが示されます。

このプロシージャは、他のユーザー・アカウントをプロキシする認可をプロキシ・ユーザーに付与します。

たとえば:

```
SELECT * FROM DBA_DV_DDL_AUTH WHERE GRANTEE = 'PRESTON';
```

次のような出力が表示されます。

```
GRANTEE SCHEMA
------- ------
PRESTON DKENT
```

列	データ型	Null	説明
GRANTEE	VARCHAR2(128)	NOT NULL	プロキシ・ユーザーの名前
SCHEMA	VARCHAR2(128)	NOT NULL	GRANTEE ユーザーによってプロキシされるスキーマの名前。

関連項目:

- AUTHORIZE_PROXY_USERプロシージャ
- UNAUTHORIZE_PROXY_USERプロシージャ

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.28 DBA_DV_PUB_PRIVSビュー

DBA_DV_PUB_PRIVSデータ・ディクショナリ・ビューは、Oracle Database Vault Administratorで使用されるOracle Database Vault権限管理レポートに反映されるデータが表示されます。

「権限管理 - サマリー・レポート」も参照してください。

たとえば:

```
SELECT USERNAME, ACCESS_TYPE FROM DBA_DV_PUB_PRIVS WHERE USERNAME = 'OE';
```

次のような出力が表示されます。

```
USERNAME     ACCESS_TYPE
------------ -----------------
OE           PUBLIC
```

関連するビュー

- DBA_DV_USER_PRIVSビュー
- DBA_DV_USER_PRIVS_ALLビュー
- DBA_DV_ROLEビュー

列	データ型	Null	説明
USERNAME	VARCHAR(128)	NOT NULL	現在のデータベース・インスタンスのデータベース・スキーマ。
ACCESS_TYPE	VARCHAR(128)	NULL	USERNAME 列に表示されたユーザーに付与されているアクセス・タイプ(PUBLIC など)。
PRIVILEGE	VARCHAR(40)	NOT NULL	USERNAME 列に表示されたユーザーに付与されている権限。
OWNER	VARCHAR(128)	NOT NULL	USERNAME のユーザーが権限を付与されているデータベース・スキーマの所有者。
OBJECT_NAME	VARCHAR(128)	NOT NULL	OWNER 列に表示されたスキーマ内のオブジェクトの名前。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.29 DBA_DV_REALMビュー

DBA_DV_REALMデータ・ディクショナリ・ビューには、現行のデータベース・インスタンスで作成されたレルムが表示されます。

たとえば:

```
SELECT NAME, AUDIT_OPTIONS, ENABLED, COMMON FROM DBA_DV_REALM
  WHERE AUDIT_OPTIONS = '1';
```

次のような出力が表示されます。

```
NAME                           AUDIT_OPTIONS    ENABLED  COMMON
------------------------------ ---------------- -------- ------
```

```
Performance Statistics Realm  1                    Y         NO
```

関連するビュー

- DBA_DV_REALM_AUTHビュー
- DBA_DV_REALM_OBJECTビュー

列	データ型	Null	説明
NAME	VARCHAR(128)	NOT NULL	作成されたレルムの名前。デフォルトのレルムのリストは、「デフォルトのレルム」を参照してください。
DESCRIPTION	VARCHAR(1024)	NOT NULL	作成されたレルムの説明。
AUDIT_OPTIONS	NUMBER	NOT NULL	監査が有効になっているかどうかを指定します。使用される値は、次のとおりです。 ● 0: レルムの監査なし。 ● 1: 認可されていないユーザーがレルムによって保護されているオブジェクトの変更を試行するというようなレルム違反が発生した場合に、監査レコードが作成されます。 ● 2: レルムで保護されているオブジェクトに対する認可されたアクティビティに関する監査レコードが作成されます。 ● 3: レルムで保護されているオブジェクトに対する認可および無認可の両方のアクティビティに関する監査レコードが作成されます。
REALM_TYPE	VARCHAR(9)	NULL	レルムのタイプ: 通常のレルムと必須レルムのどちらになるか。可能な値については、表 14-9 の「realm_type」を参照してください。
COMMON	VARCHAR(3)	NOT NULL	マルチテナント環境の場合は、レルムがローカルか共通かを示します。使用される値は、次のとおりです。 ● レルムが共通の場合は YES ● レルムがローカルの場合は NO
INHERITED	VARCHAR(3)	NULL	COMMON 列の出力が YES の場合は、レルムの継承ステータス

列	データ型	Null	説明
			を示します。値は次のとおりです。 ● YES は、レルムが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、レルムがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
ENABLED	VARCHAR(1)	NOT NULL	有効な値は次のとおりです。 ● Y は、レルム・チェックが有効になっていることを示します ● N はそれが無効になっていることを示します ● S は、レルムがシミュレーション・モードであることを示します
ID#	NUMBER	NOT NULL	レルムの ID 番号。これは、レルム作成時に自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR(3)	NOT NULL	レルムがデフォルトの(つまり、Oracle によって提供されている)レルムであるかユーザーが作成したコマンド・ルールであるかを示します。使用される値は、次のとおりです。 ● レルムがデフォルト・レルムである場合は YES ● レルムがユーザーが作成したレルムである場合は NO
PL_SQL_STACK	VARCHAR(3)	NULL	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックが記録されているかどうかを示します。TRUE は PL/SQL スタックが記録されていることを示し、FALSE は PL/SQL スタックが記録されていないことを示します。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.30 DBA_DV_REALM_AUTHビュー

DBA_DV_REALM_AUTHデータ・ディクショナリ・ビューには、レルム・オブジェクトにアクセスできる、データベース・ユーザー・アカウントまたはロール認可(GRANTEE)が表示されます。

詳細は、「レルム認可について」を参照してください。

たとえば:

```
SELECT REALM_NAME, GRANTEE, AUTH_RULE_SET_NAME FROM DBA_DV_REALM_AUTH;
```

次のような出力が表示されます。

```
REALM_NAME                     GRANTEE   AUTH_RULE_SET_NAME
------------------------------ --------- --------------------
Performance Statistics Realm   SYSADM    Check Conf Access
```

関連するビュー

- DBA_DV_REALMビュー
- DBA_DV_REALM_OBJECTビュー

列	データ型	Null	説明
REALM_NAME	VARCHAR(128)	NULL	レルムの名前。
COMMON_REALM	VARCHAR(3)	NULL	マルチテナント環境の場合は、レルムがローカルか共通かを示します。使用される値は、次のとおりです。 ● レルムが共通の場合は YES ● レルムがローカルの場合は NO
INHERITED_REALM	VARCHAR(3)	NULL	COMMON 列の出力が YES の場合は、レルムの継承ステータスを示します。値は次のとおりです。 ● YES は、レルムが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、レルムがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。

列	データ型	Null	説明
GRANTEE	VARCHAR(128)	NOT NULL	所有者または参加者として認可するユーザーまたはロール名。
AUTH_RULE_SET_NAME	VARCHAR(128)	NULL	認可の前にチェックするルール・セット。ルール・セットの評価が True の場合は、認可が許可されます。
AUTH_OPTIONS	VARCHAR(4000)	NULL	レルム認可のタイプ。「参加者」または「所有者」のいずれかです。
COMMON_AUTH	VARCHAR(3)	NULL	マルチテナント環境の場合は、共通レルムに対する認可がローカルか共通かを示します。使用される値は、次のとおりです。 ● 認可が共通の場合は YES ● 認可がこの PDB に対してローカルである場合は NO
INHERITED_AUTH	VARCHAR(3)	NULL	COMMON_AUTH 列の出力が YES の場合は、レルム認可の継承ステータスを示します。値は次のとおりです。 ● YES は、レルム認可が、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、Database Vault ポリシー適用時にこのコンテナで継承されたことを意味します。 ● NO は、レルム認可がローカルであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED_AUTH 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED_AUTH 値は YES になります。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.31 DBA_DV_REALM_OBJECTビュー

DBA_DV_REALM_OBJECTデータ・ディクショナリ・ビューには、データベース・スキーマ、またはレルムによって保護されているスキーマのサブセットが表示されます。

詳細は、「レルム・セキュア・オブジェクトについて」を参照してください。

たとえば:

```
SELECT REALM_NAME, OWNER, OBJECT_NAME, COMMON_REALM FROM DBA_DV_REALM_OBJECT;
```

次のような出力が表示されます。

```
REALM_NAME                    OWNER    OBJECT_NAME COMMON_REALM
----------------------------- -------- ----------- ------------
Performance Statistics Realm OE       ORDERS      NO
```

関連するビュー

- DBA_DV_REALMビュー
- DBA_DV_REALM_AUTHビュー

列	データ型	Null	説明
REALM_NAME	VARCHAR(128)	NOT NULL	レルムの名前。
COMMON_REALM	VARCHAR(3)	NOT NULL	このレルムが共通レルムかローカル・レルムかを示します。使用される値は、次のとおりです。 ● レルムが共通の場合は YES ● レルムがローカルの場合は NO
INHERITED_REALM	VARCHAR(3)	NOT NULL	COMMON 列の出力が YES の場合は、レルムの継承ステータスを示します。値は次のとおりです。 ● YES は、レルムが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、レルムがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
OWNER	VARCHAR(128)	NOT NULL	オブジェクトを所有するデータベース・スキーマ所有者。
OBJECT_NAME	VARCHAR(128)	NOT NULL	レルムによって保護されるオブジェクトの名前。
OBJECT_TYPE	VARCHAR(32)	NOT NULL	レルムによって保護されるオブジェクトのタイプ(データベース表、ビュー、索引、ロールなど)。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.32 DBA_DV_ROLEビュー

DBA_DV_ROLEデータ・ディクショナリ・ビューには、権限管理で使用されるOracle Database Vaultセキュア・アプリケーション・ロールが表示されます。

たとえば:

```
SELECT ROLE, RULE_NAME FROM DBA_DV_ROLE;
```

次のような出力が表示されます。

```
ROLE                 RULE_NAME
-------------------- --------------------
Sector2_APP_MGR      Check App2 Access
Sector2_APP_DBA      Check App2 Access
```

関連するビュー

- DBA_DV_PUB_PRIVSビュー
- DBA_DV_USER_PRIVSビュー
- DBA_DV_USER_PRIVS_ALLビュー

列	データ型	Null	説明
ROLE	VARCHAR(128)	NOT NULL	セキュア・アプリケーション・ロールの名前。
RULE_NAME	VARCHAR(128)	NOT NULL	セキュア・アプリケーション・ロールに関連付けられているルール・セットの名前。
ENABLED	VARCHAR(1)	NOT NULL	セキュア・アプリケーション・ロールが有効になっているかどうかを示します。使用される値は、次のとおりです。 ● ロールが有効になっている場合は Y (Yes) ● ロールが無効になっている場合は N (No)
ID#	NUMBER	NOT NULL	コマンド・ルールの ID 番号。これは、コマンド・ルール作成時に自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR(3)	NOT NULL	コマンド・ルールがデフォルト(つまり、Oracle によって提供されている)コマンド・ルールであるかユーザーが作成したコマンド・ルールであるかを示します。使用される値は、次のとおりです。 ● コマンド・ルールがデフォルト・コマンド・ルールである場合は YES ● コマンド・ルールがユーザーが作成したコマンド・ルールで

列	データ型	Null	説明
			ある場合は NO

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.33 DBA_DV_RULEビュー

DBA_DV_RULEデータ・ディクショナリ・ビューには、定義済のルールが表示されます。

たとえば:

```
SELECT NAME, RULE_EXPR FROM DBA_DV_RULE WHERE NAME = 'Maintenance Window';
```

次のような出力が表示されます。

```
NAME                 RULE_EXP
-------------------- ----------------------------------------------
Maintenance Window   TO_CHAR(SYSDATE,'HH24') BETWEEN '10' AND '12'
```

特定のルールを使用するルール・セットを検索する場合は、DBA_DV_RULE_SET_RULEビューに問い合せます。

関連するビュー

- DBA_DV_RULE_SETビュー
- DBA_DV_RULE_SET_RULEビュー

列	データ型	Null	説明
NAME	VARCHAR(128)	NOT NULL	ルールの名前。
RULE_EXPR	VARCHAR(1024)	NOT NULL	ルール用の PL/SQL 式。
COMMON	VARCHAR(3)	NOT NULL	マルチテナント環境の場合は、ルールがローカルか共通かを示します。使用される値は、次のとおりです。 ● ルールが共通の場合は YES ● ルールがローカルの場合は NO
INHERITED	VARCHAR(3)	NULL	COMMON 列の出力が YES の場合は、ルールの継承ステータスを示します。値は次のとおりです。 ● YES は、ルールが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。

列	データ型	Null	説明
			● NO は、ルールがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
ID#	NUMBER	NOT NULL	ルールの ID 番号。これは、ルール作成時に自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR(3)	NULL	ルールがデフォルト(つまり、Oracle によって提供されている)ルールであるかユーザーが作成したルールであるかを示します。使用される値は、次のとおりです。 ● ルールがデフォルト・ルールである場合は YES ● ルールがユーザーが作成したルールである場合は NO

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.34 DBA_DV_RULE_SETビュー

DBA_DV_RULE_SETデータ・ディクショナリ・ビューには、作成済のルール・セットが表示されます。

たとえば:

```
SELECT RULE_SET_NAME, HANDLER_OPTIONS, HANDLER FROM DBA_DV_RULE_SET
 WHERE RULE_SET_NAME = 'Maintenance Period';
```

次のような出力が表示されます。

```
RULE_SET_NAME       HANDLER_OPTIONS  HANDLER
------------------- ---------------- ----------------------
Maintenance Period                   1 dbavowner.email_alert
```

関連するビュー

- DBA_DV_RULEビュー
- DBA_DV_RULE_SET_RULEビュー

列	データ型	Null	説明
RULE_SET_NAME	VARCHAR(128)	NOT NULL	ルール・セットの名前。
DESCRIPTION	VARCHAR(1024)	NULL	ルール・セットの説明。

列	データ型	Null	説明
ENABLED	VARCHAR(1)	NOT NULL	ルール・セットが有効になっているかどうかを示します。Y(YES)の場合、ルール・セットは有効になり、N(NO)の場合、ルール・セットは無効になります。
EVAL_OPTIONS_MEANING	VARCHAR(4000)	NULL	複数のルールが含まれるルール・セットの場合、評価されるルールの数が決まります。使用される値は、次のとおりです。 ● すべて True: ルール・セット自体が TRUE と評価されるために、ルール・セットのルールはすべて True と評価される必要があります。 ● いずれか True: ルール・セット自体が TRUE と評価されるために、少なくともルール・セットの 1 つのルールが True と評価される必要があります。
AUDIT_OPTIONS	NUMBER	NOT NULL	監査が使用される時期を示します。使用される値は、次のとおりです。 ● 0: 監査なし ● 1: 失敗時に監査 ● 2: 成功時に監査 ● 3: 失敗時と成功時の両方に監査
FAIL_OPTIONS_MEANING	VARCHAR(4000)	NULL	ルール・セットに対して監査レコードが作成される時期が決定されます。使用される値は、次のとおりです。 ● エラー・メッセージを表示しない。 ● エラー・メッセージを表示
FAIL_MESSAGE	VARCHAR(80)	NULL	FAIL_CODE 列に表示された失敗コードに関連付けられている失敗に対するエラー・メッセージ。
FAIL_CODE	VARCHAR(10)	NULL	FAIL_MESSAGE 列に表示されたメッセージに関連付けられているエラー・メッセージ番号。考えられる

列	データ型	Null	説明
			値の範囲は、-20000 から-20999 と 20000 から 20999 です。
HANDLER_OPTIONS	NUMBER	NOT NULL	エラー処理の使用方法が決まります。使用される値は、次のとおりです。 ● 0: エラー処理を無効にします。 ● 1: ルール・セット失敗時にハンドラをコールします。 ● 2: ルール・セット成功時にハンドラをコールします。
HANDLER	VARCHAR(1024)	NULL	カスタム・イベント・ハンドラ・ロジックを定義する PL/SQL ファンクションまたはプロシージャの名前。
IS_STATIC	VARCHAR2(5)	NULL	ユーザー・セッション中にルール・セットが評価される頻度を示します。使用される値は、次のとおりです。 ● TRUE: ルール・セットは 1 回評価され、ルール・セットの結果はユーザー・セッションで再利用されます。 ● FALSE(デフォルト): ルール・セットは、ユーザー・セッション中にアクセスされるたびに評価されます。
COMMON	VARCHAR2(3)	NULL	マルチテナント環境の場合は、ルール・セットがローカルか共通かを示します。使用される値は、次のとおりです。 ● ルール・セットが共通の場合は YES ● ルール・セットがローカルの場合は NO
INHERITED	VARCHAR2(3)	NULL	COMMON 列の出力が YES の場合は、ルール・セットの継承ステータスを示します。値は次のとおりです。 ● YES は、ルール・セットが、コンテナ・ツリー階層の上位にある別のコンテナで定義され

列	データ型	Null	説明
			ており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、ルール・セットがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
ID#	NUMBER)	NOT NULL	ルール・セットの ID 番号。これは、ルール・セット作成時に自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR2(3)	NULL	ルール・セットがデフォルト(つまり、Oracle によって提供されている)ルール・セットであるかユーザーが作成したルール・セットであるかを示します。使用される値は、次のとおりです。 ● ルール・セットがデフォルト・ルール・セットである場合は YES ● ルール・セットがユーザーが作成したルール・セットである場合は NO

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.35 DBA_DV_RULE_SET_RULEビュー

DBA_DV_RULE_SET_RULEデータ・ディクショナリ・ビューには、既存のルール・セットに関連付けられているルールが表示されます。

たとえば:

```
SELECT RULE_SET_NAME, RULE_NAME, RULE_EXPR FROM DBA_DV_RULE_SET_RULE
 WHERE RULE_NAME = 'Is Security Officer';
```

次のような出力が表示されます。

```
RULE_SET_NAME                RULE_NAME          RULE_EXP
---------------------------- ------------------ --------------------------------
Can Grant VPD Administration Is Security Owner  DBMS_MACUTL.USER_HAS_ROLE_VARCHAR
                                                ('DV_OWNER',
                                                  dvsys.dv_login_user) = 'Y'
```

関連するビュー

- DBA_DV_RULEビュー
- DBA_DV_RULE_SETビュー

列	データ型	Null	説明
RULE_SET_NAME	VARCHAR(128)	NOT NULL	ルールが含まれるルール・セットの名前。
RULE_NAME	VARCHAR(128)	NOT NULL	ルールの名前。
RULE_EXPR	VARCHAR(1024)	NOT NULL	RULE_NAME 列に表示されたルールを定義する PL/SQL 式。
ENABLED	VARCHAR(1)		ルールが有効になっているか無効になっているかを示します。Y(YES)の場合、ルール・セットは有効になり、N(NO)の場合、ルール・セットは無効になります。
RULE_ORDER	NUMBER	NOT NULL	ルール・セット内でルールが使用される順序。このリリースには適用されません。
COMMON	VARCHAR(3)	NOT NULL	マルチテナント環境の場合は、ルールがローカルか共通かを示します。使用される値は、次のとおりです。 ● ルールが共通の場合は YES ● ルールがローカルの場合は NO
INHERITED	VARCHAR(3)	NOT NULL	COMMON 列の出力が YES の場合は、ルールの継承ステータスを示します。値は次のとおりです。 ● YES は、ルールが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、ルールがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。

24.36 DBA_DV_SIMULATION_LOGビュー

DBA_DV_SIMULATION_LOGデータ・ディクショナリ・ビューでは、シミュレーション・モードが有効になっているレルムおよびコマンド・ルールのシミュレーション・ログ情報が取得されます。

たとえば:

```
SELECT USERNAME, COMMAND
FROM DBA_DV_SIMULATION_LOG, TABLE(DBA_DV_SIMULATION_LOG.REALM_NAME) RN
WHERE RN.COLUMN_VALUE = 'HR Realm';
```

次のような出力が表示されます。

```
USERNAME      COMMAND
------------- ---------------------------------------------
PSMITH        SELECT
```

関連するビュー

- レルムのシミュレーション・モード設定については、DBA_DV_REALMビューを参照してください
- コマンド・ルールのシミュレーション・モード設定については、DBA_DV_COMMAND_RULEビューを参照してください
- Oracle Database Vaultポリシーのシミュレーション・モード設定については、DBA_DV_POLICYビューを参照してください

列	データ型	Null	説明
ID	NUMBER	NOT NULL	シミュレーション・ログ ID
USERNAME	VARCHAR2(128)	NOT NULL	情報が追跡されているユーザーの名前
COMMAND	VARCHAR2(128)	NOT NULL	追跡されているコマンド・ルール 既存のコマンド・ルールのリストを確認するには、「DBA_DV_COMMAND_RULE ビュー」で説明されている DBA_DV_COMMAND_RULE ビューを問い合せます。
VIOLATION_TYPE	VARCHAR2(4000)	NULL	違反のタイプ。詳細は、表 24-2 を参照してください。
REALM_NAME	DVSYS.DV_OBJ_NAME	NULL	追跡されているレルム。データ型 DVSYS.DV_OBJ_NAME は、問合せで複数のレルムを取得できる、ネストされたリスト・オブジェクトです。 既存のレルムのリストを確認するには、

列	データ型	Null	説明
			「DBA_DV_REALM ビュー」で説明されている DBA_DV_REALM ビューを問い合せます。
REALM_TYPE	VARCHAR2(9)	NULL	追跡されているレルムのタイプ(たとえば、必須レルム)。
OBJECT_OWNER	VARCHAR2(128)	NULL	コマンド・ルールの場合は、コマンド・ルールが適用されるデータベース・スキーマ
OBJECT_NAME	VARCHAR2(128)	NULL	コマンド・ルールの場合は、コマンド・ルールで保護されるデータベース・オブジェクト
OBJECT_TYPE	VARCHAR2(129)	NULL	コマンド・ルールの場合は、保護されているオブジェクトのタイプ
RULE_SET_NAME	DVSYS.DV_OBJ_NAME	NULL	追跡されているルール・セット。コマンド・ルールに関連付けられています。データ型 DVSYS.DV_OBJ_NAME は、問合せで複数のルール・セットを取得できる、ネストされたリスト・オブジェクトです。 既存のルール・セットのリストを確認するには、「DBA_DV_RULE_SET ビュー」で説明されている DBA_DV_RULE_SET ビューを問い合せます。
RETURNCODE	NUMBER	NOT NULL	Database Vault エンティティがシミュレーション状態ではなく有効状態だった場合に発生する Oracle Database ORA エラー
SQLTEXT	VARCHAR2(4000)	NULL	シミュレーション・モードで取得される SQL 文
AUTHENTICATION_METHOD	VARCHAR2(10)	NULL	使用される認証方法。デフォルトのファクタを参照してください。
CLIENT_IP	VARCHAR2(45)	NULL	クライアントが接続されているマシンの IP アドレス
DB_DOMAIN	VARCHAR2(128)	NULL	DB_DOMAIN 初期化パラメータで指定されているデータベースのドメイン

列	データ型	Null	説明
DATABASE_HOSTNAME	VARCHAR2(128)	NULL	インスタンスを実行しているコンピュータのホスト名
DATABASE_INSTANCE	VARCHAR2(5)	NULL	現行のインスタンスのインスタンス識別番号を戻します
DATABASE_IP	VARCHAR2(45)	NULL	インスタンスが実行されているコンピュータの IP アドレス
DATABASE_NAME	VARCHAR2(128)	NULL	DB_NAME 初期化パラメータで指定されているデータベースの名前
DOMAIN	VARCHAR2(4000)	NULL	ランタイム環境での物理、構成または実装固有のファクタの名前付きコレクション。デフォルトのファクタを参照してください。
ENTERPRISE_IDENTITY	VARCHAR2(1024)	NULL	ユーザーのエンタープライズ全体のアイデンティティ。デフォルトのファクタを参照してください。
IDENTIFICATION_TYPE	VARCHAR2(14)	NULL	データベースでのユーザー・スキーマの作成方法。デフォルトのファクタを参照してください。
LANG	VARCHAR2(10)	NULL	既存の LANGUAGE パラメータより短い形式の、言語名の ISO 略称
LANGUAGE	VARCHAR2(100)	NULL	セッションで現在使用中の言語と地域、およびデータベース文字セット。デフォルトのファクタを参照してください。
MACHINE	VARCHAR2(64)	NULL	現在のセッションを確立したデータベース・クライアントのホスト名。コンピュータがクライアントまたはサーバー・セッションに使用されていたかどうかを調べる必要がある場合には、この設定を Database_Hostname ファクタと比較して特定できます
NETWORK_PROTOCOL	VARCHAR2(4)	NULL	接続文字列の PROTOCOL=protocol の部分で指定された、通信に使用されるネットワーク・プロトコル

列	データ型	Null	説明
PROXY_ENTERPRISE_IDENTITY	VARCHAR2(1024)	NULL	プロキシ・ユーザーがエンタープライズ・ユーザーの場合の Oracle Internet Directory DN
PROXY_USER	VARCHAR2(128)	NULL	SESSION_USER のかわりに現行のセッションを開いたデータベース・ユーザー名
SESSION_USER	VARCHAR2(128)	NULL	現行ユーザーが認証されたデータベース・ユーザー名。この値は、セッションを通して同じです。
DV$_DBLINK_INFO	VARCHAR2(128)	NULL	データベース・リンク・セッションのソースを戻します。返される文字列の形式は次のとおりです。 SOURCE_GLOBAL_NAME=dblink_src_global_name, DBLINK_NAME=dblink_name, SOURCE_AUDIT_SESSIONID=dblink_src_audit_sessionid 詳細は、次のとおりです。 ● dblink_src_global_name: ソース・データベースの一意のグローバル名 ● dblink_name: ソース・データベースでのデータベース・リンクの名前 ● dblink_src_audit_sessionid: dblink_name を使用してリモート・データベースへの接続を開始したソース・データベース
DV$_MODULE	VARCHAR2(64)	NULL	DBMS_APPLICATION_INFO PL/SQL パッケージまたは Oracle Call Interface (OCI)を使用して設定されたアプリケーション名(モジュール)。
DV$_CLIENT_IDENTIFIER	VARCHAR2(64)	NULL	アプリケーションによって設定された識別子を、DBMS_SESSION.SET_IDENTIFIER プロシージャ、OCI 属性 OCI_ATTR_CLIENT_IDENTIFIER または Oracle Dynamic Monitoring Service

列	データ型	Null	説明
			(DMS)を使用して戻します。様々な Oracle Database コンポーネントが、この属性を使用して同じデータベース・ユーザーとして認証される軽量アプリケーション・ユーザーを識別します。
FACTOR_CONTEXT	VARCHAR2(4000)	NULL	監査イベントがトリガーされた時点での、現行セッションに対するすべてのファクタ識別子を含む XML 文書。
TIMESTAMP	TIMESTAMP(6)WITH TIME ZONE	NULL	UTC (協定世界時)タイムゾーンでのユーザー・アクションのタイムスタンプ
PL_SQL_STACK	CLOB	NULL	シミュレーション・モードが有効な場合に、失敗した操作の PL/SQL スタックが記録されているかどうかを示します。TRUE は PL/SQL スタックが記録されていることを示し、FALSE は PL/SQL スタックが記録されていないことを示します。

VIOLATION_TYPEコード値

表24-2に、DBA_DV_SIMULATION_LOGビューのVIOLATION_TYPEコード値を示します。

表24-2 DBA_DV_SIMULATION_LOG VIOLATION_TYPEコード値

コード	意味
1000	レルム違反
1001	コマンド・ルール違反
1002	Oracle Data Pump 認可違反
1003	シミュレーション違反
1004	Oracle Scheduler 認可違反
1005	DDL 認可違反
1006	PARSE_AS_USER 違反

関連トピック

- レルムおよびコマンド・ルール・アクティビティのログ記録のためのシミュレーション・モードの使用

24.37 DBA_DV_STATUSまたはSYS.DBA_DV_STATUSビュー

DBA_DV_STATUSまたはSYS.DBA_DV_STATUSデータ・ディクショナリ・ビューには、有効化され構成されているOracle Database Vaultのステータスが示されます。

DBA_DV_STATUSおよびSYS.DBA_DV_STATUSデータ・ディクショナリ・ビューの問合せ方法は、持っている権限によって異なります。

- DBAロールまたはSYSDBA管理権限を持つユーザーとして接続している場合は、DBA_DV_STATUSを問い合せます。たとえば:

```
SELECT * FROM DBA_DV_STATUS;
```

- DV_OWNERロールまたはDV_ADMINロールを持つユーザーとして接続している場合は、DBA_DV_STATUSの前にSYS.を付加します。たとえば:

```
SELECT * FROM SYS.DBA_DV_STATUS;
```

次のような出力が表示されます。

```
NAME                 STATUS
-------------------- --------------
DV_APP_PROTECTION    NOT CONFIGURED
DV_CONFIGURE_STATUS  TRUE
DV_ENABLE_STATUS     TRUE
```

関連するビュー

. CDB_DV_STATUSビュー

列	データ型	Null	説明
NAME	VARCHAR2(19)	NOT NULL	次の設定のいずれかを示します。 ● DV_APP_PROTECTION は、Database Vault 操作の制御が構成されているかどうかを示します ● DV_CONFIGURE_STATUS は、Oracle Database Vault が構成されているかどうか、つまり、CONFIGURE_DV プロシージャが使用されたかどうかを示します。 ● DV_ENABLE_STATUS は、Oracle Database Vault が有効になっているかどうか、つまり、DBMS_MACADM.ENABLE_DV プロシージャが使用されたかどうかを示します。
STATUS	VARCHAR2(64)	NOT NULL	TRUE は Oracle Database Vault が構成され有効になっていることを意味し、FALSE はそうでないことを意

列	データ型	Null	説明
			味します。DV_APP_PROTECTION の場合、CONFIGURED または NOT CONFIGURED のいずれかが表示されます。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.38 DBA_DV_TTS_AUTHビュー

DBA_DV_TTS_AUTHデータ・ディクショナリ・ビューには、Oracle Data Pumpのトランスポータブル操作を実行するための認可をDBMS_MACADM.AUTHORIZE_TTS_USERプロシージャによって与えられたユーザーが示されます。

詳細は、「Oracle Database VaultでのOracle Data Pumpの使用」を参照してください。

たとえば:

```
SELECT * FROM DBA_DV_TTS_AUTH;
```

次のような出力が表示されます。

```
GRANTEE  TSNAME
-------- --------
DB_MGR   HR_TS
```

関連するビュー

- DBA_DV_DATAPUMP_AUTHビュー

列	データ型	Null	説明
GRANTEE	VARCHAR(128)	NOT NULL	トランスポータブル表領域の権限を付与されたユーザーの名前
TSNAME	VARCHAR(128)	NOT NULL	GRANTEE のユーザーが権限を付与されているトランスポータブル表領域の名前

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.39 DBA_DV_USER_PRIVSビュー

DBA_DV_USER_PRIVSデータ・ディクショナリ・ビューには、PUBLICロールによって付与された権限を除くデータベース・ユーザー・アカウントの権限が表示されます。

たとえば:

```
SELECT USERNAME, ACCESS_TYPE, PRIVILEGE FROM DBA_DV_USER_PRIVS;
```

次のような出力が表示されます。

```
USERNAME  ACCESS_TYPE          PRIVILEGE
--------- -------------------- ------------
DVSYS     DV_PUBLIC            EXECUTE
DVOWNER   DV_ADMIN             SELECT
SYS       SELECT_CATALOG_ROLE  SELECT
```

```
...
```

関連するビュー

- DBA_DV_PUB_PRIVSビュー
- DBA_DV_ROLEビュー
- DBA_DV_USER_PRIVS_ALLビュー

列	データ型	Null	説明
USERNAME	VARCHAR(128)	NOT NULL	権限が定義されているデータベース・スキーマ・アカウントの名前。
ACCESS_TYPE	VARCHAR(128)	NULL	USERNAME 列に表示されたデータベース・ユーザー・アカウントがデータベースへのアクセスに使用するロール。Oracle Database Vault アカウントには直接アクセス権があります。
PRIVILEGE	VARCHAR(40)	NOT NULL	USERNAME 列に表示されたユーザーに付与されている権限。
OWNER	VARCHAR(128)	NOT NULL	データベース・ユーザー・アカウントの名前。
OBJECT_NAME	VARCHAR(128)	NOT NULL	権限の定義に使用される PL/SQL ファンクションまたはプロシージャの名前。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.40 DBA_DV_USER_PRIVS_ALLビュー

DBA_DV_USER_PRIVS_ALLデータ・ディクショナリ・ビューには、PUBLICによって付与された権限を含むデータベース・アカウントの権限が示されます。

たとえば:

```
SELECT USERNAME, ACCESS_TYPE, PRIVILEGE FROM DBA_DV_USER_PRIVS;
```

次のような出力が表示されます。

```
USERNAME             ACCESS_TYPE  PRIVILEGE
-------------------- ------------ -----------------
ACCTS_ADMIN_ACE      CONNECT      CREATE_SESSION
SEC_ADMIN_OWEN       DIRECT       CREATE PROCEDURE
...
```

関連するビュー

- DBA_DV_PUB_PRIVSビュー
- DBA_DV_ROLEビュー
- DBA_DV_USER_PRIVSビュー

列	データ型	Null	説明
USERNAME	VARCHAR(128)	NULL	権限が定義されているデータベース・スキーマ・アカウントの名前。
ACCESS_TYPE	VARCHAR(128)	NULL	USERNAME 列に表示されたデータベース・ユーザー・アカウントがデータベースへのアクセスに使用するロール。Oracle Database Vault アカウントには直接アクセス権があります。
PRIVILEGE	VARCHAR(40)	NULL	USERNAME 列に表示されたユーザーに付与されている権限。
OWNER	VARCHAR(128)	NULL	データベース・ユーザー・アカウントの名前。
OBJECT_NAME	VARCHAR(128)	NULL	権限の定義に使用される PL/SQL ファンクションまたはプロシージャの名前。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.41 DVSYS.DV$CONFIGURATION_AUDITビュー

DVSYS.DV$CONFIGURATION_AUDITデータ・ディクショナリ・ビューは、DVSYS.AUDIT_TRAIL$表の監査証跡レコードを取得します。

レルム、ルール、ルール・セット、ファクタ、その他Oracle Database Vaultポリシー構成アクティビティに加えられた構成の変更の成功および失敗に関連するレコードが含まれます。

たとえば:

```
SELECT USERNAME, ACTION_NAME FROM DVSYS.DV$CONFIGURATION_AUDIT
WHERE USERNAME = 'PSMITH';
```

次のような出力が表示されます。

```
USERNAME   ACTION_NAME
---------- --------------------
PSMITH     Realm Creation Audit
PSMITH     Rule Set Update Audit
```

関連するビュー

. AUDSYS.DV$CONFIGURATION_AUDITビュー

列	データ型	Null	説明
ID#	NUMBER	NOT NULL	監査レコードごとの数値識別子。
OS_USERNAME	VARCHAR(255)	NULL	アクションが監査対象となったユーザーのオペレーティング・

列	データ型	Null	説明
			システムのログイン・ユーザー名。
USERNAME	VARCHAR(128)	NULL	アクションが監査対象となったデータベース・ユーザーの名前。
USERHOST	VARCHAR2(128)	NULL	クライアント・コンピュータ名。
TERMINAL	VARCHAR2(30)	NULL	ユーザーの端末に対する識別子。
TIMESTAMP	DATA	NULL	監査証跡エントリの作成日時(ローカル・データベース・セッションのタイムゾーン)。
OWNER	VARCHAR2(128)	NULL	アクションの影響を受けるオブジェクトの作成者、常時 DVSYS(DVSYS でオブジェクトが作成されるため)
OBJ_NAME	VARCHAR2(128)	NULL	アクションの影響を受けるオブジェクトの名前。想定値は次のとおりです。 ● ROLE$ ● REALM$ ● CODE$ ● FACTOR$
ACTION	NUMBER	NOT NULL	数値のアクション・タイプ・コード。アクション・タイプに対応する名前は、ACTION_NAME 列に示されます。使用可能なアクションのリストは、表 24-3 を参照してください。
ACTION_NAME	VARCHAR2(128)	NULL	ACTION 列の数値コードに対応するアクション・タイプの名前。使用可能なアクションのリストは、表 24-3 を参照してください。
ACTION_OBJECT_ID	NUMBER	NULL	OBJ_NAME に指定された表のレコードの一意の識別子
ACTION_OBJECT_NAME	VARCHAR2(128)	NULL	OBJ_NAME に指定された表のレコードの一意の名前または固有のキー
ACTION_COMMAND	VARCHAR2(4000)	NULL	実行された結果、監査イベントがトリガーされたコマンド・プロシージャの SQL テキスト。

列	データ型	Null	説明
AUDIT_OPTION	VARCHAR2(4000)	NULL	結果として監査イベントがトリガーされたレコードに指定されたすべての監査オプションのラベル。たとえば、失敗または NULL になったときに監査することになっているファクタ設定操作では、これら 2 つのオプションを指します。
RULE_SET_ID	NUMBER	NULL	実行された結果、監査イベントがトリガーされたルール・セットの一意の識別子。
RULE_SET_NAME	VARCHAR2(128)	NULL	実行された結果、監査イベントがトリガーされたルール・セットの一意の名前。
RULE_ID	NUMBER	NULL	使用されていません。
RULE_NAME	VARCHAR2(128)	NULL	使用されていません。
FACTOR_CONTEXT	VARCHAR2(4000)	NULL	監査イベントがトリガーされた時点での、現行セッションに対するすべてのファクタ識別子を含む XML 文書。
COMMENT_TEXT	VARCHAR2(4000)	NULL	監査対象となった文の詳細を示す、監査証跡エントリについてのテキスト・コメント。
SESSIONID	NUMBER	NOT NULL	Oracle セッションごとの数値識別子。
ENTRYID	NUMBER	NOT NULL	ID#列の値と同じ。
STATEMENTID	NUMBER	NOT NULL	起動された結果、監査イベントが生成された文の数値識別子。ほとんどの Oracle Database Vault イベントの場合、このパラメータは空です。
RETURNCODE	NUMBER	NOT NULL	アクションによって生成された Oracle エラー・コード。起動された結果、監査イベントが生成された文またはプロシージャに対するエラー・コード。ほとんどの Oracle Database Vault イベントの場合、このパラメータは空です。
EXTENDED_TIMESTAMP	TIMESTAMP(6)WITH TIME ZONE	NULL	UTC(協定世界時)タイムゾーンの、監査証跡エントリの作成時のタイムスタンプ(エントリに対するユーザー・ログインのタイムスタンプ)。
PROXY_SESSIONID	NUMBER	NULL	エンタープライズ・ユーザーがプロキシ機構を介してログイン

列	データ型	Null	説明
			した場合の、プロキシ・セッションのシリアル番号
GLOBAL_UID	VARCHAR2(32)	NULL	ユーザーがエンタープライズ・ユーザーとしてログインした場合の、ユーザーのグローバル・ユーザー識別子。
INSTANCE_NUMBER	NUMBER	NULL	INSTANCE_NUMBER 初期化パラメータによって指定されるインスタンス番号
OS_PROCESS	VARCHAR2(16)	NULL	Oracle プロセスのオペレーティング・システム・プロセス識別子
CREATED_BY	VARCHAR2(128)	NULL	アクションが監査対象となったユーザーのデータベースのログイン・ユーザー名。
CREATE_DATE	DATE	NULL	SYSDATE の日付を基にした、アクションが発生した日付。
UPDATED_BY	VARCHAR2(128)	NULL	CREATED_BY 列の値と同じ。
UPDATE_DATE	DATE	NULL	UPDATED_BY 列の値と同じ。
GRANTEE	VARCHAR2(128)	NULL	Database Vault で保護されるロール、レルム認可、コマンドルール認可、ジョブ・スケジューラ認可または Oracle Data Pump 認可が付与されているユーザーのユーザー ID
ENABLED_STATUS	VARCHAR2(1)	NULL	構成が有効だったかどうかを示します。

表24-3に、DVSYS.DV$CONFIGURATION_AUDITビューのACTION列で使される値を示します。

表24-3 DVSYS.DV$CONFIGURATION_AUDITビューのACTIONの値

アクション・タイプ・コード	アクション名
20001	DV 強制有効化の監査
20002	DV 強制無効化の監査
20003	レルム作成の監査
20004	レルム更新の監査

アクション・タイプ・コード	アクション名
20005	レルム名変更の監査
20006	レルム削除の監査
20007	レルム認可追加の監査
20008	レルム認可削除の監査
20009	レルム認可更新の監査
20010	レルム・オブジェクト追加の監査
20011	レルム・オブジェクト更新の監査
20012	レルム・オブジェクト削除の監査
20013	イベント有効化の監査
20014	イベント無効化の監査
20015	ルール・セット作成の監査
20016	ルール・セット更新の監査
20017	ルール・セット名変更の監査
20018	ルール・セット削除の監査
20019	ルール・セットへのルール追加の監査
20020	ルール・セットからのルール削除の監査
20021	ルール作成の監査
20022	ルール更新の監査
20023	ルール名変更の監査
20024	ルール削除の監査
20025	`CommandRule` 作成の監査

アクション・タイプ・コード	アクション名
20026	CommandRule 更新の監査
20027	CommandRule 削除の監査
20028	データポンプ・ユーザー認可の監査
20029	データポンプ・ユーザー認可解除の監査
20030	ジョブ・ユーザー認可の監査
20031	ジョブ・ユーザー認可解除の監査
20032	Factor_Type 作成の監査
20033	Factor_Type 削除の監査
20034	Factor_Type 更新の監査
20035	Factor_Type 名変更の監査
20036	ファクタ作成の監査
20037	G_FACTOR_DELETION_AUDIT_CODE
20038	ファクタ更新の監査
20039	ファクタ名変更の監査
20040	ファクタ・リンク追加の監査
20041	ファクタ・リンク削除の監査
20042	ポリシー・ファクタ追加の監査
20043	ポリシー・ファクタ削除の監査
20044	アイデンティティ作成の監査
20045	アイデンティティ削除の監査
20046	アイデンティティ更新の監査

アクション・タイプ・コード	アクション名
20047	アイデンティティ・ファクタ変更の監査
20048	アイデンティティ値変更の監査
20049	アイデンティティ・マップ作成の監査
20050	アイデンティティ・マップ削除の監査
20051	ポリシー・ラベル作成の監査
20052	ポリシー・ラベル削除の監査
20053	MAC ポリシー作成の監査
20054	マップ・ポリシー更新の監査
20055	マップ・ポリシー削除の監査
20056	ロール作成の監査
20057	ロール削除の監査
20058	ロール更新の監査
20059	ロール名変更の監査
20060	ドメイン・アイデンティティ作成の監査
20061	ドメイン・アイデンティティ削除の監査
20062	Oradebug 有効化の監査
20063	Oradebug 無効化の監査
20064	プロキシ・ユーザー認可の監査
20065	プロキシ・ユーザー認可解除の監査
20066	DV ディクショナリ・アカウント有効化の監査
20067	DV ディクショナリ・アカウント無効化の監査

アクション・タイプ・コード	アクション名
20068	DDL 認可の監査
20069	DDL 認可解除の監査
20070	TTS 認可の監査
20071	TTS 認可解除の監査
20072	PREPROCESSOR 認可の監査
20073	PREPROCESSOR 認可解除の監査
20074	ポリシー作成の監査
20075	ポリシー説明更新の監査
20076	ポリシー状態更新の監査
20077	ポリシー名変更の監査
20078	ポリシー削除の監査
20079	ポリシーへのレルム追加の監査
20080	ポリシーからのレルム削除の監査
20081	ポリシーへのコマンド・ルール追加の監査
20082	ポリシーからのコマンド・ルール削除の監査
20083	ポリシー所有者追加の監査
20084	ポリシー所有者削除の監査
20085	メンテナンス認可の監査
20086	メンテナンス認可解除の監査

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.42 DVSYS.DV$ENFORCEMENT_AUDITビュー

DVSYS.DV$ENFORCEMENT_AUDITデータ・ディクショナリ・ビューでは、DVSYS.AUDIT_TRAIL$表の強制関連監査の

詳細が提供されます。

コマンド・ルール、レルムおよびファクタに関するユーザー違反を取得します。

たとえば:

```
SELECT USERNAME, ACTION_COMMMAND FROM DVSYS.DV$ENFORCEMENT_AUDIT
WHERE OWNER = 'HR';
```

次のような出力が表示されます。

```
USERNAME    ACTION_COMMMAND
----------- ------------------------------
PSMITH      CREATE_REALM
```

関連するビュー

. AUDSYS.DV$ENFORCEMENT_AUDITビュー

列	データ型	Null	説明
ID#	NUMBER	NOT NULL	監査レコードごとの数値識別子。
OS_USERNAME	VARCHAR(255)	NULL	アクションが監査対象となったユーザーのオペレーティング・システムのログイン・ユーザー名。
USERNAME	VARCHAR(128)	NULL	アクションが監査対象となったデータベース・ユーザーの名前。
USERHOST	VARCHAR(255)	NULL	クライアント・コンピュータ名。
TERMINAL	VARCHAR(255)	NULL	ユーザーの端末に対する識別子。
TIMESTAMP	DATE	NULL	監査証跡エントリの作成日時(ローカル・データベース・セッションのタイムゾーン)。
OWNER	VARCHAR(128)	NULL	アクションの影響を受けるオブジェクトの作成者、常時 DVSYS(DVSYS でオブジェクトが作成されるため)
OBJ_NAME	VARCHAR(128)	NULL	アクションの影響を受けるオブジェクトの名前。想定値は次のとおりです。 ● ROLE$ ● REALM$ ● CODE$

列	データ型	Null	説明
			• FACTOR$
ACTION	NUMBER	NOT NULL	数値のアクション・タイプ・コード。アクション・タイプに対応する名前は、ACTION_NAME 列に示されます。使用可能なアクションのリストは、表 24-4 を参照してください。
ACTION_NAME	VARCHAR(128)	NULL	ACTION 列の数値コードに対応するアクション・タイプの名前
ACTION_OBJECT_ID	NUMBER	NULL	OBJ_NAME に指定された表のレコードの一意の識別子
ACTION_OBJECT_NAME	VARCHAR(128)	NULL	OBJ_NAME に指定された表のレコードの一意の名前または固有のキー
ACTION_COMMAND	VARCHAR2(4000)	NULL	実行された結果、監査イベントがトリガーされたコマンド・プロシージャの SQL テキスト。
AUDIT_OPTION	VARCHAR2(4000)	NULL	結果として監査イベントがトリガーされたレコードに指定されたすべての監査オプションのラベル。たとえば、失敗または NULL になったときに監査することになっているファクタ設定操作では、これら 2 つのオプションを指します。
RULE_SET_ID	NUMBER	NULL	実行された結果、監査イベントがトリガーされたルール・セットの一意の識別子。
RULE_SET_NAME	VARCHAR(128)	NULL	実行された結果、監査イベントがトリガーされたルール・セットの一意の名前。
RULE_ID	NUMBER	NULL	使用されていません。
RULE_NAME	VARCHAR2(128)	NULL	使用されていません。
FACTOR_CONTEXT	VARCHAR2(4000)	NULL	監査イベントがトリガーされた時点での、現行セッションに対するすべてのファクタ識別子を含む XML 文書。

列	データ型	Null	説明
COMMENT_TEXT	VARCHAR2(4000)	NULL	監査対象となった文の詳細を示す、監査証跡エントリについてのテキスト・コメント。
SESSIONID	NUMBER	NOT NULL	Oracle セッションごとの数値識別子。
ENTRYID	NUMBER	NOT NULL	ID#列の値と同じ。
STATEMENTID	NUMBER	NOT NULL	起動された結果、監査イベントが生成された文の数値識別子。ほとんどの Oracle Database Vault イベントの場合、このパラメータは空です。
RETURNCODE	NUMBER	NOT NULL	アクションによって生成された Oracle エラー・コード。起動された結果、監査イベントが生成された文またはプロシージャに対するエラー・コード。ほとんどの Oracle Database Vault イベントの場合、このパラメータは空です。
EXTENDED_TIMESTAMP	TIMESTAMP(6)WITH TIME ZONE	NULL	UTC(協定世界時)タイムゾーンの、監査証跡エントリの作成時のタイムスタンプ(エントリに対するユーザー・ログインのタイムスタンプ)。
PROXY_SESSIONID	NUMBER	NULL	エンタープライズ・ユーザーがプロキシ機構を介してログインした場合の、プロキシ・セッションのシリアル番号
GLOBAL_UID	VARCHAR2(32)	NULL	ユーザーがエンタープライズ・ユーザーとしてログインした場合の、ユーザーのグローバル・ユーザー識別子。
INSTANCE_NUMBER	NUMBER	NULL	INSTANCE_NUMBER 初期化パラメータによって指定されるインスタンス番号
OS_PROCESS	VARCHAR2(16)	NULL	Oracle プロセスのオペレーティング・システム・プロセス識別子
CREATED_BY	VARCHAR2(128)	NULL	アクションが監査対象となったユーザーのデータベースのログイン・ユーザー名。
CREATE_DATE	DATE	NULL	SYSDATE の日付を基にした、アクションが発生し

列	データ型	Null	説明
			た日付。
UPDATED_BY	VARCHAR2(128)	NULL	CREATED_BY 列の値と同じ。
UPDATE_DATE	DATE	NULL	UPDATED_BY 列の値と同じ。

次の表では、DVSYS.DV$ENFORCEMENT_AUDITビューのACTION列で使用される値を示します。

表24-4 DVSYS.DV$ENFORCEMENT_AUDITビューのACTIONの値

アクション・タイプ・コード	アクション名
10000	ファクタ評価の監査
10001	ファクタ割当ての監査
10002	ファクタ式の監査
10003	レルム違反の監査
10004	レルム認可の監査
10005	コマンド認可の監査
10006	セキュア・ロールの監査
10007	セッション初期化の監査
10008	セキュア・コマンド認可の監査
10009	OLS セッション初期化の監査
10010	OLS ラベル・アップグレード試行の監査
10011	コマンド失敗の監査

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.43 DVSYS.DV$REALMビュー

DVSYS.DV$REALMデータ・ディクショナリ・ビューでは、Oracle Database Vaultのレルムの作成に使用された設定(割り当てられた監査オプションや、レルムが必須レルムかどうかなど)が示されます。

また、このビューでは、レルムの作成者や更新者、レルムの作成日や更新日などの情報も示されます。

たとえば:

```
SELECT NAME, CREATED_BY, TYPE FROM DVSYS.DV$REALM WHERE NAME LIKE 'Statistics';
```

次のような出力が表示されます。

```
NAME                          CREATED_BY TYPE
----------------------------- ---------- -----
Performance Statistics Realm JGODFREY   2
```

関連するビュー

. DBA_DV_REALMビュー

列	データ型	Null	説明
ID#	NUMBER	NOT NULL	レルムの ID 番号
NAME	VARCHAR2(128)	NOT NULL	レルムの名前
DESCRIPTION	VARCHAR2(1024)	NULL	レルムの説明
AUDIT_OPTIONS	NUMBER	NOT NULL	レルムに設定された監査オプション。可能な値については、表 14-9 の「audit_options」を参照してください。
REALM_TYPE	NUMBER	NULL	レルムのタイプ: 通常のレルムと必須レルムのどちらになるか。可能な値については、表 14-9 の「realm_type」を参照してください。
COMMON	VARCHAR2(3)	NULL	マルチテナント環境の場合は、レルムがローカルか共通かを示します。使用される値は、次のとおりです。 ● レルムが共通の場合は YES ● レルムがローカルの場合は NO
INHERITED	VARCHAR2(3)	NULL	COMMON 列の出力が YES の場合は、レルムの継承ステータスを示します。値は次のとおりです。 ● YES は、レルムが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、レルムがローカル・オブジェクトであるか、そ

列	データ型	Null	説明
			のコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
ENABLED	VARCHAR2(1)	NOT NULL	レルムが有効かどうか。可能な値については、表 14-9 の「enabled」を参照してください。
VERSION	NUMBER	NULL	レルムが作成された Oracle Database Vault のバージョン
CREATED_BY	VARCHAR2(128)	NULL	レルムを作成したユーザー
CREATE_DATE	DATE	NULL	レルムの作成日。
UPDATED_BY	VARCHAR2(128)	NULL	レルムを最後に更新したユーザー
UPDATE_DATE	DATE	NULL	レルムが最後に更新された日付

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.44 DVSYS.POLICY_OWNER_COMMAND_RULEビュー

DVSYS.POLICY_OWNER_COMMAND_RULEデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーに関連付けられているコマンド・ルールについて情報を確認できます。

ユーザーが確認できる情報の例としては、コマンド・ルールの名前、その関連付けられたルール・セット、およびそれが有効になっているかどうかがあります。DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。

たとえば:

```
SELECT COMMAND, OBJECT_OWNER, OBJECT_NAME FROM DVSYS.POLICY_OWNER_COMMAND_RULE;
```

次のような出力が表示されます。

```
COMMAND        OBJECT_OWNER  OBJECT_NAME
-------------- ------------- ------------
SELECT         HR            EMPLOYEES
```

関連するビュー

. DVSYS.POLICY_OWNER_POLICYビュー

列	データ型	Null	説明

列	データ型	Null	説明
COMMAND	VARCHAR(128)	NOT NULL	コマンド・ルールの名前。デフォルトのコマンド・ルールのリストは、「デフォルトのコマンド・ルール」を参照してください。
CLAUSE_NAME	VARCHAR(100)	NOT NULL	コマンド・ルールの作成に使用された、ALTER SYSTEM または ALTER SESSION SQL 文のどちらかの句。たとえば、ALTER SESSION 文の SET 句をリストできます。 使用可能な句の値をすべて示すリストについては、次のトピックを参照してください。 ● 表 16-2 ● 表 16-3
PARAMETER_NAME	VARCHAR(128)	NOT NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールの CLAUSE_NAME 設定からのパラメータ。
EVENT_NAME	VARCHAR(128)	NOT NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールで定義されているイベント。
COMPONENT_NAME	VARCHAR(128)	NOT NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールの EVENT_NAME 設定のコンポーネント。
ACTION_NAME	VARCHAR(128)	NOT NULL	ALTER SYSTEM または ALTER SESSION コマンド・ルールの EVENT_NAME 設定のアクション。
RULE_SET_NAME	VARCHAR(128)	NOT NULL	このコマンド・ルールに関連付けられたルール・セットの名前。
OBJECT_OWNER	VARCHAR(128)	NOT NULL	コマンド・ルールが影響するオブジェクトの所有者。
OBJECT_NAME	VARCHAR(128)	NOT NULL	コマンド・ルールが影響するデータベース・オブジェクト(データベース表など)の名前。
ENABLED	VARCHAR(1)	NOT NULL	Y はコマンド・ルールが有効になっていることを示し、N はコマンド・ルールが無効になっていることを示します。
PRIVILEGE_SCOPE	NUMBER	NOT NULL	廃止された列
ID#	NUMBER	NOT NULL	コマンド・ルールの ID 番号。これは、コマンド・ルール作成時に

列	データ型	Null	説明
			自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR(3)	NULL	コマンド・ルールがデフォルト(つまり、Oracle によって提供されている)コマンド・ルールであるかユーザーが作成したコマンド・ルールであるかを示します。使用される値は、次のとおりです。 ● コマンド・ルールがデフォルト・コマンド・ルールである場合は YES ● コマンド・ルールがユーザーが作成したコマンド・ルールである場合は NO

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.45 DVSYS.POLICY_OWNER_POLICYビュー

DVSYS.POLICY_OWNER_POLICYデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、他のポリシー所有者によって作成されたポリシーを含め、現在のデータベース・インスタンス内の既存のポリシーの名前、説明および状態などの情報を確認できます。

DVSYS.POLICY_OWNER_POLICYビューの列は、DBA_DV_POLICYの列と同じです。DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。

たとえば:

```
SELECT POLICY_NAME, STATE FROM DVSYS.POLICY_OWNER_POLICY
  WHERE STATE != 'ENABLED';
```

次のような出力が表示されます。

```
POLICY_NAME                       STATE
--------------------------------- --------
HR.EMPLOYEES_pol                  ENABLED
```

関連するビュー

- DBA_DV_POLICYビュー

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.46 DVSYS.POLICY_OWNER_REALMビュー

POLICY_OWNER_REALMデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーに関連付けられているレルムについて情報を確認できます。

ユーザーが確認できる情報の例としては、レルムの名前、監査オプション、タイプ、それが継承されているかどうか、およびそれが有効になっているかどうかがあります。DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。

たとえば:

```
SELECT NAME, ENABLED FROM DVSYS.POLICY_OWNER_REALM;
```

次のような出力が表示されます。

```
NAME                           ENABLED
------------------------------ --------
HR.EMPLOYEES_realm             S
```

関連するビュー

- DVSYS.POLICY_OWNER_REALM_AUTHビュー
- DVSYS.POLICY_OWNER_REALM_OBJECTビュー

列	データ型	Null	説明
NAME	VARCHAR(128)	NOT NULL	Database Vault ポリシーに関連付けられているレルムの名前。 レルムをすべて示すリストについては、DBA_DV_REALM ビューを参照してください。
DESCRIPTION	VARCHAR(1024)	NULL	レルムの説明
AUDIT_OPTIONS	NUMBER	NOT NULL	レルムに設定された監査オプション。可能な値については、表 14-9 の「audit_options」を参照してください。
REALM_TYPE	NUMBER	NULL	レルムのタイプ: 通常のレルムと必須レルムのどちらになるか。可能な値については、表 14-9 の「realm_type」を参照してください。
COMMON_REALM	VARCHAR2(3)	NULL	マルチテナント環境の場合は、レルムがローカルか共通かを示します。使用される値は、次のとおりです。 ● レルムが共通の場合は YES ● レルムがローカルの場合は NO
INHERITED_REALM	VARCHAR2(3)	NULL	COMMON 列の出力が YES の場合は、レルムの継承ステータスを示します。値は次のとおりです。 ● YES は、レルムが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、レルムがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリ

列	データ型	Null	説明
			ケーション・ルートでは、アプリケーション共通レルムのINHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
ENABLED	VARCHAR2(1)	NOT NULL	レルムの有効化ステータスを示します。使用される値は、次のとおりです。 ● YES(有効)の場合は Y ● NO(有効でない)場合は N ● シミュレーション・モードの場合は S
ID#	NUMBER	NOT NULL	レルムの ID 番号。これは、レルム作成時に自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR(3)	NOT NULL	レルムがデフォルト(つまり、Oracle によって提供されている)レルムであるかユーザーが作成したレルムであるかを示します。使用される値は、次のとおりです。 ● レルムがデフォルト・レルムである場合は YES ● レルムがユーザーが作成したレルムである場合は NO

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.47 DVSYS.POLICY_OWNER_REALM_AUTHビュー

DVSYS.POLICY_OWNER_REALM_AUTHデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーに関連付けられているレルムに与えられた認可ついて情報を確認できます。

ユーザーが確認できる情報の例としては、レルムの名前、権限受領者、および関連付けられたルール・セットがあります。DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。

たとえば:

```
SELECT REALM_NAME, INHERITED_REALM FROM DVSYS.POLICY_OWNER_REALM_AUTH;
```

次のような出力が表示されます。

```
REALM_NAME                INHERITED
------------------------- --------
HR.EMPLOYEES_realm        NO
```

関連するビュー

- DVSYS.POLICY_OWNER_REALMビュー
- DVSYS.POLICY_OWNER_REALM_OBJECTビュー

列	データ型	Null	説明
REALM_NAME	VARCHAR(128)	NOT NULL	Database Vault ポリシーに関連付けられているレルムの名前。 レルムをすべて示すリストについては、DBA_DV_REALM ビューを参照してください。
COMMON_REALM	VARCHAR2(3)	NULL	マルチテナント環境の場合は、レルムがローカルか共通かを示します。
INHERITED_REALM	VARCHAR2(3)	NULL	COMMON 列の出力が YES の場合は、レルムの継承ステータスを示します。値は次のとおりです。 ● YES は、レルムが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、レルムがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
GRANTEE	VARCHAR(128)	NOT NULL	所有者または参加者として認可するユーザーまたはロール名。
AUTH_RULE_SET_NAME	VARCHAR(128)	NULL	認可の前にチェックするルール・セット。ルール・セットの評価が True の場合は、認可が許可されます。
AUTH_OPTIONS	VARCHAR(4000)	NULL	レルム認可のタイプ。「参加者」または「所有者」のいずれかです。
COMMON_AUTH	VARCHAR(3)	NULL	マルチテナント環境の場合は、このレルムに対して認可されているユーザーがローカルか共通かを示します。使用される値は、次のとおりです。

列	データ型	Null	説明
			● ユーザーが共通ユーザーの場合は YES ● ユーザーがローカル・ユーザーの場合は NO
INHERITED_AUTH	VARCHAR(3)	NULL	使用される値は、次のとおりです。 ● YES ● NO

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.48 DVSYS.POLICY_OWNER_REALM_OBJECTビュー

DVSYS.POLICY_OWNER_REALM_OBJECTデータ・ディクショナリ・ビューでは、ユーザーが、Database Vaultポリシーに関連付けられているレルムに追加されたオブジェクトについて情報を確認できます。DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。

ユーザーが確認できる情報の例としては、レルムの名前、権限受領者、および関連付けられたルール・セットがあります。

たとえば:

```
SELECT REALM_NAME, OWNER, OBJECT_NAME, OBJECT_TYPE FROM
DVSYS.POLICY_OWNER_REALM_OBJECT;
```

次のような出力が表示されます。

```
REALM_NAME          OWNER  OBJECT_NAME OBJECT_TYPE
------------------- ------ ----------- -----------
HR.EMPLOYEES_realm HR     EMPLOYEES   TABLE
```

関連するビュー

- DVSYS.POLICY_OWNER_REALMビュー
- DVSYS.POLICY_OWNER_REALM_AUTHビュー

列	データ型	Null	説明
REALM_NAME	VARCHAR(128)	NOT NULL	Database Vault ポリシーに関連付けられているレルムの名前。 レルムをすべて示すリストについては、DBA_DV_REALMビューを参照してください。
COMMON_REALM	VARCHAR2(3)	NULL	マルチテナント環境の場合は、レルムがローカルか共通かを示します。
INHERITED_REAL	VARCHAR2(3)	NULL	COMMON 列の出力が YES の場合は、レルムの継承ステータス

列	データ型	Null	説明
M			を示します。値は次のとおりです。 ● YES は、レルムが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、レルムがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
OWNER	VARCHAR(128)	NOT NULL	オブジェクトを所有するデータベース・スキーマ所有者。
OBJECT_NAME	VARCHAR(128)	NOT NULL	レルムによって保護されるオブジェクトの名前。
OBJECT_TYPE	VARCHAR(32)	NOT NULL	レルムによって保護されるオブジェクトのタイプ(データベース表、ビュー、索引、ロールなど)。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.49 DVSYS.POLICY_OWNER_RULEビュー

DVSYS.POLICY_OWNER_RULEデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、ルール名とその式など、Database Vaultポリシー内のルール・セットに関連付けられているルールについて情報を確認できます。DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。

たとえば:

```
SELECT NAME, RULE_EXPR FROM DVSYS.POLICY_OWNER_RULE WHERE NAME = 'True';
```

次のような出力が表示されます。

```
NAME       RULE_EXPR
---------- --------
True       1=1
```

関連するビュー

- DVSYS.POLICY_OWNER_COMMAND_RULEビュー
- DVSYS.POLICY_OWNER_RULE_SETビュー

列	データ型	Null	説明
NAME	VARCHAR(128)	NOT NULL	ルールの名前。
RULE_EXPR	VARCHAR(1024)	NOT NULL	ルール用の PL/SQL 式。
COMMON	VARCHAR(3)	NOT NULL	マルチテナント環境の場合は、ルールがローカルか共通かを示します。使用される値は、次のとおりです。 ● ルールが共通の場合は YES ● ルールがローカルの場合は NO
INHERITED	VARCHAR(3)	NULL	COMMON 列の出力が YES の場合は、ルールの継承ステータスを示します。値は次のとおりです。 ● YES は、ルールが、コンテナ・ツリー階層の上位にある別のコンテナで定義されており、アプリケーション PDB でのアプリケーション同期プロセスの間の Database Vault ポリシー同期時にこのコンテナで継承されたことを意味します。 ● NO は、ルールがローカル・オブジェクトであるか、そのコンテナの共通であることを意味します。たとえば、アプリケーション・ルートでは、アプリケーション共通レルムの INHERITED 値は NO になりますが、CDB ルートの共通コマンド・ルールでは、INHERITED 値は YES になります。
ID#	NUMBER	NOT NULL	ルールの ID 番号。これは、ルール作成時に自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR(3)	NULL	ルールがデフォルト(つまり、Oracle によって提供されている)ルールであるかユーザーが作成したルールであるかを示します。使用される値は、次のとおりです。 ● ルールがデフォルト・ルールである場合は YES ● ルールがユーザーが作成したルールである場合は NO

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.50 DVSYS.POLICY_OWNER_RULE_SETビュー

DVSYS.POLICY_OWNER_RULE_SETデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーに関連付けられているルール・セットについて情報を確認できます。

ユーザーが確認できる情報の例としては、ルール・セットの名前、そのハンドラ情報、およびそれが有効になっているかどうかがあります。DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。

たとえば:

```
SELECT RULE_SET_NAME, ENABLED FROM DVSYS.POLICY_OWNER_RULE_SET;
```

次のような出力が表示されます。

```
RULE_SET_NAME  ENABLED
-------------- --------
Allow Sessions Y
```

関連するビュー

- DVSYS.POLICY_OWNER_COMMAND_RULEビュー
- DVSYS.POLICY_OWNER_RULEビュー
- DVSYS.POLICY_OWNER_RULE_SET_RULEビュー

列	データ型	Null	説明
RULE_SET_NAME	VARCHAR(128)	NOT NULL	ルール・セットの名前。
DESCRIPTION	VARCHAR(1024)	NULL	ルール・セットの説明。
ENABLED	VARCHAR(1)	NOT NULL	ルール・セットが有効になっているかどうかを示します。Y(YES)の場合、ルール・セットは有効になり、N(NO)の場合、ルール・セットは無効になります。
EVAL_OPTIONS_MEANING	VARCHAR(4000)	NULL	複数のルールが含まれるルール・セットの場合、評価されるルールの数が決まります。使用される値は、次のとおりです。 ● すべて True: ルール・セット自体が TRUE と評価されるために、ルール・セットのルールはすべて True と評価される必要があります。 ● いずれか True: ルール・セット自体が TRUE と評価されるために、少なくともルール・セットの 1 つのルールが True と評価される必要があります。
AUDIT_OPTIONS	NUMBER	NOT NULL	監査が使用される時期を示します。使用される値は、次のとおりです。

列	データ型	Null	説明
			● 0: 監査なし ● 1: 失敗時に監査 ● 2: 成功時に監査 ● 3: 失敗時と成功時の両方に監査
FAIL_OPTIONS_MEANING	VARCHAR(4000)	NULL	ルール・セットに対して監査レコードが作成される時期が決定されます。使用される値は、次のとおりです。 ● エラー・メッセージを表示しない。 ● エラー・メッセージを表示
FAIL_MESSAGE	VARCHAR(80)	NULL	FAIL_CODE 列に表示された失敗コードに関連付けられている失敗に対するエラー・メッセージ。
FAIL_CODE	VARCHAR(10)	NULL	FAIL_MESSAGE 列に表示されたメッセージに関連付けられているエラー・メッセージ番号。考えられる値の範囲は、-20000 から-20999 と 20000 から 20999 です。
HANDLER_OPTIONS	NUMBER	NOT NULL	エラー処理の使用方法が決まります。使用される値は、次のとおりです。 ● 0: エラー処理を無効にします。 ● 1: ルール・セット失敗時にハンドラをコールします。 ● 2: ルール・セット成功時にハンドラをコールします。
HANDLER	VARCHAR(1024)	NULL	カスタム・イベント・ハンドラ・ロジックを定義する PL/SQL ファンクションまたはプロシージャの名前。
IS_STATIC	VARCHAR2(5)	NULL	ユーザー・セッション中にルール・セットが評価される頻度を示します。使用される値は、次のとおりです。 ● TRUE: ルール・セットは 1 回評価され、ルール・セットの結果はユーザー・セッションで再利用されます。 ● FALSE(デフォルト): ルール・セットは、ユーザー・セッション中にアクセスされるたびに評価されます。

列	データ型	Null	説明
ID#	NUMBER)	NOT NULL	ルール・セットの ID 番号。これは、ルール・セット作成時に自動的に生成されます。
ORACLE_SUPPLIED	VARCHAR2(3)	NULL	ルール・セットがデフォルト(つまり、Oracle によって提供されている)ルール・セットであるかユーザーが作成したルール・セットであるかを示します。使用される値は、次のとおりです。 ● ルール・セットがデフォルト・ルール・セットである場合は YES ● ルール・セットがユーザーが作成したルール・セットである場合は NO

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.51 DVSYS.POLICY_OWNER_RULE_SET_RULEビュー

DVSYS.POLICY_OWNER_RULE_SET_RULEデータ・ディクショナリ・ビューでは、DV_POLICY_OWNERロールを付与されたユーザーが、Database Vaultポリシーで使用されるルールを含むルール・セットについて情報を確認できます。

ユーザーが確認できる情報の例としては、ルール・セットの名前、およびそれが有効になっているかどうかがあります。

DV_POLICY_OWNERロールを付与されたユーザーのみ、このビューを問合せできます。

たとえば:

```
SELECT ENABLED FROM DVSYS.POLICY_OWNER_RULE_SET_RULE WHERE RULE_SET_NAME = 'Can
Maintain Own Account';
```

次のような出力が表示されます。

```
ENABLED
----------
Y
```

関連するビュー

- DVSYS.POLICY_OWNER_COMMAND_RULEビュー
- DVSYS.POLICY_OWNER_RULE_SETビュー
- DVSYS.POLICY_OWNER_RULEビュー

列	データ型	Null	説明
RULE_SET_NAME	VARCHAR(128)	NOT NULL	ルールが含まれるルール・セットの名前。
RULE_NAME	VARCHAR(128)	NOT NULL	ルールの名前。

列	データ型	Null	説明
RULE_EXPR	VARCHAR(1024)	NOT NULL	RULE_NAME 列に表示されたルールを定義する PL/SQL 式。
ENABLED	VARCHAR(1)		ルールが有効になっているか無効になっているかを示します。Y(YES)の場合、ルール・セットは有効になり、N(NO)の場合、ルール・セットは無効になります。
RULE_ORDER	NUMBER	NOT NULL	ルール・セット内でルールが使用される順序。このリリースには適用されません。

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.52 AUDSYS.DV$CONFIGURATION_AUDITビュー

AUDSYS.DV$CONFIGURATION_AUDITビューは、統合監査証跡のDatabase Vault監査レコードを取得する点を除き、DVSYS.DV$CONFIGURATION_AUDITビューとほぼ同じです。

関連トピック

- DVSYS.DV$CONFIGURATION_AUDITビュー

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

24.53 AUDSYS.DV$ENFORCEMENT_AUDITビュー

AUDSYS.DV$ENFORCEMENT_AUDITビューは、統合監査証跡のDatabase Vault監査レコードを取得する点を除き、DVSYS.DV$ENFORCEMENT_AUDITビューとほぼ同じです。

関連トピック

- DVSYS.DV$ENFORCEMENT_AUDITビュー

親トピック: Oracle Database Vaultのデータ・ディクショナリ・ビュー

25 Oracle Database Vaultの監視

Database Vault構成への違反のチェックおよびポリシーの変更の追跡を行うことにより、Oracle Database Vaultを監視できます。

- Oracle Database Vaultの監視について
  Oracle Enterprise Manager Cloud ControlでDatabase Vaultホーム・ページを使用して、Database Vault対応データベースを監視できます。
- セキュリティ違反と構成変更の監視
  適切なロールを付与されているユーザーは、Oracle Database Vault Administratorを使用してセキュリティ違反および構成変更を監視できます。

25.1 Oracle Database Vaultの監視について

Oracle Enterprise Manager Cloud ControlでDatabase Vaultホーム・ページを使用して、Database Vault対応データベースを監視できます。

この機能では、違反未遂の上位5件と違反未遂者の上位5件が表示されます。違反未遂には、レルムに対する違反とコマンド・ルールに対する違反が含まれます。違反未遂者は、ユーザーとクライアント・ホストに分けられます。「違反未遂トップ5」の下の「Oracle Database Vault」リンクをクリックすることで、違反のタイプ、その発生時間、違反を試みたユーザーなど、詳細を表示できます。同様に、「違反未遂者トップ5」の下のユーザー・リンク(SYSなど)をクリックすると、実行したアクション、アクション発生元のホスト名、違反の発生時間など、各違反者について詳細情報を表示できます。データを手動でリフレッシュすることや、直近24時間以内などにデータ・ビューを制限することができます。このページには、生成されたすべてのアラートをリストする表も示されます。

これらのイベントを表示する前に、データベースを統合監査に移行していない場合は、AUDIT_TRAIL初期化パラメータがDBまたはDB, EXTENDEDに設定されていることを確認する必要があります。統合監査を使用するようにデータベースを移行した場合、追加の設定を構成する必要はありません。セキュリティ違反を確認する準備ができました。

関連トピック

- Oracle Database Vaultレポート

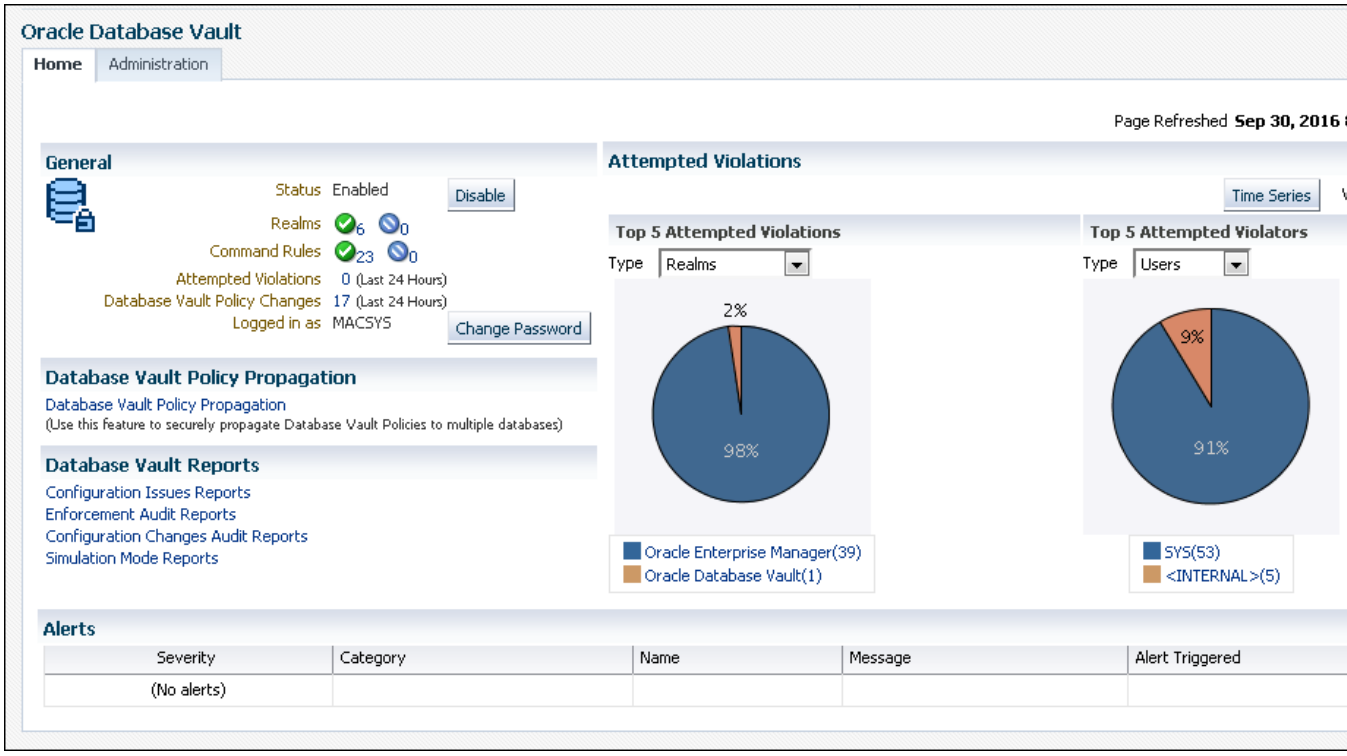
親トピック: Oracle Database Vaultの監視

25.2 セキュリティ違反と構成変更の監視

適切なロールを付与されているユーザーは、Oracle Database Vault Administratorを使用してセキュリティ違反および構成変更を監視できます。

1. DV_OWNER、DV_ADMINまたはDV_SECANALYSTロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「ホーム」タブを選択します。

   次のようなページが表示されます。



3. 直近7日間など、特定の時間の違反未遂を見つけるには、右上隅にある「時系列」ボタンの下のメニューから選択します。

   「時系列」ボタンをクリックすることで、円グラフをグラフに変更することもできます。

4. 構成問題レポート、強制監査レポート、構成変更の監査レポートおよびシミュレーション・モード・レポートを確認するには、Database Vaultレポートの下の該当するリンクを選択します。

   Database Vaultレポートの詳細は、Oracle Database Vaultレポートを参照してください。

親トピック: Oracle Database Vaultの監視

26 Oracle Database Vaultレポート

Oracle Database Vaultには、Database Vaultの構成設定などのアクティビティを追跡するレポートが用意されています。

- Oracle Database Vaultレポートについて
  Oracle Database Vaultには、データベースからのセキュリティに関する情報を示す一連のレポートが用意されています。
- Oracle Database Vaultレポートを実行できるユーザー
  Oracle Database Vaultレポートを実行するには、ユーザーに`DV_OWNER`、`DV_ADMIN`または`DV_SECANALYST`ロールが必要です。
- Oracle Database Vaultレポートの実行
  適切なロールを付与されているユーザーは、Database Vault AdministratorからOracle Database Vaultレポートを実行できます。
- Oracle Database Vault構成の問題のレポート
  構成の問題のレポートは、コマンド・ルール、ルール・セット、レルムおよびその他のOracle Database Vault構成を追跡します。
- Oracle Database Vaultの監査レポート
  統合監査を有効にしている場合、Oracle Database Vault監査レポートは、統合監査ポリシーの結果を取得します。
- Oracle Database Vaultの一般セキュリティ・レポート
  一般的なセキュリティ・レポートは、`PUBLIC`に関連したオブジェクト権限やデータベース・アカウントまたはロールに付与された権限などの情報を追跡します。

26.1 Oracle Database Vaultレポートについて

Oracle Database Vaultには、データベースからのセキュリティに関する情報を示すレポートが用意されています。

これらのレポートには、Oracle Database Vaultのカスタム監査イベント情報も表示されます。統合監査が有効な場合、レポートは、統合監査ポリシーの結果を取得します。

レポートは、次の2つのカテゴリに分類されます。

- Database Vaultレポート。このレポートでは、レルム、コマンド・ルール、ファクタ、ファクタのアイデンティティ、ルール・セットおよびセキュア・アプリケーション・ロールの構成上の問題をチェックできます。レルム違反、監査結果なども明らかになります。
- 一般セキュリティ・レポート。このレポートでは、オブジェクト権限、データベース・アカウントのシステム権限、機密オブジェクト、権限管理、強力なデータベース・アカウントおよびロール、初期化パラメータ、プロファイル、アカウントのパスワード、セキュリティ監査、その他のセキュリティ脆弱性レポートのステータスをチェックできます。

関連トピック

- Enterprise Manager Cloud ControlにおけるOracle Database Vault固有レポート
- Oracle Database Vaultのデータ・ディクショナリ・ビュー

親トピック: Oracle Database Vaultレポート

26.2 Oracle Database Vaultレポートを実行できるユーザー

Oracle Database Vaultレポートを実行するには、ユーザーにDV_OWNER、DV_ADMINまたはDV_SECANALYSTロールが必要です。

関連トピック

- DV_OWNER Database Vault所有者ロール
- DV_ADMIN Database Vault構成管理者ロール
- DV_SECANALYST Database Vaultセキュリティ分析者ロール

親トピック: Oracle Database Vaultレポート

26.3 Oracle Database Vaultレポートの実行

適切なロールを付与されているユーザーは、Database Vault AdministratorからOracle Database Vaultレポートを実行できます。

1. DV_OWNER、DV_ADMINまたはDV_SECANALYSTロールおよびSELECT ANY DICTIONARY権限を付与されているユーザーとして、Cloud ControlからOracle Database Vault Administratorにログインします。ログイン方法については、「Oracle Enterprise Cloud ControlからのOracle Database Vaultへのログイン」を参照してください。
2. 「ホーム」ページの「レポート」で、「Database Vaultレポート」を選択します。
3. 左側で、目的のレポートのカテゴリを選択します。
    - Database Vault構成の問題
    - Database Vault強制監査レポート
    - Database Vault構成変更
4. 「レポート」ページで、レポートを含むカテゴリを展開します。

    たとえば、「ルール・セット構成の問題」レポートを検索するには、「Database Vault構成の問題」を展開する必要があります。
5. レポート(「ルール・セット構成の問題」など)を選択します。

    レポートが右ペインに表示されます。
6. オプションで、「検索」フィールドを使用してレポートを絞り込みます。

    たとえば、特定のルール・セットを含むレポート済インシデントを検索できます。「検索」フィールドの内容は、レポートによって異なります。
7. レポートの表示が完了したら、「OK」ボタンをクリックします。

親トピック: Oracle Database Vaultレポート

26.4 Oracle Database Vault構成の問題のレポート

構成の問題のレポートは、コマンド・ルール、ルール・セット、レルムおよびその他のOracle Database Vault構成用の設定を追跡します。

- 「コマンド・ルール構成の問題」レポート

「コマンド・ルール構成の問題」レポートは、構成に問題があるコマンド・ルールを示します。

- 「ルール・セット構成の問題」レポート
  「ルール・セット構成の問題」レポートは、Oracle Database Vaultルール・セットの構成の問題を示します。
- 「レルム認可構成の問題」レポート
  「レルム認可構成の問題」レポートは、Oracle Database Vaultレルム構成の問題を示します。
- 「ファクタ構成の問題」レポート
  「ファクタ構成の問題」レポートは、Oracle Database Vaultファクタ構成の問題を示します。
- 「アイデンティティのないファクタ」レポート
  「アイデンティティのないファクタ」レポートは、アイデンティティが定義されていないOracle Database Vaultファクタを示します。
- 「アイデンティティ構成の問題」レポート
  「アイデンティティ構成の問題」レポートは、Oracle Database Vaultファクタ・アイデンティティ構成の問題を示します。
- 「セキュア・アプリケーション構成の問題」レポート
  「セキュア・アプリケーション構成の問題」レポートは、Database Vaultのセキュアなアプリケーション・ロールの構成の問題を示します。

親トピック: Oracle Database Vaultレポート

26.4.1 「コマンド・ルール構成の問題」レポート

「コマンド・ルール構成の問題」レポートは、構成に問題があるコマンド・ルールを示します。

これらの問題は次のとおりです。

- コマンド・ルールのルール・セットが無効である。
- コマンド・ルールのルール・セットが不完全である。
- コマンド・ルールのオブジェクト所有者が存在しない。この状況は、オブジェクトのユーザー・アカウントが削除された場合に発生する可能性があります。

親トピック: Oracle Database Vault構成の問題のレポート

26.4.2 「ルール・セット構成の問題」レポート

「ルール・セット構成の問題」レポートは、Oracle Database Vaultルール・セットの構成の問題を示します。

このレポートは、ルールが定義されていないかルール・セットに対して有効になっていない場合に追跡します。

親トピック: Oracle Database Vault構成の問題のレポート

26.4.3 「レルム認可構成の問題」レポート

「レルム認可構成の問題」レポートは、Oracle Database Vaultレルム構成の問題を示します。

これらの問題は次のとおりです。

- レルム認可のルール・セットが無効である。
- レルム認可に対して権限受領者が存在しない。
- レルム・セキュア・オブジェクトに対して所有者が存在しない。この状況は、ユーザー・アカウントが削除された場合に発生する可能性があります。

しかし、ほとんどの場合、このような問題はレルムの構成時および検証時に発見されます。

親トピック: Oracle Database Vault構成の問題のレポート

26.4.4 「ファクタ構成の問題」レポート

「ファクタ構成の問題」レポートは、Oracle Database Vaultファクタ構成の問題を示します。

これらの問題は次のとおりです。

- ファクタ割当てのルール・セットが無効である。
- ファクタ割当てのルール・セットが不完全である。
- ファクタの監査オプションが無効である。
- ファクタ取得メソッドまたは定数が存在しない。
- サブファクタ(子ファクタ)がファクタ・アイデンティティにリンクされていない。
- サブファクタ(子ファクタ)がラベル・ファクタにリンクされていない。
- ファクタに対してOracle Label Securityポリシーが存在しない。

親トピック: Oracle Database Vault構成の問題のレポート

26.4.5 「アイデンティティのないファクタ」レポート

「アイデンティティのないファクタ」レポートは、アイデンティティが定義されていないOracle Database Vaultファクタを示します。

`Background_Job_Id`などの一部のファクタの場合、これは本当の問題ではない可能性がありますが、アクセス制御の構成が完全かどうか、すべてのファクタの構成を考慮に入れているかどうかの判断にこのレポートを役立てることができます。

親トピック: Oracle Database Vault構成の問題のレポート

26.4.6 「アイデンティティ構成の問題」レポート

「アイデンティティ構成の問題」レポートは、Oracle Database Vaultファクタ・アイデンティティ構成の問題を示します。

これらの問題は次のとおりです。

- ラベル・アイデンティティ(このアイデンティティのOracle Label Securityラベル)が削除されていて、すでに存在しない。
- アイデンティティに対してマップが存在しない。

親トピック: Oracle Database Vault構成の問題のレポート

26.4.7 「セキュア・アプリケーション構成の問題」レポート

「セキュア・アプリケーション構成の問題」レポートは、Database Vaultのセキュアなアプリケーション・ロールの構成の問題を示します。

これらの問題は次のとおりです。

- データベース・ロールが存在しない。この状況は、データベース・ロールが削除された場合に発生する可能性があります。
- ロールのルール・セットが無効である。
- ロールのルール・セットが不完全である。

親トピック: Oracle Database Vault構成の問題のレポート

26.5 Oracle Database Vaultの監査レポート

統合監査を有効にしている場合、Oracle Database Vault監査レポートは、統合監査ポリシーの結果を取得します。

- 「レルムの監査」レポート
  「レルムの監査」レポートは、レルム保護およびレルム認可の操作によって生成される監査レコードを示します。
- 「コマンド・ルールの監査」レポート
  「コマンド・ルールの監査」レポートは、コマンド・ルール処理の操作によって生成される監査レコードを示します。
- 「ファクタの監査」レポート
  「ファクタの監査」レポートは、評価できなかった、または様々な条件下で監査レコードを作成するように設定されたファクタを示します。
- 「Label Security統合の監査」レポート
  「Label Security統合の監査」レポートは、セッション初期化の操作によって生成される監査レコード、およびLabel Securityのセッション・ラベル割当ての操作を示します。
- 「コアDatabase Vault監査証跡」レポート
  「コアDatabase Vault監査証跡」レポートは、コア・アクセス・セキュリティ・セッション初期化の操作によって生成される監査レコードを示します。
- 「セキュア・アプリケーション・ロールの監査」レポート
  「セキュア・アプリケーション・ロールの監査」レポートは、Oracle Database Vaultセキュア・アプリケーション・ロールを有効にする操作によって生成される監査レコードを示します。

親トピック: Oracle Database Vaultレポート

26.5.1 「レルムの監査」レポート

「レルムの監査」レポートは、レルム保護およびレルム認可の操作によって生成される監査レコードを示します。

レルム認可はルール・セットを使用して管理し、そのルール・セットの処理結果を監査できます。レルム違反は、レルムで保護されているオブジェクトに対してアクションを実行するデータベース・アカウントに、そのアクションを実行する権限がない場合に発生します。レルムに関連付けられているルール・セットを指定しない場合でも、Oracle Database Vaultによって違反が監査されます。レルムを構成する際に、レルム違反のインスタンスを監査するように設定できます。この情報を使用して、セキュリティ違反の試行を調査できます。

親トピック: Oracle Database Vaultの監査レポート

26.5.2 「コマンド・ルールの監査」レポート

「コマンド・ルールの監査」レポートは、コマンド・ルール処理の操作によって生成される監査レコードを示します。

コマンド・ルールを構成する際に、ルール・セットの処理結果を監査するように設定できます。

親トピック: Oracle Database Vaultの監査レポート

26.5.3 「ファクタの監査」レポート

「ファクタの監査」レポートは、評価できなかった、または様々な条件下で監査レコードを作成するように設定されたファクタを示します。

また、このレポートにはファクタ設定の失敗も表示されます。

ファクタ・アイデンティティを解決できない、または割り当てることができない(データが見つからない、行が多すぎるなど)インスタンスを監査できます。ファクタには、実行時にアイデンティティをファクタに割り当てるルール・セットを関連付けることができます。ファクタを構成する際に、ルール・セットの処理結果を監査するように設定できます。

親トピック: Oracle Database Vaultの監査レポート

26.5.4 「Label Security統合の監査」レポート

「Label Security統合の監査」レポートは、セッション初期化の操作によって生成される監査レコード、およびLabel Securityのセッション・ラベル割当ての操作を示します。

Label Securityセッションの初期化に失敗したインスタンス、およびセッションで最大セッション・ラベルを超えるラベルの設定がLabel Securityコンポーネントによって妨げられているインスタンスを監査できます。

親トピック: Oracle Database Vaultの監査レポート

26.5.5 「コアDatabase Vault監査証跡」レポート

「コアDatabase Vault監査証跡」レポートは、コア・アクセス・セキュリティ・セッションの初期化操作によって生成された監査レコードを示します。

アクセス・セキュリティ・セッションの初期化に失敗したインスタンスを監査できます。次のデータが表示されます。

データA-R	データR-U
アカウント	ルール・セット
コマンド	タイムスタンプ
インスタンス番号	ルール・セット
オブジェクト名	ユーザー・ホスト
戻りコード	-

親トピック: Oracle Database Vaultの監査レポート

26.5.6 「セキュア・アプリケーション・ロールの監査」レポート

「セキュア・アプリケーション・ロールの監査」レポートは、Oracle Database Vaultセキュア・アプリケーション・ロールを有効にする操作によって生成される監査レコードを示します。

関連トピック

- Oracle Database Vaultのセキュア・アプリケーション・ロールの構成

親トピック: Oracle Database Vaultの監査レポート

26.6 Oracle Database Vaultの一般セキュリティ・レポート

一般的なセキュリティ・レポートは、PUBLICに関連したオブジェクト権限やデータベース・アカウントまたはロールに付与された権限などの情報を追跡します。

- オブジェクト権限レポート
  オブジェクト権限レポートは、PUBLICの影響を受ける権限、直接オブジェクト権限およびオブジェクトの依存性を追跡します。
- データベース・アカウントのシステム権限レポート
  データベース・アカウントのシステム権限レポートは、直接、間接、階層的およびANYシステム権限などのアクティビティを追跡します。
- 機密オブジェクト・レポート
  機密オブジェクト・レポートは、SYSスキーマ・オブジェクトのEXECUTE権限の付与や機密オブジェクトへのアクセスなどのアクティビティを追跡します。
- 権限管理 - サマリー・レポート
  権限管理サマリー・レポートは、権限受領者、所有者および権限別の権限配布を追跡します。
- 強力なデータベース・アカウントおよびロールのレポート
  強力なデータベース・アカウントおよびロールのレポートは、WITH ADMIN権限などの強力な権限を付与されたユーザーに関する情報を追跡します。
- 初期化パラメータおよびプロファイルのレポート
  初期化パラメータおよびプロファイルのレポートは、データベース・パラメータ、リソース・プロファイルおよびシステム制限を追跡します。
- データベース・アカウント・パスワードのレポート
  データベース・アカウント・パスワードのレポートは、デフォルト・パスワード、およびデータベース・アカウントのアカウント・ステータスを追跡します。
- セキュリティ監査レポート: コア・データベース監査レポート
  コア・データベース監査レポートは、データベース監査証跡レコードを示します。
- その他のセキュリティ脆弱性レポート
  その他のセキュリティ脆弱性レポートは、Javaポリシーの付与またはオペレーティング・システムのディレクトリ・オブジェクトなど、アクティビティで発生する可能性のある脆弱性の問題を追跡します。

親トピック: Oracle Database Vaultレポート

26.6.1 オブジェクト権限レポート

オブジェクト権限レポートは、PUBLICの影響を受ける権限、直接オブジェクト権限およびオブジェクトの依存性を追跡します。

- 「PUBLICを使用したオブジェクト・アクセス」レポート
  「PUBLICを使用したオブジェクト・アクセス」レポートは、アクセス権がPUBLICに付与されているすべてのオブジェクトを示します。
- 「PUBLIC以外でのオブジェクト・アクセス」レポート
  「PUBLIC以外でのオブジェクト・アクセス」レポートは、「レポート・パラメータ」ページでアカウントによって使用されるオブジェクト・アクセスを説明します。
- 「直接オブジェクト権限」レポート
  「直接オブジェクト権限」レポートには、非システム・データベース・アカウントに付与されている直接オブジェクト権限が表示されます。

- 「オブジェクトの依存性」レポート
  「オブジェクトの依存性」レポートは、データベース内のプロシージャ、パッケージ、ファンクション、パッケージ本体、およびトリガー間の依存性を説明しています。

親トピック: Oracle Database Vaultの一般セキュリティ・レポート

26.6.1.1 「PUBLICでのオブジェクト・アクセス」レポート

「PUBLICでのオブジェクト・アクセス」レポートは、PUBLICにアクセス権が付与されているすべてのオブジェクトをリストします。

「レポート・パラメータ」ページで指定されたデータベース・アカウントについて、PUBLICへのオブジェクト付与によるすべてのオブジェクト・アクセスの詳細を示します。レポート・パラメータ・ページでは、権限、オブジェクト所有者またはオブジェクト名に基づいて結果をフィルタ処理できます。

ノート:

このレポートは、デフォルトを選択すると非常に大きなサイズになることがあります。

親トピック: オブジェクト権限レポート

26.6.1.2 「PUBLIC以外でのオブジェクト・アクセス」レポート

「PUBLIC以外でのオブジェクト・アクセス」レポートは、「レポート・パラメータ」ページでアカウントによって使用されるオブジェクト・アクセスを説明します。

PUBLICへの付与を除く、アカウントへの付与を直接またはロールを通じてチェックします。

「レポート・パラメータ」ページでは、権限、オブジェクト所有者またはオブジェクト名に基づいて結果をフィルタにかけることができます。

ノート:

このレポートは、デフォルトを選択すると非常に大きなサイズになることがあります。

親トピック: オブジェクト権限レポート

26.6.1.3 「直接オブジェクト権限」レポート

「直接オブジェクト権限」レポートは、非システム・データベース・アカウントに付与された直接オブジェクト権限を示します。

次のデータベース・アカウントは、このレポートの対象外です。

アカウントC-O	アカウントP-W
CTXSYS	PUBLIC
DMSYS	SYS
DVSYS	SYSMAN

アカウントC-O	アカウントP-W
LBACSYS	SYSTEM
MDSYS	WKSYS
ORDSYS	WMSYS

親トピック: オブジェクト権限レポート

26.6.1.4 「オブジェクトの依存性」レポート

「オブジェクトの依存性」レポートは、データベース内のプロシージャ、パッケージ、ファンクション、パッケージ本体、およびトリガー間の依存性を説明しています。

このレポートには、データベース・リンクなしで作成されたビューへの依存性が含まれます。

このレポートは、最低限の権限の原則を使用するセキュリティ・ポリシーを既存のアプリケーション用に開発するために役立ちます。データベース・オブジェクト(UTL_FILEパッケージなど)がPUBLICに付与された権限または他のなんらかのグローバル・ロールを保持する場合、「オブジェクトの依存性」レポートを使用して、そのオブジェクトに依存するアカウントを割り出し、アカウントでオブジェクトをどのように使用しているかを判断できます。レポートを実行するには、依存性を調べるデータベース・アカウントと、そのアカウントが依存するオブジェクトを、レポート・パラメータ・ページに入力します。

「レポート結果」ページには、依存オブジェクトとオブジェクト・タイプの他に、ソース・オブジェクトの名前とタイプが表示されます。このレポートには、機密オブジェクトが使用されている可能性がある場所が示されます。いくつかのアカウントを調べることで、制限付きロールの開発に役立つパターンを発見できることがあります。広く使用されている機密オブジェクトに対するPUBLIC権限を、これらの制限付きロールで置換えできます。

親トピック: オブジェクト権限レポート

26.6.2 データベース・アカウントのシステム権限レポート

データベース・アカウントのシステム権限レポートは、直接、間接、階層的およびANYシステム権限などのアクティビティを追跡します。

- 「データベース・アカウントごとの直接システム権限」レポート
  「データベース・アカウントごとの直接システム権限」レポートは、「レポート・パラメータ」ページで選択されたデータベース・アカウントに直接付与されているシステム権限を示します。
- 「データベース・アカウントごとの直接および間接システム権限」レポート
  「データベース・アカウントごとの直接および間接システム権限」レポートには、「レポート・パラメータ」ページで選択されたデータベース・アカウントのシステム権限が表示されます。
- 「データベース・アカウントごとの階層システム権限」レポート
  「データベース・アカウントごとの階層システム権限」レポートは、ロールベースのシステム権限および直接システム権限の階層の内訳を示します。
- 「データベース・アカウントのANYシステム権限」レポート
  「データベース・アカウントのANYシステム権限」レポートには、指定されたデータベース・アカウントまたはロールに付与されたすべてのANYシステム権限が表示されます。
- 「権限ごとのシステム権限」レポート
  「権限ごとのシステム権限」レポートは、「レポート・パラメータ」ページで選択されたシステム権限を持つデータベース・ア

カウントおよびロールを示します。

親トピック: Oracle Database Vaultの一般セキュリティ・レポート

26.6.2.1 「データベース・アカウントごとの直接システム権限」レポート

「データベース・アカウントごとの直接システム権限」レポートは、「レポート・パラメータ」ページで選択されたデータベース・アカウントに直接付与されているシステム権限を示します。

このレポートは、権限にWITH ADMINオプションが付与されているかどうかも示します。

親トピック: データベース・アカウントのシステム権限レポート

26.6.2.2 「データベース・アカウントごとの直接および間接システム権限」レポート

「データベース・アカウントごとの直接および間接システム権限」レポートは、「レポート・パラメータ」ページで選択されたデータベース・アカウントに対するシステム権限を示します。

システム権限は、直接またはWITH ADMINステータスのデータベース・ロールを介して付与されていることがあります。

親トピック: データベース・アカウントのシステム権限レポート

26.6.2.3 「データベース・アカウントごとの階層システム権限」レポート

「データベース・アカウントごとの階層システム権限」レポートは、ロールベースのシステム権限および直接システム権限の階層の内訳を示します。

これらの権限は、「レポート・パラメータ」ページで指定されたデータベース・アカウントに対して付与されます。

親トピック: データベース・アカウントのシステム権限レポート

26.6.2.4 「データベース・アカウントのANYシステム権限」レポート

「データベース・アカウントのANYシステム権限」レポートは、指定されたデータベース・アカウントまたはロールに付与されているANYシステム権限を示します。

ANYシステム権限は、とても強力であるため、アカウントおよびロールに慎重に割り当てる必要があります。

親トピック: データベース・アカウントのシステム権限レポート

26.6.2.5 「権限ごとのシステム権限」レポート

「権限ごとのシステム権限」レポートは、「レポート・パラメータ」ページで選択されたシステム権限を持つデータベース・アカウントおよびロールを示します。

権限を制御するもう1つの方法は、権限分析ポリシーの作成による権限使用の分析です。

親トピック: データベース・アカウントのシステム権限レポート

26.6.3 機密オブジェクト・レポート

機密オブジェクト・レポートは、SYSスキーマ・オブジェクトのEXECUTE権限の付与や機密オブジェクトへのアクセスなどのアクティビティを追跡します。

- 「強力なSYSパッケージに対するEXECUTE権限」レポート
  「強力なSYSパッケージに対するEXECUTE権限」レポートは、強力なSYSパッケージに対するEXECUTE権限を持つデータベース・アカウントおよびロールを示します。

- 「機密オブジェクトへのアクセス」レポート
  「機密オブジェクトへのアクセス」レポートには、機密情報を含むシステム表またはビューのオブジェクト権限があるデータベース・アカウントおよびロールが表示されます。
- 「SYS PL/SQLプロシージャに対するPUBLIC EXECUTE権限」レポート
  「SYS PL/SQLプロシージャに対するPUBLIC EXECUTE権限」のレポートには、SYSが所有するパッケージに対してEXECUTE権限を持っているデータベース・アカウントおよびロールが表示されます。
- 「SYSDBA/SYSOPER権限を持つアカウント」レポート
  「SYSDBA/SYSOPER権限を持つアカウント」レポートは、SYS優先接続権限のあるデータベース・アカウントを示します。

親トピック: Oracle Database Vaultの一般セキュリティ・レポート

26.6.3.1 「強力なSYSパッケージに対するEXECUTE権限」レポート

「強力なSYSパッケージに対するEXECUTE権限」レポートは、強力なSYSパッケージに対するEXECUTE権限を持つデータベース・アカウントおよびロールを示します。

たとえば、これらのパッケージ・タイプを使用すると、オペレーティング・システム・リソースにアクセスできます。

次のシステムPL/SQLパッケージが対象となります。

パッケージD-D	パッケージD-U
DBMS_ALERT	DBMS_RANDOM
DBMS_BACKUP_RESTORE	DBMS_REPAIR
DBMS_CAPTURE_ADM	DBMS_REPCAT
DBMS_DDL	DBMS_REPCAT_ADMIN
DBMS_DISTRIBUTED_TRUST_ADMIN	DBMS_RESOURCE_MANAGER
DBMS_FGA	DBMS_RESOURCE_MANAGER_PRIVS
DBMS_JOB	DBMS_RLS
DBMS_LDAP	DBMS_SESSION
DBMS_LOB	DEBUG_EXTPROC
DBMS_LOGMNR	UTL_FILE
DBMS_LOGMNR_D	UTL_HTTP
DBMS_OBFUSCATION_TOOLKIT	UTL_SMTP
DBMS_ORACLE_TRACE_AGENT	UTL_TCP

パッケージD-D	パッケージD-U
DBMS_PIPE	-

親トピック: 機密オブジェクト・レポート

26.6.3.2 「機密オブジェクトへのアクセス」レポート

「機密オブジェクトへのアクセス」レポートは、機密情報を含むシステム表またはビューに対してオブジェクト権限を持つデータベース・アカウントおよびロールを示します。

このレポートには、次のシステム表およびビューが含まれます。

表/ビューA-O	表/ビューP-S
ALL_SOURCE	PROFILE$
ALL_USERS	PROXY_ROLE_DATA$
APPROLE$	PROXY_ROLE_INFO$
AUD$	ROLE_ROLE_PRIVS
AUDIT_TRAIL$	SOURCE$
DBA_ROLE_PRIVS	STATS$SQLTEXT
DBA_ROLES	STATS$SQL_SUMMARY
DBA_TAB_PRIVS	SYSTEM_PRIVILEGE_MAP
DBMS_BACKUP_RESTORE	TABLE_PRIVILEGE_MAP
DEFROLE$	TRIGGER$
FGA_LOG$	USER$
LINK$	USER_HISTORY$
OBJ$	USER_TAB_PRIVS
OBJAUTH$	SYSTEM_PRIVILEGE_MAP
OBJPRIV$	-

親トピック: 機密オブジェクト・レポート

26.6.3.3 「SYS PL/SQLプロシージャに対するPUBLIC EXECUTE権限」レポート

「SYS PL/SQLプロシージャに対するPUBLIC EXECUTE権限」レポートは、SYSが所有する対象に対してEXECUTE権限を持つデータベース・アカウントおよびロールを示します。

このレポートは、どの権限をPUBLICまたはその他のアカウントやロールから取り消すことができるかの判断に使用できます。これにより、最低限の権限の原則を使用するセキュリティ・ポリシーの全体的な実装の一部としての脆弱性が軽減されます。

親トピック: 機密オブジェクト・レポート

26.6.3.4 「SYSDBA/SYSOPER権限を持つアカウント」レポート

「SYSDBA/SYSOPER権限を持つアカウント」レポートは、SYS権限が付与された接続の権限を持つデータベース・アカウントを示します。

このレポートは、そのアカウントが外部パスワードを使用するかどうかも示します。ただし、このレポートには、SYSDBAになり得るオペレーティング・システム・ユーザーは含まれないことに注意してください。

親トピック: 機密オブジェクト・レポート

26.6.4 権限管理 - サマリー・レポート

権限管理サマリー・レポートは、権限受領者、所有者および権限別の権限配布を追跡します。

- 「権限受領者ごとの権限の配布」レポート
  「権限受領者ごとの権限の配布」レポートは、データベース・アカウントまたはロールに付与された権限の総数を示します。
- 「権限受領者、所有者ごとの権限の配布」レポート
  「権限受領者、所有者ごとの権限の配布」レポートには、権限受領者およびオブジェクトの所有者ごとに権限数が表示されます。
- 「権限受領者、所有者、権限ごとの権限の配布」レポート
  「権限受領者、所有者、権限ごとの権限の配布」レポートは、権限、権限受領者、オブジェクトの所有者ごとの権限数を示します。

関連項目:

これらのレポートに表示された件数の基になっている値を調べるには、「DBA_DV_PUB_PRIVSビュー」を参照してください

親トピック: Oracle Database Vaultの一般セキュリティ・レポート

26.6.4.1 「権限受領者ごとの権限の配布」レポート

「権限受領者ごとの権限の配布」レポートは、データベース・アカウントまたはロールに付与された権限の総数を示します。

このレポートを使用すると、強力な権限を持つ可能性のあるアカウントおよびロールを認識できます。

親トピック: 権限管理 - サマリー・レポート

26.6.4.2 「権限受領者、所有者ごとの権限の配布」レポート

「権限受領者、所有者ごとの権限の配布」レポートは、オブジェクトの権限受領者および所有者に基づいて権限の総数を示します。

このレポートを使用すると、強力な権限を持つ可能性のあるアカウントまたはロールを認識できます。このレポートは、潜在的な

侵入者または内部関係者によって、攻撃または侵害するためのアカウントとして強力な権限を持つアカウントが狙われていると疑われる場合に使用できます。たとえば、侵入者がパスワードを推測することでアカウントを侵害できる場合、侵入者はすでに保持している権限よりも多くの権限を入手できます。

親トピック: 権限管理 - サマリー・レポート

26.6.4.3 「権限受領者、所有者、権限ごとの権限の配布」レポート

「権限受領者、所有者、権限ごとの権限の配布」レポートは、権限、権限受領者およびオブジェクト所有者に基づいて権限の総数を示します。

このレポートを使用すると、強力な権限を持つ可能性のあるアカウントまたはロールを認識できます。

親トピック: 権限管理 - サマリー・レポート

26.6.5 強力なデータベース・アカウントおよびロールのレポート

強力なデータベース・アカウントおよびロールのレポートは、WITH ADMIN権限などの強力な権限を付与されたユーザーに関する情報を追跡します。

- 「WITH ADMIN権限の付与」レポート
  「WITH ADMIN権限の付与」レポートは、WITH ADMIN句によって権限を付与されているデータベース・アカウントおよびロールをすべて示します。
- 「DBAロールを持つアカウント」レポート
  「DBAロールを持つアカウント」レポートは、DBAロールが付与されているデータベース・アカウントをすべて示します。
- 「セキュリティ・ポリシー除外」レポート
  「セキュリティ・ポリシー除外」レポートは、EXEMPT ACCESS POLICYシステム権限が付与されている(Oracle Database Vault以外の)データベース・アカウントおよびロールを示します。
- BECOME USERレポート
  BECOME USERレポートは、BECOME USERシステム権限を持つデータベース・アカウント・ロールを示します。
- 「ALTER SYSTEM」または「ALTER SESSION」レポート
  「ALTER SYSTEM」または「ALTER SESSION」レポートには、ALTER SYSTEMまたはALTER SESSION権限のあるデータベース・アカウントおよびロールが表示されます。
- 「パスワード履歴へのアクセス」レポート
  「パスワード履歴へのアクセス」レポートは、各アカウントで前に使用されていたハッシュ・パスワードが格納されているUSER_HISTORY$表へのアクセス権を持つデータベース・アカウントを示します。
- WITH GRANT権限の付与
  「WITH GRANT権限の付与」レポートは、WITH GRANT句を含む権限を付与されているデータベース・アカウントを示します。
- 「指定されたロールを持つロールとアカウント」レポート
  このレポートは、ロールが付与されているデータベース・アカウントおよびロールを示します。
- 「カタログ・ロールを持つデータベース・アカウント」レポート
  「カタログ・ロールを持つデータベース・アカウント」レポートは、カタログ関連のロールが付与されているすべてのデータベース・アカウントおよびロールを示します。
- 「AUDIT権限」レポート
  「AUDIT権限」レポートは、AUDIT ANYまたはAUDIT SYSTEM権限を持つデータベース・アカウントおよびロールをすべて示します。
- 「OSセキュリティ脆弱性に関する権限」レポート

「OSセキュリティ脆弱性に関する権限」レポートは、機密情報をオペレーティング・システムにエクスポートするための権限を持つデータベース・アカウントおよびロールを示します。

親トピック: Oracle Database Vaultの一般セキュリティ・レポート

26.6.5.1 「WITH ADMIN権限の付与」レポート

「WITH ADMIN権限の付与」レポートは、WITH ADMIN句によって権限を付与されているデータベース・アカウントおよびロールをすべて示します。

この権限は、別のアカウントに必要以上のシステム権限を付与するために悪用される可能性があります。

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.2 「DBAロールを持つアカウント」レポート

「DBAロールを持つアカウント」レポートは、DBAロールが付与されているデータベース・アカウントをすべて示します。

DBAロールは、不正に利用される可能性のある特権ロールです。時間を節約し、アカウントに本当に必要な最小権限を判断しないで済むように、このロールがデータベース・アカウントに付与されることはよくあります。このレポートは、最低限の権限の原則を使用するポリシーを、既存のデータベースに適用する際に役立ちます。

関連項目:

権限ロールを持つユーザーの決定に関するガイドラインは、「Oracle Database Vaultセキュリティ・ガイドライン」を参照してください

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.3 「セキュリティ・ポリシー除外」レポート

「セキュリティ・ポリシー除外」レポートは、EXEMPT ACCESS POLICYシステム権限を持つ(Oracle Database Vault以外の)データベース・アカウントおよびロールを示します。

この権限を持つアカウントは、すべてのVirtual Private Database(VPD)のポリシー・フィルタと、Oracle Virtual Private Databaseを間接的に使用するOracle Label Securityポリシーを無視できます。この権限は、Oracle Virtual Private DatabaseまたはOracle Label Securityによって保護される表の機密情報にアクセスするターゲットを示すため、どうしても必要な場合にかぎり付与すべき強力なシステム権限です。「Oracle Database Vaultの監査」で説明するように監査ポリシーを使用すると、この権限の使用を監査できます。

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.4 「BECOME USER」レポート

BECOME USERレポートは、BECOME USERシステム権限を持つデータベース・アカウントおよびロールを示します。

BECOME USER権限は、非常に強力なシステム権限で、Oracle Data Pumpで使用するIMP_FULL_DATABASEロールおよびEXP_FULL_DATABASEロールを有効にします。この権限を持つアカウントは、機密情報を入手したり、アプリケーションを侵害するために悪用されたりする可能性があります。

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.5 ALTER SYSTEMまたはALTER SESSIONレポート

「ALTER SYSTEM」レポートまたは「ALTER SESSION」レポートは、ALTER SYSTEMまたはALTER SESSION権限を持つデータベース・アカウントおよびロールをすべて示します。

これらの権限は、本当に必要なアカウントおよびロール(SYSアカウントやDV_ADMINロールなど)に限定することをお薦めします。ALTER SYSTEM文を使用すると、Oracle Database Vaultのセキュリティ強化サービスの一部として推奨値に設定されているセキュリティ関連のデータベース初期化パラメータを変更できます。ALTER SYSTEM文とALTER SESSION文の両方を使用すると、機密構成情報が含まれる可能性があるデータベース・トレース・ファイルをオペレーティング・システムにダンプできます。

関連項目:

ALTER SYSTEMおよびALTER SESSION権限の使用のガイドラインは、「ALTER SYSTEMおよびALTER SESSION権限のセキュリティの考慮事項について」を参照してください

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.6 「パスワード履歴へのアクセス」レポート

「パスワード履歴へのアクセス」レポートは、各アカウントで前に使用されていたハッシュ・パスワードが格納されているUSER_HISTORY$表へのアクセス権を持つデータベース・アカウントを示します。

この表には、各アカウントで以前使用されたハッシュ・パスワードが格納されています。

この表へのアクセスにより、データベースをハッキングする何者かがアカウントの既存のパスワードを簡単に推測できるようになります。

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.7 WITH GRANT権限レポート

WITH GRANT権限レポートは、WITH GRANT句によって権限を付与されているデータベース・アカウントを示します。

WITH GRANTは、オブジェクトレベルの権限に使用されることに注意してください。WITH GRANTオプションを使用して権限が付与されているアカウントは、別のアカウントにオブジェクト権限を付与するために悪用される可能性があります。

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.8 「指定されたロールを持つロールとアカウント」レポート

このレポートは、ロールが付与されているデータベース・アカウントおよびロールを示します。

このレポートは、依存性の分析のために用意されています。

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.9 「カタログ・ロールを持つデータベース・アカウント」レポート

「カタログ・ロールを持つデータベース・アカウント」レポートは、カタログ関連のロールが付与されたデータベース・アカウントおよびロールをすべて示します。

これらのロールは次のとおりです。

- DELETE_CATALOG_ROLE

- EXECUTE_CATALOG_ROLE
- RECOVERY_CATALOG_OWNER
- SELECT_CATALOG_ROLE

これらのカタログベースのロールは、非常に多くの強力な権限を保持します。これらを使用するDBAロールと同様に、慎重に付与する必要があります。

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.10 「AUDIT権限」レポート

「AUDIT権限」レポートは、AUDIT ANYまたはAUDIT SYSTEM権限を持つデータベース・アカウントおよびロールをすべて示します。

この権限を使用すると監査を無効にできるため、システムを侵害した侵入者の監査証跡レコードを削除するために使用される可能性があります。この権限を持つアカウントは、侵入者のターゲットになる恐れがあります。

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.5.11 「OSセキュリティ脆弱性に関する権限」レポート

「OSセキュリティ脆弱性に関する権限」レポートは、機密情報をオペレーティング・システムにエクスポートするための権限を持つデータベース・アカウントおよびロールを示します。

このレポートは、オペレーティング・システムに関連する重要な脆弱性を示す可能性があります。

親トピック: 強力なデータベース・アカウントおよびロールのレポート

26.6.6 初期化パラメータおよびプロファイルのレポート

初期化パラメータおよびプロファイルのレポートは、データベース・パラメータ、リソース・プロファイルおよびシステム制限を追跡します。

- 「セキュリティ関連のデータベース・パラメータ」レポート
  「セキュリティ関連のデータベース・パラメータ」レポートは、データベース・パラメータの設定が正しくない場合に、セキュリティの脆弱性の原因となる可能性があるこれらのパラメータを示します。
- 「リソース・プロファイル」レポート
  「リソース・プロファイル」レポートは、リソース使用量を無制限に許可している可能性があるリソース・プロファイルを示します。
- 「システム・リソース制限」レポート
  「システム・リソース制限」レポートを使用すると、データベースごとに現在のシステム・リソースの使用率を認識できます。

親トピック: Oracle Database Vaultの一般セキュリティ・レポート

26.6.6.1 「セキュリティ関連のデータベース・パラメータ」レポート

「セキュリティ関連のデータベース・パラメータ」レポートは、データベース・パラメータの設定が正しくない場合に、セキュリティの脆弱性の原因となる可能性があるこれらのパラメータを示します。

このレポートを使用すると、推奨される設定とデータベース・パラメータ値の現在の状態を比較できます。

親トピック: 初期化パラメータおよびプロファイルのレポート

26.6.6.2 「リソース・プロファイル」レポート

「リソース・プロファイル」レポートは、リソース使用量を無制限に許可している可能性があるリソース・プロファイルを示します。

リソース・プロファイルの例は、CPU_PER_SESSIONおよびIDLE_TIMEです。リソースの潜在的使用の抑制が必要なプロファイルは、見直す必要があります。

親トピック: 初期化パラメータおよびプロファイルのレポート

26.6.6.3 「システム・リソース制限」レポート

「システム・リソース制限」レポートを使用すると、データベースごとに現在のシステム・リソースの使用率を認識できます。

このレポートは、既存のアプリケーションの負荷の下で、これらのリソースのいずれかが限界に達しつつあるかどうかを判断するために役立ちます。短期間で大幅な増加を示すリソースは、DoS攻撃を示している可能性があります。リソースの上限を下げて、この先の状況を回避することが必要な場合があります。

親トピック: 初期化パラメータおよびプロファイルのレポート

26.6.7 データベース・アカウント・パスワードのレポート

データベース・アカウント・パスワードのレポートは、デフォルト・パスワード、およびデータベース・アカウントのアカウント・ステータスを追跡します。

- 「データベース・アカウントのデフォルト・パスワード」レポート
  「データベース・アカウントのデフォルト・パスワード」レポートは、デフォルト・パスワードを持つデータベース・アカウントを示します。
- 「データベース・アカウントのステータス」レポート
  「データベース・アカウントのステータス」レポートは、既存のデータベース・アカウントを示します。

親トピック: Oracle Database Vaultの一般セキュリティ・レポート

26.6.7.1 「データベース・アカウントのデフォルト・パスワード」レポート

「データベース・アカウントのデフォルト・パスワード」レポートは、デフォルト・パスワードを持つデータベース・アカウントを示します。

デフォルト・パスワードは、Oracle Databaseのインストール時に指定されます。

データベースを保護するために、このレポートに含まれるアカウントのパスワードをデフォルト以外の複雑なパスワードに変更する必要があります。

親トピック: データベース・アカウント・パスワードのレポート

26.6.7.2 「データベース・アカウントのステータス」レポート

「データベース・アカウントのステータス」レポートは、既存のデータベース・アカウントを示します。

このレポートにはアカウントごとにアカウント・ステータスが示されるため、ロックする必要があるアカウントを特定するために役立ちます。ロック日および有効日は、パスワード・エイジングの結果としてアカウントがロックされたかどうかの判断に役立つ情報となります。特殊なパスワードおよびリソース・セキュア・プロファイルが使用されている場合は、それらを使用していないアカウントを特定できます。体系立てて定義されたデフォルト表領域を使用していないアカウントも特定でき、アカウントの一時表領域を割り出すことができます。また、このレポートでは、外部パスワードを使用するアカウントも識別されます。

親トピック: データベース・アカウント・パスワードのレポート

26.6.8 セキュリティ監査レポート: コア・データベース監査レポート

コア・データベース監査レポートは、データベース監査証跡レコードを示します。

このレポートは非統合監査環境に適用されます。

コア・データベース監査レポートは、「Oracle Database Vaultの監査」で定義されている監査ポリシーに対する監査レコード、およびユーザーが定義した監査文に対して生成された監査レコードを返します。

このレポートには、データベース初期化パラメータAUDIT_TRAILがDBに設定されている場合(統合監査は無効になっている)に取得される監査レコードのみが表示されます。

関連項目:

AUDIT_TRAIL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: Oracle Database Vaultの一般セキュリティ・レポート

26.6.9 その他のセキュリティ脆弱性レポート

その他のセキュリティ脆弱性レポートは、Javaポリシーの付与またはオペレーティング・システムのディレクトリ・オブジェクトなど、アクティビティで発生する可能性のある脆弱性の問題を追跡します。

- 「Javaポリシーの付与」レポート
  「Javaポリシーの付与」レポートは、データベースに格納されるJavaポリシーの権限を示します。
- 「OSディレクトリ・オブジェクト」レポート
  「OSディレクトリ・オブジェクト」レポートは、データベース内のディレクトリ・オブジェクト、その権限、およびPUBLICで使用できるか否かを示します。
- 「動的SQLに依存するオブジェクト」レポート
  「動的SQLに依存するオブジェクト」レポートは、動的SQLを使用するオブジェクトをリストします。
- 「アンラップされたパッケージ本体」レポート
  「アンラップされたパッケージ本体」レポートは、ラップされていないPL/SQLパッケージ・プロシージャを示します。
- 「ユーザー名またはパスワード表」レポート
  「ユーザー名またはパスワード表」レポートは、ユーザー名とパスワード文字列を格納するデータベース内のアプリケーション表を識別します。
- 「表領域割当て制限」レポート
  「表領域割当て制限」レポートは、1つ以上の表領域に割当て制限があるすべてのデータベース・アカウントを示します。
- 「所有者でないオブジェクトのトリガー」レポート
  「所有者でないオブジェクトのトリガー」レポートは、所有者ではないトリガーを示します。

親トピック: Oracle Database Vaultの一般セキュリティ・レポート

26.6.9.1 「Javaポリシーの付与」レポート

「Javaポリシーの付与」レポートは、データベースに格納されるJavaポリシーの権限を示します。

このレポートは、最低限の権限の原則に対する違反の検出に役立ちます。必ずしも権限を必要としないPUBLICおよびその他のアカウントやロールに対するGRANT、READまたはWRITE権限を見つけます。データベースでJavaが必要ない場合は、PUBLICのJavaロード権限を無効にすることをお薦めします。

ノート:

「Java ポリシーの付与」レポートを実行するには、Oracle JVM(Oracle Database Vault に用意された Java 仮想マシン・オプション)をインストールする必要があります。

親トピック: その他のセキュリティ脆弱性レポート

26.6.9.2 「OSディレクトリ・オブジェクト」レポート

「OSディレクトリ・オブジェクト」レポートは、データベース内のディレクトリ・オブジェクト、その権限、およびPUBLICで使用できるか否かを示します。

ディレクトリ・オブジェクトは保護されたオペレーティング・システム(OS)に対してのみ存在し、データベース内でのディレクトリ・オブジェクトへのアクセスは保護する必要があります。リモート・データベース・セッションでデバイス上のすべてのファイルが参照できるようになるため、オペレーティング・システムのルート・ディレクトリ(/など)を任意のストレージ・デバイス上で使用しないでください。

親トピック: その他のセキュリティ脆弱性レポート

26.6.9.3 「動的SQLに依存するオブジェクト」レポート

「動的SQLに依存するオブジェクト」レポートは、動的SQLを使用するオブジェクトを示します。

パラメータ・チェックまたはバインド変数が使用されない場合、潜在的な侵入者がこのチャネルを使用する可能性が大きくなります。このレポートにより、動的SQLを使用しているユーザーが明らかになり、問題を探す範囲を狭めることができます。このようなオブジェクトは、SQLインジェクション攻撃のターゲットとなる可能性があり、この種の攻撃を回避するために保護する必要があります。動的SQLを使用するオブジェクトを特定したら、次のようにします。

- そのオブジェクトに対してクライアント・アプリケーション(Webアプリケーションなど)が保持する権限を確認します。
- そのオブジェクト用にPUBLICまたはより広範なアカウント・ベースに付与されたアクセス権を確認します。
- パラメータを確認します。
- 可能な場合は、バインド変数を使用します。

親トピック: その他のセキュリティ脆弱性レポート

26.6.9.4 アンラップされたPL/SQLパッケージ本体レポート

「アンラップされたパッケージ本体」レポートは、ラップされていないPL/SQLパッケージ・プロシージャを示します。

データ・ディクショナリまたはデータ・ディクショナリ・ビューで読み取ることができない程度にコードを不明瞭化するラップ・ユーティリティが用意されています。このユーティリティを使用すると、データを操作するソース・コードの読取りを不可能にすることで、侵入者がデータ保護を回避する能力を削ぐことができます。

親トピック: その他のセキュリティ脆弱性レポート

26.6.9.5 「ユーザーまたはパスワード表」レポート

「ユーザー名またはパスワード表」レポートは、ユーザー名とパスワード文字列を格納するデータベース内のアプリケーション表を識別します。

これらの表を調査して、情報が暗号化されているかどうかを判断する必要があります。(%USER%NAME%や%PASSWORD%などの列名を検索してください。)暗号化されていない場合は、これらの表を使用するコードおよびアプリケーションを変更して、データベース・セッションから見えないように保護してください。

親トピック: その他のセキュリティ脆弱性レポート

26.6.9.6 「表領域割当て制限」レポート

「表領域割当て制限」レポートは、1つ以上の表領域に割当て制限があるすべてのデータベース・アカウントを示します。

これらの表領域は、DoS攻撃の潜在的なターゲットになり得ます。

親トピック: その他のセキュリティ脆弱性レポート

26.6.9.7 「所有者でないオブジェクトのトリガー」レポート

「所有者でないオブジェクトのトリガー」レポートは、所有者ではないトリガーを示します。

これらは、トリガーが動作するデータベース・オブジェクトを所有するアカウントとは別のデータベース・アカウントが所有するトリガーです。

トリガーが信頼できるデータベース・アプリケーションに属していない場合、Oracle Label SecurityやVirtual Private Database(VPD)によって保護されている表などから機密データが盗まれ、保護されていない表に挿入されて、後で表示やエクスポートが行われる可能性があります。

親トピック: その他のセキュリティ脆弱性レポート

A Oracle Database Vaultの監査

ポリシー構成への変更など、Oracle Database Vaultのアクティビティを監査できます。

- Oracle Database Vaultの監査について
  Oracle Database Vaultのすべてのアクティビティは、Database Vault Administratorのアクティビティを含めて監査できます。
- Oracle Database Vault環境での統合監査証跡の保護
  デフォルトでは、統合監査証跡を含むAUDSYSスキーマはレルムで保護されません。
- Oracle Database Vault固有の監査イベント
  Oracle Database Vaultの監査イベントは、レルムで試行されたアクションが成功したかどうかなど、アクティビティを追跡します。
- Oracle Database Vault監査証跡のアーカイブおよびパージ
  統合監査に移行していない場合、Oracle Database Vault監査証跡を定期的にアーカイブおよびパージする必要があります。
- Oracle Database Vaultに作成されたOracle Database監査設定
  Oracle Database Vaultをインストールすると、データベース内に複数のAUDIT設定が作成されます。

A.1 Oracle Database Vaultでの監査について

Oracle Database Vaultのすべてのアクティビティは、Database Vault Administratorのアクティビティを含めて監査できます。

オプションで、レルム、ルール・セットおよびファクタに作成する個別のポリシーを監査できます。監査は、ユーザーのアクションの成功(ポリシーによってユーザーがタスクを実現できたかどうか)、またはユーザーのアクションの失敗(ポリシー違反したかどうか)のいずれかを示します。このようなアクションは監査ログに書き込まれます。この内容を確認するには、適切なデータ・ディクショナリ・ビューを問い合せるか、「Oracle Database Vaultレポート」に記載されているレポートを実行します。

Database Vaultへのすべての構成変更については監査が必須となっており、これには、権限が与えられていないユーザーによってDatabase Vaultポリシーの変更が試行されるアクションが含まれます。

新しいデータベースをインストールして、Oracle Database Vaultを使用するようにそのデータベースを構成する場合、デフォルトで混合モード環境(統合監査と事前移行済監査の混合)が使用されます。以前のリリースからアップグレードした場合、Database Vaultは、そのリリースで使用できた監査を使用します。

完全な統合監査環境に移行する前に、監査ポリシーを次のように作成できます。

- Database VaultのAPIの使用: DBMS_MACADM PL/SQLパッケージまたはEnterprise ManagerのDatabase Vaultページを使用します。この場合、監査レコードはDatabase Vault監査証跡に書き込まれます。この証跡は、DVSYS.AUDIT_TRAIL$表に格納されます。これらの監査レコードについて、DVSYS.DV$CONFIGURATION_AUDITビューとDVSYS.DV$ENFORCEMENT_AUDITビューに問い合せることができます。
- 統合監査ポリシーのSQL文の使用: これらの文は、CREATE AUDIT POLICY、ALTER AUDIT POLICY、DROP AUDIT POLICY、AUDITおよびNO AUDIT文です。これらは統合監査証跡に書き込まれます。この証跡は、UNIFIED_AUDIT_TRAIL、AUDSYS.DV$CONFIGURATION_AUDITおよびAUDSYS.DV$ENFORCEMENT_AUDITデータ・ディクショナリ・ビューで取得されます。

統合監査に移行する場合、Database VaultのAPIの監査機能は、無効となります。「Oracle Database Vault監査証跡のアーカイブおよびパージ」の説明に従い、これらの監査レコードをアーカイブおよびパージする必要があります。それ以降、統合監査ポリシーのPL/SQL文によってDatabase Vault監査ポリシーを管理できます。

記述した内容を除き、この章の残りの項では、Database Vault監査の非統合または混合モードの監査環境での動作について説明します。

関連項目:

- Oracle Database Vaultでの統合環境の動作と統合監査ポリシ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
- Oracle Database Vault監査証跡レコードの書式
- Database Vault統合監査に固有の次のデータ・ディクショナリ・ビュー。

  DVSYS.DV$CONFIGURATION_AUDITビュー

  DVSYS.DV$ENFORCEMENT_AUDITビュー
- 統合監査にデータベースを移行する場合は、『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: Oracle Database Vaultの監査

A.2 Oracle Database Vault環境での統合監査証跡の保護

デフォルトでは、統合監査証跡を含むAUDSYSスキーマはレルムで保護されません。

統合監査証跡をより適切に保護するために、次のことを実行することをお薦めします。

- AUDSYSスキーマの周囲に通常の(必須でない)レルムを作成し、認可されたユーザー(AUDIT_ADMINおよびAUDIT_VIEWERロールを付与されているユーザー)のみが統合監査証跡ビューを問い合せ、DBMS_AUDIT_MGMT PL/SQLパッケージを使用して監査証跡を管理できるようにします。このレルムは、SYSを含む多くの権限を持つユーザーが、そのレルムの認可リストに追加されるまでこれらのアクションを実行できないようにします。
- CREATE AUDIT POLICY、ALTER AUDIT POLICYおよびDROP AUDIT POLICY SQL文のコマンド・ルールを作成し、認可されたユーザーのみがこれらの文を実行できるようにします。

関連トピック

- レルムの作成
- コマンド・ルールの作成

親トピック: Oracle Database Vaultの監査

A.3 Oracle Database Vault固有の監査イベント

Oracle Database Vaultの監査イベントは、レルムで試行されたアクションが成功したかどうかなど、アクティビティを追跡します。

- Oracle Database Vaultポリシー監査イベント
  Oracle Database Vaultでは、監査イベントを使用して、構成アクティビティを追跡します。
- Oracle Database Vault監査証跡レコードの書式
  統合監査を使用しない場合、Oracle Database Vaultは、監査レコードをDVSYS.AUDIT_TRAIL$表に書き込みます。

親トピック: Oracle Database Vaultの監査

A.3.1 Oracle Database Vaultポリシーの監査イベント

Oracle Database Vaultでは、監査イベントを使用して構成アクティビティを追跡します。

これらのアクティビティは、次のとおりです。

- レルムの監査。レルムの作成時に設定した監査オプションに基づいて、正常終了したアクションおよび失敗したアクションの両方を監査できます。ただし、スキーマ所有者によって実行されたアクションは監査できません。
- ルール・セットの監査。ルール・セットの処理結果を監査します。正常終了した処理および失敗した処理の両方を監査できます。レルム認可は、ルール・セットを使用して管理できます。ルール・セットの処理結果を監査できます。ファクタ割当ておよびセキュア・アプリケーション・ロール監査は、ルール・セットを使用して管理できます。
- ファクタの監査。正常終了したファクタ処理および失敗したファクタ処理の両方を監査できます。失敗したファクタ処理の場合、「取得エラー」、「取得がNULL」、「検証エラー」、「検証がFalse」、「信頼レベルがNULL」、「信頼レベルがゼロ未満」のすべてまたはいずれかについて監査できます。
- Oracle Label Securityセッション初期化の失敗。Oracle Label Securityセッションの初期化に失敗したインスタンスを監査します。
- Oracle Label Securityセッション・ラベル・アップグレード試行の失敗。Oracle Label Securityコンポーネントによって最大セッション・ラベルを超えるラベルの設定が妨げられているセッションのインスタンスを監査します。

関連トピック

- ファクタの監査オプションの設定
- レルム認可について
- Oracle Database Vaultレポート

親トピック: Oracle Database Vault固有の監査イベント

A.3.2 Oracle Database Vault監査証跡レコードの書式

統合監査を使用しない場合、Oracle Database Vaultは、監査レコードを`DVSYS.AUDIT_TRAIL$`表に書き込みます。

これらの監査レコードはOracle Database Vault監査証跡の一部ではないため、データベースで監査がどのように有効になっても、Oracle Database Vaultが`DVSYS.AUDIT_TRAIL$`表から監査データをどのように収集するかには影響しません。実際には、Oracle Databaseで監査が無効になっている場合でも、Oracle Database Vaultの監査機能は`DVSYS.AUDIT_TRAIL$`表への書込みを続けます。

`DV_OWNER`、`DV_ADMIN`、`DV_SECANALYST`または`DV_MONITOR`ロールが付与されているユーザーは、`DVYS.AUDIT_TRAIL$`表に直接問い合せることができます。

表A-1では、監査証跡の書式を説明しています。`DVSYS.AUDIT_TRAIL$`表を使用するカスタム・レポートの作成を予定している場合、これらの書式を理解する必要があります。

表A-1 Oracle Database Vault監査証跡の書式

列	データ型	Null	説明
ID#	NUMBER	NOT NULL	監査レコードごとの数値識別子。

列	データ型	Null	説明
OS_USERNAME	VARCHAR2(255)	NULL	アクションが監査対象となったユーザーのオペレーティング・システムのログイン・ユーザー名。
USERNAME	VARCHAR2(30)	NULL	アクションが監査対象となったデータベース・ユーザーの名前。
USERHOST	VARCHAR2(128)	NULL	クライアント・コンピュータ名。
TERMINAL	VARCHAR2(255)	NULL	ユーザーの端末に対する識別子。
TIMESTAMP	DATE	NULL	監査証跡エントリの作成日時(ローカル・データベース・セッションのタイムゾーン)。
OWNER	VARCHAR2(30)	NULL	アクションの影響を受けるオブジェクトの作成者、常時DVSYS(DVSYS でオブジェクトが作成されるため)
OBJ_NAME	VARCHAR2(128)	NULL	アクションの影響を受けるオブジェクトの名前。想定値は次のとおりです。 ● ROLE$ ● REALM$ ● CODE$ ● FACTOR$
ACTION	NUMBER	NOT NULL	数値のアクション・タイプ・コード。アクション・タイプに対応する名前は、ACTION_NAME 列に示されます。予想される ACTION と ACTION_NAME の値のリストについては、表 24-3 を参照してください。
ACTION_NAME	VARCHAR2(128)	NULL	ACTION 列の数値コードに対応するアクション・タイプの名前
ACTION_OBJECT_ID	NUMBER	NULL	OBJ_NAME に指定された表のレコードの一意の識別子。レルムの場合、このフィールドには、「失敗時に監査」監査オプションのあるすべてのレルム ID のカンマ区切りの値のリストが含まれます。
ACTION_OBJECT_NAME	VARCHAR2(128)	NULL	OBJ_NAME に指定された表のレコードの一意の名前または固有のキー。レルムの場合、このフィールドには、

列	データ型	Null	説明
			「失敗時に監査」監査オプションのあるすべてのレルム名のカンマ区切りの値のリストが含まれます。
ACTION_COMMAND	VARCHAR2(4000)	NULL	実行された結果、監査イベントがトリガーされたコマンド・プロシージャの SQL テキスト。
AUDIT_OPTION	VARCHAR2(4000)	NULL	結果として監査イベントがトリガーされたレコードに指定されたすべての監査オプションのラベル。たとえば、失敗または NULL になったときに監査することになっているファクタ設定操作では、これら 2 つのオプションを指します。
RULE_SET_ID	NUMBER	NULL	実行された結果、監査イベントがトリガーされたルール・セットの一意の識別子。
RULE_SET_NAME	VARCHAR2(30)	NULL	実行された結果、監査イベントがトリガーされたルール・セットの一意の名前。
RULE_ID	NUMBER	NULL	使用されていません。
RULE_NAME	VARCHAR2(30)	NULL	使用されていません。
FACTOR_CONTEXT	VARCHAR2(4000)	NULL	監査イベントがトリガーされた時点での、現行セッションに対するすべてのファクタ識別子を含む XML 文書。
COMMENT_TEXT	VARCHAR2(4000)	NULL	監査対象となった文の詳細を示す、監査証跡エントリについてのテキスト・コメント。
SESSIONID	NUMBER	NOT NULL	Oracle セッションごとの数値識別子。
ENTRYID	NUMBER	NOT NULL	ID#列の値と同じ。
STATEMENTID	NUMBER	NOT NULL	起動された結果、監査イベントが生成された文の数値識別子。ほとんどの Oracle Database Vault イベントの場合、このパラメータは空です。
RETURNCODE	NUMBER	NOT NULL	アクションによって生成された Oracle エラー・コード。起動された結果、監査イベントが生成された文またはプロシージャに対するエラー・コード。ほとんどの Oracle

列	データ型	Null	説明
			Database Vault イベントの場合、このパラメータは空です。
EXTENDED_TIMESTAMP	TIMESTAMP(6)WITH TIME ZONE	NULL	UTC(協定世界時)タイムゾーンの、監査証跡エントリの作成時のタイムスタンプ(エントリに対するユーザー・ログインのタイムスタンプ)。
PROXY_SESSIONID	NUMBER	NULL	エンタープライズ・ユーザーがプロキシ機構を介してログインした場合の、プロキシ・セッションのシリアル番号
GLOBAL_UID	VARCHAR2(32)	NULL	ユーザーがエンタープライズ・ユーザーとしてログインした場合の、ユーザーのグローバル・ユーザー識別子。
INSTANCE_NUMBER	NUMBER	NULL	INSTANCE_NUMBER 初期化パラメータによって指定されるインスタンス番号
OS_PROCESS	VARCHAR2(16)	NULL	Oracle プロセスのオペレーティング・システム・プロセス識別子
CREATED_BY	VARCHAR2(30)	NULL	アクションが監査対象となったユーザーのデータベースのログイン・ユーザー名。
CREATE_DATE	DATE	NULL	SYSDATE の日付を基にした、アクションが発生した日付。
UPDATED_BY	VARCHAR2(30)	NULL	CREATED_BY 列の値と同じ。
UPDATE_DATE	DATE	NULL	UPDATED_BY 列の値と同じ。

親トピック: Oracle Database Vault固有の監査イベント

A.4 Oracle Database Vault監査証跡のアーカイブおよびパージ

統合監査に移行していない場合、Oracle Database Vault監査証跡を定期的にアーカイブおよびパージする必要があります。

- Oracle Database Vault監査証跡のアーカイブおよびパージ
  非統合監査環境では、DVSYS.AUDIT_TRAIL$表をダンプ・ファイルにエクスポートすることで、Oracle Database Vault監査証跡のアーカイブを作成できます。
- Oracle Database Vault監査証跡のアーカイブ
  SQL*PlusおよびOracle Data Pumpを使用すると、Oracle Database Vault監査証跡をアーカイブできます。

- Oracle Database Vault監査証跡のパージ
  SQL*PlusでOracle Database Vault監査証跡をパージできます。

親トピック: Oracle Database Vaultの監査

A.4.1 Oracle Database Vault監査証跡のアーカイブおよびパージについて

非統合監査環境では、DVSYS.AUDIT_TRAIL$表をダンプ・ファイルにエクスポートすることで、Oracle Database Vault監査証跡をアーカイブできます。

監査証跡は、大きくなりすぎないように定期的にアーカイブし、パージする必要があります。

統合監査に移行するように選択する場合、移行の完了後、このプロシージャを使用してDatabase Vault監査証跡レコードをアーカイブおよびパージします。統合監査がレコードの収集を開始すると、新しいレコードは、UNIFIED_AUDIT_TRAIL、AUDSYS.DV$CONFIGURATION_AUDITおよびAUDSYS.DV$ENFORCEMENT_AUDITデータ・ディクショナリ・ビューの表示に使用できるようになります。

親トピック: Oracle Database Vault監査証跡のアーカイブおよびパージ

A.4.2 Oracle Database Vault監査証跡のアーカイブ

SQL*PlusおよびOracle Data Pumpを使用すると、Oracle Database Vault監査証跡をアーカイブできます。

1. SYSDBA管理権限を持つユーザーSYSとして、データベース・インスタンスにログインします。

   ```
   sqlplus sys as sysdba
   Enter password: password
   ```

2. アーカイブを実行するユーザーに適切な権限があることを確認します。

   たとえば:

   ```
   GRANT CREATE ANY DIRECTORY, EXP_FULL_DATABASE, UNLIMITED TABLESPACE TO psmith;
   ```

3. DV_OWNERまたはDV_AUDIT_CLEANUPロールを付与されているユーザーとして接続します。

   たとえば:

   ```
   connect ebrown
   Enter password: password
   ```

4. Oracle Database Vault監査証跡を、該当するスキーマの新しい表にアーカイブします。

   たとえば:

   ```
   CREATE TABLE psmith.dv_audit_trail nologging ¥
   AS SELECT * FROM DVSYS.AUDIT_TRAIL$;
   ```

5. スキーマがレルムによってすでに保護されている場合、担当者またはエクスポート操作を実行するユーザーに、Database Vault環境でOracle Data Pumpを使用する適切な認可が付与されていることを確認します。

   たとえば、ユーザーpsmithがそのスキーマに対してData Pump操作を実行できるように、このユーザーを認可するとします。

   ```
   EXEC DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('PSMITH', 'PSMITH');
   ```

6. Data Pumpユーザーとして接続します。

たとえば:

```
CONNECT psmith
Enter password: password
```

7. Database Vault監査証跡用のディレクトリを作成します。

```
CREATE DIRECTORY dv_audit_dir AS 'dv_audit_trail_directory';
```

8. SQL*Plusを終了します。

```
EXIT
```

9. Data Pumpを使用して、Database Vault監査証跡を、作成したディレクトリ・オブジェクトにエクスポートします。

```
expdp psmith directory=dv_audit_dir tables=psmith.dv_audit_trail ¥
dumpfile=dv_audit.dmp log=dv_audit_exp.log
```

10. DV_OWNERロールを付与されているユーザーとしてSQL*Plusに接続します。

```
sqlplus ebrown
Enter password: password
```

11. このようにしなかった場合、Database Vault監査証跡を含むスキーマの周辺にレルムを作成します。

    a. レルムを作成します。たとえば:

```
BEGIN
 DBMS_MACADM.CREATE_REALM(
realm_name    => 'DV Audit Trail Realm',
description   => 'Realm to protect the DV audit trail',
enabled       => DBMS_MACUTL.G_YES,
audit_options => DBMS_MACUTL.G_REALM_AUDIT_FAIL +
DBMS_MACUTL.G_REALM_AUDIT_SUCCESS,
realm_type    => 1);
END;
/
```

    b. 監査証跡を含むスキーマをこのレルムに追加します。たとえば:

```
BEGIN
 DBMS_MACADM.ADD_OBJECT_TO_REALM(
realm_name   => 'DV Audit Trail Realm',
object_owner => 'psmith',
object_name  => '%',
object_type  => '%');
END;
/
```

    c. このレルムに対して信頼されるユーザーを認可します。

```
BEGIN
 DBMS_MACADM.ADD_AUTH_TO_REALM(
realm_name  => 'DV Audit Trail Realm',
grantee     => 'PSMITH',
auth_options => DBMS_MACUTL.G_REALM_AUTH_OWNER);
END;
/
```

関連項目:

- Database Vault環境でのユーザーへのOracle Data Pump権限の付与の詳細は、「Oracle Database

VaultでのOracle Data Pumpの使用」を参照してください

- CREATE DIRECTORY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
- Oracle Data Pumpのexpdpユーティリティの詳細は、『Oracle Databaseユーティリティ』を参照してください
- DBMS_MACADMのレルム関連プロシージャの詳細は、「Oracle Database VaultレルムのAPI」を参照してください

親トピック: Oracle Database Vault監査証跡のアーカイブおよびパージ

A.4.3 Oracle Database Vault監査証跡のパージ

SQL*PlusでOracle Database Vault監査証跡をパージできます。

1. DV_OWNERロールまたはDV_AUDIT_CLEANUPロールを付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus psmith
   Enter password: password
   ```

   DV_OWNERロールとDV_AUDIT_CLEANUPロールでは、その権限受領者はDVSYS.AUDIT_TRAIL$システム表を切り捨てることができない点に注意してください。

   ユーザーに付与されているロールを検索するには、DBA_ROLE_PRIVSデータ・ディクショナリ・ビューを問い合せることができます。

2. Database Vault監査証跡をパージします。

   ```
   DELETE FROM DVSYS.AUDIT_TRAIL$;
   ```

関連トピック

- DV_AUDIT_CLEANUP監査証跡クリーンアップ・ロール

親トピック: Oracle Database Vault監査証跡のアーカイブおよびパージ

A.5 Oracle Database Vault用に作成されるOracle Database監査設定

Oracle Database Vaultをインストールすると、データベース内に複数のAUDIT設定が作成されます。

非統合監査環境でこれらの監査設定を実施にするには、このデータベースで監査を有効にする必要があります。監査が有効かどうかは、SHOW PARAMETERコマンドを使用して、AUDIT_TRAIL初期化パラメータの値を検索することで確認できます。デフォルトでは、監査はOracle Databaseで有効になっています。

表A-2に、Oracle Database Vaultによりデータベースに追加されるAUDITの設定を示します。

表A-2 Oracle Database VaultによりOracle Databaseに追加される監査ポリシーの設定

監査設定タイプ	監査対象文(特に断りのない場合はBY ACCESSかつ成功時または失敗時)
DVSYS/DVF のユーザー監査設定	ADMINISTER DATABASE TRIGGER

監査設定タイプ	監査対象文(特に断りのない場合はBY ACCESSかつ成功時または失敗時)
LBACSYS のユーザー監査設定 これらのアカウントの詳細は、表 13-1 を参照してください。 また、DVSYS スキーマおよび DVF スキーマの詳細は、次の項を参照してください。 ● DVSYS スキーマ ● DVF スキーマ	ALTER オブジェクト AUDIT SYSTEM BECOME USER CLUSTER COMMENT CONTEXT CREATE オブジェクト DATABASE LINK DEBUG DIRECTORY DROP オブジェクト EXECUTE LIBRARY (WHENEVER NOT SUCCESSFUL) EXECUTE PROCEDURE (WHENEVER NOT SUCCESSFUL) EXEMPT ACCESS POLICY EXPORT FULL DATABASE GRANT オブジェクト IMPORT FULL DATABASE INDEX MANAGE SCHEDULER MANAGE TABLESPACE MATERIALIZED VIEW(マテリアライズド・ビューへのアクセスと作成の両方を監査) SELECT SEQUENCE (WHENEVER NOT SUCCESSFUL) SELECT TABLE (WHENEVER NOT SUCCESSFUL)
DVF のオブジェクト監査設定	AUDIT PACKAGE/PROCEDURE/FUNCTION/SEQUENCE/TABLE

監査設定タイプ	監査対象文(特に断りのない場合はBY ACCESSかつ成功時または失敗時)
	COMMENT TABLE/VIEW DELETE TABLE/VIEW EXECUTE PACKAGE/PROCEDURE/FUNCTION (WHENEVER NOT SUCCESSFUL) GRANT PACKAGE/PROCEDURE/FUNCTION/SEQUENCE/TABLE RENAME PACKAGE/PROCEDURE/FUNCTION/SEQUENCE/VIEW/TABLE SELECT SEQUENCE/TABLE/VIEW (WHENEVER NOT SUCCESSFUL)
DVSYS のオブジェクト監査設定 LBACSYS のオブジェクト監査設定	AUDIT PACKAGE/PROCEDURE/FUNCTION/SEQUENCE/TABLE COMMENT TABLE/VIEW DELETE TABLE/VIEW EXECUTE PACKAGE/PROCEDURE/FUNCTION (WHENEVER NOT SUCCESSFUL) GRANT PACKAGE/PROCEDURE/FUNCTION/SEQUENCE/TABLE INSERT TABLE/VIEW RENAME PACKAGE/PROCEDURE/FUNCTION/SEQUENCE/VIEW/TABLE SELECT SEQUENCE/TABLE/VIEW (WHENEVER NOT SUCCESSFUL) UPDATE TABLE/VIEW

親トピック: Oracle Database Vaultの監査

B Oracle Database Vaultの無効化および有効化

Oracle Databaseのオプション製品または機能のインストールなどのアクティビティのために、定期的にOracle Database Vaultを無効化してから再有効化する必要があります。

- Oracle Database Vaultを無効にする必要がある場合
  アップグレード・タスクを実行する際、または誤った構成を修正する際には、Oracle Database Vaultを無効にする必要があります。
- ステップ1: Oracle Database Vaultの無効化
  Oracle Database Vaultを無効化した後も、Database Vaultの実行に必要なOracle Label Securityは有効なままです。
- ステップ2: 必要なタスクの実行
  この段階では、Oracle Database Vaultは無効になっており、必要な作業を実行できます。
- ステップ3: Oracle Database Vaultの有効化
  SQL*PlusからOracle Database VaultおよびOracle Label Securityを有効にできます。

B.1 Oracle Database Vaultを無効にする必要がある場合

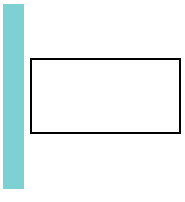
アップグレード・タスクを実行する際、または誤った構成を修正する際には、Oracle Database Vaultを無効にする必要があります。

修正タスクが終了したら、Oracle Database Vaultを再度有効にできます。

Oracle Database Vaultの有効化ステータスおよび構成ステータスを確認するには、Database Vaultが構成および有効化されていることの確認を参照してください。

次の場合に、Oracle Database Vaultを無効にする必要があります。

- Oracle Databaseのオプションの製品または機能(Oracle Spatialなど)のいずれかを、Database Configuration Assistant (DBCA)を使用してインストールする必要がある場合。
- Oracle Database Vaultの構成時および有効化時にDV_OWNERおよびDV_ACCTMGRのバックアップ・アカウントを構成しておらず、これらのアカウントが不注意でロックされたか、それらのパスワードを忘れた場合。使用サイトにDV_OWNERユーザーが1人しかおらず、このユーザーが自身のパスワードを失くした場合、担当者はOracle Database Vaultを無効にすることができません。ただし、サイトで唯一のDV_ACCTMGRユーザーがパスワードを失くした場合は、Database Vaultを無効にできます。ベスト・プラクティスとして、DV_OWNERロールとDV_ACCTMGRロールを、新しいまたは既存のユーザー・アカウントに付与し、Database Vaultをバックアップ・アカウントとして構成して有効化したときに作成したDatabase Vault所有者およびアカウント・マネージャのアカウントを使用します。(以後この問題を回避するためのガイドラインについては、「バックアップOracle Database Vaultアカウント」を参照)。
- Oracle Database Configuration Assistant (DBCA)を使用してOracle Internet Directory (OID)を構成する場合。



ノート:

- Oracle Database Vault を無効にすると、構成中に既存のユーザーおよびロールから取り消された権限が有効のままになることに注意してください。取り消される権限のリストは、「既存のユーザーお

よびロールから取り消される権限」を参照してください。

- Oracle Database Vault を無効にする場合、一部の Database Vault 機能は引き続き使用できます。
- Oracle Database Vault の削除はサポートされていません。

親トピック: Oracle Database Vaultの無効化および有効化

B.2 ステップ1: Oracle Database Vaultの無効化

Oracle Database Vaultを無効化した後も、Database Vaultの実行に必要なOracle Label Securityは有効なままです。

1. SQL*Plusでは、Oracle Database所有者(DV_OWNER)アカウントとしてログインしてから、Oracle Database Vaultを無効にします。

```
sqlplus psmith
Enter password: password
EXEC DBMS_MACADM.DISABLE_DV;
```

2. マルチテナント環境で、適切なプラガブル・データベース(PDB)に接続します。

   たとえば:

```
CONNECT psmith@hrpdb
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。

3. データベースを再起動します。

```
CONNECT SYS AS SYSOPER -- Or, CONNECT SYS@hrpdb AS SYSOPER
Enter password: password
SHUTDOWN IMMEDIATE
STARTUP
```

4. Oracle RACインストールの場合は、データベースがインストールされているノードごとに、これらのステップを繰り返します。

親トピック: Oracle Database Vaultの無効化および有効化

B.3 ステップ2: 必要なタスクの実行

この段階では、Oracle Database Vaultは無効になっており、必要な作業を実行できます。

次のようなアクティビティを実行できます。

- Oracle Database VaultのPL/SQLパッケージおよびファンクションを使用します。たとえば、ログインまたはCONNECTルール・セット・エラーを修正するには、DBMS_MACADM PL/SQLパッケージまたはEnterprise Manager Cloud ControlのOracle Database Vaultページを使用します。CONNECTコマンド・ルールでは、DV_OWNERまたはDV_ADMINロールを持つユーザーのデータベースへの接続は阻止できません。このため、Database Vault管理者はDatabase Vaultを無効化することなく、間違って構成された保護を修正できます。
- パスワードの作成または変更、あるいはアカウントのロックおよびロック解除などのタスクを実行するために、SYSTEMアカ

ウントまたはSYSアカウントを使用します。標準のデータベースおよび管理ユーザー・アカウントの変更に加えて、DV_ADMINまたはDV_ACCTMGRロールを付与されているユーザーなど、任意のOracle Database Vault固有のアカウントのパスワードおよびロック・ステータスを変更できます。(以後この問題を回避するためのガイドラインについては、「登録中に作成されるOracle Database Vaultアカウント」でヒントを参照)。

- インストールまたはセキュリティ保護を無効にする必要のあるその他のタスクを実行します。

親トピック: Oracle Database Vaultの無効化および有効化

B.4 ステップ3: Oracle Database Vaultの有効化

SQL*PlusからOracle Database VaultおよびOracle Label Securityを有効にできます。

1. SQL*Plusで、Oracle Database所有者(DV_OWNER)アカウントとして接続してから、Database Vaultを有効にします。

   非マルチテナント環境またはPDBからDatabase Vaultを有効にしている場合は、次のようにします。

   ```
   CONNECT psmith -- Or, CONNECT psmith@hrpdb for a PDB
   Enter password: password
   EXEC DBMS_MACADM.ENABLE_DV;
   ```

   CDBルートからDatabase Vaultを有効にしている場合。たとえば:

   ```
   CONNECT c##sec_admin_owen
   Enter password: password
   ```

   次の設定のどちらかを選択します。

   ```
   EXEC DBMS_MACADM.ENABLE_DV (strict_mode => 'n');
    -- For regular mode
   EXEC DBMS_MACADM.ENABLE_DV (strict_mode => 'y');
    -- For strict mode
   ```

2. Oracle Label Securityが有効がどうかを確認します。

   ```
   SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Oracle Label Security';
   ```

   Database Vaultを使用する前に、Oracle Label Securityを有効にする必要があります。有効でない場合、この問合せからFALSEが返されます。

3. Oracle Label Securityが有効でない場合は、有効にしてください。

   ```
   CONNECT SYS AS SYSDBA -- Or, CONNECT SYS@hrpdb AS SYSDBA
   Enter password: password
   EXEC LBACSYS.CONFIGURE_OLS;
   EXEC LBACSYS.OLS_ENFORCEMENT.ENABLE_OLS;
   ```

4. データベースを再起動します。

   ```
   CONNECT SYS AS SYSOPER -- Or, CONNECT SYS@hrpdb AS SYSOPER
   Enter password: password
   SHUTDOWN IMMEDIATE
   STARTUP
   ```

5. Oracle RACインストールの場合は、データベースがインストールされているノードごとに、これらのステップを繰り返します。

親トピック: Oracle Database Vaultの無効化および有効化

C Oracle Database Vaultのインストール後の手順

Oracle Database Vaultを構成および有効化した後は、言語の追加やOracle Database Vaultのアンインストールおよび再インストールなどの特別なタスクを実行できます。

- Oracle Database Vaultへの言語の追加
  デフォルトでは、Oracle Database Vaultによって英語表のみがロードされます。
- Oracle Database Vaultのアンインストール
  単一インスタンスとOracle RACの両方のインストールの場合、Oracle DatabaseインストールからOracle Database Vaultを削除できます。
- Oracle Database Vaultの再インストール
  Database Configuration Assistantを使用してOracle Database Vaultを再インストールしてから、Database Vaultを構成して有効化できます。

関連トピック

- スタンドアロンのOracle DatabaseをPDBに変換してCDBにプラグイン

C.1 Oracle Database Vaultへの言語の追加

デフォルトでは、Oracle Database Vaultによって英語表のみがロードされます。

言語を追加するには、追加する新しい言語ごとにDBMS_MACADM.ADD_NLS_DATAプロシージャを実行します。複数の言語をDatabase Vaultに追加できます。

1. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして、データベース・インスタンスにログインします。
2. 次のプロシージャを実行します。

   ```
   EXEC DBMS_MACADM.ADD_NLS_DATA('language');
   ```

   language設定は、大文字と小文字のいずれを使用しても指定できます。たとえば:

   ```
   EXEC DBMS_MACADM.ADD_NLS_DATA('french');
   EXEC DBMS_MACADM.ADD_NLS_DATA('JAPANESE');
   ```

   languageを、サポートされている次のいずれかの言語で置き換えます。

   - ENGLISH
   - GERMAN
   - SPANISH
   - FRENCH
   - ITALIAN
   - JAPANESE
   - KOREAN
   - BRAZILIAN PORTUGUESE
   - SIMPLIFIED CHINESE
   - TRADITIONAL CHINESE

親トピック: Oracle Database Vaultのインストール後の手順

C.2 Oracle Database Vaultのアンインストール

単一インスタンスとOracle RACの両方のインストールの場合、Oracle DatabaseインストールからOracle Database Vaultを削除できます。

ただし、マルチテナント環境のデータベースからDatabase Vaultをアンインストールすることはできません。この手順は、従来の非CDB Oracle Database環境にのみ適用されます。

削除プロセスによって、初期化パラメータ設定(インストール・プロセスで変更された設定であっても)やOracle Label Securityが影響を受けることはありません。

1. SYSDBA管理権限を持つユーザーSYSとして、またはALTER SYSTEMシステム権限を持つユーザーとしてデータベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus psmith -- Or, sqlplus psmith@hrpdb for a pluggable database (PDB)
   Enter password: password
   ```

2. リサイクルビンが無効になっていることを確認します。

   ```
   SHOW PARAMETER RECYCLEBIN
   ```

3. リサイクルビンがオンの場合は、次のいずれかの文を使用して無効にします。

   ```
   ALTER SYSTEM SET RECYCLEBIN = OFF;
   ALTER SESSION SET recyclebin = OFF SCOPE = SPFILE;
   ```

4. DV_OWNERまたはDV_ADMINロールを付与されているユーザーとして接続します。

   たとえば:

   ```
   CONNECT sec_admin_owen -- Or, CONNECT sec_admin_owen@hrpdb
   Enter password: password
   ```

5. 次のプロシージャを実行してOracle Database Vaultを無効にします。

   ```
   EXEC DBMS_MACADM.DISABLE_DV;
   ```

6. SYSOPER権限を使用してSYSとして接続し、データベースを再起動します。

   たとえば:

   ```
   CONNECT SYS AS SYSOPER -- Or, CONNECT SYS@hrpdb AS SYSOPER
   Enter password: password
   SHUTDOWN IMMEDIATE
   STARTUP
   ```

   Oracle RACインストールの場合は、次のように各データベース・インスタンスをシャットダウンしてから再起動します。

   ```
   $ srvctl stop database -db db_name
   $ srvctl start database -db db_name
   ```

7. dvremov.sqlスクリプトを実行してOracle Database Vaultを削除します。

   たとえば:

   ```
   $ORACLE_HOME/rdbms/admin/dvremov.sql
   ```

8. 必要に応じて、SQL*PlusでSYSDBA管理権限を持つユーザーSYSとして、DV_OWNERロールを付与されたユーザー

からDBMS_RLS PL/SQLパッケージのEXECUTE権限を手動で取り消します。
Oracle Database Vaultを構成する場合、DV_OWNERユーザーに付与される権限の1つがこの権限です。ただし、Oracle Database Vaultを削除しても、DV_OWNERユーザーにはこの権限があります。必要に応じて、取り消すことができます。

```
REVOKE EXECUTE ON DBMS_RLS FROM dbv_owner_backup;
```

その後、SQL*Plusにログインして次の文を入力することで、Oracle Database Vaultが本当にアンインストールされたことを二重にチェックできます。

```
SELECT * FROM V$OPTION WHERE PARAMETER = 'Oracle Database Vault';
```

Oracle Database Vaultがアンインストールされている場合、次の出力結果が表示されます。

```
PARAMETER                      VALUE
------------------------------ -----------------------
Oracle Database Vault          FALSE
```

親トピック: Oracle Database Vaultのインストール後の手順

C.3 Oracle Database Vaultの再インストール

Database Configuration Assistantを使用してOracle Database Vaultを再インストールしてから、Database Vaultを構成して有効化できます。

1. SYSDBA管理権限を持つユーザーSYSとして、データベース・インスタンスにログインします。

   ```
   sqlplus sys as sysdba -- Or, sqlplus sys@hrpdb as sysdba
   Enter password: password
   ```

2. Database Configuration Assistant(DBCA)を開始します。
   - UNIX: シェル・ウィンドウで次のコマンドを入力します。

     ```
     dbca
     ```

   - Windows: 次のいずれかの方法を使用して、WindowsでDBCAを起動します。
     - 「スタート」をクリックし、「プログラム」(または「すべてのプログラム」)、「Oracle - HOME_NAME」、「Configuration and Migration Tools」、および「Database Configuration Assistant」の順に選択します。
     - コマンド・プロンプトで次のコマンドを入力します。

       ```
       dbca
       ```

3. DBCAを使用し、新規または既存のデータベースに対してDatabase Vaultを構成します。
4. Oracle Database Vaultを構成して有効化します。

関連トピック

- Database Vaultユーザーの登録
- マルチテナント環境におけるOracle DatabaseでのOracle Database Vaultの構成および有効化

親トピック: Oracle Database Vaultのインストール後の手順

D Oracle Database Vaultセキュリティ・ガイドライン

すべてのOracle Database製品と同様に、Oracle Database Vaultインストールを適切に保護するためにセキュリティ・ガイドラインに従う必要があります。

- 職務分離のガイドライン
  Oracle Database Vaultは、職務分離のガイドラインを容易に実現できるように設計されています。
- Oracle Database管理アカウントの管理
  Oracleでは、SYSTEMなどの管理アカウント、またはSYSDBA管理権限を持つユーザーのセキュリティを管理するためのガイドラインを提供しています。
- Oracle Database Vaultによって信頼されるアカウントおよびロール
  Oracle Database Vaultは、データベース内の権限を与えられた多くのユーザーおよびロールからのアプリケーション・データへのアクセスを制限します。
- 信頼できる人物に制限する必要のあるアカウントおよびロール
  強力なアカウントおよびロールは、信頼できる人物に制限する必要があります。
- Oracle Database Vaultを本番環境で使用するためのガイドライン
  Oracle Database Vaultを本番環境で実行する際は、特定のガイドラインに従う必要があります。
- セキュアな構成のガイドライン
  PL/SQLパッケージ、権限およびリサイクルビンのセキュリティの考慮事項を認識する必要があります。

D.1 職務分離のガイドライン

Oracle Database Vaultは、職務分離のガイドラインを容易に実現できるように設計されています。

- Oracle Database Vaultによる職務分離の処理
  職務分離とは、各ユーザーの権限をそのユーザーが担当するタスクのみに制限し、それ以外のタスクの権限は付与しないことを意味します。
- Oracle Database Vault環境でのタスクの分離
  Oracle Database Vaultでは、いくつかの主な職責が定義されます。
- Oracle Database Vaultの職務分離マトリクス
  職務分離を適切に行うためには、環境で基本的な管理タスクを実行するユーザーおよびその管理タスクの内容を理解する必要があります。
- データベース・ユーザーのタスクの識別および文書化
  組織で必要となる次のタスク範囲を文書化する必要があります。

親トピック: Oracle Database Vaultセキュリティ・ガイドライン

D.1.1 Oracle Database Vaultによる職務分離の処理

職務分離は、各ユーザーの権限をそのユーザーが担当するタスクのみに制限し、それ以外のタスクの権限は付与しません。

多くの権限を1人のユーザーに付与するのではなく、特定の権限カテゴリを特定のユーザーに割り当てます。簡単に言えば、組織で求められる各タスクのアカウンタビリティが職務分離によって構築されます。

職務分離は、過去10年で非常に重要視されるようになりました。多くの組織にとって、職務分離は今後も展開し続ける新しい概念です。データベースの統合、法令順守およびアウトソーシングは、職務分離を推し進める要因の一部にすぎません。
Oracle Database Vaultの職務分離では、セキュリティ関連の管理を日常的なDBA操作と分離することによって、セキュリ

ティが強化されます。Database Vaultの職務分離は、現在および将来のビジネス要件に容易に対処できるようにカスタマイズが可能です。特に小規模組織では、限られたリソースでセキュリティ・プロファイルを向上させるために、柔軟性が必要となります。

親トピック: 職務分離のガイドライン

D.1.2 Oracle Database Vault環境でのタスクの分離

Oracle Database Vaultでは、いくつかの主な職責が定義されます。

これらの職責は次のとおりです。

- アカウント管理。アカウント管理では、ユーザー・アカウントの作成、変更、および削除を行います。DV_ACCTMGRロールがこれらの権限を提供します。日常用プライマリDV_ACCTMGRユーザーおよびバックアップDV_ACCTMGRユーザーは、Oracle Database Vault登録プロセスの間に作成されます。安全対策として、プライマリDV_ACCTMGRアカウント所有者が自分のパスワードを忘れた場合や企業を退職した場合に備えて、バックアップ・アカウントを保持し続けます。
- セキュリティ管理。セキュリティ管理では、基本的なセキュリティ・タスクを行います。たとえば、レルムやコマンド・ルールの作成、データベース・ユーザーのアクセスに対するセキュリティ・ポリシーの設定、実行が許可されたジョブに対するデータベース・ユーザーの認可などがあります。セキュリティ管理者は、セキュリティ監査レポートの実行も行います。DV_OWNERおよびDV_ADMINロールにこれらの権限が含まれています。日常用プライマリDV_OWNERユーザーおよびバックアップDV_OWNERユーザーは、Oracle Database Vault登録プロセスの間に作成されます。

  重要:

  安全対策として、プライマリ DV_OWNER アカウント所有者が自分のパスワードを忘れた場合や企業を退職した場合に備えて、バックアップ・ユーザー・アカウントを保持し続ける必要があります。DV_OWNER ロールを付与されているすべてのユーザー・アカウントへのアクセスを失わないことも重要です。DV_OWNER ロールを持つアカウントへのアクセスを失った場合(パスワードの紛失やスタッフの離職など)、DV_OWNER ロールをリカバリする方法はありません。DV_OWNER ロールへのアクセスを失った場合、Database Vault の制御を変更したり、Database Vault を無効にすることはできません。この問題に対処するには、データベースが Database Vault 所有者アカウントを所有している最後の既知のポイントまでデータベースをリカバリします。

  オプションで、アカウント管理とセキュリティ管理の職責を統合することもできます。
- データベース管理。データベース管理とは、データベース・システムの管理のことで、ビジネス・データへのアクセスのことではありません。次のような操作があります。
  - バックアップ操作では、事前定義されたツールを使用してバックアップを実行するための事前定義時刻が必要です。
  - チューニングおよび監視操作では、継続したパフォーマンス監視および分析が必要です。
  - パッチ適用操作では、パッチが適用されている間のみの一時アクセスが必要です。

  職務分離との関連においてデータベース管理アカウントを確認することをお薦めします。様々なデータベース管理者が、様々な権限およびロールを必要とする様々な職務を担当できます。同様に、経験豊かなデータベース管理者ほど、より多くのロールおよび権限がある可能性があります。ユーザーにデフォルトのDBAロールを付与するかわりに、組織での特定の地位および勤続年数にあわせてデータベース管理ロールを調整することを検討してください。名前付きアカウントの

みを日常的な活動のために使用することが重要です。SYSなどのアカウント、およびSYSDBA管理権限を使用するアカウントは、特権アカウント管理(PAM)システムで管理され、使用時にチェック(および監査)される必要があります。バックアップOracle Database Vault所有者およびアカウント管理アカウントをPAMシステムで管理する必要もあります。オペレーティング・システム内では、rootおよびoracleアカウントは、強力な権限があるため、チェックアウト・システムによってのみ使用できるようにする必要があります。

データベース・アカウント管理とデータベース・セキュリティ管理のための個別アカウントと、バックアップ操作のためのその他の名前付きアカウントが必要です。監査者は、各職責のための個別のデータベース・アカウントをチェックし、各アカウントのアクションを追跡できます。特定タスクに割り当てられているユーザーの数は、あまり重要ではありません。Oracle Database Vault監査イベントは保護されており、Database Vaultレポートにはすべての違反未遂が表示されます。

関連項目:

- Oracle Database Vaultロールがどのように職務分離を提供するかの詳細は、Oracle Database Vaultロールを参照してください
- デフォルトのOracle Database Vaultアカウントをすべて示すリストは、登録中に作成されるOracle Database Vaultアカウントを参照してください
- バックアップ・アカウントの重要性の詳細は、バックアップOracle Database Vaultアカウントを参照してください

親トピック: 職務分離のガイドライン

D.1.3 Oracle Database Vaultの職務分離マトリクス

職務分離を適用するには、環境で基本的な管理タスクを実行するユーザーおよびその管理タスクの内容を理解する必要があります。

1人のデータベース管理者が新しいデータベース・アカウントのプロビジョニングとアプリケーションのパッチ適用の両方の管理を担当する場合でも、これらの各タスクについて文書化および計画することが重要となります。これらのタイプのタスクに別々の管理アカウントを使用すると、アカウンタビリティが向上し、悪質なユーザーによって単一のアカウントが侵害された場合に関連するリスクが軽減されます。中規模から大規模の組織では、データベース管理者は通常、一般的な管理タスクを実行する必要はありますが、アプリケーションで管理されるビジネス・データにアクセスする必要はありません。職務分離マトリクスを作成すると、Database Vaultデプロイメントを計画する際に役立ちます。必要に応じて、追加タスクおよび関連するユーザーをリストに含めることができます。この情報は、組織の全体的なエンタープライズ・セキュリティ・ドキュメントの一部となります。

表D-1は、職務分離マトリクスの例を示しています。

表D-1 職務分離マトリクスの例

		データベース管理					
ユーザー、プロセスまたはアプリケーション	**アカウント作成**	**SYSDBA**	**バックアップ**	**チューニング**	**パッチ適用**	**監視**	**セキュリティ管理者**
JSMITH	あり	不可	不可	不可	不可	不可	不可
SHARDY	不可	不可	不可	不可	不可	不可	あり

		データベース管理					
ユーザー、プロセスまたはアプリケーション	**アカウント作成**	**SYSDBA**	**バックアップ**	**チューニング**	**パッチ適用**	**監視**	**セキュリティ管理者**
PKESTNER	不可	不可	あり	不可	不可	不可	不可
RTYLER	不可	不可	不可	不可	あり	不可	不可
SANDERSON	不可	不可	不可	あり	不可	あり	不可
SYSTEM	不可	不可	不可	不可	可(EBSのパッチ適用の場合)	不可	不可
RMAN	不可	あり	あり	不可	不可	不可	不可

システム管理タスクで、特定のツールおよびプログラムからデータへの一時的なアクセスが必要なこともあります。この場合、Oracle Database Vaultルールおよびルール・セットに、この一時アクセスまたは緊急アクセスに対するプロビジョニングを構築します。

親トピック: 職務分離のガイドライン

D.1.4 データベース・ユーザーのタスクの識別および文書化

組織で必要となる次のタスク範囲を文書化する必要があります。

これらの範囲は次のとおりです。

- 各管理ユーザーの職責。
- ユーザーが必要とするアクセス権の種類。たとえば、アプリケーション所有者はデータ・アクセス、開発者は開発インスタンスに対するアクセスのみを必要とします。
- ビジネス・データにアクセスせずにシステムを管理する必要のあるユーザー(バックアップ、パッチ適用、チューニングおよび監視操作を実行するユーザーなど)。
- 各タスク・カテゴリの職務(バックアップの必要なファイル、パッチ適用の必要なアプリケーション、監視の正確な対象など)。これらの各タスクの代替ユーザー・アカウントも含まれます。
- 保護が必要なデータベースとアプリケーション。これには、Oracle Applications、パートナ・アプリケーションおよびカスタム・アプリケーションが含まれます。
- ビジネス・データベースへのアクセスを認可する必要のあるユーザー。次のユーザーが含まれます。
    - 中間層プロセスを利用するアプリケーション所有者
    - アプリケーション・インタフェースを利用するビジネス・ユーザー
- セキュリティ侵害の対処方法など、緊急時のWhat-Ifシナリオ。

- 本番環境でのレポート作成。次の内容が含まれます。
    - レポートの実行者
    - 実行が必要なレポート
    - 各レポートの実行頻度
    - 各レポートのコピーの受取りを必要とするユーザー
- 職務分離マトリクス以外に、次のマトリクスの作成があります。
    - Oracle Database Vault固有マトリクス。Database Vaultロールを付与されているユーザーの名前およびタスクを含めることができます。
    - アプリケーション保護マトリクス。保護対象アプリケーションおよび設定した保護タイプを含めることができます。

表D-2は、OracleがPeopleSoftアプリケーション向けに作成した保護の例を示しています。SYSADM、PSFTDBA、SYSTEMおよびDBAはすべて、適切なルール・セットに対して認可されています。

表D-2 アプリケーション保護マトリクスの例

保護タイプ	SYSADM	PSFTDBA	SYSTEM	DBA
PeopleSoft レルム	所有者	所有者	アクセス権なし	アクセス権なし
SELECT コマンド・ルール	制限なし	制限 PSFTDB ルール・セット	アクセス権なし	アクセス権なし
CONNECT コマンド・ルール	PeopleSoftAccess ルール・セット	制限なし	制限なし	制限なし
DROP TABLESPACE コマンド・ルール	無効なルール・セット	無効なルール・セット	無効なルール・セット	無効なルール・セット

親トピック: 職務分離のガイドライン

D.2 Oracle Database管理アカウントの管理

Oracleでは、SYSTEMなどの管理アカウント、またはSYSDBA管理権限を持つユーザーのセキュリティを管理するためのガイドラインを提供しています。

- 一般的な管理目的のためのSYSTEMユーザー・アカウント
  理想的には、SYSTEMアカウントは、使用中にチェックアウトおよび監査されるバックアップとしてのみ使用可能になる必要があります。
- アプリケーション表のSYSTEMスキーマ
  SYSTEMスキーマ内にアプリケーション表がある場合は、SYSTEMアカウントをこれらの表に対するレルム認可に追加します。
- SYSDBA管理権限の制限
  SYSDBA管理権限は、接続時にこの権限の使用がどうしても必要なユーザーや、引き続きSYSDBAアクセスを必要と

するアプリケーションに制限してください。

- Oracle Database Vaultへのルートおよびオペレーティング・システムのアクセス
  セキュリティ向上のために、ルートおよびオペレーティング・システムのOracle Databaseへのアクセスを注意深く監視する必要があります。Vault。

親トピック: Oracle Database Vaultセキュリティ・ガイドライン

D.2.1 一般的な管理目的のためのSYSTEMユーザー・アカウント

理想的には、SYSTEMアカウントは、使用中にチェックアウトおよび監査されるバックアップとしてのみ使用可能になる必要があります。

共有アカウントではなく、名前付きアカウントのみが、通常のデータベース管理タスクのために使用される必要があります。これにより、データベース内の管理アクションに対するアカウンタビリティが向上します。

親トピック: Oracle Database管理アカウントの管理

D.2.2 アプリケーション表のSYSTEMスキーマ

SYSTEMスキーマ内にアプリケーション表がある場合は、SYSTEMアカウントをこれらの表に対するレルム認可に追加します。

これにより、これらのアプリケーションは引続き正常に動作します。

これらのアプリケーションのセキュリティを向上または微調整するため、SYSTEMアカウントに制限を加えることもできます。たとえば、SYSTEMユーザーのアクセスを特定のIPアドレスに制限するDatabase Vaultルール・セットを作成できます。

親トピック: Oracle Database管理アカウントの管理

D.2.3 SYSDBA管理権限の制限

SYSDBA管理権限は、接続時にこの権限の使用がどうしても必要なユーザーや、引き続きSYSDBAアクセスを必要とするアプリケーションに制限してください。

たとえば、必須のパッチ適用プロセスには、SYSDBAアクセスが必要です。

他のすべてのケースでは、日常的なデータベース管理を実行するための名前付きデータベース・アカウントを作成します。また、OSDBAユーザー・グループのメンバーにはSYSDBA管理権限が付与されます。オペレーティング・システムの`root`および`oracle`アカウントの他に、データベースSYSアカウント、およびSYSDBA権限があるアカウントは、特権アカウント管理(PAM)システムで管理され、要求時のみチェックアウトされる必要があります。

関連トピック

- SYSDBAアクセスの管理

親トピック: Oracle Database管理アカウントの管理

D.2.4 Oracle Database Vaultへのルートおよびオペレーティング・システムのアクセス

セキュリティ向上のために、ルートおよびオペレーティング・システムの次へのアクセスを注意深く監視する必要があります: Oracle Database Vault。

Oracle Database Vaultは、多くの権限を持つデータベース・ユーザーが機密データにアクセスすることを防ぎます。また、Oracle Database自体を使用している場合は、「透過的データ暗号化」を使用することで、最大の権限を持つオペレーティング・システム・ユーザーが機密データにアクセスできないようにできます。透過的データ暗号化では、表領域および表の列を暗号

化できます。これにより、オペレーティング・システム・ユーザーが、オペレーティング・システムのデータベース・ファイルを参照することや、機密データを見つけることができなくなります。オペレーティング・システムへの直接アクセスを常に注意深く確認し、制限することをお薦めします。

オペレーティング・システムにアクセスするには、アカウントをパーソナライズしておく必要があります。これらのパーソナライズされたアカウントは、LinuxまたはUNIX環境で、必要に応じてsudoを使用してoracleソフトウェア所有者にログインします。sudoでは、パーソナライズされた各ユーザーが実行できる特定のコマンドを制御できます。これらのユーザーについては、make、relink、gdb、またはデータベースで問題が発生する可能性があるその他のコマンドの使用を禁止してください。ただし、管理ユーザーがパッチをインストール、またはその他の緊急操作を実行する必要がある場合は、一定の時間のみmakeおよびrelinkコマンドを有効にし、この時間中のアクションを監査できます。

関連項目:

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。

親トピック: Oracle Database管理アカウントの管理

D.3 Oracle Database Vaultによって信頼されるアカウントおよびロール

Oracle Database Vaultは、データベース内の権限を与えられた多くのユーザーおよびロールからのアプリケーション・データへのアクセスを制限します。

ただし、場合によっては、Oracle Database Vaultsは特定のロールおよび権限を信頼します。

表D-3に、信頼できるロールおよび権限を示します。これらのロールおよび権限は、Oracle Database Vaultのインストール時に作成されます。

表D-3 信頼できるOracle Database Vaultロールおよび権限

ロールまたは権限	ステータス	説明
DV_ACCTMGR ロール	オープン	登録時に作成され、新規データベース・アカウントの作成に使用されるロール。安全対策として、DV_ACCTMGR ロールがあるバックアップ・ユーザーを保持し、特権アカウント管理(PAM)システムを使用してこのアカウントを管理します。 DV_OWNER ロールがあるユーザーは、このユーザーを変更できません。 DV_ACCTMGR ロールがあるすべてのアカウントを失った場合(パスワードの紛失や組織からのユーザーの退職などが原因)、回復できません。必ずバックアップ DV_ACCTMGR アカウントをこの目的のために作成するようにしてください。
DV_OWNER ロール	オープン	登録時に作成され、レルム、ファクタおよびコマンド・ルールの管理に使用されるロール。このユーザーは、自分自身をレルム認可に追加できます。安全対策として、DV_OWNER ロールがあるバックアップ・ユーザーを保持し、特権アカウント管理(PAM)システムを使用してこのアカウントを管理します。

ロールまたは権限	ステータス	説明
		DV_OWNER ロールがあるユーザーは、このユーザーを変更できません。 DV_OWNER ロールがあるすべてのアカウントを失った場合(パスワードの紛失や組織からのユーザーの退職などが原因)、回復できません。必ずバックアップ DV_OWNER アカウントをこの目的のために作成するようにしてください。
SYSDBA 権限	有効	Oracle Database のインストール時に作成される権限。一部の Oracle 機能に必要です。
SYSOPER 権限	有効	Oracle Database のインストール時に作成される権限。データベースの起動および停止。デフォルトでは SYS にのみ付与されます。

関連トピック

- バックアップOracle Database Vaultアカウント
- SYSDBAアクセスの管理
- SYSOPERアクセスの管理

親トピック: Oracle Database Vaultセキュリティ・ガイドライン

D.4 信頼できる人物に制限する必要のあるアカウントおよびロール

強力なアカウントおよびロールは、信頼できる人物に制限する必要があります。

- オペレーティング・システムへのルート・アクセス権を持つユーザーの管理
  ルート・ユーザー・アクセス権を持つユーザーは、システムを完全に制御できます。
- Oracleソフトウェア所有者の管理
  システムにOracleソフトウェア所有者としてのアクセス権を持つユーザーは、Oracleソフトウェアを制御できます。
- SYSDBAアクセスの管理
  通常のデータベース・メンテナンス・タスクではSYSアカウントおよびSYSDBA権限を使用しないようにする必要があります。
- SYSOPERアクセスの管理
  デフォルトでは、Oracle Databaseは、SYSOPERアクセスをOSOPERグループのオペレーティング・システム・ユーザーおよびユーザーSYSに限定します。

親トピック: Oracle Database Vaultセキュリティ・ガイドライン

D.4.1 オペレーティング・システムへのルート・アクセス権を持つユーザーの管理

ルート・ユーザー・アクセス権を持つユーザーは、システムを完全に制御できます。

これらのユーザーが実行できるアクティビティには、次のものがあります。

- 暗号化されていないファイルの読取り
- ファイルの移動および削除
- システムでのプログラムの起動または停止

- Oracle Databaseインストール環境を所有するユーザーを含む任意のユーザーとしてのログイン

Oracle Database Vaultでは、オペレーティング・システム・ルート・アクセスに対して保護されません。rootおよびoracleアカウントを特権アカウント管理(PAM)システムで管理します。これらのアカウントは、特定のタスクに必要な場合のみチェックアウトします。強い権限があるオペレーティング・システム・アカウントが使用されている場合は、キーストローク取得およびビデオ取得まで、監査レベルを高くします。

親トピック: 信頼できる人物に制限する必要のあるアカウントおよびロール

D.4.2 Oracleソフトウェア所有者の管理

システムにOracleソフトウェア所有者としてのアクセス権を持つユーザーは、Oracleソフトウェアを制御できます。

これらのユーザーが実行できるアクティビティには、次のものがあります。

- 暗号化されていないデータベース・ファイルの読取り
- データベース・ファイルの移動および削除
- システムでのOracleプログラムの起動または停止

Oracle Database Vaultでは、Oracleソフトウェア所有者のオペレーティング・システム・アクセスから保護されません。Oracleソフトウェア所有者アカウントを特権アカウント管理(PAM)システムで管理します。このアカウントは、特定のタスクに必要な場合のみチェックアウトします。強い権限があるオペレーティング・システム・アカウントが使用されている場合は、キーストローク取得およびビデオ取得まで、監査レベルを高くします。

親トピック: 信頼できる人物に制限する必要のあるアカウントおよびロール

D.4.3 SYSDBAアクセスの管理

通常のデータベース・メンテナンス・タスクではSYSアカウントおよびSYSDBA権限を使用しないようにする必要があります。

かわりに、必要なシステム権限、またはSYSBACKUP、SYSDG、SYSKMなどの特定の管理権限がある名前付きアカウントを使用します。ただし、パッチの実行、データベースのアップグレードまたは問題のトラブルシューティング(たとえば、停止したデータベースへの接続)のためにSYSDBA権限が必要な場合があります。

SYSDBA権限のあるユーザーは、直接的または間接的(たとえば、診断、データベース・アップグレードおよびパッチ適用による)に機密アプリケーション・データにアクセスできるため、SYSDBA権限およびアカウントの使用は、厳密に制限する必要があります。強力な権限のあるアカウントのリストには、SYS、およびデータベース内のSYSDBA権限のあるユーザー・アカウント、およびオペレーティング・システム内のrootおよびoracleアカウントが含まれます。データベースおよびオペレーティング・システム内の強力な権限のあるアカウントへのアクセスは、例外ベースである必要があり、ユーザーが、これらのアカウントおよび権限へのアクセスをロック解除するプロセスを経る必要があります。これらのアカウントを特権アカウント管理(PAM)システムで管理することをお薦めします。これらのアカウントは、特定のタスクに必要な場合のみチェックアウトします。強い権限があるオペレーティング・システム・アカウント(rootおよびoracle)およびデータベース・アカウント(SYSアカウントおよびSYSDBA管理権限)が使用されている場合は、キーストローク取得およびビデオ取得まで、監査レベルを高くします。これらの強い権限のあるアカウントがデータベースにアクセスするときには、SYSアカウントを監査してそれらのアクティビティを監視します。SYS (またはSYSDBA管理者権限を持つユーザー)にDV_PATCH_ADMINロールが付与されている場合は、バッチ適用操作の際にENABLE_DV_PATCH_ADMIN_AUDITプロシージャを使用することをお薦めします。

関連トピック

- ENABLE_DV_PATCH_ADMIN_AUDITプロシージャ

親トピック: 信頼できる人物に制限する必要のあるアカウントおよびロール

D.4.4 SYSOPERアクセスの管理

デフォルトでは、Oracle Databaseは、SYSOPERアクセスをOSOPERグループのオペレーティング・システム・ユーザーおよびユーザーSYSに限定します。

これによって、SYSOPERがOracleデータ・ディクショナリを直接変更することを防ぎます。SYSOPER権限は、データベース内の制限された権限を持ちますが、このロールを持つユーザーは、Oracleデータベースの起動と停止を行うことができます。SYSOPER権限は、信頼できるユーザーにのみ付与してください。

親トピック: 信頼できる人物に制限する必要のあるアカウントおよびロール

D.5 Oracle Database Vaultを本番環境で使用するためのガイドライン

Oracle Database Vaultを本番環境で実行する際は、特定のガイドラインに従う必要があります。

これらのガイドラインは、次のとおりです。

- アプリケーションの全テストを実行して、作成したDatabase Vaultポリシーが予測どおりに機能していることを確認します。
- アプリケーションのパフォーマンスを監視し、必要に応じてルール式を調整します。
- 適切な製品サポートおよびセキュリティ・グループに職責を次のように割り当てます。
    - データベース・セキュリティ管理者にセキュリティ職責を割り当てます。
    - データベース・アカウント・マネージャにアカウント管理を割り当てます。
    - データベース管理者にリソース管理タスクを割り当てます。
- Database Vault APIスクリプトをセキュア・サーバーにバックアップします。

親トピック: Oracle Database Vaultセキュリティ・ガイドライン

D.6 セキュアな構成のガイドライン

PL/SQLパッケージ、権限およびリサイクルビンのセキュリティの考慮事項を認識する必要があります。

- 一般的なセキュア構成のガイドライン
  一般的なセキュア構成のガイドラインには、パッチおよび取消し操作が含まれます。
- UTL_FILEおよびDBMS_FILE_TRANSFERパッケージのセキュリティの考慮事項
  UTL_FILEおよびDBMS_FILE_TRANSFER PL/SQLパッケージへのアクセスを慎重に制限する必要があります。
- CREATE ANY JOB権限のセキュリティの考慮事項
  CREATE ANY JOB権限はDBAロールおよびSCHEDULER_ADMINロールから取り消されています。
- CREATE EXTERNAL JOB権限のセキュリティの考慮事項
  CREATE EXTERNAL JOB権限は、Oracle Database 10gリリース2 (10.2)で導入されました。
- LogMinerパッケージのセキュリティの考慮事項
  ロールEXECUTE_CATALOG_ROLEには、いくつかのLogMinerパッケージに対するEXECUTE権限がデフォルトでは付与されません。
- ALTER SYSTEMおよびALTER SESSION権限のセキュリティの考慮事項について
  強力なALTER SYSTEMおよびALTER SESSIONシステム権限を保護する方法を認識する必要があります。

親トピック: Oracle Database Vaultセキュリティ・ガイドライン

D.6.1 一般的なセキュア構成のガイドライン

一般的なセキュア構成のガイドラインには、パッチおよび取消し操作が含まれます。

- パッチおよび新規アプリケーションをインストールすると、この項で取り消した権限のうち、Oracleが推奨する一部の権限が再付与されることがあります。パッチおよび新規アプリケーションをインストールした後で、これらの権限を確認し、取り消されたままになっていることを確認してください。
- パッケージに対するEXECUTE権限を取り消した場合は、所有者にパッケージに対するEXECUTEが付与されていることを確認し、パッケージの依存関係を確認し、取消し後に無効なパッケージを再コンパイルしてください。

  パッケージにアクセスできるユーザーを確認するには、名前付きデータベース管理者としてデータベース・インスタンスにログインして、次の問い合せを発行します。

  ```
  SELECT * FROM DBA_TAB_PRIVS WHERE TABLE_NAME = package_name;
  ```

  package_nameは、検索するパッケージの名前です。

  パッケージに依存するユーザー、パッケージ、プロシージャおよびファンクションを検索するには、次の問合せを発行します。

  ```
  SELECT OWNER, NAME, TYPE  FROM ALL_DEPENDENCIES
  WHERE REFERENCED_NAME = package_name;
  ```

  この2つの問合せは、動的SQLで行われたパッケージへの参照を識別しません。

親トピック: セキュアな構成のガイドライン

D.6.2 UTL_FILEおよびDBMS_FILE_TRANSFERパッケージのセキュリティの考慮事項

UTL_FILEおよびDBMS_FILE_TRANSFER PL/SQLパッケージへのアクセスを慎重に制限する必要があります。

- UTL_FILEおよびDBMS_FILE_TRANSFERパッケージのセキュリティの考慮事項について
  UTL_FILEパッケージは、SYSによって所有され、PUBLICに付与されます。
- DBMS_FILE_TRANSFERパッケージへのアクセスの保護
  DBMS_FILE_TRANSFER PL/SQLパッケージへのアクセスを様々な方法で保護できます。
- 例: CREATE DATABASE LINKへのアクセスを拒否するコマンド・ルールの作成
  DBMS_MACADM.CREATE_COMMAND_RULEでは、コマンド・ルールを作成して、CREATE DATABASE LINK SQL文へのアクセスを拒否できます。
- 例: CREATE DATABASE LINKへのアクセスを有効にするコマンド・ルールの作成
  DBMS_MACADM.UPDATE_COMMAND_RULEプロシージャは、既存のコマンド・ルールを変更するために使用できます。
- 例: CREATE DIRECTORYへのアクセスを無効および有効にするコマンド・ルール

親トピック: セキュアな構成のガイドライン

D.6.2.1 UTL_FILEおよびDBMS_FILE_TRANSFERパッケージのセキュリティの考慮事項について

UTL_FILEパッケージはSYSによって所有され、PUBLICに付与されます。

ただし、ユーザーがオペレーティング・システム・ディレクトリ内のファイルを操作するためには、そのディレクトリ・オブジェクトへのアク

セス権が必要です。

DBMS_FILE_TRANSFERパッケージは、SYSによって所有され、EXECUTE_CATALOG_ROLEに付与されます。このパッケージに対するEXECUTEアクセス権を持つユーザーは、同じファイル・システムの1つの場所から別の場所にファイルを移動できます。リモート・システム上のデータベースを含め、データベース・インスタンス間でファイルを移動することもできます。

関連項目:

UTL_FILEパッケージの安全な構成については、『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: UTL_FILEおよびDBMS_FILE_TRANSFERパッケージのセキュリティの考慮事項

D.6.2.2 DBMS_FILE_TRANSFERパッケージへのアクセスの保護

DBMS_FILE_TRANSFER PL/SQLパッケージへのアクセスを様々な方法で保護できます。

- 次の方法のいずれかを使用して、DBMS_FILE_TRANSFER PL/SQLパッケージを保護します。
  - DBMS_FILE_TRANSFERパッケージのEXECUTE権限を取り消し、この権限を必要とする信頼できるユーザーに対してのみEXECUTE権限を付与します。
  - コマンド・ルールを作成して、CREATE DATABASE LINKおよびCREATE DIRECTORY SQL文を制御します。Oracle Database Vault Administratorを使用したコマンド・ルールの作成の詳細は、「コマンド・ルールの作成」を参照してください。
  - Oracle Database Vaultコマンド・ルールを作成して、CREATE DATABASE LINKおよびCREATE DIRECTORY文へのアクセスを制限および有効化します。これらの文は、リモート・データベースへの接続を確立するために使用します。

関連項目:

CREATE DATABASE LINK文の使用を保護するために作成できるコマンド・ルールの例は、次の項を参照してください。

- 例: CREATE DATABASE LINKへのアクセスを拒否するコマンド・ルールの作成
- 例: CREATE DATABASE LINKへのアクセスを有効にするコマンド・ルールの作成
- 例: CREATE DIRECTORYへのアクセスを無効および有効にするコマンド・ルール

親トピック: UTL_FILEおよびDBMS_FILE_TRANSFERパッケージのセキュリティの考慮事項

D.6.2.3 例: CREATE DATABASE LINKへのアクセスを拒否するコマンド・ルールの作成

DBMS_MACADM.CREATE_COMMAND_RULEでは、コマンド・ルールを作成して、CREATE DATABASE LINK SQL文へのアクセスを拒否できます。

例D-1に、CREATE DATABASE LINK権限へのアクセスを拒否するコマンド・ルールの作成方法を示します。

例D-1 CREATE DATABASE LINKへのアクセスを拒否するコマンド・ルールの作成

```
BEGIN
 DBMS_MACADM.CREATE_COMMAND_RULE (
  command       => 'CREATE DATABASE LINK',
  rule_set_name => 'Disabled',
  object_owner  => '%',
```

```
  object_name    => '%',
  enabled        => DBMS_MACUTL.G_YES);
  END;
  /
COMMIT;
```

親トピック: UTL_FILEおよびDBMS_FILE_TRANSFERパッケージのセキュリティの考慮事項

D.6.2.4 例: CREATE DATABASE LINKへのアクセスを有効にするコマンド・ルールの作成

DBMS_MACADM.UPDATE_COMMAND_RULEプロシージャは、既存のコマンド・ルールを変更するために使用できます。

例D-2に、CREATE DATABASE LINK権限へのアクセスを有効にするコマンド・ルールの作成方法を示します。

有効なユーザーがCREATE DATABASE LINK文を使用する必要がある場合、Oracle Database Vault所有者は、Oracle Database Vault Administratorから再び有効にするか、SQL*Plusで次のコマンドを発行できます。

例D-2 CREATE DATABASE LINKへのアクセスを有効にするコマンド・ルールの作成

```
BEGIN
 DBMS_MACADM.UPDATE_COMMAND_RULE (
  command        => 'CREATE DATABASE LINK',
  rule_set_name => 'Enabled',
  object_owner   => '%',
  object_name    => '%',
  enabled        => DBMS_MACUTL.G_YES);
 END;
 /
COMMIT;
```

親トピック: UTL_FILEおよびDBMS_FILE_TRANSFERパッケージのセキュリティの考慮事項

D.6.2.5 例: CREATE DIRECTORYへのアクセスを無効および有効にするコマンド・ルール

例D-3に、CREATE DIRECTORYへのアクセスを無効および有効にするコマンド・ルールを示します。

例D-3 CREATE DIRECTORYへのアクセスを無効および有効にするコマンド・ルール

```
-- Disable access to CREATE DIRECTORY
BEGIN
 DBMS_MACADM.CREATE_COMMAND_RULE (
  command        => 'CREATE DIRECTORY',
  rule_set_name => 'Disabled',
  object_owner   => '%',
  object_name    => '%',
  enabled        => dbms_macutl.g_yes);
 END;
  /
COMMIT;
-- Enable access to CREATE DIRECTORY
BEGIN
 dbms_macadm.update_command_rule (
  command        => 'CREATE DIRECTORY',
  rule_set_name => 'Enabled',
  object_owner   => '%',
  object_name    => '%',
  enabled        => dbms_macutl.g_yes);
 END;
 /
COMMIT;
```

親トピック: UTL_FILEおよびDBMS_FILE_TRANSFERパッケージのセキュリティの考慮事項

D.6.3 CREATE ANY JOB権限のセキュリティの考慮事項

CREATE ANY JOB権限はDBAロールおよびSCHEDULER_ADMINロールから取り消されています。

使用しているアプリケーションにこの変更による影響がないことを確認してください。

関連トピック

- Oracle Database VaultでのOracle Schedulerの使用

親トピック: セキュアな構成のガイドライン

D.6.4 CREATE EXTERNAL JOB権限のセキュリティの考慮事項

CREATE EXTERNAL JOB権限は、Oracle Database 10gリリース2 (10.2)で導入されました。

この権限は、データベースの外部のオペレーティング・システムで稼働するジョブを実行するデータベース・ユーザーに必要です。デフォルトでは、CREATE EXTERNAL JOB権限は、CREATE JOB権限が付与されたすべてのユーザーに付与されます。セキュリティを高めるために、この権限を必要としないユーザーからこの権限を取り消し、必要とするユーザーにのみ付与してください。

親トピック: セキュアな構成のガイドライン

D.6.5 LogMinerパッケージのセキュリティの考慮事項

ロールEXECUTE_CATALOG_ROLEには、いくつかのLogMinerパッケージに対するEXECUTE権限がデフォルトでは付与されません。

これらのパッケージは次のとおりです。

- DBMS_LOGMNR
- DBMS_LOGMNR_D
- DBMS_LOGMNR_LOGREP_DICT
- DBMS_LOGMNR_SESSION

使用しているアプリケーションにこの変更による影響がないことを確認してください。

親トピック: セキュアな構成のガイドライン

D.6.6 ALTER SYSTEMおよびALTER SESSION権限のセキュリティの考慮事項

強力なALTER SYSTEMおよびALTER SESSIONシステム権限を保護する方法を認識する必要があります。

- ALTER SYSTEMおよびALTER SESSION権限のセキュリティの考慮事項について
  トレースおよびデバッグ・コマンドは、Oracleデータベース・メモリー情報を表示する可能性があることに注意してください。
- 例: 既存のALTER SYSTEMコマンド・ルールへのルールの追加
  ALTER SYSTEM権限のあるユーザーがALTER SYSTEM文を発行できないようにするルールを作成できます。

親トピック: セキュアな構成のガイドライン

D.6.6.1 ALTER SYSTEMおよびALTER SESSION権限のセキュリティの考慮事項について

トレースおよびデバッグ・コマンドは、Oracleデータベース・メモリー情報を表示する可能性があることに注意してください。

Oracle Database Vaultは、これらのコマンドに対する保護を行いません。Oracleデータベース・メモリー情報を保護するために、ALTER SYSTEM権限およびALTER SESSION権限へのアクセスを厳密に制御することをお薦めします。これらの権限は、

SYSDBAとして接続しているときにユーザーSYSによって、またDBAロールを付与されている任意のユーザーによって付与できます。

また、ALTER SYSTEM文の既存のコマンド・ルールにツールを追加することをお薦めします。Oracle Database Vault Administratorを使用してルールを作成し、そのルールをルール・セットに追加できます。ALTER SESSION権限は、信頼できるユーザーにのみ付与してください。(たとえば、ALTER SESSION文はトレースを有効にできます。)

親トピック: ALTER SYSTEMおよびALTER SESSION権限のセキュリティの考慮事項

D.6.6.2 例: 既存のALTER SYSTEMコマンド・ルールへのルールの追加

ALTER SYSTEM権限のあるユーザーがALTER SYSTEM文を発行できないようにするルールを作成できます。

例D-4では、ALTER SYSTEM権限のあるユーザーがALTER SYSTEM DUMP文を発行できないようにするルールを作成する方法を示します。このコマンド・ルールを作成するときに、Oracle Database Vault所有者アカウントとしてデータベース・インスタンスにログインします。

または、Oracle Database Vault Administratorを使用してルールを作成し、そのルールをルール・セットに追加できます。詳細は、「ルール・セットに追加するルールの作成」を参照してください。

例D-4 既存のALTER SYSTEMコマンド・ルールへのルールの追加

```
CONNECT accts_admin_ace
Enter password: password
BEGIN
 DBMS_MACADM.CREATE_RULE('NO_SYSTEM_DUMP',
 '(INSTR(UPPER(DV_SQL_TEXT),''DUMP'') = 0)');
 END;
/
EXEC DBMS_MACADM.ADD_RULE_TO_RULE_SET
  ('Allow Fine Grained Control of System Parameters','NO_SYSTEM_DUMP');
COMMIT;
```

親トピック: ALTER SYSTEMおよびALTER SESSION権限のセキュリティの考慮事項

E Oracle Database Vaultのトラブルシューティング

トレース・ファイルなどのツールを使用したり、特定のOracle Database Vaultレポートを確認したりして、Oracle Database Vaultのトラブルシューティングを行うことができます。

- トレース・ファイルを使用したOracle Database Vaultイベントの診断
  データベースによって生成されるトレース・ファイルは、エラーのデバッグに役立つ重要な情報を取得します。
- 一般的な診断のヒント
  Oracleでは、レルム、ファクタおよびルール・セットの問題を診断する一般的なヒントを提供しています。
- Oracle Database Vaultコンポーネントにかかわる構成の問題
  Oracle Database Vaultには、レルム、コマンド・ルール、ファクタ、ルール・セットまたはセキュア・アプリケーション・ロールの構成の問題をチェックするためのレポートが用意されています。
- Oracle Database Vaultのアカウント・パスワードのリセット
  バックアップ・アカウントは、DV_OWNERおよびDV_ACCTMGRロールが付与されているユーザーの紛失したパスワードをリセットする際に役立ちます。

E.1 トレース・ファイルを使用したOracle Database Vaultイベントの診断

データベースによって生成されるトレース・ファイルは、エラーのデバッグに役立つ重要な情報を取得します。

- トレース・ファイルを使用したOracle Database Vaultイベントの診断について
  Oracle Database Vaultデータベース・インスタンスのトレース・ファイルを有効化および確認して、サーバーやバックグラウンド・プロセスのイベントについてOracle Database Vaultのデータベース・インスタンスを監視できます。
- Oracle Database Vaultで追跡できるトレース・イベントと追跡できないイベントのタイプ
  トレース・ファイルを使用して、様々なOracle Database Vaultアクティビティを追跡できます。
- Oracle Database Vaultトレース・イベントのレベル
  Oracle Database Vaultのトレース・イベントにいくつかのレベルを使用できます。
- Oracle Database Vaultトレース・ファイルを有効にしたときのパフォーマンスへの影響
  トレース・ファイルの有効化には注意してください。
- Oracle Database Vaultトレース・イベントの有効化
  ALTER SESSIONまたはALTER SYSTEM SQL文を使用して、Oracle Database Vaultトレース・イベントを有効化できます。
- Oracle Database Vaultトレース・ファイル・データの検索
  Linux grepコマンドおよびADRコマンド・インタプリタ(ADRCI)コマンドライン・ユーティリティを使用すると、Oracle Database Vaultトレース・ファイル・データを検索できます。
- 例: 低レベルのOracle Database Vaultレルム違反を示すトレース・ファイル
  トレース・ファイル・データを使用して、低レベルのレルム違反を追跡できます。
- 例: 高レベルのトレースを有効にしたOracle Database Vault権限
  高レベルのトレースが有効になっているトレース・ファイルで、Oracle Database Vault認可を追跡できます。
- 例: レルム保護されたオブジェクトに対する違反の最高レベルのトレース
  トレース・ファイルを使用して高レベルの違反を追跡できます。
- Oracle Database Vaultトレース・イベントの無効化
  Oracle Database Vaultイベントのトレースを無効化できます。

親トピック: Oracle Database Vaultのトラブルシューティング

E.1.1 トレース・ファイルを使用したOracle Database Vaultイベントの診断について

Oracle Database Vaultデータベース・インスタンスのトレース・ファイルを有効化および確認して、Oracle Database Vaultのデータベース・インスタンスで、サーバーやバックグラウンド・プロセスのイベントを監視することができます。

トレース・ファイルを見れば、Oracle Database Vaultのポリシー認可が成功したか失敗したかを確認できます。バグおよびその他の問題が発生したとき、その解決に有効な情報を得ることができます。

Oracle Database Vaultのトレースを設定するには、DV_ADMINロールが必要です。この構成を実行するには、ALTER SESSION SET EVENTSまたはALTER SYSTEM SET EVENTSというSQL文を使用します。

関連項目:

トレース・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

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.2 Oracle Database Vaultで追跡できるトレース・イベントと追跡できないイベントのタイプ

トレース・ファイルを使用して、様々なOracle Database Vaultアクティビティを追跡できます。

表E-1に、これらのアクティビティを示します。

表E-1 Oracle Database Vaultトレース・ファイルの内容

Database Vaultの機能	説明
レルムの認可	このトレース・ファイルは、ロールに対するルール・セットとレルム認可を使用して、レルム認可の各ケースを追跡します。このタイプのトレース・ファイルの例は、「例: 低レベルのOracle Database Vault レルム違反を示すトレース・ファイル」を参照してください。
ルール・セットの評価	このトレース・ファイルには、レルム認可、コマンド・ルールの CONNECT コマンド・ルール、およびファクタからのルール・セット評価に関する情報が含まれます。
Oracle Data Pump の認可	このトレース・ファイルには、Database Vault の Data Pump の認可結果と、ユーザー、オブジェクト、SQL テキストに関するその他の情報が含まれます。
Oracle Scheduler ジョブの認可	このトレース・ファイルには、Database Vault の Oracle Scheduler ジョブの認可結果、ジョブ名、ジョブ所有者、現在の文などが含まれます。
オブジェクト権限のバイパス	このトレース・ファイルは、直接の権限付与と、ロールを介した権限付与の両方を追跡します。このタイプのトレースは、必須レルムが有効になっておらず、オブジェクト権限を持つユーザーがレルム保護されたオブジェクトにアクセスできる場合に役に立ちます。
ファクタのロード	このトレース・ファイルは、ロードされた各ファクタの式と値を追跡します。

Database Vaultの機能	説明
その他	オブジェクト所有者がバイパスしたレルム保護や、Database Vault で失敗した操作、成功した操作

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.3 Oracle Database Vaultトレース・イベントのレベル

Oracle Database Vaultのトレース・イベントにいくつかのレベルを使用できます。

これらのレベルは次のとおりです。

- 低: Oracle Database Vaultで失敗したすべての認可に関する情報をトレース・ファイルに出力します。このタイプのトレース・ファイルには、レルム認可の失敗、ファクタのロードの失敗、ルール・セット評価の失敗などが記録されます。Oracleデータベースのパフォーマンスに対する影響は小さくなります。
- 高: 認可の成功と失敗の両方を含むトレース・レコードを出力します。このタイプのトレースは、すべての認可を追跡するので、低レベルのトレースよりオーバーヘッドが大きくなります。また、トレース・ファイルのサイズも通常は大きくなります。
- 最高: PL/SQLスタックとファンクション・コール・スタック、および高レベルでトレースされる内容(表E-1を参照)をトレース・ファイルに出力します。Oracleデータベースのパフォーマンスに対する影響は最も大きくなります。

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.4 Oracle Database Vaultトレース・ファイルを有効にしたときのパフォーマンスへの影響

トレース・ファイルの有効化には注意してください。

有効化により、データベース・インスタンス操作のオーバーヘッドが増加し、パフォーマンスが低下する可能性があります。

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.5 Oracle Database Vaultトレース・イベントの有効化

`ALTER SESSION`または`ALTER SYSTEM` SQL文を使用して、Oracle Database Vaultトレース・イベントを有効化できます。

- 現在のデータベース・セッションに対するトレース・イベントの有効化
  `ALTER SESSION SET EVENTS` SQL文を使用して、現在のデータベース・セッションのトレース・イベントを有効化できます。
- すべてのデータベース・セッションに対するトレース・イベントの有効化
  `ALTER SYSTEM SET EVENTS` SQL文を使用して、すべてのデータベース・セッションのDatabase Vaultトレース・イベントを有効化できます。
- マルチテナント環境でのトレース・イベントの有効化
  マルチテナント環境では、トレース・イベントは、現在のユーザー・セッションとすべてのデータベース・セッションの両方に影響します。

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.5.1 現在のデータベース・セッションに対するトレース・イベントの有効化

ALTER SESSION SET EVENTS SQL文を使用して、現在のデータベース・セッションのトレース・イベントを有効化できます。

1. DV_ADMINロールおよびALTER SESSIONシステム権限を付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus sec_admin_owen
   Enter password: password
   Connected.
   ```

2. 「Oracle Database Vaultトレース・イベントのレベル」で説明されているように、ALTER SESSION SET EVENTS SQL文を入力してトレースを低、高または最高に設定します。
   - 影響が小さい操作の失敗に対してトレースを有効にするには、次のいずれかの文を入力します。

     ```
     ALTER SESSION SET EVENTS 'TRACE[DV] DISK=LOW';
     ALTER SESSION SET EVENTS '47998 TRACE NAME CONTEXT FOREVER, LEVEL 1';
     ```

   - 影響が大きい操作の失敗と成功に対してトレースを有効にするには、次のいずれかの文を入力します。

     ```
     ALTER SESSION SET EVENTS 'TRACE[DV] DISK=HIGH';
     ALTER SESSION SET EVENTS '47998 TRACE NAME CONTEXT FOREVER, LEVEL 3';
     ```

   - 最も影響が大きいファンクションとPL/SQLコール・スタックを使用する操作の失敗と成功に対してトレースを有効にするには、次のいずれかの文を入力します。

     ```
     ALTER SESSION SET EVENTS 'TRACE[DV] DISK=HIGHEST';
     ALTER SESSION SET EVENTS '47998 TRACE NAME CONTEXT FOREVER, LEVEL 4';
     ```

親トピック: Oracle Database Vaultトレース・イベントの有効化

E.1.5.2 すべてのデータベース・セッションに対するトレース・イベントの有効化

ALTER SYSTEM SET EVENTS SQL文を使用して、現在のデータベース・セッションのDatabase Vaultトレース・イベントを有効化できます。

1. DV_ADMINロールおよびALTER SYSTEMシステム権限を付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus sec_admin_owen
   Enter password: password
   Connected.
   ```

2. 「現在のデータベース・セッションに対するトレース・イベントの有効化」のステップ2に示されている構文を使用して、ALTER SYSTEM SET EVENTS SQL文を入力します。

   たとえば:

   ```
   ALTER SYSTEM SET EVENTS 'TRACE[DV] DISK=LOW';
   ```

すべてのデータベース・セッションに対してトレース・イベントを有効にするには、init.oraファイルに次の行を追加してデータベースを再起動する方法もあります。

```
event="47998 trace name context forever, level [trace_level]"
```

trace_levelを、次のいずれかの値に置き換えます。

- 1: 最低レベルのトレース
- 3: 高レベルのトレース
- 4: 最高レベルのトレース

たとえば:

```
event="47998 trace name context forever, level [1]"
```

関連トピック

- すべてのデータベース・セッションに対するトレース・イベントの無効化
- Oracle Database Vaultトレース・イベントのレベル

親トピック: Oracle Database Vaultトレース・イベントの有効化

E.1.5.3 マルチテナント環境でのトレース・イベントの有効化

マルチテナント環境では、トレース・イベントは、現在のユーザー・セッションとすべてのデータベース・セッションの両方に影響します。

- 現在のユーザー・セッションのトレース・イベント: マルチテナント環境で、ルートまたはプラガブル・データベース(PDB)からALTER SESSION SET EVENTS SQL文を実行すると、現在のユーザー・セッションに対するトレースが有効になります。1つのPDBから別のPDBに切り替えた(ALTER SESSION SET CONTAINER文を使用して)場合、切り替え後のPDBに対して引き続きトレースが有効です。マルチテナント・コンテナ・データベース(CDB)におけるトレースの有効化は単一のPDBに対してはできず、トレースはすべてのPDBおよびルートに適用されます。PDBを切り替えるには、ALTER SESSION SET CONTAINERシステム権限が必要です。
- すべてのデータベース・セッションのトレース・イベント: マルチテナント環境で、ルートまたは特定のPDBからALTER SYSTEM SET EVENTS文を実行すると、コンテナ・データベースのすべてのPDBに対するトレースが有効になります。

親トピック: Oracle Database Vaultトレース・イベントの有効化

E.1.6 Oracle Database Vaultトレース・ファイル・データの検索

Linux grepコマンドおよびADRコマンド・インタプリタ(ADRCI)コマンドライン・ユーティリティを使用すると、Oracle Database Vaultトレース・ファイル・データを検索できます。

- Database Vaultトレース・ファイルのディレクトリの場所の検索
  トレース・ファイルのディレクトリの完全な場所は、V$DIAG_INFO動的ビューを問い合せて検索することができます。
- Linuxのgrepコマンドを使用してトレース・ファイルから文字列を検索
  トレース・ファイルの問合せ、または処理を行うとき、Linuxのgrepコマンドを使用して文字列を検索することができます。
- ADRコマンド・インタプリタ(ADRCI)ユーティリティを使用してトレース・ファイルを問合せ
  ADRコマンド・インタプリタ(ADRCI)コマンドライン・ユーティリティを使用して、トレース・ファイルを問い合せることができます。

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.6.1 Database Vaultトレース・ファイルのディレクトリの場所の検索

トレース・ファイルのディレクトリの完全な場所は、V$DIAG_INFO動的ビューを問い合せて検索することができます。

- 次のように、V$DIAG_INFO動的ビューに問い合せます。

```
SELECT VALUE FROM V$DIAG_INFO WHERE NAME = 'Default Trace File';
```

次のような出力が表示されます。

```
VALUE
--------------------------------------------------------------------------------
/u01/app/oracle/product/12.1.0/log/diag/rdbms/orcl/orcl/trace/orcl_ora_7174.trc
```

親トピック: Oracle Database Vaultトレース・ファイル・データの検索

E.1.6.2 Linuxのgrepコマンドを使用してトレース・ファイルから文字列を検索

トレース・ファイルの問合せ、または処理を行うとき、Linuxのgrepコマンドを使用して文字列を検索することができます。

- たとえば、レルム認可の失敗を表示するトレース・ファイルを検索するには、次のコマンドを入力します。

```
grep 'Result=Realm Authorization Failed'  *.trc
```

親トピック: Oracle Database Vaultトレース・ファイル・データの検索

E.1.6.3 ADRコマンド・インタプリタ(ADRCI)ユーティリティを使用してトレース・ファイルを問合せ

ADRコマンド・インタプリタ(ADRCI)コマンドライン・ユーティリティを使用して、トレース・ファイルを問い合せることができます。

- ADRCIユーティリティを使用してトレース・ファイル情報を検索するには、SHOWコマンドを使用します。

たとえば、ADRCIを使用してトレース・ファイルを検索するには、SHOW TRACEFILEコマンドを入力します。

```
adrci --To start ACRCI from the command line
adrci> show tracefile
diag/rdbms/orcl/orcl/trace/orcl_m002_14551.trc
diag/rdbms/orcl/orcl/trace/orcl_tmon_13450.trc
diag/rdbms/orcl/orcl/trace/orcl_vktm_963.trc
diag/rdbms/orcl/orcl/trace/alert_orcl.log
...
```

すべてのトレース・インシデントの数を検索するには、次のようにします。

```
adrci> show incident
ADR Home = /u01/app/oracle/product/12.1.0/log/diag/rdbms/orcl/orcl:
*************************************************************************
234 rows fetched
```

次のADRCIコマンドは、名前にoraという語が含まれているすべてのトレース・ファイルのリストを返します。

```
adrci> show tracefile %ora%
/u01/app/oracle/product/12.1.0/log/diag/rdbms/orcl/orcl/trace/orcl_ora_18841.trc
/u01/app/oracle/product/12.1.0/log/diag/rdbms/orcl/orcl/trace/orcl_ora_12017.trc
/u01/app/oracle/product/12.1.0/log/diag/rdbms/orcl/orcl/trace/orcl_ora_19372.trc
/u01/app/oracle/product/12.1.0/log/diag/rdbms/orcl/orcl/trace/orcl_ora_12221.trc
/u01/app/oracle/product/12.1.0/log/diag/rdbms/orcl/orcl/trace/orcl_ora_1600.trc
...
```

次のADRCIコマンドは、Realm Authorization Failedというフレーズを含むトレース・ファイルを検索します。

```
adrci> show trace %trc -xp "[payload like '%Realm Authorization Failed%']"
```

関連項目:

- ADRCIユーティリティの詳細は、『Oracle Databaseユーティリティ』を参照してください

- ADRCIユーティリティでのレポートの表示については、『Oracle Database管理者ガイド』を参照してください

親トピック: Oracle Database Vaultトレース・ファイル・データの検索

E.1.7 例: 低レベルのOracle Database Vaultレルム違反を示すトレース・ファイル

トレース・ファイル・データを使用して、低レベルのレルム違反を追跡できます。

例E-1に、低レベルのレルム違反の追跡の例を示します。

例E-1 低レベルのOracle Database Vaultレルム違反を示すトレース・ファイル

```
*** 2010-02-05 18:35:31.438
*** SESSION ID:(34.559) 2010-02-05 18:35:31.438
*** CLIENT ID:() 2010-02-05 18:35:31.438
*** SERVICE NAME:(SYS$USERS) 2010-02-05 18:35:31.438
*** MODULE NAME:(SQL*Plus) 2010-02-05 18:35:31.438
*** ACTION NAME:() 2010-02-05 18:35:31.438

Result=Realm Authorization Failed
        Realm_Name=realm 3      Required_Auth_Level=0
        Current_User=116
        Object_Owner=U1 Object_Name=T1  Object_Type=TABLE
        SQL_Text=INSERT INTO U1.T1 VALUES(30)

Result=Realm Authorization Failed
        Realm_Name=realm 3      Required_Auth_Level=0
        Current_User=116
        Object_Owner=U1 Object_Name=T1  Object_Type=TABLE
        SQL_Text=DELETE FROM U1.T1

Result=Realm Authorization Failed
        Realm_Name=realm 3      Required_Auth_Level=0
        Current_User=116
        Object_Owner=U1 Object_Name=T3  Object_Type=TABLE
        SQL_Text=CREATE TABLE U1.T3(C INT)

*** 2010-02-05 18:35:34.465

Result=Realm Authorization Failed
        Realm_Name=realm 3      Required_Auth_Level=0
        Current_User=116
        Object_Owner=U1 Object_Name=T1  Object_Type=TABLE
        SQL_Text=INSERT INTO U1.T1 VALUES(30)

Result=Realm Authorization Failed
        Realm_Name=realm 3      Required_Auth_Level=0
        Current_User=116
        Object_Owner=U1 Object_Name=T1  Object_Type=TABLE
        SQL_Text=DELETE FROM U1.T1
```

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.8 例: 高レベルのトレースを有効にしたOracle Database Vault権限

高レベルのトレースが有効になっているトレース・ファイルで、Oracle Database Vault認可を追跡できます。

例E-2に、このタイプのトレース・ファイルの例を示します。

例E-2 高レベルのトレースを有効にしたOracle Database Vault権限

```
Result= Realm Authorization Passed
        Reason=Current user is the object owner
        Current_User=70 Command=SELECT
```

```
        Object_Owner=LBACSYS    Object_Name=LBAC$AUDIT  Object_Type=TABLE

Result= Realm Authorization Passed
        Reason=Current user is the object owner
        Current_User=70 Command=SELECT
        Object_Owner=LBACSYS    Object_Name=LBAC$AUDIT  Object_Type=TABLE

Result= Realm Authorization Passed
        Reason=Current user is the object owner
        Current_User=70 Command=SELECT
        Object_Owner=LBACSYS    Object_Name=LBAC$POL    Object_Type=TABLE

Result= Realm Authorization Passed
        Reason=Current user is the object owner
        Current_User=70 Command=SELECT
        Object_Owner=LBACSYS    Object_Name=LBAC$USER_LOGON     Object_Type=VIEW

……

Result= Realm Authorization Passed
        Reason=Current user is the object owner
        Current_User=70 Command=SELECT
        Object_Owner=LBACSYS    Object_Name=LBAC$POL    Object_Type=TABLE

Result=Set Factor Value
        Factor_Name=Sensitive_Treatments
Factor_Expression=/SURGERY/PSYCHOLOGICAL

Result=Set Factor Value
        Factor_Name=Database_Instance
Factor_Expression=UPPER(SYS_CONTEXT('USERENV','INSTANCE'))      Factor_Value=1

Result=Set Factor Value
        Factor_Name=Client_IP
Factor_Expression=UPPER(SYS_CONTEXT('USERENV','IP_ADDRESS'))    Factor_Value=

Result=Set Factor Value
        Factor_Name=Authentication_Method
Factor_Expression=UPPER(SYS_CONTEXT('USERENV','AUTHENTICATION_METHOD'))
Factor_Value=PASSWORD
……

*** ACTION NAME:() 2010-02-05 18:47:19.540

Result=Rule Set Evaluation Failed
        Command=SELECT  RuleSet_ID=2    RuleSet_Name=Disabled
        Current_User=SYSTEM
        Object_Owner=U1 Object_Name=T1  Object_Type=TABLE
        SQL_Text=SELECT * FROM U1.T1

Result=Rule Set Evaluation Succeeded
        Command=SELECT  RuleSet_ID=1    RuleSet_Name=Enabled
        Current_User=SYSTEM
        Object_Owner=U1 Object_Name=T1  Object_Type=TABLE
        SQL_Text=SELECT * FROM U1.T1
```

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.9 例: レルム保護されたオブジェクトに対する違反の最高レベルのトレース

トレース・ファイルを使用して高レベルの違反を追跡できます。

例E-3では、最高レベルのトレースを有効にしたときのトレース・ファイルでOracle Schedulerのジョブ認可に関連する最高レベルの違反が示されています。

例E-3 レルム保護されたオブジェクトに対する違反の最高レベルのトレース

```
------ Call Stack Trace ------
kzvdvechk<-kzvdveqau<-kksfbc<-opiexe<-kpoal8<-opiodr<-ttcpip<-opitsk<-opiino<-
opiodr<-opidrv<-sou2o<-opimai_real<-ssthrdmain<-main<-__libc_start_main<-_start

Result=Object Privilege check passed
        Current_User=INVOKER2   Used_Role=1
        Object_Owner=SYSTEM     Object_Name=PRODUCT_PRIVS       Object_Type=VIEW
        SQL_Text=SELECT CHAR_VALUE FROM SYSTEM.PRODUCT_PRIVS WHERE (UPPER('SQL*PLUS')
LIKE UPPER(PRODUCT)) AND ((USER LIKE USERID) OR (USERID = 'PUBLIC')) AND
(UPPER(ATTRIBUTE) = 'ROLES')
*** MODULE NAME:(SQL*Plus) 2010-02-05 18:57:53.973
*** ACTION NAME:() 2010-02-05 18:57:53.973

----- Current SQL Statement for this session (sql_id=2sr63rjm45yfh) -----
UPDATE INVOKER1.T1 SET A = 20
----- PL/SQL Stack -----
----- PL/SQL Call Stack -----
  object      line  object
  handle    number  name
0x26a00e34         1  anonymous block
0x2495b000       185  package body SYS.DBMS_ISCHED
0x24958fb8       486  package body SYS.DBMS_SCHEDULER
0x247bbb34         1  anonymous block

------ Call Stack Trace ------
kzvdvechk<-kzvdveqau<-kksfbc<-opiexe<-opipls<-opiodr<-__PGOSF151_rpidrus<-skgmstack<-
rpidru<-rpiswu2<-rpidrv<-psddr0<-psdnal<-pevm_EXECC<-pfrinstr_EXECC<-pfrrun_no_tool<-
pfrrun<-plsql_run<-peicnt<-kkxexe<-opiexe<-kpoal8<-opiodr<-kpoodr<-upirtrc<-kpurcsc<-
kpuexec
<-OCIStmtExecute<-jslvec_execcb<-jslvswu<-jslve_execute0<-jskaJobRun<-jsiRunJob<-
jsaRunJob<-spefcmpa<-spefmccallstd<-pextproc<-__PGOSF495_peftrusted<-
__PGOSF522_psdexsp<-rpiswu2<-psdextp<-pefccal<-pefcal<-pevm_FCAL<-pfrinstr_FCAL<-
pfrrun_no_tool<-pfrrun<-plsql_run
<-peicnt<-kkxexe<-opiexe<-kpoal8<-opiodr<-ttcpip<-opitsk<-opiino<-opiodr<-opidrv<-
sou2o<-opimai_real<-ssthrdmain<-main<-__libc_start_main<-_start

Result=Realm Authorization Succeeded
        Realm_Name=jobowner realm       Used_Auth_Level=0
        Current_User=119
        Object_Owner=INVOKER1   Object_Name=T1  Object_Type=TABLE
        SQL_Text=UPDATE INVOKER1.T1 SET A = 20

Result=Scheduler Job Authorization Succeeded
        Current_User=JOBOWNER   Logon_User=INVOKER2
        Job_Owner=JOBOWNER      Job_Name=DMLJOB1
        Object_Owner=INVOKER1   Object_Name=T1  Object_Type=TABLE
        SQL_Text=UPDATE INVOKER1.T1 SET A = 20
```

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.10 Oracle Database Vaultトレース・イベントの無効化

Oracle Database Vaultイベントのトレースを無効化できます。

- 現在のデータベース・セッションに対するトレース・イベントの無効化
  `ALTER SESSION SET EVENTS` SQL文を使用して、現在のデータベース・セッションのDatabase Vaultのトレースを無効化できます。
- すべてのデータベース・セッションに対するトレース・イベントの無効化
  `ALTER SYSTEM SET EVENTS` SQL文を使用して、すべてのデータベース・セッションのDatabase Vaultトレースを無効化できます。

- マルチテナント環境でのトレース・イベントの無効化
  マルチテナント環境でのトレース・イベントの無効化は、現在のユーザー・セッションとすべてのデータベース・セッションの両方に影響します。

親トピック: トレース・ファイルを使用したOracle Database Vaultイベントの診断

E.1.10.1 現在のデータベース・セッションに対するトレース・イベントの無効化

ALTER SESSION SET EVENTS SQL文を使用して、現在のデータベース・セッションのDatabase Vaultのトレースを無効化できます。

1. DV_ADMINロールおよびALTER SESSIONシステム権限を付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus sec_admin_owen
   Enter password: password
   Connected.
   ```

2. トレースを無効にするには、次のSQL文の両方を入力します。

   ```
   ALTER SESSION SET EVENTS 'TRACE[DV] OFF';
   ALTER SESSION SET EVENTS '47998 trace name context off';
   ```

   あるいは、ALTER SYSTEM文を使用する方法もあります。

   ```
   ALTER SYSTEM SET EVENTS 'TRACE[DV] OFF';
   ALTER SYSTEM SET EVENTS '47998 trace name context off';
   ```

親トピック: Oracle Database Vaultトレース・イベントの無効化

E.1.10.2 すべてのデータベース・セッションに対するトレース・イベントの無効化

ALTER SYSTEM SET EVENTS SQL文を使用して、すべてのデータベース・セッションのDatabase Vaultトレースを無効化できます。

1. DV_ADMINロールおよびALTER SYSTEMシステム権限を付与されているユーザーとして、データベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus sec_admin_owen
   Enter password: password
   Connected.
   ```

2. 「現在のデータベース・セッションに対するトレース・イベントの無効化」のステップ2に示されている構文を使用して、ALTER SYSTEM SET EVENTS SQL文を入力します。

   たとえば:

   ```
   ALTER SYSTEM SET EVENTS 'TRACE[DV] OFF';
   ```

すべてのデータベース・セッションに対してトレース・イベントを無効にするには、init.oraファイルに次の行を追加してデータベースを再起動する方法もあります。

```
event="47998 trace name context off"
```

init.oraファイルで、47998行目との競合がないようにしてください。event="47998 trace name context forever, level [1]"などです。

親トピック: Oracle Database Vaultトレース・イベントの無効化

E.1.10.3 マルチテナント環境でのトレース・イベントの無効化

マルチテナント環境でのトレース・イベントの無効化は、現在のユーザー・セッションとすべてのデータベース・セッションの両方に影響します。

- 現在のユーザー・セッションのトレース・イベント: マルチテナント環境で、ルートまたはPDBからALTER SESSION SET EVENTS SQL文を実行すると、現在のユーザー・セッションに対するトレースが無効になります。1つのPDBから別のPDBに切り替えた(ALTER SESSION SET CONTAINER文を使用して)場合、切り替え後のPDBに対して引き続きトレースが無効です。CDB内の1つのPDBに対してトレースを無効にすることはできず、トレースはすべてのPDBとルートに適用されます。PDBを切り替えるには、ALTER SESSION SET CONTAINERシステム権限が必要です。
- すべてのデータベース・セッションのトレース・イベント: マルチテナント環境で、ルートまたは特定のPDBからALTER SYSTEM SET EVENTS文を実行すると、CDBのすべてのPDBに対するトレースが無効になります。

親トピック: Oracle Database Vaultトレース・イベントの無効化

E.2 一般的な診断のヒント

Oracleでは、レルム、ファクタおよびルール・セットの問題を診断する一般的なヒントを提供しています。

これらのガイドラインは、次のとおりです。

- レルム保護の場合、コマンドに作用する基礎となるシステム権限またはオブジェクト権限をユーザーが保持している(直接またはロールを介して付与されている)ことを確認します。
- レルム認可が機能しない場合は、アカウントのロールが正しく設定されていることを確認します。
- ファクタおよびルール・セットで使用されるPL/SQL式の場合、これらの式で使用されるPL/SQLパッケージ・ファンクションに対するEXECUTE権限をアカウントに直接付与し、結果が正しいと思われるかどうかを判断します。
- 一般に、監査レポートを使用して問題を診断します。詳細は、「Oracle Database Vaultの監査レポート」を参照してください。

親トピック: Oracle Database Vaultのトラブルシューティング

E.3 Oracle Database Vaultコンポーネントにかかわる構成の問題

Oracle Database Vaultには、レルム、コマンド・ルール、ファクタ、ルール・セットまたはセキュア・アプリケーション・ロールの構成の問題をチェックするためのレポートが用意されています。

詳細は、次の各項を参照してください。

- 「コマンド・ルール構成の問題」レポート
- 「ファクタ構成の問題」レポート
- 「アイデンティティのないファクタ」レポート
- 「アイデンティティ構成の問題」レポート

- 「レルム認可構成の問題」レポート
- 「ルール・セット構成の問題」レポート
- 「セキュア・アプリケーション構成の問題」レポート

これらのレポートを実行するには、「Oracle Database Vaultレポートの実行」を参照してください。

親トピック: Oracle Database Vaultのトラブルシューティング

E.4 Oracle Database Vaultのアカウント・パスワードのリセット

バックアップ・アカウントを使用すると、DV_OWNERロールとDV_ACCTMGRロールを付与されているユーザーの紛失したパスワードをリセットできます。

- DV_OWNERユーザー・パスワードのリセット
  DV_OWNERバックアップ・アカウントを使用して、DV_OWNERユーザー・パスワードをリセットできます。
- DV_ACCTMGRユーザー・パスワードのリセット
  DV_ACCTMGRバックアップ・アカウントを使用して、DV_ACCTMGRユーザー・パスワードをリセットできます。

親トピック: Oracle Database Vaultのトラブルシューティング

E.4.1 DV_OWNERユーザー・パスワードのリセット

DV_OWNERバックアップ・アカウントを使用して、DV_OWNERユーザー・パスワードをリセットできます。

DV_OWNERユーザーのパスワードをリセットするには、このユーザーから一時的にDV_OWNERロールを取り消し、パスワードをリセットしてから、もう一度ユーザーにロールを付与します。

1. DV_OWNERユーザー・アカウントのバックアップ・ユーザーとしてデータベース・インスタンスにログインします。

   たとえば:

   ```
   sqlplus dbv_owner_backup
   Enter password: password
   ```

2. パスワードを紛失したDV_OWNERユーザーからDV_OWNERロールを取り消します。

   たとえば:

   ```
   REVOKE DV_OWNER FROM sec_admin_owen;
   ```

3. DV_ACCTMGRロールを付与されているユーザーとして接続します。

   たとえば:

   ```
   CONNECT accts_admin_ace
   Enter password: password
   ```

4. DV_OWNERユーザーのパスワードをリセットします。

   ```
   ALTER USER sec_admin_owen IDENTIFIED BY password;
   ```

   『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

5. バックアップDV_OWNERユーザーとして接続します。

   ```
   CONNECT dbv_owner_backup
   ```

```
Enter password: password
```

6. DV_OWNERロールをもう一度DV_OWNERユーザーに付与します。

```
GRANT DV_OWNER TO sec_admin_owen WITH ADMIN OPTION;
```

ノート:

バックアップ DV_OWNER アカウントを、もう一度必要になる場合に備えて、必ず安全に格納するようにしてください。

親トピック: Oracle Database Vaultのアカウント・パスワードのリセット

E.4.2 DV_ACCTMGRユーザー・パスワードのリセット

DV_ACCTMGRバックアップ・アカウントを使用して、DV_ACCTMGRユーザー・パスワードをリセットできます。

DV_ACCTMGRユーザーのパスワードをリセットするには、バックアップDV_ACCTMGRを使用してこのユーザーのパスワードをリセットします。

1. DV_ACCTMGRユーザー・アカウントのバックアップ・ユーザーとしてデータベース・インスタンスにログインします。

   たとえば:

```
sqlplus dbv_acctmgr_backup
Enter password: password
```

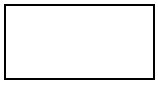
2. DV_ACCTMGRユーザーのパスワードをリセットします。

   たとえば:

```
ALTER USER accts_admin_ace IDENTIFIED BY password;
```

   『Oracle Databaseセキュリティ・ガイド』のガイドラインに従って、安全なパスワードでパスワードを置き換えてください。

ノート:



バックアップ DV_ACCTMGR アカウントを、もう一度必要になる場合に備えて、必ず安全に格納するようにしてください。

親トピック: Oracle Database Vaultのアカウント・パスワードのリセット

索引

A

B

C

D

- DBMS_MACADM.CREATE_CONNECT_COMMAND_RULEプロシージャ 16.2
- DBMS_MACADM.CREATE_DOMAIN_IDENTITYプロシージャ 17.1.5
- DBMS_MACADM.CREATE_FACTOR_TYPEプロシージャ 17.1.7
- DBMS_MACADM.CREATE_FACTORプロシージャ 17.1.6
- DBMS_MACADM.CREATE_IDENTITY_MAPプロシージャ 17.1.9
- DBMS_MACADM.CREATE_IDENTITYプロシージャ 17.1.8
- DBMS_MACADM.CREATE_MAC_POLICYプロシージャ 19.1
- DBMS_MACADM.CREATE_POLICY_LABELプロシージャ 19.2
- DBMS_MACADM.CREATE_POLICYプロシージャ 22.4
- DBMS_MACADM.CREATE_REALMプロシージャ 14.3
- DBMS_MACADM.CREATE_ROLEプロシージャ 18.1.1
- DBMS_MACADM.CREATE_RULE_SETプロシージャ 15.1.3
- DBMS_MACADM.CREATE_RULEプロシージャ 15.1.2
- DBMS_MACADM.CREATE_SESSION_EVENT_CMD_RULEプロシージャ 16.3
- DBMS_MACADM.CREATE_SYSTEM_EVENT_CMD_RULEプロシージャ 16.4
- DBMS_MACADM.DELETE_APP_EXCEPTIONプロシージャ 21.1.17
- DBMS_MACADM.DELETE_AUTH_FROM_REALMプロシージャ 14.4
- DBMS_MACADM.DELETE_COMMAND_RULEプロシージャ 16.5
- DBMS_MACADM.DELETE_CONNECT_COMMAND_RULEプロシージャ 16.6
- DBMS_MACADM.DELETE_FACTOR_LINKプロシージャ 17.1.11
- DBMS_MACADM.DELETE_FACTOR_TYPEプロシージャ 17.1.12
- DBMS_MACADM.DELETE_FACTORプロシージャ 17.1.10
- DBMS_MACADM.DELETE_IDENTITY_MAPプロシージャ 17.1.14
- DBMS_MACADM.DELETE_IDENTITYプロシージャ 17.1.13
- DBMS_MACADM.DELETE_MAC_POLICY_CASCADEプロシージャ 19.3
- DBMS_MACADM.DELETE_OBJECT_FROM_REALMプロシージャ 14.5
- DBMS_MACADM.DELETE_OWNER_FROM_POLICYプロシージャ 22.6
- DBMS_MACADM.DELETE_POLICY_FACTORプロシージャ 19.4
- DBMS_MACADM.DELETE_POLICY_LABELプロシージャ 19.5
- DBMS_MACADM.DELETE_REALM_CASCADEプロシージャ 14.7
- DBMS_MACADM.DELETE_REALM_FROM_POLICYプロシージャ 22.7
- DBMS_MACADM.DELETE_REALMプロシージャ 14.6
- DBMS_MACADM.DELETE_ROLEプロシージャ 18.1.2
- DBMS_MACADM.DELETE_RULE_FROM_RULE_SETプロシージャ 15.1.5
- DBMS_MACADM.DELETE_RULE_SETプロシージャ 15.1.6
- DBMS_MACADM.DELETE_RULEプロシージャ 15.1.4
- DBMS_MACADM.DELETE_SESSION_EVENT_CMD_RULEプロシージャ 16.7
- DBMS_MACADM.DELETE_SYSTEM_EVENT_CMD_RULEプロシージャ 16.8
- DBMS_MACADM.DISABLE_APP_PROTECTIONプロシージャ 21.1.18
- DBMS_MACADM.DISABLE_DV_DICTIONARY_ACCTSプロシージャ 21.1.20
- DBMS_MACADM.DISABLE_DV_PATCH_ADMIN_AUDITプロシージャ 21.1.21
- DBMS_MACADM.DISABLE_DVプロシージャ 21.1.19
- DBMS_MACADM.DISABLE_ORADEBUGプロシージャ 21.1.22

E

F

G

H

I

J

L

M

N

O

P

Q

R

T

U

V

- AUDSYS.DV$CONFIGURATION_AUDIT 24.52
- AUDSYS.DV$ENFORCEMENT_AUDIT 24.53
- CDB_DV_STATUS 24.2
- DBA_DV_APP_EXCEPTION 24.3
- DBA_DV_CODE 24.4
- DBA_DV_COMMAND_RULE 24.5
- DBA_DV_DATAPUMP_AUTH 24.6
- DBA_DV_DBCAPTURE_AUTH 24.7
- DBA_DV_DBREPLAY_AUTH 24.8
- DBA_DV_DDL_AUTH 24.9
- DBA_DV_DICTIONARY_ACCTS 24.10
- DBA_DV_FACTOR 24.11
- DBA_DV_FACTOR_TYPE 24.12
- DBA_DV_IDENTITY 24.14
- DBA_DV_IDENTITY_MAP 24.15
- DBA_DV_JOB_AUTH 24.16
- DBA_DV_MAINTENANCE_AUTH 24.19
- DBA_DV_ORADEBUG 24.20
- DBA_DV_PATCH_ADMIN_AUDIT 24.21
- DBA_DV_POLICY 24.22
- DBA_DV_POLICY_LABEL 24.23
- DBA_DV_POLICY_OBJECT 24.24
- DBA_DV_POLICY_OWNER 24.25
- DBA_DV_PREPROCESSOR_AUTH 24.26
- DBA_DV_PROXY_AUTH 24.27
- DBA_DV_PUB_PRIVS 24.28
- DBA_DV_REALM 24.29
- DBA_DV_REALM_AUTH 24.30
- DBA_DV_REALM_OBJECT 24.31
- DBA_DV_ROLE 24.32
- DBA_DV_RULE_SET 24.34
- DBA_DV_RULE_SET_RULE 24.35
- DBA_DV_SIMULATION_LOG 24.36
- DBA_DV_STATUS 24.37
- DBA_DV_TTS_AUTH 24.38
- DBA_DV_USER_PRIVS 24.39
- DBA_DV_USER_PRIVS_ALL 24.40
- DVSYS.DV$CONFIGURATION_AUDIT 24.41
- DVSYS.DV$ENFORCEMENT_AUDIT 24.42
- DVSYS.DV$REALM 24.43
- DVSYS.POLICY_OWNER_COMMAND_RULE 24.44
- DVSYS.POLICY_OWNER_POLICY 24.45
- DVSYS.POLICY_OWNER_REALM 24.46

W

X