

SQL*Plus®

ユーザーズ・ガイドおよびリファレンス 19c

F16155-05(原本部品番号:E96459-06)

2021年7月

タイトルおよび著作権情報

SQL*Plusユーザズ・ガイドおよびリファレンス, 19c

F16155-05

[Copyright ©](#) 1996, 2021, Oracle and/or its affiliates.

原著者: Celin Cherian

原簿協力者: Luan Nim、Andrei Souleimanian、Senthilprabhu Dhamotharan、Mahantesh Savanur

目次

- [タイトルおよび著作権情報](#)
- [はじめに](#)
 - [対象読者](#)
 - [ドキュメントのアクセシビリティについて](#)
 - [関連ドキュメント](#)
 - [表記規則](#)
- [このリリースにおけるSQL*Plusの変更点](#)
 - [SQL*Plusリリース19cの変更点](#)
 - [新機能](#)
 - [サポート対象外機能](#)
- [SQL*Plusのクイック・スタート](#)
 - [SQL*Plusのリソース](#)
 - [SQL*Plus概要](#)
 - [SQL*Plusの対象ユーザー](#)
 - [SQL*Plusの学習方法](#)
 - [このマニュアルの使用方法](#)
 - [SQL*Plusコマンドラインのアーキテクチャ](#)
 - [SQL*Plusクライアント](#)
 - [Oracle Database](#)
 - [SQL*Plusの前提条件](#)
 - [SQL*Plus日付書式](#)
 - [SQL*Plusコマンドラインの起動](#)
 - [SQL*Plus Instant Clientの起動について](#)
 - [他のデータベースへの接続について](#)
 - [サンプル・スキーマおよびSQL*Plusについて](#)
 - [サンプル表のロック解除](#)
 - [最初の間合せの実行](#)
 - [SQL*Plusの終了について](#)
- [第I部 SQL*Plusスタート・ガイド](#)
 - [1 SQL*Plusユーザー・インタフェース](#)
 - [1.1 コマンドラインの画面について](#)
 - [1.2 コマンドラインのフォントおよびフォント・サイズの変更](#)
 - [2 SQL*Plusの構成](#)
 - [2.1 SQL*Plusの環境変数](#)
 - [2.1.1 SQLPATHレジストリ・エントリ](#)
 - [2.2 SQL*Plusの構成](#)
 - [2.2.1 サイト・プロファイル](#)
 - [2.2.1.1 デフォルトのサイト・プロファイル・スクリプト](#)
 - [2.2.2 ユーザー・プロファイル](#)
 - [2.2.2.1 LOGINファイルの変更](#)
 - [2.2.3 SQL*Plusシステム変数の格納およびリストア](#)
 - [2.2.3.1 システム変数のリストア](#)

- [2.2.4 コマンドライン・ヘルプのインストールについて](#)
 - [2.2.4.1 hlpbld.sqlスクリプトを実行したコマンドライン・ヘルプのインストール](#)
 - [2.2.4.2 helpdrop.sqlスクリプトを実行したコマンドライン・ヘルプの削除](#)
 - [2.2.5 Oracle Net Servicesの構成について](#)
 - [3 SQL*Plusの起動](#)
 - [3.1 ユーザー名およびパスワードのログイン](#)
 - [3.1.1 セキュアな外部パスワード・ストア](#)
 - [3.1.2 パスワードの期限切れ](#)
 - [3.1.3 パスワードの変更について](#)
 - [3.2 データベースへの接続について](#)
 - [3.2.1 ネット・サービス名](#)
 - [3.2.2 完全な接続識別子](#)
 - [3.2.3 簡単な接続識別子](#)
 - [3.2.4 /NOLOGを使用した接続なしのセッション](#)
 - [3.3 SQL*Plusの起動について](#)
 - [3.3.1 SQL*Plusコマンドラインの起動について](#)
 - [3.3.2 コマンドライン・ヘルプの表示について](#)
 - [3.4 SQL*Plusコマンドラインの終了について](#)
 - [3.5 SQL*Plusプログラムの構文](#)
 - [3.5.1 オプション](#)
 - [3.5.1.1 HELPオプション](#)
 - [3.5.1.2 VERSIONオプション](#)
 - [3.5.1.3 COMPATIBILITYオプション](#)
 - [3.5.1.4 LOGONオプション](#)
 - [3.5.1.5 FASTオプション](#)
 - [3.5.1.6 MARKUPオプション](#)
 - [3.5.1.7 MARKUPの使用上のノート](#)
 - [3.5.1.8 No Login Timeオプション](#)
 - [3.5.1.9 RESTRICTオプション](#)
 - [3.5.1.10 SILENTオプション](#)
 - [3.5.2 ログオン](#)
 - [3.5.3 起動](#)
- [第II部 SQL*Plusの使用方法](#)
 - [4 SQL*Plusの基本](#)
 - [4.1 コマンドの入力と実行について](#)
 - [4.1.1 SQLバッファ](#)
 - [4.1.2 コマンドの実行について](#)
 - [4.2 表定義の表示について](#)
 - [4.3 PL/SQL定義の表示について](#)
 - [4.4 SQLコマンドの実行](#)
 - [4.4.1 SQLコマンドの構文について](#)
 - [4.4.1.1 1つのSQLコマンドの複数行への分割について](#)
 - [4.4.1.2 SQLコマンドの終了について](#)
 - [4.5 PL/SQLブロックの実行について](#)
 - [4.5.1 ストアド・プロシージャの作成について](#)

- [4.6 SQL*Plusコマンドの実行](#)
 - [4.6.1 SQL*Plusコマンドの構文について](#)
 - [4.6.1.1 長いSQL*Plusコマンドを次の行に続ける方法について](#)
- [4.7 コマンドの実行に影響するシステム変数](#)
- [4.8 実行中のコマンドの停止について](#)
- [4.9 オペレーティング・システムのコマンドの実行について](#)
- [4.10 表示の一時停止について](#)
- [4.11 データベースへの変更の自動保存について](#)
- [4.12 エラー・メッセージの解釈について](#)
- [5 SQL*Plusでのスクリプトの使用](#)
 - [5.1 スクリプトの編集について](#)
 - [5.1.1 システム・エディタでのスクリプトの作成](#)
 - [5.2 SQL*Plusコマンドラインでのスクリプトの編集について](#)
 - [5.2.1 バッファ内容の表示](#)
 - [5.2.2 カレント行の編集](#)
 - [5.2.3 行へのテキストの追加](#)
 - [5.2.4 新しい行の追加](#)
 - [5.2.5 行の削除](#)
 - [5.3 スクリプトへのコメントの挿入について](#)
 - [5.3.1 REMARKコマンドの使用法](#)
 - [5.3.2 /*...*/の使用](#)
 - [5.3.3 -- の使用](#)
 - [5.3.4 コメント挿入時のノート](#)
 - [5.4 スクリプトの実行](#)
 - [5.4.1 SQL*Plus起動時のスクリプトの実行](#)
 - [5.5 スクリプトのネスト](#)
 - [5.6 リターン・コードを伴うスクリプトの終了について](#)
 - [5.7 置換変数の定義](#)
 - [5.8 事前定義変数の使用について](#)
 - [5.9 置換変数の使用](#)
 - [5.9.1 置換変数を使用する位置および方法](#)
 - [5.9.2 値の入力を求める不要なプロンプトの回避](#)
 - [5.9.3 制限](#)
 - [5.9.4 システム変数](#)
 - [5.10 STARTコマンドを使用したパラメータの受渡し方法](#)
 - [5.11 ユーザーとの対話について](#)
 - [5.11.1 置換変数値の受信](#)
 - [5.11.2 置換変数の入力を求めるプロンプトのカスタマイズ](#)
 - [5.11.3 メッセージの送信および入力としての\[Return\]のアクセプト](#)
 - [5.11.4 画面のクリア](#)
 - [5.12 バインド変数の使用方法について](#)
 - [5.12.1 バインド変数の作成](#)
 - [5.12.2 バインド変数の参照](#)
 - [5.12.3 バインド変数の表示](#)
 - [5.12.4 入力バインドの実行](#)

- [5.13 REFCURSORバインド変数の使用方法](#)
- [5.14 PL/SQLブロックに含まれるSELECT文の結果の反復的なフェッチ](#)
- [6 SQL*Plusレポートの書式設定](#)
 - [6.1 列の書式設定について](#)
 - [6.1.1 列ヘッダーの変更について](#)
 - [6.1.1.1 デフォルトのヘッダー](#)
 - [6.1.1.2 デフォルトのヘッダーの変更](#)
 - [6.1.2 NUMBER列の書式設定について](#)
 - [6.1.2.1 デフォルトの表示](#)
 - [6.1.2.2 デフォルト表示の変更](#)
 - [6.1.3 データ型の書式設定について](#)
 - [6.1.3.1 デフォルトの表示](#)
 - [6.1.3.2 デフォルト表示の変更](#)
 - [6.1.4 列の表示属性のコピー](#)
 - [6.1.5 列の表示属性の表示およびリセット](#)
 - [6.1.6 列の表示属性の非表示および表示について](#)
 - [6.1.7 折り返した列の値の後で文字の行を出力する方法](#)
 - [6.2 間隔の設定およびサマリー行によるレポートの明確化について](#)
 - [6.2.1 ブレーク列内の値の重複の抑止](#)
 - [6.2.2 ブレーク列の値が変更されたときの空白の挿入](#)
 - [6.2.3 各行の後に空白を挿入する方法](#)
 - [6.2.4 複数の間隔設定技法の使用法](#)
 - [6.2.5 ブレーク定義の表示および取消し](#)
 - [6.2.6 ブレーク列の値が変更されたときにサマリー行を計算する方法](#)
 - [6.2.7 レポートの終わりでのサマリー行の計算](#)
 - [6.2.8 複数のサマリー値およびサマリー行の計算](#)
 - [6.2.9 COMPUTE定義の表示および取消し](#)
 - [6.3 ページおよびレポートのタイトルとサイズの定義について](#)
 - [6.3.1 上部タイトルと下部タイトルおよびヘッダーとフッターの設定](#)
 - [6.3.1.1 タイトル要素の位置決定](#)
 - [6.3.1.2 タイトル要素のインデント](#)
 - [6.3.1.3 長いタイトルの入力](#)
 - [6.3.2 タイトル内にシステム管理値を表示する方法](#)
 - [6.3.3 ページ・タイトル定義の表示と非表示の切替え](#)
 - [6.3.4 タイトル内に列の値を表示する方法](#)
 - [6.3.5 タイトル内に現在の日付を表示する方法について](#)
 - [6.3.6 ページ・サイズの設定](#)
 - [6.4 問合せ結果の格納および印刷について](#)
 - [6.4.1 フラット・ファイルの作成](#)
 - [6.4.2 ファイルへの結果の格納](#)
 - [6.4.3 プリンタへの結果の出力](#)
- [7 SQL*Plusからのレポートの生成](#)
 - [7.1 SQL*Plusコマンドラインを使用したレポートの作成について](#)
 - [7.1.1 HTMLレポートの作成](#)
 - [7.1.1.1 HTMLエンティティ](#)

- [7.1.2 CSVレポートの作成](#)
 - [7.1.3 レポートでのSQL*Plusコマンドの非表示について](#)
- [8 SQL*Plusのチューニング](#)
 - [8.1 文のトレースについて](#)
 - [8.1.1 自動トレース・レポートの制御](#)
 - [8.1.2 実行計画](#)
 - [8.1.3 統計](#)
 - [8.2 タイミング統計の収集について](#)
 - [8.3 パラレル問合せおよび分散問合せのトレース](#)
 - [8.4 以前のデータベースでの実行計画の出力](#)
 - [8.5 SQL*Plusスクリプトのチューニングについて](#)
 - [8.5.1 COLUMN NOPRINT](#)
 - [8.5.2 SET APPINFO OFF](#)
 - [8.5.3 SET ARRAYSIZE](#)
 - [8.5.4 SET DEFINE OFF](#)
 - [8.5.5 SET FLUSH OFF](#)
 - [8.5.6 SET LINESIZE](#)
 - [8.5.7 SET LONGCHUNKSIZE](#)
 - [8.5.8 SET PAGESIZE](#)
 - [8.5.9 SET SERVEROUTPUT](#)
 - [8.5.10 SET SQLPROMPT](#)
 - [8.5.11 SET TAB](#)
 - [8.5.12 SET TERMOUT](#)
 - [8.5.13 SET TRIMOUT ON SET TRIMSPool ON](#)
 - [8.5.14 UNDEFINE](#)
- [9 SQL*Plusのセキュリティ](#)
 - [9.1 SQL*Plus、SQLおよびPL/SQLコマンドの使用禁止](#)
 - [9.2 ロールの作成および制御について](#)
 - [9.2.1 SET ROLEの使用禁止について](#)
 - [9.2.2 ユーザー・ロールの使用禁止について](#)
 - [9.3 SQLPLUS -RESTRICTでのコマンドの使用禁止について](#)
 - [9.4 プログラム引数のセキュリティについて](#)
- [10 SQL*Plusによるデータベース管理](#)
 - [10.1 概要](#)
 - [10.2 データベースの起動および停止の概要](#)
 - [10.2.1 データベースの起動](#)
 - [10.2.2 PDB起動](#)
 - [10.2.3 データベースの停止](#)
 - [10.2.4 PDB停止](#)
 - [10.3 REDOLOG・ファイル](#)
 - [10.3.1 ARCHIVELOGモード](#)
 - [10.4 データベースのリカバリ](#)
- [11 SQL*Plusのグローバリゼーション・サポート](#)
 - [11.1 SQL*Plusコマンドラインでのグローバリゼーション・サポートの構成について](#)
 - [11.1.1 SQL*Plusクライアント](#)

- [11.1.2 Oracle Database](#)
 - [11.2 環境変数NLS_LANG](#)
 - [11.2.1 NLS_LANGの設定表示](#)
 - [11.3 NLS_LANGの設定](#)
- [第III部 SQL*Plusリファレンス](#)
 - [12 SQL*Plusコマンド・リファレンス](#)
 - [12.1 SQL*Plusコマンド一覧](#)
 - [12.2 @ \(アットマーク\)](#)
 - [12.3 @@ \(二重アットマーク\)](#)
 - [12.4 / \(スラッシュ\)](#)
 - [12.5 ACCEPT](#)
 - [12.6 APPEND](#)
 - [12.7 ARCHIVE LOG](#)
 - [12.8 ATTRIBUTE](#)
 - [12.9 BREAK](#)
 - [12.10 BTITLE](#)
 - [12.11 CHANGE](#)
 - [12.12 CLEAR](#)
 - [12.13 COLUMN](#)
 - [12.14 COMPUTE](#)
 - [12.15 CONNECT](#)
 - [12.16 COPY](#)
 - [12.17 DEFINE](#)
 - [12.17.1 事前定義変数](#)
 - [12.18 DEL](#)
 - [12.19 DESCRIBE](#)
 - [12.20 DISCONNECT](#)
 - [12.21 EDIT](#)
 - [12.22 EXECUTE](#)
 - [12.23 EXIT](#)
 - [12.24 GET](#)
 - [12.25 HELP](#)
 - [12.26 HISTORY](#)
 - [12.27 HOST](#)
 - [12.28 INPUT](#)
 - [12.29 LIST](#)
 - [12.30 PASSWORD](#)
 - [12.31 PAUSE](#)
 - [12.32 PRINT](#)
 - [12.33 PROMPT](#)
 - [12.34 RECOVER](#)
 - [12.35 REMARK](#)
 - [12.36 REPFOOTER](#)
 - [12.37 REPHEADER](#)
 - [12.38 RUN](#)

- [12.39 SAVE](#)
- [12.40 SET](#)
- [12.41 SETシステム変数の一覧](#)
 - [12.41.1 SET APPI\[NFO\]{ON | OFF | text}](#)
 - [12.41.2 SET ARRAY\[SIZE\] {15 | n}](#)
 - [12.41.3 SET AUTO\[COMMIT\]{ON | OFF | IMM\[EDIATE\] | n}](#)
 - [12.41.4 SET AUTOP\[RINT\] {ON | OFF}](#)
 - [12.41.5 SET AUTORECOVERY \[ON | OFF\]](#)
 - [12.41.6 SET AUTOT\[RACE\] {ON | OFF | TRACE\[ONLY\]} \[EXP\[LAIN\]\] \[STAT\[ISTICS\]\]](#)
 - [12.41.7 SET BLO\[CKTERMINATOR\] {. | c | ON | OFF}](#)
 - [12.41.8 SET CMDS\[EP\] {; | c | ON | OFF}](#)
 - [12.41.9 SET COLINVI\[SIBLE\] \[ON | OFF\]](#)
 - [12.41.10 SET COLSEP { | text}](#)
 - [12.41.11 SET CON\[CAT\] {. | c | ON | OFF}](#)
 - [12.41.12 SET COPYC\[OMMIT\] {0 | n}](#)
 - [12.41.13 SET COPYTYPECHECK {ON | OFF}](#)
 - [12.41.14 SET DEF\[INE\] {& | c | ON | OFF}](#)
 - [12.41.15 SET DESCRIBE \[DEPTH {1 | n | ALL}\] \[LINENUM {ON | OFF}\] \[INDENT {ON | OFF}\]](#)
 - [12.41.16 SET ECHO {ON | OFF}](#)
 - [12.41.17 SET EDITF\[ILE\] file_name\[.ext\]](#)
 - [12.41.18 SET EMB\[EDDED\] {ON | OFF}](#)
 - [12.41.19 SET ERRORL\[OGGING\] {ON | OFF} \[TABLE \[schema.\]tablename\] \[TRUNCATE\] \[IDENTIFIER identifier\]](#)
 - [12.41.20 SET ESC\[APE\] {¥ | c | ON | OFF}](#)
 - [12.41.21 SET ESCCHAR](#)
 - [12.41.22 SET EXITC\[OMMIT\] {ON | OFF}](#)
 - [12.41.23 SET FEED\[BACK\] {6 | n | ON | OFF | ONLY} \[SQL_ID\]](#)
 - [12.41.24 SET FLAGGER {OFF | ENTRY | INTERMED\[IATE\] | FULL}](#)
 - [12.41.25 SET FLU\[SH\] {ON | OFF}](#)
 - [12.41.26 SET HEA\[DING\] {ON | OFF}](#)
 - [12.41.27 SET HEADS\[EP\] { | c | ON | OFF}](#)
 - [12.41.28 SET HIST\[ORY\] {ON | OFF | n}](#)
 - [12.41.29 SET INSTANCE \[instance_path | LOCAL\]](#)
 - [12.41.30 SET LIN\[ESIZE\] {80 | n | WINDOW}](#)
 - [12.41.31 SET LOBOF\[FSET\] {1 | n}](#)
 - [12.41.32 SET LOBPREFETCH {0 | n}](#)
 - [12.41.33 SET LOGSOURCE \[パス名\]](#)
 - [12.41.34 SET LONG {80 | n}](#)
 - [12.41.35 SET LONGC\[HUNKSIZE\] {80 | n}](#)
 - [12.41.36 SET MARK\[UP\]](#)
 - [12.41.37 SET NEWP\[AGE\] {1 | n | NONE}](#)
 - [12.41.38 SET NULLテキスト](#)
 - [12.41.39 SET NUMF\[ORMAT\] format](#)

- [12.41.40 SET NUM\[WIDTH\] {10 | n}](#)
- [12.41.41 SET PAGES\[IZE\] {14 | n}](#)
- [12.41.42 SET PAU\[SE\] {ON | OFF | text}](#)
- [12.41.43 SET RECSEP {WR\[APPED\] | EA\[CH\] | OFF}](#)
- [12.41.44 SET RECSEPCHAR { | c}](#)
- [12.41.45 SET ROWLIMIT {n | OFF}](#)
- [12.41.46 SET ROWPREFETCH {1 | n}](#)
- [12.41.47 SET SECUREDCOL {OFF | ON} \[UNAUTH\[ORIZED\] text\] \[UNK\[NOWN\] text\]](#)
- [12.41.48 SET SERVEROUT\[PUT\] {ON | OFF} \[SIZE {n | UNL\[IMITED\]}\] \[FOR\[MAT\] {WRA\[PPED\] | WOR\[D_ WRAPPED\] | TRU\[NCATED\]}\]](#)
- [12.41.49 SET SHIFT\[INOUT\] {VIS\[IBLE\] | INV\[ISIBLE\]}](#)
- [12.41.50 SET SHOW\[MODE\] {ON | OFF}](#)
- [12.41.51 SET SQLBL\[ANKLINES\] {ON | OFF}](#)
- [12.41.52 SET SQLC\[ASE\] {MIX\[ED\] | LO\[WER\] | UP\[PER\]}](#)
- [12.41.53 SET SQLCO\[NTINUE\] {> | text}](#)
- [12.41.54 SET SQLN\[UMBER\] {ON | OFF}](#)
- [12.41.55 SET SQLPLUSCOMPAT\[IBILITY\] {x.y\[.z\]}](#)
 - [12.41.55.1 SQL*Plus互換性マトリックス](#)
- [12.41.56 SET SQLPRE\[FIX\] {# | c}](#)
- [12.41.57 SET SQLP\[ROMPT\] {SQL> | text}](#)
- [12.41.58 SET SQLT\[ERMINATOR\] {; | c | ON | OFF}](#)
- [12.41.59 SET STATEMENTC\[ACHE\] {0 | n}](#)
- [12.41.60 SET SUF\[FIX\] {SQL | text}](#)
- [12.41.61 SET TAB {ON | OFF}](#)
- [12.41.62 SET TERM\[OUT\] {ON | OFF}](#)
- [12.41.63 SET TI\[ME\] {ON | OFF}](#)
- [12.41.64 SET TIMI\[NG\] {ON | OFF}](#)
- [12.41.65 SET TRIM\[OUT\] {ON | OFF}](#)
- [12.41.66 SET TRIMS\[POOL\] {ON | OFF}](#)
- [12.41.67 SET UND\[ERLINE\] {- | c | ON | OFF}](#)
- [12.41.68 SET VER\[IFY\] {ON | OFF}](#)
- [12.41.69 SET WRA\[P\] {ON | OFF}](#)
- [12.41.70 SET XMLOPT\[IMIZATIONCHECK\] \[ON|OFF\]](#)
- [12.41.71 SET XQUERY BASEURI {text}](#)
- [12.41.72 SET XQUERY ORDERING {UNORDERED | ORDERED | DEFAULT}](#)
- [12.41.73 SET XQUERY NODE {BYVALUE | BYREFERENCE | DEFAULT}](#)
- [12.41.74 SET XQUERY CONTEXT {text}](#)
- [12.42 SHOW](#)
- [12.43 SHUTDOWN](#)
- [12.44 SPOOL](#)
- [12.45 START](#)
- [12.46 STARTUP](#)

- [12.47 STORE](#)
 - [12.48 TIMING](#)
 - [12.49 TTITLE](#)
 - [12.50 UNDEFINE](#)
 - [12.51 VARIABLE](#)
 - [12.52 WHENEVER OSERROR](#)
 - [12.53 WHENEVER SQLERROR](#)
 - [12.54 XQUERY](#)
- [13 SQL*Plusのエラー・メッセージ](#)
 - [13.1 SQL*Plusのエラー・メッセージ](#)
 - [13.2 COPYコマンドのメッセージ](#)
- [第IV部 SQL*Plusの付録](#)
- [A SQL*Plusの制限](#)
- [B SQL*PlusのCOPYコマンド](#)
 - [B.1 COPYコマンドの構文](#)
 - [B.1.1 用語](#)
 - [B.1.2 使用方法](#)
 - [B.1.3 例](#)
 - [B.2 データベース間でのデータのコピー](#)
 - [B.2.1 COPYコマンド構文について](#)
 - [B.2.2 宛先の表の扱いを制御する方法について](#)
 - [B.2.3 COPYで表示されるメッセージの解釈について](#)
 - [B.2.4 別ユーザーの表の指定](#)
 - [B.3 1つのデータベース上にある表の間でのデータのコピーについて](#)
- [C 廃止されたSQL*Plusコマンド](#)
 - [C.1 SQL*Plusで廃止されたコマンドとその代替コマンド](#)
 - [C.2 BTI\[TLE\] text \(旧形式は廃止\)](#)
 - [C.3 COL\[UMN\] {column|expr} DEF\[AULT\] \(廃止\)](#)
 - [C.4 DOC\[UMENT\] \(廃止\)](#)
 - [C.5 NEWPAGE \[1|n\] \(廃止\)](#)
 - [C.6 SET BUF\[FER\] {buffer|SQL} \(廃止\)](#)
 - [C.7 SET COM\[PATIBILITY\]{V7 | V8 | NATIVE} \(廃止\)](#)
 - [C.8 SET CLOSECUR\[SOR\] {ON|OFF} \(廃止\)](#)
 - [C.9 SET DOC\[UMENT\] {ON|OFF} \(廃止\)](#)
 - [C.10 SET MAXD\[ATA\] n \(廃止\)](#)
 - [C.11 SET SCAN {ON|OFF} \(廃止\)](#)
 - [C.12 SET SPACE {1|n} \(廃止\)](#)
 - [C.13 SET TRU\[NCATE\] {ON|OFF} \(廃止\)](#)
 - [C.14 TTI\[TLE\] text \(旧形式は廃止\)](#)
- [D SQL*Plus Instant Client](#)
 - [D.1 インストールするSQL*Plus Instant Clientの選択について](#)
 - [D.1.1 基本Instant Client](#)
 - [D.1.2 軽量Instant Client](#)
 - [D.1.2.1 サポートされていない文字セットによる軽量SQL*Plus Instant Clientエラー](#)
 - [D.2 OTNからのダウンロードによるSQL*Plus Instant Clientのインストールについて](#)

- [D.2.1 Linux RPMパッケージからのSQL*Plus Instant Clientのインストール](#)
 - [D.2.2 UNIXまたはWindows zipファイルからのSQL*Plus Instant Clientのインストール](#)
 - [D.2.3 SQL*Plus Instant Clientに必要なファイルのリスト](#)
- [D.3 19c Clientリリース・メディアからのSQL*Plus Instant Clientのインストール](#)
 - [D.3.1 UNIXまたはLinuxでのSQL*Plus Instant Clientのインストール](#)
 - [D.3.2 WindowsでのSQL*Plus Instant Clientのインストール](#)
- [D.4 SQL*Plus Instant Clientの構成について](#)
 - [D.4.1 Linuxでの\(RPMからの\)SQL*Plus Instant Clientの構成](#)
 - [D.4.2 Linux \(Clientメディアまたはzipファイルから\)およびUNIXでのSQL*Plus Instant Clientの構成](#)
 - [D.4.3 WindowsでのSQL*Plus Instant Clientの構成](#)
- [D.5 SQL*Plus Instant Clientによるデータベースへの接続について](#)
- [D.6 SQL*Plus Instant ClientによるAS SYSDBAまたはAS SYSOPER接続](#)
- [D.7 Instant Clientのアンインストールについて](#)
 - [D.7.1 SQL*Plus Instant Clientのアンインストール](#)
 - [D.7.2 Instant Clientの完全なアンインストール](#)
- [索引](#)

はじめに

『SQL*Plusユーザーズ・ガイドおよびリファレンス』では、SQL*Plusプログラムとその使用方法、および各SQL*Plusコマンドについて説明します。

ここでは、次の項目について説明します。

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

対象読者

『SQL*Plusユーザーズ・ガイドおよびリファレンス』は、次の作業を行うビジネス・ユーザー、テクニカル・ユーザーおよびシステム管理者の方を対象としています。

- バッチ・スクリプトの開発および実行
- 問合せ結果からの書式設定、計算、格納、印刷およびWebへの表示
- 表定義およびオブジェクト定義の検証
- データベースの管理

このマニュアルを使用するには、SQL言語の基礎知識が必要です。SQLの基礎知識については、[『Oracle Database SQL言語リファレンス』](#)を参照してください。SQL*PlusでのPL/SQLの使用を予定している場合は、[『Oracle Database PL/SQL言語リファレンス』](#)を参照してください。

ドキュメントのアクセシビリティについて

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

Oracle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。

関連ドキュメント

詳細は、次のOracleドキュメントを参照してください。

- [『SQL*Plusクイック・リファレンス』](#)
- [『Oracle Database PL/SQL言語リファレンス』](#)
- [『Oracle Database SQL言語リファレンス』](#)
- [『Oracle Call Interfaceプログラマーズ・ガイド』](#)
- [Oracle Database概要](#)

- [『Oracle Database管理者ガイド』](#)
- [『Oracle Databaseバックアップおよびリカバリ・アドバンスト・ユーザーズ・ガイド』](#)
- [『Oracle Databaseアドバンスト・アプリケーション開発者ガイド』](#)
- [『Oracle XML DB開発者ガイド』](#)
- [『Oracle Databaseグローバリゼーション・サポート・ガイド』](#)
- [『Oracle Database Heterogeneous Connectivity管理者ガイド』](#)
- [『Oracle Databaseエラー・メッセージ』](#)
- [『Oracle Databaseアップグレード・ガイド』](#)
- [Oracle Databaseリファレンス](#)
- [『Oracle Databaseパフォーマンス・チューニング・ガイド』](#)
- [『Oracle Database Net Services管理者ガイド』](#)
- [『Pro*COBOLプログラマーズ・ガイド』](#)
- [『Pro*C/C++プログラマーズ・ガイド』](#)
- ご使用のオペレーティング・システムのOracle Databaseのインストレーション・ガイドおよび管理者リファレンス

このマニュアルに記載されている多くの例で、Oracle Databaseのインストールで「基本インストール」オプションを選択した場合にデフォルトでインストールされるサンプル・スキーマを使用しています。スキーマの作成方法および使用方法については、[『Oracle Databaseサンプル・スキーマ』](#)を参照してください。

SQL*Plusエラー・メッセージ・マニュアルは、「[SQL*Plusのエラー・メッセージ](#)」を参照してください。Oracle Databaseエラー・メッセージ・マニュアルは、HTMLでのみ利用可能です。Oracle Databaseドキュメント・メディアのみが利用可能な場合は、分野ごとにOracle Databaseエラー・メッセージを参照できます。特定の範囲を検出したら、ブラウザの「このページの検索」機能を使用して特定のメッセージを検索します。インターネットに接続している場合、Oracle Databaseオンライン・マニュアルのエラー・メッセージ検索機能を使用すると、特定のエラー・メッセージを検索できます。

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、ドキュメントのタイトル、強調また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。

このリリースにおけるSQL*Plusの変更点

ここでは、『SQL*Plusユーザーズ・ガイドおよびリファレンス』での変更点を示します。

SQL*Plusリリース19cの変更点

SQL*Plusリリース19cの『SQL*Plusユーザーズ・ガイドおよびリファレンス』では、次の点が変更されています。

新機能

このリリースの新機能は次のとおりです。

- Oracle Databaseへの接続にアプリケーションで使用する簡易接続構文が拡張され、簡易接続プラスと呼ばれます。

[簡易接続プラス](#)に関する項を参照してください

サポート対象外機能

次の機能はサポートされなくなりました。

- PRODUCT_USER_PROFILE表のサポート終了

Oracle Database19c以降、SQL*Plus表PRODUCT_USER_PROFILE (PUP表)はサポート対象外です。

SQL*Plus製品レベルのセキュリティ機能は、Oracle Database19cでは使用できません。Oracle Databaseの設定を使用してデータを保護し、すべてのクライアント・アプリケーション間で一貫したセキュリティを確保することをお勧めします。

SQL*Plusのクイック・スタート

ここでは、SQL*Plusのインストール後、ログインしてデータベースに接続する手順について説明します。インストール中に作成したデフォルトのデータベースに接続したり、他の既存のOracle Databaseにも接続できます。

- [SQL*Plusのリソース](#)
- [SQL*Plus概要](#)
- [SQL*Plusの前提条件](#)
- [SQL*Plusコマンドラインの起動](#)
- [SQL*Plus Instant Clientの起動について](#)
- [他のデータベースへの接続について](#)
- [サンプル・スキーマおよびSQL*Plusについて](#)
- [最初の間合せの実行](#)
- [SQL*Plusの終了について](#)

SQL*Plusのリソース

- Oracleドキュメント・ライブラリ(<http://www.oracle.com/technetwork/>)。

SQL*Plus概要

SQL*Plusは、Oracle Databaseのインストール時にインストールされる対話型のバッチ問合せツールです。コマンドライン・ユーザー・インタフェースがあります。

また、SQL*Plus Instant Clientもあります。これは、OCI Instant Client対応のプラットフォームで利用できるスタンドアロン・コマンドライン・インタフェースです。SQL*Plus Instant Clientは使用可能なすべてのOracle Databaseに接続されますが、専用のOracle Databaseをインストールする必要はありません。OCI Instant Clientの詳細は、『[Oracle Call Interfaceプログラマーズ・ガイド](#)』を参照してください。

SQL*Plusには専用のコマンドおよび環境があり、Oracle Databaseへのアクセスを提供します。SQL、PL/SQL、SQL*Plusおよびオペレーティング・システム・コマンドを入力すると、次のことを実行できます。

- 問合せ結果の書式設定、計算の実行、格納および印刷
- 表定義およびオブジェクト定義の検証
- バッチ・スクリプトの開発および実行
- データベースの管理

SQL*Plusを使用して、対話的なレポート生成、バッチ処理としてのレポート生成、およびテキスト・ファイル、スクリーンまたはインターネットでの閲覧用のHTMLファイルへの結果の出力が可能です。SQL*PlusのHTML出力機能を使用して、動的にレポートを生成できます。

SQL*Plusの対象ユーザー

SQL*Plus、SQLおよびPL/SQLコマンド言語は、データベースの操作経験があるユーザーのニーズに応える十分な機能を備えている一方、Oracle Databaseの初心者にも十分にわかりやすい設計になっています。

SQL*Plus言語は簡単に使用できます。たとえば、LAST_NAMEというラベルが付いた列をヘッダー「Family Name」に変更するには、次のコマンドを入力します。

```
COLUMN LAST_NAME HEADING 'Family Name'
```

同様に、EMPLOYEES表の列定義を表示するには、次のコマンドを入力します。

```
DESCRIBE EMPLOYEES
```

SQL*Plusの学習方法

SQL*Plusを学習する場合、次のソースが有効です。

- このマニュアルの第II部の「[SQL*Plusの使用方法](#)」
- SQL*Plusのヘルプ、コマンドライン・ヘルプ
- Oracle Database 12c: SQL基礎

オラクル社が開設している、インストラクタが指導を行うコースです。このコースは、包括的な実地コースであり、SQL*Plusを使用してOracle Databaseへアクセスする方法を学習します。

- その他のOracle Database 12cトレーニング

オラクル社のその他のコースについては、<http://www.oracle.com/education>を参照してください。

このマニュアルの使用方法

このマニュアルでは、SQL*Plusについて、すべてのオペレーティング・システムに共通する情報を説明します。また、WindowsおよびUNIXに固有の情報も示します。SQL*Plusのいくつかの点は、オペレーティング・システムごとに異なる場合があります。オペレーティング・システム固有の情報については、ご使用のシステムのOracle Databaseのインストール・ガイドを参照してください。これらのオペレーティング・システム固有のガイドを、この『SQL*Plusユーザーズ・ガイドおよびリファレンス』とともに使用してください。

コマンドの入力方法を示す例では、このマニュアルの全編を通じて共通のコマンド構文および共通のサンプル表セットを使用しています。表については、[サンプル・スキーマおよびSQL*Plusについて](#)を参照してください。

SQL*Plusコマンドラインのアーキテクチャ

SQL*Plusコマンドラインでは、次の層で構成される2層モデルが使用されます。

- クライアント(コマンドライン・ユーザー・インタフェース)
- データベース(Oracle Database)

2つの層が同じコンピュータ上に存在する場合があります。

SQL*Plusクライアント

コマンドライン・ユーザー・インタフェースは、文字ベースの端末環境です。

Oracle Database

Oracle Netコンポーネントによって、SQL*PlusクライアントとOracle Database間の通信が可能です。

SQL*Plusの前提条件

SQL*Plusは、Oracle Databaseのコンポーネントです。SQL*Plusは、Oracle Databaseのインストール時にデフォルトでインストールされます。

Oracle DatabaseおよびSQL*Plusには、コンピュータおよびオペレーティング・システムによって異なる点があります。これらの相違点の詳細は、SQL*Plusでサポートされているオペレーティング・システムに固有のOracle Databaseのインストール・ガイドを参照してください。

SQL*Plusを実行する前に、次の準備が必要です。

- Oracle DatabaseまたはOracle Clientをインストールします。<http://www.oracle.com/technetwork/>から、ご使用のオペレーティング・システムのOracle Databaseのインストール・ガイドを参照してください。
- インストール中に、またはデータベース管理者から、Oracle Databaseのログイン・ユーザー名およびパスワードを取得します。「[ユーザー名およびパスワードのログイン](#)」を参照してください。
- サンプル・データベースがインストールされていることと、Oracle Databaseのインストール時に必要なログイン・ユーザー名およびパスワードを取得済であることを確認します。[サンプル・スキーマおよびSQL*Plusについて](#)を参照してください。
- インストール時にデフォルトのデータベースを作成するか、接続するOracle Databaseの接続識別子をデータベース管理者から取得します。[データベースへの接続について](#)を参照してください。
- 接続するデータベースが起動していることを確認します。「[STARTUP](#)」コマンドを参照してください。

SQL*Plus日付書式

SQL*Plusのデフォルトの日付書式は、データベースのNLS_DATE_FORMATパラメータによって決定され、2桁の年号で表示する日付書式を使用することもできます。また、SELECT文の中で、SQLのTO_CHARファンクションまたはSQL*PlusのCOLUMN FORMATコマンドを使用して、レポート内での日付の表示方法を制御できます。

SQL*Plusコマンドラインの起動

通常、SQL*Plusの実行可能ファイルは\$ORACLE_HOME/binにインストールされます。通常、このディレクトリは、ご使用のオペレーティング・システムの環境変数PATHに含まれます。SQL*Plusを起動するには、\$ORACLE_HOME/binディレクトリに移動する必要があります。

次の各例では、データベース・アカウントのパスワードを入力するように求められます。

簡単な接続識別子を使用して、mymachineで実行されているMYDBデータベース内のHRスキーマに接続する例は、次のとおりです。

```
sqlplus hr@¥"//mymachine.mydomain:port/MYDB¥"
```

ネット・サービス名を使用する例は、次のとおりです。

```
sqlplus hr@MYDB
```

ネット・サービス名は、Oracle Namesなどいくつかの場所に格納できます。詳細は、『Oracle Database Net Services管理者ガイド』を参照してください。

ローカルのOracle Net tnsnames.oraファイル内に構成されたネット・サービス名を使用する場合は、tnsnames.oraファイルを含むディレクトリに環境変数TNS_ADMINを設定します。たとえば、UNIXで、/home/user1内にあるtnsnames.oraファイルによってネット・サービス名MYDB2が定義されている場合は、次のように設定します。

```
TNS_ADMIN=/home/user1
```

```
export TNS_ADMIN  
sqlplus hr@MYDB2
```

この例は、環境変数ORACLE_HOMEが設定されていて\$ORACLE_HOME/network/admin/tnsnames.oraまたはORACLE_HOME¥network¥admin¥tnsnames.oraファイルによってネット・サービス名MYDB3が定義されていることを前提としています。

```
sqlplus hr@MYDB3
```

環境変数TNS_ADMIN(UNIXの場合)またはLOCAL(Windowsの場合)には接続識別子を設定できます。これによって、SQL*PlusまたはSQL*Plus Instant Clientで接続が行われるたびに接続識別子を明示的に入力する必要がなくなります。このUNIXの例では、MYDB4という名前のデータベースに接続します。

```
TNS_ADMIN=/home/user1  
export TNS_ADMIN  
TWO_TASK=MYDB4  
export TWO_TASK  
sqlplus hr
```

SQL*Plusを起動し、デフォルトのデータベースに接続する手順

1. UNIX端末またはWindows端末を起動し、次のSQL*Plusコマンドを入力します。

```
sqlplus
```

2. プロンプトが表示された場合、Oracle Databaseのユーザー名およびパスワードを入力します。Oracle Databaseのユーザー名およびパスワードが不明な場合は、データベース管理者に問い合わせてください。
3. または、次の形式でSQL*Plusコマンドを入力します。

```
sqlplus username
```

パスワードを入力するように求められます。

4. SQL*Plusが起動し、デフォルトのデータベースに接続します。

これで、「SQL>」プロンプトで、SQL、PL/SQLおよびSQL*Plusの文やコマンドを入力して実行できるようになります。

例 - SQL*Plusを起動し、デフォルト以外のデータベースに接続する手順

UNIX端末またはWindows端末を起動し、次のSQL*Plusコマンドを入力します。

```
sqlplus username@connect_identifier
```

パスワードを入力するように求められます。

SQL*Plus Instant Clientの起動について

SQL*Plus Instant Clientは、SQL*Plusコマンドラインです。Oracle Databaseをインストールする必要はありません。使用方法については、「[SQL*Plusコマンドラインの起動](#)」を参照してください。

SQL*Plus Instant Clientは、データベースを含まないため、すべてのデータベース・サーバーから常にリモートの状態になります。データベースに接続するには、Oracle Net接続識別子を使用して、データベースを指定する必要があります。

TNS_ADMINが設定されていない場合は、オペレーティング・システム依存のディレクトリ内でtnsnames.oraが検索されます。この検索パスにはnetwork/admin/tnsnames.oraも含まれ、環境変数ORACLE_HOMEで指定されているディレクトリ内も検索されます。環境変数ORACLE_HOMEは、このためにのみ、SQL*Plus Instant Clientに設定します。Instant Clientアプリケーションの実行時にORACLE_HOMEを設定する場合は、既存のディレクトリに設定する必要があります。

他のデータベースへの接続について

既存のコマンドライン・セッションから、次の形式でCONNECTコマンドを入力します。

```
SQL> connect username@connect_identifier
```

パスワードを入力するように求められます。

サンプル・スキーマおよびSQL*Plusについて

Oracle Databaseにはサンプル・スキーマが含まれています。このマニュアルの例には、人事管理(HR)サンプル・スキーマのEMP_DETAILS_VIEWビューを使用します。このスキーマには、架空の会社の人事記録が含まれています。

EMP_DETAILS_VIEWビューの列の詳細を表示するには、次のように入力します。

```
DESCRIBE EMP_DETAILS_VIEW
```

サンプル・スキーマの詳細は、[『Oracle Databaseサンプル・スキーマ』](#)を参照してください。

サンプル表のロック解除

人事管理(HR)サンプル・スキーマは、デフォルトでOracle Databaseインストールの一部としてインストールされます。デフォルトでは、HRアカウントはロックされています。

HRサンプル・スキーマを使用するには、HRアカウントのロックを解除する必要があります。HRアカウントのロックを解除するには、SYSTEMユーザーとしてログインしてから次のコマンドを入力します。your_passwordは、ユーザーHRに対して定義するパスワードです。

```
ALTER USER HR IDENTIFIED BY your_password ACCOUNT UNLOCK;
```

HRアカウントのロック解除の詳細は、[『Oracle Databaseサンプル・スキーマ』](#)を参照してください。HRユーザーは、主にHRサンプル・スキーマへのアクセスを可能にします。また、このマニュアルの例を実行するために必要です。

データベース内のそれぞれの表は、特定のユーザーによって所有されています。このガイドの例を試すときに、サンプル表を独自にコピーして使用することもできます。HR表を独自にコピーする場合は、DBAに連絡するか、または[『Oracle Databaseサンプル・スキーマ』](#)を参照してください。また、UNIXの場合は、次のディレクトリにあるHR_MAIN.SQLスクリプトを使用しても、HR表を作成できます。

```
$ORACLE_HOME/demo/schema/human_resources/hr_main.sql
```

Windowsの場合は、次のディレクトリにあります。

```
ORACLE_HOME\demo\schema\human_resources\hr_main.sql
```

コマンドラインのSQL*PlusからHR表を作成するには、次の手順に従います。

1. Oracle Databaseアカウントのユーザー名およびパスワードについてDBAに問い合わせます。
2. SQL*Plusにログインします。
3. UNIXの場合、SQL*Plusプロンプトで次のコマンドを入力します。

```
SQL> @?/demo/schema/human_resources/hr_main.sql
```

Windowsの場合、SQL*Plusプロンプトで次のコマンドを入力します。

```
SQL> @?%demo%schema%human_resources%hr_main.sql
```

サンプル表を削除するには、HR_MAIN.SQLをHR_DROP.SQLと置き換えて、同じステップを実行します。

最初の間合せの実行

データベース・オブジェクト(たとえば、EMP_DETAILS_VIEWの列の詳細)を表示するには、次のようにDESCRIBEコマンドを入力します。

```
DESCRIBE EMP_DETAILS_VIEW
```

これによって、次の出力が表示されます。

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
JOB_ID	NOT NULL	VARCHAR2(10)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)
LOCATION_ID		NUMBER(4)
COUNTRY_ID		CHAR(2)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
JOB_TITLE	NOT NULL	VARCHAR2(35)
CITY	NOT NULL	VARCHAR2(30)
STATE_PROVINCE		VARCHAR2(25)
COUNTRY_NAME		VARCHAR2(40)
REGION_NAME		VARCHAR2(25)

列ヘッダーの名前を変更して、HRサンプル・スキーマ・ビューEMP_DETAILS_VIEWからデータを選択するには、次のように入力します。

```
COLUMN FIRST_NAME HEADING "First Name"
COLUMN LAST_NAME HEADING "Family Name"
SELECT FIRST_NAME, LAST_NAME
FROM EMP_DETAILS_VIEW
WHERE LAST_NAME LIKE 'K%';
```

これによって、次の出力が表示されます。

First Name	Family Name
Payam	Kaufling
Steven	King
Neena	Kochhar
Alexander	Khoo
Janette	King
Sundita	Kumar

SQL*Plusの終了について

SQL*Plusコマンドラインを終了するには、EXITを入力します。

第I部 SQL*Plusスタート・ガイド

第I部では、SQL*Plusを使用する場合に理解しておく必要がある情報を示します。コマンドライン・ユーザー・インタフェースについて説明し、構成情報およびSQL*Plusにログインして実行するために必要な情報を示します。

第I部の内容は次のとおりです。

- [SQL*Plusユーザー・インタフェース](#)
- [SQL*Plusの構成](#)
- [SQL*Plusの起動](#)

1 SQL*Plusユーザー・インタフェース

この章では、SQL*Plusコマンドライン・ユーザー・インタフェースについて説明します。次の項目が含まれます。

- [コマンドラインの画面について](#)
- [コマンドラインのフォントおよびフォント・サイズの変更](#)

1.1 コマンドラインの画面について

次の画像は、SQL*Plusコマンドライン・ユーザー・インタフェースを示しています。

SQL*Plusコマンドライン・ユーザー・インタフェースはすべてのオペレーティング・システムの標準です。

```
SQL*Plus: Release 18.0.0.0.0 Production on Sun Nov 19 17:04:22 2017  
Version 18.1.0.0.0
```

```
Copyright (c) 1982, 2017, Oracle. All rights reserved.
```

```
Last Successful login time: Sun Nov 19 2017 17:01:31 -07:00
```

```
Connected to:
```

```
Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production  
Version 18.1.0.0.0
```

```
SQL>|
```

SQL*Plusを起動すると、SQL*Plusプロンプトが表示される前に、日付、時刻、バージョン情報および著作権情報が表示されます。SQL*Plusコマンドラインのデフォルトのプロンプトは次のとおりです。

```
SQL>
```

1.2 コマンドラインのフォントおよびフォント・サイズの変更

Windowsでは、SQL*Plusコマンドライン・インタフェースで使用するフォントおよびフォント・サイズを設定するには、コマンド・プロンプトから「"コマンド プロンプト"のプロパティ」ダイアログ・ボックスを開きます。

コマンドライン・インタフェースのフォントおよびフォント・サイズを変更する手順

1. コマンドライン・インタフェースのタイトル・バーを右クリックします。
2. 「プロパティ」をクリックします。「ウィンドウのプレビュー」ボックスには、フォントとフォント・サイズの選択に応じたモニター上で
の現行ウィンドウの相対的なサイズが表示されます。「選択したフォント:」ボックスには、現行フォントのサンプルが表示
されます。
3. 「フォント」タブをクリックします。
4. 「サイズ」ボックスから、使用するフォント・サイズを選択します。ラスターフォントのサイズでは、ピクセル単位で幅と高さが
表示されます。TrueTypeフォントのサイズでは、ピクセル単位で高さが表示されます。
5. 「フォント」ボックスから、使用するフォントを選択します。
6. 太字フォントを使用するには、「ボールド フォント」チェック・ボックスを選択します。

コマンド・プロンプトのプロパティの詳細は、Windowsのヘルプを参照するか、「"コマンド・プロンプト"のプロパティ」ダイアログ・ボックスで「ヘルプ」をクリックしてください。

2 SQL*Plusの構成

この章では、SQL*Plusのコマンドライン環境の構成方法について説明します。内容は次のとおりです。

- [SQL*Plusの環境変数](#)
- [SQL*Plusの構成](#)

2.1 SQL*Plusの環境変数

ここで説明する環境変数は、SQL*Plusが使用するファイルの位置またはパスを指定します。SQL*Plusの動作に影響する他の環境変数については、『Oracle Database管理者リファレンス』を参照してください。

表2-1 SQL*Plusに影響するパラメータまたは環境変数

パラメータまたは変数	説明
LD_LIBRARY_PATH	UNIX および Linux 上のライブラリの検索に使用するパスを指定する環境変数。一部のオペレーティング・システムでは別の名前(Apple Mac OS での DYLD_LIBRARY_PATH、IBM/AIX-5L での LIBPATH、HP-UX での SHLIB_PATH など)である場合があります。Windows オペレーティング・システムでは使用できません。 例 \$ORACLE_HOME/lib
LOCAL	接続文字列を指定する Windows 環境変数。UNIX の TWO_TASK と同じ機能を実行します。
NLS_LANG	グローバリゼーション機能を指定する環境変数。 例 american_america.utf8
ORACLE_HOME	SQL*Plus のインストール場所を指定する環境変数。SQL*Plus により、メッセージ・ファイルの配置場所を指定するためにも使用されます。 例: d:\oracle\10g /u01/app/oracle/product/v10g
ORA_EDITION	使用するデータベース・エディションを指定する環境変数。CONNECT または SQLPLUS コマンドのオプション(edition=value)を使用してエディションを指定した場合は、ORA_EDITION ではなくこのオプションが使用されます。CONNECT または SQLPLUS コマンドのオプションあるいは ORA_EDITION でエディションを指定しない場合、SQL*Plus はデフォルトのエディションに接続します。 ORA_EDITION が設定されると、セッション内で後続の STARTUP コマンドが ORA-38802 エラーとなります。これを修正するには、ORA_EDITION を設定解

パラメータまたは変数	説明
	除し、再度データベースに接続してこれを停止し、さらにもう一度データベースを起動する必要があります。
ORA_NLS10	SQL*Plus リリース 2(10.2)の NLS データおよびユーザーのブート・ファイルの位置を指定する環境変数。デフォルトの位置は\$ORACLE_HOME/nls/data です。 Oracle9i および Oracle10g の両方を使用するシステム、またはアップグレードされたシステムの場合は、10g 用に ORA_NLS10、9i 用に ORA_NLS33 を設定する必要があります。Oracle9i では、デフォルトの NLS の位置は \$ORACLE_HOME/common/nls/admin/data でした。
ORACLE_PATH	SQL スクリプトの位置を指定する環境変数。SQL*Plus が ORACLE_PATH でスクリプト・ファイルを検出できない場合、または ORACLE_PATH が設定されていない場合は、現行の作業ディレクトリが検索されます。 Windows では使用しません。
ORACLE_SID	データベース・インスタンスを指定する環境変数(オプション)。
PATH	実行可能ファイルおよび Windows の DLL を検索するパスを指定する環境変数。通常は、ORACLE_HOME/bin を含めます。
SQLPATH	SQL スクリプトの位置を指定する環境変数または Windows レジストリ・エントリ。SQL*Plus は、 <i>login.sql</i> などの SQL スクリプトを、SQLPATH で指定されたディレクトリ内で検索します。SQLPATH は、コロンで区切られたディレクトリのリストです。UNIX 版のインストールでは、デフォルト値は設定されません。 Windows 版では、SQLPATH がインストール時にレジストリ・エントリに定義されます。SQLPATH レジストリ・エントリの詳細は、「 SQLPATH レジストリ・エントリ 」を参照してください。
TNS_ADMIN	tnsnames.ora ファイルの位置を指定する環境変数。この環境変数を指定しない場合は、\$ORACLE_HOME/network/admin が使用されます。 例 h:\network /var/opt/oracle
TWO_TASK	接続文字列を指定する UNIX 環境変数。データベースを指定しない接続は、TWO_TASK に指定したデータベースに接続されます。 例 TWO_TASK=MYDB export TWO_TASK sqlplus hr

パラメータまたは変数	説明
	これは、次のように解釈されます。
	sqlplus hr@MYDB

2.1.1 SQLPATHレジストリ・エントリ

SQLPATHレジストリ・エントリは、SQLスクリプトの位置を指定します。SQL*Plusでは、カレント・ディレクトリ、次にSQLPATHレジストリ・エントリで指定されたディレクトリ、SQLPATHディレクトリのサブディレクトリの順に、SQLスクリプトが検索されます。

HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥HOME0レジストリ・サブキー(または、ORACLE_HOMEに関連付けられているHOME n ディレクトリ)には、SQLPATHレジストリ・エントリが含まれます。SQLPATHは、デフォルト値ORACLE_HOME¥DBSを使用して作成されます。SQLPATHの有効な値として、任意のドライブの任意のディレクトリを指定できます。

SQLPATHレジストリ・エントリを設定する場合に、ディレクトリをセミicolon(;)で区切って複数指定することもできます。次に例を示します。

```
c:¥oracle¥ora12¥database;c:¥oracle¥ora12¥dbs
```

SQLPATHレジストリ・エントリの編集方法の説明は、レジストリ・エディタのヘルプ・システムを参照してください。

2.2 SQL*Plusの構成

各セッションで同じ設定を使用するように、SQL*Plus環境を設定できます。

この設定には、次の2つのオペレーティング・システム・ファイルを使用します。

- *glogin.sql*などのサイト・プロファイル・ファイル。サイト全体の設定用です。
- *login.sql*などのユーザー・プロファイル・ファイル。ユーザー固有の設定値を設定します。

これらのファイルの正確な名前は、システムによって異なります。

ノート:



サイト・プロファイル・ファイルとユーザー・プロファイル・ファイルは、SQLPLUS または CONNECT コマンドで Oracle Database に正常に接続した後で、あるいは/NOLOG を指定したときに実行されます。ALTER SESSION SET CONTAINER を使用して別の PDB に切り替えると、サイト・プロファイル・ファイルとユーザー・プロファイル・ファイルは実行されません。

一部の権限での接続では、サイト・プロファイルまたはユーザー・プロファイルにSET SERVEROUTPUTまたはSET APPINFO コマンドが入力されていると、エラーが生成される場合があります。

次の表に、プロファイル・スクリプトを示し、コマンドライン・ユーザー・インタフェースに影響する一部のコマンドと設定を示します。

表2-2 SQL*Plusユーザー・インタフェースの設定に影響するプロファイル・スクリプト

スクリプトの内容	コマンドラインで実行されるタイミング
----------	--------------------

スクリプトの内容	コマンドラインで実行されるタイミング
サイト・プロファイル (glogin.sql) SQL*Plus スクリプトに含めることができる内容(システム変数設定や DBA が実装する他のグローバル設定など)はすべて含めることができます。	SQLPLUS コマンドまたは CONNECT コマンドから Oracle Database への正常な接続後 /NOLOG が指定された場合
ユーザー・プロファイル (login.sql) SQL*Plus スクリプトに含めることができる内容はすべて含めることができます。ただし、設定はユーザー・セッションにのみ適用可能です。	サイト・プロファイルの直後

表2-3 SQL*Plusユーザー・インタフェースの設定に影響するプロファイル・スクリプトのコマンド

プロファイル・スクリプト内のコマンド	コマンドラインに影響する操作
SET SQLPLUSCOMPAT[IBILITY]{x.y[.z]} 「 SQL*Plus 互換性マトリックス 」を参照。	DBA がこのサイトで希望する動作を実現するために、SQL*Plus の互換モードを設定した場合
SQLPLUS コマンド COMPATIBILITY オプション	SET SQLPLUSCOMPATIBILITY コマンドが実行された場合(SQLPLUS コマンドの COMPATIBILITY オプション付きで設定された場合以外)
SQLPLUS コマンド RESTRICT オプション	RESTRICT オプションを 3 に設定して SQL*Plus を起動すると、ユーザー・プロファイル・スクリプトが読み取られません。

2.2.1 サイト・プロファイル

サイト・プロファイル・スクリプトは、インストール中に作成されます。これは、SQL*Plusコマンドライン接続に対するサイト全体の動作を構成するために、データベース管理者が使用します。インストール中にインストールされたサイト・プロファイル・スクリプトは、空のスクリプトです。

サイト・プロファイル・スクリプトは、通常、glogin.sqlという名前です。ユーザーが、SQL*Plusセッションを開始し、Oracle Databaseとの接続が正常に確立されるたびに、SQL*Plusはこのスクリプトを実行します。

サイト・プロファイルを使用すると、DBAは、特定のSQL*Plusをインストールするすべてのユーザーに対して、サイト全体にわたるSQL*Plus環境のデフォルト値を設定できます。

ユーザーはサイト・プロファイルに直接アクセスできません。

2.2.1.1 デフォルトのサイト・プロファイル・スクリプト

サイト・プロファイル・スクリプトは、UNIXでは\$ORACLE_HOME/sqlplus/admin/glogin.sql、WindowsではORACLE_HOME¥sqlplus¥admin¥glogin.sqlにあります。この場所にすでにサイト・プロファイルが存在する場合は、

SQL*Plusインストール時に上書きされます。SQL*Plusが削除されると、サイト・プロファイル・スクリプトも削除されます。

2.2.2 ユーザー・プロファイル

SQL*Plusのコマンドライン接続の場合、SQL*Plusはユーザー・プロファイル・スクリプトもサポートします。ユーザー・プロファイルは、サイト・プロファイルの後に実行され、ユーザー・セッションの特別なカスタマイズを可能にします。ユーザー・プロファイル・スクリプトは、通常、login.sqlという名前です。SQL*Plusでは、ORACLE_PATH環境変数で指定されるディレクトリを検索します。SQL*Plusでは、このコロンで区切られたディレクトリとそれらのサブディレクトリのリストが検索される際、ディレクトリとサブディレクトリがリストされている順で検索されます。



ノート:

SQL*Plus では、カレント・ディレクトリの login.sql を検索しません。

すべてのSQLコマンド、PL/SQLブロックまたはSQL*Plusコマンドをユーザー・プロファイルに追加できます。SQL*Plusを起動すると、自動的にユーザー・プロファイルが検索され、ユーザー・プロファイル内のコマンドが実行されます。

2.2.2.1 LOGINファイルの変更

LOGINファイルは、他のすべてのスクリプトと同様に、変更が可能です。次のユーザー・プロファイル・スクリプトの例に、ユーザーが追加できる変更の一部を示します。

```
-- login.sql
-- SQL*Plus user login startup file.
--
-- This script is automatically run after glogin.sql
--
-- To change the SQL*Plus prompt to display the current user,
-- connection identifier and current time.
-- First set the database date format to show the time.
ALTER SESSION SET nls_date_format = 'HH:MI:SS';

-- SET the SQLPROMPT to include the _USER, _CONNECT_IDENTIFIER
-- and _DATE variables.
SET SQLPROMPT "_USER'@'_CONNECT_IDENTIFIER _DATE> "

-- To set the number of lines to display in a report page to 24.
SET PAGESIZE 24

-- To set the number of characters to display on each report line to 78.
SET LINESIZE 78

-- To set the number format used in a report to $99,999.
SET NUMFORMAT $99,999
```

関連項目:

- SQL*PlusのLOGINファイルに設定できる各種のSETコマンド変数の詳細は、「[SET](#)」コマンドを参照してください。
- 事前定義変数の詳細は、「[事前定義変数の使用について](#)」を参照してください。

2.2.3 SQL*Plusシステム変数の格納およびリストア

STOREコマンドを使用すると、コマンドラインから、現行のSQL*Plusのシステム変数をスクリプトに格納できます。変数を変更した場合は、このスクリプトを実行して元の値に戻すことができます。この機能は、システム変数を変更するレポートを実行した後でシステム変数の値をリセットする場合に便利です。また、SQL*Plusを起動するたびにこれらのシステム変数が設定されるように、このスクリプトをユーザー・プロファイル・スクリプトに含めることもできます。

すべてのシステム変数の現行の設定を格納するには、次のように入力します。

```
STORE SET file_name
```

ファイル名とファイル拡張子を入力するか、またはデフォルトの拡張子.SQLを使用する場合は、ファイル名のみを入力します。[SET SUF\[FIX\] {SQL | text}](#)コマンドを使用して、デフォルトのファイル拡張子を変更することもできます。

2.2.3.1 システム変数のリストア

格納済のシステム変数をリストアするには、次のように入力します。

```
START file_name
```

ファイルの拡張子がデフォルト([SET SUF\[FIX\] {SQL | text}](#)コマンドで指定したもの)の場合は、ファイル名にピリオドおよび拡張子を追加する必要はありません。

@(アットマーク)または@@(二重アットマーク)コマンドを使用して、スクリプトを実行することもできます。

```
Created file plusenv
```

この時点で、任意のシステム変数の値を変更できます。

```
SHOW PAGESIZE
```

```
PAGESIZE 24
```

```
SET PAGESIZE 60
```

```
SHOW PAGESIZE
```

```
PAGESIZE 60
```

その後、システム変数の元の値をスクリプトからリストアできます。

```
START plusenv
```

```
SHOW PAGESIZE
```

```
PAGESIZE 24
```

例2-1 SQL*Plusシステム変数の格納およびリストア

SQL*Plusシステム変数の現行の値を新しいスクリプトplusenv.sqlに格納するには:

```
STORE SET plusenv
```

2.2.4 コマンドライン・ヘルプのインストールについて

通常、コマンドライン・ヘルプは、Oracle Databaseのインストール中にインストールされます。インストールされていない場合は、データベース管理者がSQL*Plusコマンドライン・ヘルプ表を作成し、指定したSQLスクリプトをSQL*Plusから実行することによって、SQL*Plusヘルプ・データを作成した表に移入できます。

また、データベース管理者はSQL*PlusからSQLスクリプトを実行して、SQL*Plusコマンドライン・ヘルプ表を削除することもできます。

SQL*Plusヘルプをインストールまたは削除する前に、次のことを確認してください。

- SQL*Plusがインストールされている。
- 環境変数ORACLE_HOMEが設定されている。
- 次のSQL*Plusヘルプ・スクリプト・ファイルがある。
 - HLPBLD.SQL - 新しいヘルプ表の削除および作成
 - HELPDROP.SQL - 既存のヘルプ表の削除
 - HELPUS.SQL - ヘルプ・データへのヘルプ表の移入

2.2.4.1 hlpbld.sqlスクリプトを実行したコマンドライン・ヘルプのインストール

コマンドライン・ヘルプをロードするために提供されているSQLスクリプトHLPBLD.SQLを実行します。

1. SYSTEMユーザーとして次のように入力し、SQL*Plusにログインします。

```
SQLPLUS SYSTEM
```

SYSTEMユーザーに対して定義したパスワードの入力を求めるプロンプトが表示されます。

2. UNIXでは、SQL*Plusから次のように入力してSQLスクリプトHLPBLD.SQLを実行します。

```
@$ORACLE_HOME/sqlplus/admin/help/hlpbld.sql helpus.sql
```

Windowsでは、SQL*Plusから次のように入力してSQLスクリプトHLPBLD.SQLを実行します。

```
@%ORACLE_HOME%\sqlplus\admin\help\hlpbld.sql helpus.sql
```

HLPBLD.SQLスクリプトによって、ヘルプ表が作成されロードされます。

2.2.4.2 helpdrop.sqlスクリプトを実行したコマンドライン・ヘルプの削除

コマンドライン・ヘルプを削除するために提供されているSQLスクリプトHELPDROP.SQLを実行します。

1. SYSTEMユーザーとして次のように入力し、SQL*Plusにログインします。

```
SQLPLUS SYSTEM
```

SYSTEMユーザーに対して定義したパスワードの入力を求めるプロンプトが表示されます。

2. UNIXでは、SQL*Plusから次のように入力してSQLスクリプトHELPDROP.SQLを実行します。

```
@$ORACLE_HOME/sqlplus/admin/help/helpdrop.sql
```

Windowsでは、SQL*Plusから次のように入力してSQLスクリプトHELPDROP.SQLを実行します。

```
@%ORACLE_HOME%\sqlplus\admin\help\helpdrop.sql
```

HELPDROP.SQLスクリプトは、ヘルプ表を削除してから切断します。

2.2.5 Oracle Net Servicesの構成について

デフォルト以外のデータベースに接続する場合、同一のコンピュータか別のコンピュータかに関係なく、Oracle Netがインストールされ、データベース・リスナーが構成されて実行中である必要があります。Oracle Net Servicesは、SQL*Plusによって使用

されます。

Oracle Net Servicesおよびデータベース・リスナーは、Oracle Databaseのインストール中に、デフォルトでインストールされます。Oracle Netのインストールおよび構成の詳細は、<http://www.oracle.com/technology/documentation>にあるOracle Databaseのドキュメントを参照してください。

3 SQL*Plusの起動

この章では、SQL*Plusの起動方法、ログイン方法、データベースへの接続方法、ヘルプの使用方法および終了方法について説明します。

ここでは、次の項目について説明します。

- [ユーザー名およびパスワードのログイン](#)
- [データベースへの接続について](#)
- [SQL*Plusの起動について](#)
- [SQL*Plusコマンドラインの終了について](#)
- [SQLPLUSプログラムの構文](#)

3.1 ユーザー名およびパスワードのログイン

SQL*Plusの起動時、Oracle Databaseスキーマへのログインにユーザー名およびパスワードが必要です。ユーザー名およびパスワードによって、Oracle Databaseスキーマの認証されたユーザーとして識別されます。

データベース管理者(DBA)には、必要な権限を付与してデータベース・アカウントを作成し、アカウントにアクセスするためのユーザー名とパスワードを提供する責任があります。

Oracle Databaseのインストール中にデフォルト・ログインが作成され、対応するパスワードの入力を求めるプロンプトが表示されます。作成されるデフォルト・ログインのユーザー名の一部には、次のものがあります。

- SYS
- SYSTEM
- HR

Oracle Databaseのインストール中に、ユーザーが作成されたことを示すメッセージが表示されます。

デフォルトのログインの詳細は、「[Oracle Databaseユーザーのタイプ](#)」を参照してください。

一度ログインすると、CONNECTコマンドを使用して別のユーザー名で接続できるようになります。ユーザー名およびパスワードは、そのデータベースに対して有効である必要があります。たとえば、パスワードFOXを使用してユーザー名TODDでデフォルトのデータベースに接続するには、次のように入力します。

```
CONNECT TODD
```

パスワードFOXを入力するように求められます。

コマンドライン・ユーザー・インタフェースでユーザー名およびパスワードを省略した場合、それらの入力を求めるSQL*Plusプロンプトが表示されます。CONNECTコマンドは、最初にユーザーを現行のデータベースから切断するため、CONNECTコマンドに無効なユーザー名およびパスワードを使用した場合、どのデータベースにも接続していない状態になります。

データベースへの接続の際にアカウントの有効期限が切れたユーザーとしてログオンまたは接続した場合は、接続の前にパスワードを変更するように求めるプロンプトが表示されます。

データベースへの接続時にアカウントがロックされている場合は、メッセージが表示され、アカウントのロックがDBAによって解除されるまで、そのユーザーとして接続できません。

DISCONNECTコマンドを使用すると、SQL*Plusを終了せずに、データベースから切断できます。

3.1.1 セキュアな外部パスワード・ストア

アプリケーションでデータベースへの接続にパスワード資格証明が使用される大規模なデプロイメントでは、コマンドラインを使用するかわりに、そのような資格証明をクライアント側のOracle Walletに格納できます。Oracle Walletは、資格証明の認証および署名の格納に使用されるセキュアなソフトウェア・コンテナです。

クライアント側のOracleウォレットにデータベース・パスワード資格証明を格納することで、ユーザー名とパスワードをアプリケーション・コード、バッチ・ジョブまたはスクリプトに埋め込む必要がなくなります。この結果、スクリプトやアプリケーション・コードに記述したパスワードが外部にさらされる危険性が低くなり、ユーザー名とパスワードを変更するたびにコードを変更する必要がないため、メンテナンスが簡素化されます。また、アプリケーション・コードを変更する必要がないため、これらのユーザー・アカウントのパスワード管理ポリシーをさらに簡単に規定できるようになります。

外部パスワード・ストアを使用するようにクライアントを構成すると、アプリケーションでは、次の構文を使用してパスワード認証を使用しているデータベースに接続できます。

```
CONNECT /@database_alias
```

このCONNECT文では、データベース・ログイン資格証明を指定する必要はありません。かわりに、データベース・ログイン資格証明はクライアントのウォレットで検索されます。

関連項目:

セキュアな外部パスワード・ストアを使用するようにクライアントを構成する方法およびそのパスワード・ストア内の資格証明を管理する方法については、『[Oracle Database管理者ガイド](#)』を参照してください。

3.1.2 パスワードの期限切れ

コマンドライン・インタフェースで、入力したパスワードが期限切れの場合は、ログイン時にパスワードの変更を要求するSQL*Plusプロンプトが表示されます。パスワードを正常に変更した後は、ログインできます。

3.1.3 パスワードの変更について

コマンドライン・ユーザー・インタフェースで、PASSWORDコマンドを使用してパスワードを変更できます。[\[PASSWORD\]](#)を参照してください。

3.2 データベースへの接続について

データベース内のデータの問合せまたは変更を実行するには、Oracle Database(インスタンス)に接続する必要があります。デフォルトのデータベースおよびネットワーク上の他の利用可能なデータベースに接続できます。ネットワークを介して別のデータベースに接続するには、両方のデータベースにOracle Netが構成され、互換性のあるネットワーク・ドライバを備えている必要があります。デフォルト以外のデータベースに接続するには、接続識別子またはネット・サービス名を入力してください。

接続識別子またはネット・サービス名の入力方法は次のとおりです。

- コマンドライン・セッションの開始時に、[SQLPLUSプログラムの構文](#)への引数として入力する。
- カレント・セッションから[CONNECT](#)コマンドへの引数として入力する。使用方法の詳細は、『[Oracle Database管理者ガイド](#)』の[CDB内のコンテナへのアクセスに関する項](#)を参照してください。

3.2.1 ネット・サービス名

DBAは、使用するデータベースを作成して、そのデータベースのネット・サービス名を`tnsnames.ora`ファイルに定義します。

`tnsnames.ora`ファイルにネット・サービス名を定義する構文は次のとおりです。

```
net_service_name=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=host) (PORT=port) )
  (CONNECT_DATA=
    (SERVICE_NAME=service_name) ) )
```

ネット・サービス名(別名)を使用するには、SQL*Plusが実行されているマシンの`tnsnames.ora`ファイルに、このサービス名のエントリが存在する必要があります。接続識別子を使用する場合、`tnsnames.ora`ファイルのエントリは必要ありません。

例3-1 salesデータベースの`tnsnames.ora`エントリ

```
SALES1 =
(DESCRIPTION =
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521) )
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com) ) )
```

例3-2 ネット・サービス名を使用したsalesデータベースへのコマンドライン・セッションの開始

```
SQLPLUS hr@SALES1
```

データベース接続およびネット・サービス名の定義の詳細は、「[構成パラメータ](#)」および「[Oracle Net Servicesの構成および管理](#)」を参照してください。

3.2.2 完全な接続識別子

ご使用の構成に応じて、完全な接続識別子を次の構文で使用できます。

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=host) (PORT=port) )
  (CONNECT_DATA=
    (SERVICE_NAME=service_name) ) )
```

`SERVICE_NAME`は、データベース作成中に入力したグローバル・データベース名です。これは、データベース名をドメイン名と結合したものです。たとえば、`SERVICE_NAME sales.us.acme.com`では、`sales`がデータベース名で、`us.acme.com`がドメインです。

`INSTANCE_NAME`は、作成中に指定したデータベース・インスタンス名です。デフォルトは、データベース作成中に入力したSIDです。

Oracleシステム識別子(SID)によって、特定のOracleリリース8.0データベース・インスタンスが識別されます。

オプションで、`SERVICE_NAME`句のかわりに`INSTANCE_NAME`を使用できます。

Oracleリリース8.0以下のデータベースに接続する場合は、`SERVICE_NAME`のかわりにSIDを使用します。

例3-3 SALES1の完全な接続識別子

```
SQLPLUS hr@¥" (DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521) )
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com) ) )¥"
```

3.2.3 簡単な接続識別子

省略した簡単な接続識別子の構文は次のとおりです。

```
[//[ ]host[:port] [/service_name]
```

例3-4 簡単な接続識別子を使用したsalesデータベースへのコマンドライン・セッションの開始

```
sqlplus hr@¥"sales-server:1521/sales.us.acme.com¥"
```

例3-5 簡単な接続識別子を使用したsalesデータベースへの接続

パスワードを省略した場合、接続文字列を引用符で囲む必要があります。

```
connect hr@"sales-server:1521/sales.us.acme.com"
```

簡単な接続識別子は、完全な接続識別子やネット・サービス名を使用できる状況であれば、どこでも使用できます。また、簡単な構文は単純で、*tnsnames.ora* エントリは必要ありません。

ノート:



Oracle Database リリース 19c 以降、簡易接続構文は機能が向上し、簡易接続プラスと呼ばれています。詳細は、[簡易接続プラス](#)に関する項を参照してください。

3.2.4 /NOLOGを使用した接続なしのセッション

コマンドライン・インタフェースでは、データベースに接続せずにSQL*Plusを起動できます。この機能は、データベース管理作業の実行、トランSPORTABLE・スクリプトの書込み、あるいはSQL*Plusの編集コマンドを使用したスクリプトの書込みまたは編集に有効です。

SQLPLUSコマンドで/NOLOG引数を使用して、接続なしのコマンドライン・セッションを開始できます。SQL*Plusの起動後は、CONNECTコマンドでデータベースに接続できます。

例3-6 /NOLOGを使用した接続なしのSQL*Plusセッションの開始

```
SQLPLUS /NOLOG
```

3.3 SQL*Plusの起動について

Remote Oracle Databaseに接続する場合は、Oracle Netソフトウェアがインストールされ、正常に動作することを確認してください。詳細は、「[Oracle Net Servicesのテストおよびトラブルシューティング](#)」を参照してください。

SQL*Plusのコマンドライン・セッションの開始時、そのセッションでCONNECTコマンドを実行すると、サイト・プロファイル *glogin.sql* およびユーザー・プロファイル・ファイル *login.sql* が、次の時点で処理されます。

- SQL*Plusが起動して接続した後で、最初のプロンプトが表示される前。
- SQL*Plusが起動して接続した後で、コマンドラインに指定したスクリプトが実行される前。
- コマンドラインで/NOLOGを指定し、かつ接続が確立していない場合に最初のプロンプトが表示される前。

最初にサイト・プロファイル *glogin.sql* が処理され、次にユーザー・プロファイル *login.sql* が処理されます。

3.3.1 SQL*Plusコマンドラインの起動について

SQL*Plusを使用する前に、SQL*Plusの起動および終了方法を理解しておく必要があります。

1. コンピュータにSQL*Plusがインストールされていることを確認します。
2. オペレーティング・システムにログオンします(必要な場合)。
3. SQLPLUSコマンドを入力し、[Return]を押します。

ノート:

一部のオペレーティング・システムでは、コマンドを小文字で入力する必要があります。そのようなシステムをご使用の場合は、SQLPLUS コマンドを小文字で入力します。



SQLPLUS

次のように、リリース番号、現在の日付および著作権情報が表示され、ユーザー名の入力を求めるプロンプトが表示されます(システム上に表示されるテキストは少し異なる場合があります)。

```
SQL*Plus: Release 19.0.0.0.0 Production on Tue Nov 14 22:12:47 2018
```

```
Version 19.1.0.0.0 Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

4. ユーザー名を入力し、[Return]を押します。「パスワードを入力してください」というプロンプトが表示されます。
5. パスワードを入力し、再度[Return]を押します。ユーザーを保護するため、パスワードは画面に表示されません。

ユーザー名およびパスワードを入力するプロセスは、ログインと呼ばれます。接続先のOracle Databaseのリリース、PL/SQLなどの使用可能なツール製品のリリースおよびローカル時間形式による最終ログオン時間が表示されます。

```
SQL*Plus: Release 19.0.0.0.0 Production on Tue Nov 14 22:12:47 2018  
Version 19.1.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
Last Successful login time: Wed Nov 01 2018 23:35:38 -07:00
```

```
Connected to:
```

```
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - 64bit Production
```

続いて、次のようなコマンド・プロンプトが表示されます。

```
SQL>
```

このSQL*Plusコマンド・プロンプトは、SQL*Plusがコマンド入力可能な状態であることを示します。

SQL*Plusが起動しない場合は、問題の修正に有効なメッセージが表示されます。

例3-7 SQL*Plusの起動

ここでは、SQL*Plusの起動方法を示します。

3.3.2 コマンドライン・ヘルプの表示について

SQL*Plusコマンドのコマンドライン・ヘルプにアクセスするには、SQLコマンド・プロンプトまたはSQL*Plusの「ワークスペースの入力」領域でHELPまたは?と入力し、続けてコマンド名を入力します。詳細は、「[HELP](#)」コマンドを参照してください。次に例を

示します。

HELP ACCEPT

SQL*Plusコマンドのリストを表示するには、HELPの後にTOPICSまたはINDEXを入力します。HELP TOPICSでは、SQL*Plusコマンドが1列に表示されます。HELP INDEXでは、SQL*Plusコマンドが標準画面内に収まるように4列に表示されます。次に例を示します。

HELP INDEX

3.4 SQL*Plusコマンドラインの終了について

無効なユーザー名またはパスワードを指定したか、あるいはその他の理由でSQL*Plusへログインできない場合は、EXIT FAILUREコマンドと同じエラー状態が戻されます。詳細は、「[EXIT](#)」コマンドを参照してください。

SQL*Plusでの作業が終了し、オペレーティング・システムに戻る場合は、SQL*PlusのプロンプトでEXITまたはQUITを入力するか、またはファイルを終了するキー（UNIXの場合は[Ctrl]を押しながら[D]、Windowsの場合は[Ctrl]を押しながら[Z]）を入力します。

オペレーティング・システムのプロンプトに戻る前に、切断したOracle Databaseのリリース番号、およびSQL*Plusを介して使用可能なツール製品のリリース番号が表示されます。

3.5 SQL*Plusプログラムの構文

SQL*Plusコマンドラインを起動するには、オペレーティング・システムのプロンプトで、次のようにSQLPLUSコマンドを使用します。

```
SQLPLUS [ [Options] [Logon]/NOLOG] [Start] ]
```

[Options](#)の構文は、次のとおりです。

```
-H[ELP] | -V[ERSION]  
| [[-C[OMPATIBILITY] {x.y.z} [-F[ast]] [-M[ARKUP] markup_option] [-L[OGON]]  
  [-NOLOGINTIME] [-R[ESTRICT] {1|2|3}] [-S[ILENT]]]
```

また、*markup_option*の構成は、次のとおりです。

- *csv_option*
- *html_option*

*csv_option*の構文は、次のとおりです。

```
CSV {ON|OFF} [DELIMI[TER] character] [QUOTE {ON|OFF}]
```

*html_option*の構文は、次のとおりです。

```
HTML {ON|OFF} [HEAD text] [BODY text] [TABLE text] [ENTMAP {ON|OFF}] [SPOOL {ON|OFF}] [PRE[FORMAT]  
{ON|OFF}]
```

[Logon](#)の構文は、次のとおりです。

```
{username[/password] [@connect_identifier] | / }  
[AS {SYSASM|SYSBACKUP|SYSDBA|SYSDG|SYSOPER|SYSRAC|SYSKM}] [edition=value]
```

[Start](#)の構文は、次のとおりです。

```
@{url|file_name.[ext]} [arg ...]
```


警告:



パスワードをプレーン・テキストで指定すると、セキュリティ上の危険があります。パスワードを省略し、パスワードの入力を求めるプロンプトが表示された場合にのみ入力することで、この危険を回避できます。

Logonの入力は、オプションです。Logonを指定しないでStartを指定した場合、SQL*Plusでは、スクリプトの1行目に有効なログインが含まれているとみなされます。StartおよびLogonのいずれも指定しない場合、SQL*Plusに、ログイン情報の入力を求めるプロンプトが表示されます。

3.5.1 オプション

次の項で、SQLPLUSコマンドのオプションについて説明します。

3.5.1.1 HELPオプション

```
-H[ELP]
```

SQLPLUSコマンドでの使用方法および構文を表示し、オペレーティング・システムに制御を戻します。

3.5.1.2 VERSIONオプション

```
-V[ERSION]
```

SQL*Plusの現行のバージョンおよびレベル番号を表示し、オペレーティング・システムに制御を戻します。

3.5.1.3 COMPATIBILITYオプション

```
-C[OMPATIBILITY] {x.y[.z]}
```

SQLPLUSCOMPATIBILITYシステム変数の値を、x.y[.z]で指定したSQL*Plusのリリースに設定します。xはバージョン番号を、yはリリース番号を、zはアップデート番号を示します。たとえば、9.0.1、10.2などです。詳細は、[SET SQLPLUSCOMPAT\[IBILITY\] {x.y\[.z\]}](#)システム変数を参照してください。

3.5.1.4 LOGONオプション

```
-L[OGON]
```

初期接続に失敗した場合にユーザー名またはパスワードを求めるプロンプトを表示しないように指定します。これは、実行結果が明確に成功または失敗のいずれかになるオペレーティング・システム・スクリプトで有効です。また、データベース・サーバーが実行されていないときに、接続の詳細を求めるプロンプトが再度表示されないようにする場合にも有効です。

3.5.1.5 FASTオプション

```
-F[ast]
```

FASTオプションにより、一般パフォーマンスが向上します。このコマンドライン・オプションにより、次のデフォルト設定の値が変更されます。

- ARRAYSIZE = 100
- LOBPREFETCH = 16384

- PAGESIZE = 50000
- ROWPREFETCH = 2
- STATEMENTCACHE = 20

3.5.1.6 MARKUPオプション

-M[ARKUP]

問合せまたはスクリプトでMARKUPオプションを使用して、HTMLまたはCSV (文字で区切られた値)形式で出力を生成できます。

MARKUPは現在、HTML 4.0 TransitionalおよびCSV形式をサポートしています。

SQLPLUS -MARKUPを使用して、出力をHTMLまたはCSV形式で生成できます。

ノート:



ご使用のオペレーティング・システムによっては、SQL*PLUS コマンドの *markup_option* 句全体を引用符で囲む必要があります。

HTML出力の場合、SQLPLUS -MARKUP HTML ONまたはSQLPLUS -MARKUP HTML ON SPOOL ONを使用して、スタンドアロンWebページを生成します。SQL*Plusによって、<HTML>および<BODY>タグでカプセル化された完全なHTMLページが自動的に生成されます。SPOOL OFFを実行するか、またはSQL*Plusを終了すると、スプール・ファイルのHTMLタグが閉じます。

-SILENTおよび-RESTRICTコマンドライン・オプションは、-MARKUPと組み合わせて使用すると有効です。

MARKUP HTML ONを使用すると、<PRE>タグまたはHTML表のいずれかにHTML出力を生成できます。表への出力では、標準のHTMLタグ<TABLE>、<TR>および<TD>が使用され、問合せ結果の行および列が自動的にエンコードされます。HTMLオプションの設定がONの場合、デフォルトではHTML表に出力されます。PREFORMAT ONを設定すると、HTMLタグ<PRE>を使用して出力を生成できます。

CSV出力の場合、SQLPLUS -MARKUP CSV ONを使用して、出力をCSV形式で生成します。DELIMITERオプションを使用して、デリミタ文字を指定できます。QUOTE OFFを使用して、引用符なしでテキストを出力することもできます。

MARKUPオプションの状態を表示するには、SHOW MARKUPコマンドを使用します。

SQLPLUS -MARKUPコマンドには、SET MARKUPコマンドと同じ機能があります。これらのオプションについては、この項で説明しています。SET MARKUPコマンドの詳細は、「[SET](#)」コマンドを参照してください。

CSV {ON|OFF}

CSVは、生成される出力タイプをCSVに指定するMARKUPの必須引数です。CSV引数ONおよびOFFは、CSV出力を生成するかどうかを指定します(オプション)。デフォルトはOFFです。CSV出力のONおよびOFFは、セッション中、必要に応じて切り替えることができます。

HTML {ON|OFF}

HTMLは、生成される出力タイプをHTMLに指定するMARKUPの必須引数です。HTML引数ONおよびOFFは、HTML出力を生成するかどうかを指定します(オプション)。デフォルトはOFFです。

MARKUP HTML ONを使用すると、指定したMARKUPオプションに従ってHTML出力が生成されます。

HTML出力のONおよびOFFは、セッション中、必要に応じて切り替えることができます。

HEAD *text*

HEAD *text*オプションでは、<HEAD>タグの内容を指定できます。デフォルトでは、*text*にデフォルトのインライン・カスケード・スタイルシートとタイトルが含まれます。

*text*に空白が含まれる場合は、引用符で囲む必要があります。SQL*Plusでは、このような自由文の入力について、HTMLの妥当性をテストしません。入力するテキストがHTMLの<HEAD>タグの内容として妥当であるかどうかは、各自が確認する必要があります。これによって、出力を、ご使用のブラウザや固有のニーズに応じて柔軟にカスタマイズできます。

BODY *text*

BODY *text*オプションでは、<BODY>タグの属性を指定できます。デフォルトでは、属性は付いていません。*text*に空白が含まれる場合は、引用符で囲む必要があります。SQL*Plusでは、このような自由文の入力について、HTMLの妥当性をテストしません。入力するテキストがHTMLの<BODY>タグの内容として妥当であるかどうかは、各自が確認する必要があります。これによって、出力を、ご使用のブラウザや固有のニーズに応じて柔軟にカスタマイズできます。

TABLE *text*

TABLE *text*オプションでは、<TABLE>タグの属性を入力できます。TABLE *text*を使用して、HTMLの<TABLE>タグのBORDER、CELLPADDING、CELLSPACING、WIDTHなどの属性を設定できます。デフォルトでは、<TABLE>のWIDTH属性は90%、BORDER属性は1に設定されます。

*text*に空白が含まれる場合は、引用符で囲む必要があります。SQL*Plusでは、このような自由文の入力について、HTMLの妥当性をテストしません。入力するテキストがHTMLの<TABLE>タグの内容として妥当であるかどうかは、各自が確認する必要があります。これによって、出力を、ご使用のブラウザや固有のニーズに応じて柔軟にカスタマイズできます。

ENTMAP {ON|OFF}

ENTMAP ONまたはOFFでは、特殊文字「<」、「>」、「"」および「&」を、SQL*PlusでそれぞれのHTMLエンティティ「<」、「>」、「"」および「&」に置き換えるかどうかを指定できます。ENTMAPは、デフォルトでONに設定されています。

ENTMAPのONおよびOFFは、セッション中、必要に応じて切り替えることができます。たとえば、ENTMAP OFFを指定すると、SQL*Plus画面出力は次のようになります。

```
SQL>PROMPT A > B
A > B
```

ENTMAP ONを指定すると、SQL*Plus画面出力は次のようになります。

```
SQL&gt; PROMPT A > B
A &gt; B
```

<HEAD>タグおよび<BODY>タグのエンティティがマップされていない場合、MARKUP HEADオプションおよびMARKUP BODYオプションで有効なエンティティが使用されていることを確認する必要があります。

エンティティがマップされていない場合、Webブラウザは、データを無効なHTMLとして処理するため、それ以降すべての出力が正しく表示されなくなります。ENTMAP OFFを使用すると、独自のHTMLタグを書き込んで出力をカスタマイズできます。



ENTMAP は、HTML オプションが ON に設定されている場合にのみ有効です。出力でのエンティティの使用方法的詳細は、「[COLUMN](#)」コマンドを参照してください。

SPOOL {ON|OFF}

SPOOL ON または OFF では、SQL*Plus の SPOOL filename コマンドで作成された各ファイルの始めと終わりに、HTML 開始タグの<HTML>タグと<BODY>タグ、および終了タグの</BODY>タグと</HTML>タグを書き込むかどうかを指定できます。デフォルトは OFF です。

SPOOLのONおよびOFFは、セッション中、必要に応じて切り替えることができます。

ノート:

SET MARKUP HTML SPOOL オプションと SQLPLUS の SPOOL filename コマンドの違いを十分に理解する必要があります。



SET MARKUP HTML SPOOL ON オプションでは、スプール・ファイルへの<HTML>タグの書き込みを指定できます。SQLPLUS の SPOOL filename コマンドを実行するまで、スプール・ファイルは作成されず、SET MARKUP HTML SPOOL ON オプションで有効化されたヘッダーおよびフッター・タグをスプール・ファイルに書き込むこともできません。詳細は、「[SPOOL](#)」コマンドを参照してください。

SPOOL filenameコマンドを実行すると、SQL*Plusによって複数のHTMLタグがスプール・ファイルに書き込まれます。

EXIT、SPOOL OFFまたはSPOOL filenameのいずれかのSQL*Plusコマンドを実行すると、SQL*Plusによって次の終了タグが書き込まれ、ファイルが閉じます。

```
</BODY>  
</HTML>
```

<HEAD>タグの内容および<BODY>の属性は、HEADおよびBODYオプションを使用して指定できます。

PRE [FORMAT] {ON|OFF}

PREFORMAT ONまたはOFFでは、出力を、SQL*Plusによって<PRE>タグまたはHTML表に書き出すかどうかを指定できます。デフォルトはOFFで、出力はHTML表に書き出されます。PREFORMATのONおよびOFFは、セッション中、必要に応じて切り替えることができます。

ノート:

HTML の<PRE>タグを使用したレポート出力を生成するには、PREFORMAT ON を設定する必要があります。次に例を示します。



```
SQLPLUS -M "HTML ON PREFORMAT ON"
```

または

3.5.1.7 MARKUPの使用上のノート

MARKUP HTML ON

MARKUP HTML ON PREFORMAT OFFが指定されている場合、基本的に紙のレポートの書式を設定するためのコマンドには、HTML表のレポートに対する場合とは異なる、次のような意味があります。

- PAGESIZEは、HTML表での行数であり、単なる行数ではありません。各行には、複数の行が含まれる場合があります。TTITLE、BTITLEおよび列ヘッダーは、各PAGESIZE行で繰り返されます。
- LINESIZEは行を折り返す場合、または非常に長いデータの場合に有効になります。データ・サイズによっては、出力は別々の行に生成され、ブラウザによって空白文字として解釈される場合があります。
- TTITLEおよびBTITLEの内容は、3つの行位置(左、中央および右)に出力され、最大行幅はブラウザのウィンドウの90%に設定されています。これらの要素は、Webでの出力とは揃わない場合があります。TTITLEおよびBTITLEでのエンティティのマッピングは、MARKUPコマンドで指定する一般的なENTMAPの設定と同じです。
- 独自の出力でタイトルを使用する場合、SQL*Plusによって新しいHTML表が開始され、タイトルの後に出力行が表示されます。各列のデータの幅によって、ご使用のブラウザでは、各表の列幅が異なる書式に設定される場合があります。
- PREFORMATがONの場合、SET COLSEP、RECSEPおよびUNDERLINEのみが、HTMLレポート出力を生成します。

MARKUP CSV ON

MARKUP CSV ONが使用されている場合、問合せからの出力はCSV形式で表示されます。

SQL*Plusコマンドラインで-M[ARKUP] CSV ONオプションを使用することにより、ユーザー・セッションにログイン中にCSVマークアップを有効にできます。詳細は、[SQL*Plusプログラムの構文](#)を参照してください。ユーザー・セッションにログイン中は、SET MARKUP CSV ONコマンドを使用してCSVマークアップを有効にできます。

DELIMITERオプションを使用して、デリミタ文字を指定できます。QUOTE OFFを使用して、引用符なしでテキストを出力することもできます。

3.5.1.8 No Login Timeオプション

```
-nologintime
```

ログオン時に、SYS以外のユーザーの最終ログイン時間が表示されます。この機能はデフォルトで有効化されています。最終ログイン時間はローカル時間の形式で表示されます。このセキュリティ機能は、-nologintimeオプションを使用して無効にできます。ログイン後に、最終ログイン情報が表示されます。

```
SQL*Plus: Release 19.0.0.0.0 Production on Tue Nov 14 22:12:47 2018
Version 19.1.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
Last Successful login time: Wed Nov 01 2017 23:35:38 -07:00
```

```
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.1.0.0.0
```

CONNECTコマンドを使用して接続を確立した場合、最終ログイン時間は表示されません。

3.5.1.9 RESTRICTオプション

-R[ESTRICT] {1|2|3}

オペレーティング・システムと対話する特定のコマンドを使用禁止にすることができます。これは、PUP(Product User Profile)表の同じコマンドを使用禁止にする場合と同様です。ただし、-RESTRICTオプションで使用禁止にされたコマンドは、サーバーに接続していない場合も使用不可能で、SQL*Plusを終了するまで使用禁止のままです。

-RESTRICTオプションが指定されていない場合は、PUP表で使用禁止に設定されていないかぎり、すべてのコマンドが使用可能です。

-RESTRICTを指定すると、LOGIN.SQLは読み込まれません。GLOGIN.SQLは読み込まれますが、制限付きのコマンドを使用すると、正常に実行されません。

表3-1 各制限レベルでの使用禁止コマンド

コマンド	レベル1	レベル2	レベル3
EDIT	使用禁止	使用禁止	使用禁止
GET			使用禁止
HOST	使用禁止	使用禁止	使用禁止
SAVE		使用禁止	使用禁止
SPOOL		使用禁止	使用禁止
START、@、@@			使用禁止
STORE		使用禁止	使用禁止

3.5.1.10 SILENTオプション

-S[ILENT]

すべてのSQL*Plus情報メッセージおよびプロンプト・メッセージを非表示にします。コマンド・プロンプト、コマンドのエコーおよびSQL*Plusの起動時に通常表示されるバナーも非表示になります。usernameまたはpasswordを省略すると、SQL*Plusによって入力が求められますが、プロンプトは表示されません。ユーザーには見えないようにSQL*Plusを使用するには、SILENTを使用して別のプログラム内でSQL*Plusを起動します。

SILENTは、CGIを含むSQLPLUS -MARKUPコマンドやオペレーティング・システムのスクリプトを使用して、Web用のレポートを作成する場合に有効なモードです。SILENTオプションを使用して作成されたレポートでは、SQL*Plusのバナーおよびプロンプトは表示されません。

3.5.2 ログオン

```
username[/password]
```

SQL*Plusを起動し、Oracle Databaseに接続するためのユーザー名およびパスワードを指定します。

警告:



パスワードをプレーン・テキストで指定すると、セキュリティ上の危険があります。パスワードを省略し、パスワードの入力を求めるプロンプトが表示された場合にのみ入力することで、この危険を回避できます。

usernameおよびpasswordを省略すると、SQL*Plusは入力を求めるプロンプトを表示します。passwordのみを省略すると、passwordの入力を求めるプロンプトが表示されます。サイレント・モードでは、usernameおよびpasswordのプロンプトは表示されません。usernameを入力した場合は表示されますが、パスワードは表示されません。

```
@connect_identifier
```

Oracle Net接続識別子で構成されます。正確な構文は、使用するOracle Net通信プロトコルによって異なります。詳細は、Oracle Netのマニュアルを参照するか、またはデータベース管理者に問い合わせてください。

```
edition=va/ue
```

Oracleエディションの値です。1つのエディションによって、データベース内に1つのオブジェクトの2つ以上のバージョンが存在できます。これにより、変更したオブジェクトをデータベースにロードし、コンパイルして稼働時間中に実行するためのステージング領域が提供されます。このことは、特に、アプリケーションへのパッチ適用に関連する停止時間を短縮する場合に有用です。

eedition=valueは、ORA_EDITION環境変数で指定されたエディション値よりも優先されます。詳細は、「[エディションベースの再定義の使用](#)」を参照してください。

```
/
```

オペレーティング・システム認証を使用するデフォルト・ログオンを表します。デフォルト・ログオンを使用する場合、接続識別子は指定できません。デフォルト・ログオンでは通常、SQL*Plusはユーザー名OPS\$nameを使用してユーザーをログインさせようとします。nameは、ご使用のオペレーティング・システムのユーザー名です。接頭辞「OPS\$」は、他のテキスト文字列に設定される場合もあります。たとえば、INIT.ORAパラメータ・ファイルの設定をLOGONnameまたはUSERIDnameに変更した場合です。オペレーティング・システム認証の詳細は、「[オペレーティング・システム認証の使用](#)」を参照してください。

```
AS {SYSASM | SYSBACKUP | SYSDBA | SYSDG | SYSOPER | SYSRAC | SYSKM}
```

AS 句を使用すると、SYSASM、SYSBACKUP、SYSDBA、SYSDG、SYSOPER、SYSRACまたはSYSKMシステム権限が付与されているユーザーによる特権付き接続が可能です。

```
/NOLOG
```

Oracle Databaseへの初期接続を確立しません。SQLコマンドを実行する前に、CONNECTコマンドを実行して有効なログインを確立する必要があります。ユーザー名、パスワードまたはデータベースを指定するためのプロンプトを、SQL*Plusスクリプトから表示するには、「/NOLOG」を使用します。このスクリプトの1行目には、ログインが含まれていないものとみなされます。

3.5.3 起動

```
@{ur|}file_name[.ext]} [arg ...]
```

スクリプトの名前および実行する引数を指定します。スクリプトは、ローカル・ファイル・システムまたはWebサーバーからコールされます。

SQL*Plusでは、SQL*PlusのSTARTコマンドを使用してファイルを実行した場合と同様に、引数がスクリプトに渡されます。ファイルに拡張子が指定されていない場合は、SET SUFFIXコマンドで定義された拡張子を使用されます。デフォルトの拡張子は、.sqlです。

詳細は、「[START](#)」コマンドを参照してください。

第II部 SQL*Plusの使用方法

第II部では、SQL*Plusの使用方法、パフォーマンス向上のためのSQL*Plusのチューニング方法、表およびコマンドへのアクセスの制限方法、データベース管理ツールおよびグローバリゼーション・サポートの概要について学習します。

第II部は、次の章で構成されます。

- [SQL*Plusの基本](#)
- [SQL*Plusでのスクリプトの使用](#)
- [SQL*Plusレポートの書式設定](#)
- [SQL*PlusからのHTMLレポートの生成](#)
- [SQL*Plusのチューニング](#)
- [SQL*Plusのセキュリティ](#)
- [SQL*Plusによるデータベース管理](#)
- [SQL*Plusのグローバリゼーション・サポート](#)

4 SQL*Plusの基本

この章では、SQL*Plusの基本的な使用方法を学習します。内容は次のとおりです。

- [コマンドの入力と実行について](#)
- [表定義の表示について](#)
- [PL/SQL定義の表示について](#)
- [SQLコマンドの実行](#)
- [PL/SQLブロックの実行について](#)
- [SQL*Plusコマンドの実行](#)
- [コマンドの実行に影響するシステム変数](#)
- [実行中のコマンドの停止について](#)
- [オペレーティング・システムのコマンドの実行について](#)
- [表示の一時停止について](#)
- [データベースへの変更の自動保存について](#)
- [エラー・メッセージの解釈について](#)

4.1 コマンドの入力と実行について

特に指定がないかぎり、コマンドの使用方法についての説明はすべてのユーザー・インタフェースに適用できます。

コマンドラインでは、SQL*Plusプロンプトでコマンドを入力し、[Return]を押して実行します。

通常、コマンド内の単語と単語を区切るには、空白かタブを使用します。また、コマンドを読みやすくするために、単語と単語の間に空白またはタブを追加することもできます。

大文字と小文字の区別は、オペレーティング・システムによって異なります。このマニュアルでは、わかりやすくするため、表名、列名およびコマンドをすべて大文字で表記しています。

次に示す3つのタイプのコマンドを入力できます。

- データベース内の情報を処理するためのSQLコマンド
- データベース内の情報を処理するためのPL/SQLブロック
- 問合せ結果の書式設定、オプションの設定、SQLコマンドとPL/SQLブロックの編集および格納を行うためのSQL*Plusコマンド

コマンドを次の行に続ける方法、コマンドの終了方法、コマンドの実行方法は、入力して実行するコマンドのタイプによって異なります。次の各ページでは、それらのタイプのコマンドを実行する例を示します。

4.1.1 SQLバッファ

SQLバッファには、最後に入力されたSQLコマンドまたはPL/SQLブロック(SQL*Plusコマンドではなく)が格納されます。コマンドやブロックは、次のSQLコマンドまたはPL/SQLブロックに置換されるまでバッファに格納されます。LISTコマンドを使用して、バッファの内容を表示できます。

RUNまたは/(スラッシュ)コマンドを使用して、SQLバッファ内のコマンドまたはブロックを実行できます。RUNを使用すると、バッ

ファ内のコマンドまたはブロックが表示されてから実行されます。/(スラッシュ)を使用すると、バッファ内のコマンドまたはブロックは表示されずに実行されます。バッファ内に格納されたコマンドやブロックの編集については、[SQL*Plusコマンドラインでのスクリプトの編集について](#)を参照してください。

SQL*Plusでは、SQL*Plusコマンドまたはコマンドを実行するために入力したセミコロンやスラッシュは、SQLバッファに格納されません。

4.1.2 コマンドの実行について

SQL*Plusコマンドラインでは、コマンドを入力し、[Return]キーを押してSQL*Plusにそのコマンドを実行するように指示します。SQL*Plusでコマンドが処理され、別のコマンドに対する準備が整うと、コマンド・プロンプトが再表示されます。

4.2 表定義の表示について

指定された表またはビューの各列の定義を表示するには、SQL*PlusのDESCRIBEコマンドを使用します。

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER (6)
JOB_ID	NOT NULL	VARCHAR2 (10)
MANAGER_ID		NUMBER (6)
DEPARTMENT_ID		NUMBER (4)
LOCATION_ID		NUMBER (4)
COUNTRY_ID		CHAR (2)
FIRST_NAME		VARCHAR2 (20)
LAST_NAME	NOT NULL	VARCHAR2 (25)
SALARY		NUMBER (8, 2)
COMMISSION_PCT		NUMBER (2, 2)
DEPARTMENT_NAME	NOT NULL	VARCHAR2 (30)
JOB_TITLE	NOT NULL	VARCHAR2 (35)
CITY	NOT NULL	VARCHAR2 (30)
STATE_PROVINCE		VARCHAR2 (25)
COUNTRY_NAME		VARCHAR2 (40)
REGION_NAME		VARCHAR2 (25)

ノート:



DESCRIBE は、Oracle Database データ・ディクショナリ内の情報へのアクセスに使用します。SQL の SELECT コマンドを使用すると、データベース内のこれらの情報や他の情報にもアクセスできます。詳細は、[『Oracle Database SQL 言語リファレンス』](#)を参照してください。

例4-1 DESCRIBEコマンドの使用

EMP_DETAILS_VIEWというサンプル・ビューの列定義を表示するには、次のように入力します。

```
DESCRIBE EMP_DETAILS_VIEW
```

4.3 PL/SQL定義の表示について

ファンクションまたはプロシージャの定義を表示するには、SQL*PlusのDESCRIBEコマンドを使用します。

例4-2 DESCRIBEコマンドの使用

AFUNCというファンクションの定義を作成および表示するには、次のように入力します。

```

create or replace function afunc (f1 varchar2, f2 number) return number as
begin
    if (length(f1) > f2) then
        return 1;
    else
        return 0;
    end if;
end;
/

```

FUNCTION created.

DESCRIBE afunc

FUNCTION afunc RETURNS NUMBER

Argument	Name	Type	In/Out	Default?
F1		VARCHAR2	IN	
F2		NUMBER	IN	

4.4 SQLコマンドの実行

SQLコマンド言語を使用すると、データベース内のデータを操作できます。個々のSQLコマンドについては、[『Oracle Database SQL言語リファレンス』](#)を参照してください。

1. コマンド・プロンプトで、コマンドの1行目を次のように入力します。

```
SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, SALARY
```

誤って入力した場合は、[Back Space]を使用して消去し、再度入力します。入力終了後、[Return]を押して次の行へ移動します。

2. 2行目の入力を求める「2」が表示されます。コマンドの2行目を次のように入力します。

```
FROM EMP_DETAILS_VIEW WHERE SALARY > 12000;
```

セミコロン(;)は、コマンドの終わりを意味します。[Return]を押すか「実行」をクリックします。コマンドが処理され、結果が表示されます。

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	\$24,000
101	Kochhar	AD_VP	\$17,000
102	De Haan	AD_VP	\$17,000
145	Russell	SA_MAN	\$14,000
146	Partners	SA_MAN	\$13,500
201	Hartstein	MK_MAN	\$13,000

6 rows selected.

結果および取得した行数が表示された後、SQL*Plusコマンドラインに再度コマンド・プロンプトが表示されます。入力に誤りがあって結果が表示されなかった場合は、コマンドを再入力します。

ヘッダーは、PAGESIZEというシステム変数の設定に応じて、繰り返し出力される場合があります。問合せの結果が、使用可能なページ幅に収まらない場合があります。システム変数LINESIZEを使用して、文字の出力幅を設定できます。「[ページ・サイズの設定](#)」を参照してください。通常、LINESIZEはコマンドラインでは80に設定されています。取得したレコード件数を示すメッセージが表示されるかどうかは、システム変数FEEDBACKの設定によって異なります。

詳細は、「[コマンドの実行に影響するシステム変数](#)」を参照してください。

例4-3 SQLコマンドの入力

この例では、SQLコマンドを入力して実行し、各従業員の従業員番号、名前、職種および給与をEMP_DETAILS_VIEWビューに表示します。

4.4.1 SQLコマンドの構文について

話し言葉に、単語を組み合わせて文にする方法を規定する構文規則があるように、SQL*Plusにも、単語を組み合わせてコマンドにする方法を規定する構文規則があります。このようなコマンドをSQL*Plusで実行するには、それらの規則に従う必要があります。

4.4.1.1 1つのSQLコマンドの複数行への分割について

SQLコマンドは、個々の単語の途中で改行しないかぎり、任意の位置で複数行に分割できます。したがって、[例4-3](#)で入力した問合せは、次のように3行で入力できます。

```
SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000;
```

このマニュアルでは、ほとんどのSQLコマンドが句に分割されており、1つの句が1行になっています。たとえば、[例4-3](#)では、SELECT句とFROM句が個別の行に配置されています。多くの場合、これが最も見やすい方法ですが、コマンドを自分にとって最も読みやすくするためには、どのように行を分割してもかまいません。

4.4.1.2 SQLコマンドの終了について

SQLコマンドは、次に示す3つの方法のいずれかによって終了できます。

- セミコロン(;)を入力する
- 1行にスラッシュ(/)のみを1つ入力する
- 空白行を入力する

セミコロン(;)は、コマンドを実行することをSQL*Plusに通知するために使用します。[例4-3](#)に示すように、コマンドの最終行の末尾にセミコロンを入力して、[Return]を押すか「実行」をクリックします。SQL*Plusによって、コマンドが処理され、SQLバッファに格納されます。詳細は、「[SQLバッファ](#)」を参照してください。セミコロンを入力する前に誤って[Return]を押した場合は、コマンドの次の行の入力を促す行番号がSQL*Plusプロンプトに表示されます。コマンドを実行するには、セミコロンを入力して、再度[Return]を押すか「実行」をクリックします。

スラッシュ(/)のみの行を使用しても、SQL*Plusにコマンドの実行を指示できます。コマンドの最終行の末尾で[Return]を押します。SQL*Plusプロンプトに次の行の行番号が表示されます。スラッシュを入力し、再度[Return]を押すか「実行」をクリックします。SQL*Plusで、コマンドが実行され、バッファに格納されます。

SQL文またはスクリプト内に空白行を使用すると、コマンド入力の終了後、SQL*Plusでそのコマンドの実行が待機されます。コマンドの最終行の末尾で[Return]を押します。SQL*Plusプロンプトに次の行の行番号が表示されます。

ノート:



SET SQLBLANKLINES コマンドを使用して、SQL 文内での空白行の表示形式および動作を変更できます。空白行の動作の変更方法の詳細は、「[SET](#)」コマンドを参照してください。

このようにコマンドを実行するには、再度[Return]を押して、SQL*Plusのコマンド・プロンプトを表示します。コマンドは、実行されないままSQLバッファに格納されます。詳細は、「[SQLバッファ](#)」を参照してください。引き続き別のSQLコマンドを入力すると、バッファ内にある前のコマンドは上書きされます。

4.5 PL/SQLブロックの実行について

データベース内のデータを操作するためにPL/SQLサブプログラム(ブロック)を使用することもできます。個々のPL/SQL文については、『[Oracle Database PL/SQL言語リファレンス](#)』を参照してください。

SQL*Plusでは、SQLコマンドと同じ方法でPL/SQLサブプログラムを処理できますが、セミicolon(;)または空白行でブロックを終了および実行できません。PL/SQLサブプログラムを終了するには、新しい行にピリオド(.)のみを入力します。新しい行にスラッシュ(/)のみを入力しても終了および実行が可能です。

PL/SQL文を入力するためのモードに入る方法は、次のとおりです。

- DECLAREまたはBEGINを入力します。この方法でPL/SQLモードに入った後、残りのPL/SQLサブプログラムを入力します。
- ストアド・プロシージャを作成するSQLコマンド(CREATE PROCEDUREなど)を入力します。この方法でPL/SQLモードに入った後、作成するストアド・プロシージャを入力します。

入力したサブプログラムは、SQLバッファに格納されます。RUNコマンドまたはスラッシュ(/)を入力して、現行のサブプログラムを実行します。セミicolon(;)はPL/SQLサブプログラムの一部とみなされるため、コマンドは実行されません。

処理のために(SQLコマンドの場合と同様に)、PL/SQLサブプログラム全体がSQL*PlusからOracle Databaseに送信されます。詳細は、『[Oracle Database PL/SQL言語リファレンス](#)』を参照してください。

たとえば、次のようなPL/SQLサブプログラムを入力して実行できます。

```
DECLARE
  x  NUMBER := 100;
BEGIN
  FOR i IN 1..10 LOOP
    IF MOD (i, 2) = 0 THEN    --i is even
      INSERT INTO temp VALUES (i, x, 'i is even');
    ELSE
      INSERT INTO temp VALUES (i, x, 'i is odd');
    END IF;
    x := x + 100;
  END LOOP;
END;
```

4.5.1 ストアド・プロシージャの作成について

ストアド・プロシージャは、PL/SQLのファンクション、パッケージまたはプロシージャです。ストアド・プロシージャを作成するには、次のSQLのCREATEコマンドを使用します。

- CREATE FUNCTION
- CREATE LIBRARY
- CREATE PACKAGE

- CREATE PACKAGE BODY
- CREATE PROCEDURE
- CREATE TRIGGER
- CREATE TYPE

これらのいずれかのコマンドを入力すると、PL/SQLモードに入り、PL/SQLサブプログラムを入力できます。詳細は、[PL/SQLブロックの実行について](#)を参照してください。PL/SQLサブプログラムの入力終了後、ピリオド(.)のみの行を入力してPL/SQLモードを終了します。SQLコマンドを実行し、ストアド・プロシージャを作成するには、RUNまたはスラッシュ(/)を入力する必要があります。セミコロン(;)を入力すると、CREATEコマンドは実行されません。

CREATEコマンドを使用してストアド・プロシージャを作成する場合、コンパイル・エラーが発生するとメッセージが表示されます。それらのエラーを表示するには、SHOW ERRORSを使用します。次に例を示します。

```
SHOW ERRORS PROCEDURE ASSIGNVL
```

詳細は、「[SHOW](#)」を参照してください。

ストアド・プロシージャを参照するPL/SQL文を実行するには、SQL*PlusのEXECUTEコマンドを使用します。EXECUTEは、このコマンドの直後に入力されたPL/SQL文を実行します。次に例を示します。

```
EXECUTE EMPLOYEE_MANAGEMENT.NEW_EMP('BLAKE')
```

詳細は、「[EXECUTE](#)」を参照してください。

4.6 SQL*Plusコマンドの実行

SQL*Plusコマンドを使用して、SQLコマンドおよびPL/SQLブロックを操作し、問合せ結果の書式を設定して出力できます。SQL*Plusでは、SQL*PlusコマンドはSQLコマンドやPL/SQLブロックとは異なる方法で処理されます。

コマンド入力を簡略化するために、多くのSQL*Plusコマンドで略称を使用できます。すべてのSQL*Plusコマンドの情報および略称については、「[SQL*Plusコマンド・リファレンス](#)」を参照してください。

1. 次のSQL*Plusコマンドを入力します。

```
COLUMN SALARY FORMAT $99,999 HEADING 'MONTHLY SALARY'
```

誤って入力した場合は、[Back Space]を使用して消去し、再度入力します。行の入力が終了したら、[Return]を押します。SQL*Plusで新しい書式が認識され、SQL*Plusコマンド・プロンプトが再度表示されて、新しいコマンドを入力できる状態になります。

2. 次の問合せを入力し、[Return]を押して実行します。

```
SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, SALARY
FROM EMP_DETAILS_VIEW WHERE SALARY > 12000;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	MONTHLY SALARY
100	King	AD_PRES	\$24,000
101	Kochhar	AD_VP	\$17,000
102	De Haan	AD_VP	\$17,000
145	Russell	SA_MAN	\$14,000
146	Partners	SA_MAN	\$13,500
201	Hartstein	MK_MAN	\$13,000

6 rows selected.

例4-4 SQL*Plusコマンドの入力

この例では、SQL*Plusコマンドを入力して、サンプル・ビューEMP_DETAILS_VIEWのSALARY列の表示に使用する書式を変更する方法を示します。

COLUMNコマンドによって、SALARY列がドル記号(\$)およびカンマ(,)を使用する書式に設定され、新しいヘッダーが付けられます。

4.6.1 SQL*Plusコマンドの構文について

SQL*Plusコマンドの構文は、SQLコマンドまたはPL/SQLブロックとは異なります。

SQL*Plusコマンドは、必ずしもセミコロンで終了する必要はありません。コマンドの入力終了後、[Return]を押すか「実行」をクリックします。SQL*Plusコマンドは、必ずしもセミコロンで終了する必要はありません。

4.6.1.1 長いSQL*Plusコマンドを次の行に続ける方法について

長いSQL*Plusコマンドは、行末にハイフン(-)を付けて[Return]を押すと、次の行に続けて入力できます。必要な場合は、ハイフンの前に空白も入力できます。SQL*Plusでは、右山カッコ(>)が次の行のプロンプトとして表示されます。

次に例を示します。

```
COLUMN SALARY FORMAT $99,999 -  
HEADING 'MONTHLY SALARY'
```

SQL*Plusでは、ハイフンは継続文字として識別されるため、SQL文内でハイフンを入力すると無視されます。SQL*Plusでは、入力処理で行が結合された後にハイフンが削除されるまで、その文はSQL文として識別されません。たとえば、次のように入力します。

```
SELECT 200 -  
100 FROM DUAL;
```

次のエラーが戻されます。

```
SELECT 200 100 FROM DUAL  
          *  
ERROR at line 1:  
ORA-00923: FROM keyword not found where expected
```

文が正しく解釈されるように、最初の行の末尾から2行目の先頭へハイフンを移動させます。

4.7 コマンドの実行に影響するシステム変数

SQL*PlusのSETコマンドを使用して、SET変数またはシステム変数と呼ばれる多くの変数を指定できます。SET変数およびシステム変数は、SQL*Plusコマンドの実行方法に影響します。システム変数には、出力のデフォルトの列幅、コマンドによって選択されたレコード数の表示、ページ・サイズなどのSQL*Plus内の様々な条件を指定できます。

このマニュアルの例は、システム変数をデフォルトの設定にしてSQL*Plusを実行することを前提としています。システム変数の設定によっては、例に示す出力とは少し異なる出力が表示される場合があります。(コンピュータにSQL*PlusのLOGINファイルが存在する場合は、デフォルトの設定と異なる可能性があります。)

システム変数およびそれらのデフォルト設定の詳細は、「[SET](#)」コマンドを参照してください。SQL*PlusのLOGINファイルの詳細は、「[SQL*Plusの構成](#)」および「[SQLPLUSプログラムの構文](#)」を参照してください。

システム変数の現行の設定を表示するには、SHOWを入力し、その後に変数名を入力します。SHOWを使用して表示できるその他の項目の詳細は、「[SHOW](#)」コマンドを参照してください。

4.8 実行中のコマンドの停止について

50ページのレポートの1ページ目を表示した後、残りのページは表示の必要がないと判断したとします。この場合は、「取消」を押します。通常、システムの割込み文字は、[Ctrl]+[C]です。SQL*Plusは表示を停止します。

ノート:



「取消」を押しても、SQL*Plus の SPOOL コマンドの OUT 句でプリンタに送信したファイルの出力は停止しません。(「[SQL*Plus レポートの書式設定](#)」で、問合せ結果の出力について学習します。)ファイルの出力は、オペレーティング・システムから停止できます。詳細は、ご使用のオペレーティング・システム固有のインストレーション・ガイドおよび管理者リファレンスを参照してください。

4.9 オペレーティング・システムのコマンドの実行について

SQL*Plusのコマンド・プロンプトからオペレーティング・システムのコマンドを実行できます。このコマンドは、既存のオペレーティング・システム・ファイルの表示などの作業を実行する場合に有効です。

オペレーティング・システムのコマンドを実行するには、SQL*PlusのHOSTコマンドに続けて、オペレーティング・システムのコマンドを入力します。たとえば、SQL*Plusコマンドで、次のDIRECTORY *.SQLコマンドを実行できます。

```
HOST DIRECTORY *.SQL
```

コマンドの実行が終わると、再度SQL*Plusのコマンド・プロンプトが表示されます。

ノート:



HOST コマンドを使用して SQL*Plus セッションから入力したオペレーティング・システムのコマンドは、現行の SQL*Plus セッションに影響を与えません。たとえば、オペレーティング・システムの環境変数を設定しても、現行の SQL*Plus セッションには影響を与えませんが、引き続き起動される SQL*Plus セッションに影響を与える可能性があります。

HOST コマンドへのアクセスを使用禁止にできます。HOST コマンドを使用禁止にする方法の詳細は、「[SQL*Plus のセキュリティ](#)」を参照してください。

4.10 表示の一時停止について

PAUSEシステム変数を使用すると、長いレポートまたは行数が多い表の定義の表示中、各ページごとに停止させて画面の内容を詳しく調べることができます。

SET PAUSEを使用して、問合せまたはレポートの各画面の表示後に出力を一時停止できます。詳細は、「[SET PAU\[SE\] {ON | OFF | text}](#)」を参照してください。

4.11 データベースへの変更の自動保存について

SQLのデータ操作言語(DML)コマンド(UPDATE、INSERTおよびDELETE)を使用して、データベース内に格納されている情報に対して行う変更を指定できます。これらのコマンドは、単独かPL/SQLブロック内で使用できます。それらの変更は、SQLのCOMMITコマンド、データ制御言語(DCL)またはデータ定義言語(DDL)コマンド(CREATE TABLEなど)を入力するか、自動コミット機能を使用するまで確定されません。SQL*Plusの自動コミット機能は、指定した数のSQL DMLトランザクションが成功した後に、保留中の変更をコミットします(SQL DMLトランザクションとは、UPDATE、INSERT、DELETEのいずれかのコマンドまたはPL/SQLブロックのことです)。

自動コミット機能は、SQL*PlusのAUTOCOMMITシステム変数で指定します。AUTOCOMMITの設定にかかわらず、変更はSQL*Plusが正常に終了するとコミットされます。

関連項目:

[SET EXITC\[OMMIT\] {ON | OFF}](#)

```
COMMIT COMPLETE
```

自動コミット機能をONにした場合は、データベースへの変更をロールバックできません。

SQLのDMLコマンドを、一定の回数(たとえば、10回)実行した後にデータベースへの変更をコミットする場合は、次のように入力します。

```
SET AUTOCOMMIT 10
```

SQL*Plusでは、SQLのDMLコマンドが実行されるたびにカウントされ、10コマンドごとに変更がコミットされます。

ノート:



この機能では、ブロックに含まれている SQL コマンドの実際の数には関係なく、1 つの PL/SQL ブロックが 1 つのトランザクションとみなされます。

自動コミット機能を再度OFFにするには、次のコマンドを入力します。

```
SET AUTOCOMMIT OFF
```

AUTOCOMMITがOFFに設定されたことを確認するには、次の形式でSHOWコマンドを入力します。

```
SHOW AUTOCOMMIT
```

```
AUTOCOMMIT OFF
```

詳細は、「[SET AUTO\[COMMIT\]{ON | OFF | IMM\[EDIATE\] | n}](#)」を参照してください。

例4-5 自動コミット機能をONにする方法

自動コミット機能をONにするには、次のように入力します。

```
SET AUTOCOMMIT ON
```

次のように入力することもできます。

AUTOCOMMITの設定を変更するまで、データベースへの変更を指定するSQLの各DMLコマンドからの変更は、SQL*Plusで自動的にコミットされます。SQL*Plusでは、自動コミットを実行するたびに次のメッセージが表示されます。

4.12 エラー・メッセージの解釈について

SQL*Plusでコマンド内にエラーが検出された場合は、エラー・メッセージが表示されます。SQL*Plusエラー・メッセージのリストについては、「[SQL*Plusのエラー・メッセージ](#)」を参照してください。

SP2-0310: unable to open file "emplyyes.sql"

多くの場合、メッセージを読むだけで問題の修正方法がわかります。詳しい説明が必要な場合は、問題の原因および修正方法を判断するため、次のいずれかのステップを実行します。

- エラー・メッセージが「SP2」という文字で始まる番号付きのものである場合は、「[SQL*Plusのエラー・メッセージ](#)」でSQL*Plusメッセージを検索してください。
- エラー・メッセージが「CPY」という文字で始まる番号付きのものである場合は、「[COPYコマンドのメッセージ](#)」でSQL*PlusのCOPYコマンド・メッセージを検索してください。
- エラー・メッセージが、「ORA」という文字で始まる番号付きのものである場合は、『[Oracle Databaseエラー・メッセージ](#)』またはご使用のオペレーティング・システムのプラットフォーム固有のOracleドキュメントで、Oracle Databaseメッセージを検索してください。
- エラー・メッセージが「PLS」という文字で始まる番号付きのものである場合は、『[Oracle Database PL/SQL言語リファレンス](#)』で、Oracle Databaseメッセージを検索してください。

エラー・メッセージに番号が付いていない場合は、エラーとなったコマンドの正しい構文を検索するため、SQL*Plusコマンドについては「[SQL*Plusコマンド・リファレンス](#)」、SQLコマンドについては『[Oracle Database SQL言語リファレンス](#)』、PL/SQLブロックについては『[Oracle Database PL/SQL言語リファレンス](#)』を参照してください。それ以外の場合は、DBAに問い合わせてください。

例4-6 エラー・メッセージの解釈

次のように入力して、存在しないファイルまたは使用可能でないファイルを実行しようとしたとします。

START EMPLOYEES.SQL

次のエラー・メッセージは、表が存在しないことを示します。

5 SQL*Plusでのスクリプトの使用

この章では、SQL*Plusコマンド、SQLコマンドおよびPL/SQLブロックの操作方法を学習します。次のトピックについて説明します。

- [スクリプトの編集について](#)
- [SQL*Plusコマンドラインでのスクリプトの編集について](#)
- [スクリプトへのコメントの挿入について](#)
- [スクリプトの実行](#)
- [スクリプトのネスト](#)
- [リターン・コードを伴うスクリプトの終了について](#)
- [置換変数の定義](#)
- [事前定義変数の使用について](#)
- [置換変数の使用](#)
- [STARTコマンドを使用したパラメータの受渡し方法](#)
- [ユーザーとの対話について](#)
- [バインド変数の使用方法について](#)
- [PL/SQLブロックに含まれるSELECT文の結果の反復的なフェッチ](#)

この章を読むときは、コンピュータで、示されている例を実際に試してみてください。始める前に、「[SQL*Plus概要](#)」で説明したサンプル・スキーマへのアクセス権限があることを確認してください。

5.1 スクリプトの編集について

SQL*Plusコマンドラインでは、外部エディタを@、@@またはSTARTコマンドと組み合わせて使用すると、共通スクリプトの作成および実行に有効です。SQL*Plus、SQLおよびPL/SQLコマンドを含むスクリプトが作成できます。この機能は、複雑なコマンドまたは頻繁に使用されるレポートの格納に特に有効です。

5.1.1 システム・エディタでのスクリプトの作成

オペレーティング・システムには、スクリプトの作成に使用可能な1つ以上のテキスト・エディタが含まれています。EDITコマンドを入力すると、SQL*Plusのコマンドラインを終了せずに、オペレーティング・システムのデフォルト・テキスト・エディタを実行できます。

デフォルトのテキスト・エディタの名前を保持するには、SQL*PlusのDEFINEコマンドを使用して変数_EDITORを定義します。たとえば、EDITで使用するエディタをviとして定義するには、次のコマンドを入力します。

```
DEFINE _EDITOR = vi
```

ユーザー定義は、SQL*Plusを起動すると常に有効になるように、ユーザー・プロファイルまたはサイト・プロファイルに含めることができます。詳細は、「[SQL*Plusの構成](#)」、「[DEFINE](#)」コマンドおよび「[EDIT](#)」コマンドを参照してください。

テキスト・エディタを使用してスクリプトを作成するには、EDITの後に編集または作成するファイルの名前を続けて入力します。たとえば、次のように入力します。

```
EDIT SALES
```

EDITを使用すると、ファイルの拡張子を指定しないかぎり、ファイル名拡張子.SQLが名前に追加されます。テキスト・エディタを使用してスクリプトを保存する場合、そのスクリプトは同じファイルに保存されます。EDITを使用すると、スクリプトを作成および編集できます。

各SQLコマンドの終わりにセミコロン、また、ファイル内の各PL/SQLブロックの後にスラッシュ(/)のみの行をそれぞれ挿入する必要があります。複数のSQLコマンドおよびPL/SQLブロックをスクリプトに挿入できます。

例5-1 システム・エディタでのSQLスクリプトの作成

販売担当者とそのコミッションのリストを表示するために問合せを作成したとします。その問合せは、それぞれの従業員の実績を追跡し記録するために月1回実行する計画です。

システム・エディタを使用して問合せを作成および保存するには、エディタを起動し、スクリプトを保存するファイルを作成します。

EDIT SALES

エディタで、次の各行を入力します。SQL文の終わりには、必ずセミコロンを挿入してください。

```
COLUMN LAST_NAME HEADING 'LAST NAME'
COLUMN SALARY HEADING 'MONTHLY SALARY' FORMAT $99,999
COLUMN COMMISSION_PCT HEADING 'COMMISSION %' FORMAT 90.90
SELECT LAST_NAME, SALARY, COMMISSION_PCT
FROM EMP_DETAILS_VIEW
WHERE JOB_ID='SA_MAN';
```

COMMISSION_PCT列の書式モデルでは、10進値に対しては最初の0(ゼロ)が表示され、行のCOMMISSION_PCTの値が0(ゼロ)の場合は空白のかわりに0(ゼロ)が表示されます。書式モデルおよびCOLUMNコマンドの詳細は、「[COLUMN](#)」コマンドおよび「[書式モデル](#)」を参照してください。

最後に、エディタの保存コマンドを使用して、SALES.SQLというファイルに問合せを格納します。

5.2 SQL*Plusコマンドラインでのスクリプトの編集について

様々なSQL*Plusコマンドを使用して、現在バッファに格納されているSQLコマンドまたはPL/SQLブロックを編集できます。

[表5-1](#)に、コマンドを再入力せずにバッファ内のコマンドを検証または変更できるSQL*Plusコマンドを示します。

表5-1 SQL*Plusの編集コマンド

コマンド	略称	用途
APPEND <i>text</i>	A <i>text</i>	カレント行の末尾にテキストを追加します。
CHANGE/ <i>old/new</i>	C/ <i>old/new</i>	カレント行内の <i>old</i> を <i>new</i> に変更します。
CHANGE/ <i>text</i>	C/ <i>text</i>	テキストをカレント行から削除します。
CLEAR BUFFER	CL BUFF	すべての行を削除します
DEL	(なし)	カレント行を削除します。
DEL <i>n</i>	(なし)	行 <i>n</i> を削除します。

コマンド	略称	用途
DEL *	(なし)	カレント行を削除します。
DEL <i>n</i> *	(なし)	行 <i>n</i> からカレント行までを削除します。
DEL LAST	(なし)	最終行を削除します。
DEL <i>m n</i>	(なし)	ある範囲(<i>m</i> から <i>n</i>)の行を削除します。
DEL * <i>n</i>	(なし)	カレント行から行 <i>n</i> までを削除します。
INPUT	I	1 つ以上の行を追加します。
INPUT <i>text</i>	I <i>text</i>	<i>text</i> で構成されている行を追加します。
LIST	; or L	SQL バッファ内のすべての行を表示します。
LIST <i>n</i>	L <i>n</i> or <i>n</i>	行 <i>n</i> を表示します。
LIST *	L *	カレント行を表示します。
LIST <i>n</i> *	L <i>n</i> *	行 <i>n</i> からカレント行までを表示します。
LIST LAST	L LAST	最終行を表示します。
LIST <i>m n</i>	L <i>m n</i>	ある範囲(<i>m</i> から <i>n</i>)の行を表示します。
LIST * <i>n</i>	L * <i>n</i>	カレント行から行 <i>n</i> までを表示します。

これらのコマンドは、入力したコマンドの修正または変更の有効です。

5.2.1 バッファ内容の表示

SQL バッファには、最新の SQL コマンドまたは PL/SQL コマンドが含まれます。LIST および DEL 以外のすべての編集コマンドは、バッファ内の 1 行にのみ影響します。その行をカレント行と呼びます。現行のコマンドまたはブロックを表示した場合、カレント行にはアスタリスクが付きます。

```
SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, SALARY
2 FROM EMP_DETAILS_VIEW
3* WHERE SALARY>12000
```

SELECT コマンドの終わりに入力したセミコロンは表示されません。このセミコロンは、入力時にはコマンドの終わりを示すために必要ですが、SQL コマンドの一部とみなされないため、SQL バッファには格納されません。

例5-2 バッファ内容の表示

現行のコマンドを表示するとします。次のように、LISTコマンドを使用します。(例4-3のステップに従った後、SQL*Plusを終了するか、別のSQLコマンドまたはPL/SQLブロックを入力した場合は、続行する前に、その例のステップを再実行してください。)

LIST

5.2.2 カレント行の編集

SQL*PlusのCHANGEコマンドを使用して、カレント行を編集できます。次のように、操作によってカレント行は異なります。

- LISTコマンドで特定の行を表示すると、その行がカレント行になります。
- バッファ内のコマンドを表示(LIST)または実行(RUN)した場合、コマンドの最終行がカレント行になります。(ただし、スラッシュ(/)コマンドを使用してバッファ内のコマンドを実行しても、カレント行は影響を受けません。)
- エラーが発生した場合は、エラーが含まれている行が自動的にカレント行になります。

```
SELECT EMPLOYEE_ID, LAST_NAME, JO_ID, SALARY
                                *
ERROR at line 1:
ORA-00904: invalid column name
```

エラー・メッセージには、問合せの1行目に無効な列名が存在することが示されています。アスタリスクは、エラーの位置(綴りを誤って入力した列であるJOB_ID)を示しています。

この場合、コマンド全体を再入力せずに、バッファ内のコマンドを編集して誤りを修正できます。この時点では、エラーが含まれている行がカレント行になっています。CHANGEコマンドを使用して、誤りを修正します。このコマンドは次の3つの部分で構成され、それぞれの部分はスラッシュまたはその他の英数字以外の文字で区切られます。

- CHANGEという単語、またはCという文字
- 変更する文字列
- 置換文字列

CHANGEコマンドを使用すると、カレント行の中で最初に現れる変更文字列を検索し、新しい文字列に変更できます。行全体を再入力するには、CHANGEコマンドを使用する必要はありません。

```
1* SELECT EMPLOYEE_ID, FIRST_NAME, JOB_ID, SALARY
```

エラー修正の終了後は、RUNコマンドを使用してコマンドを再実行できます。

RUN

SQL*Plusでは、問合せおよびその結果が正しく表示されます。

```
1 SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, SALARY
2 FROM EMP_DETAILS_VIEW
3* WHERE JOB_ID='SA_MAN'
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	MONTHLY SALARY
145	Russell	SA_MAN	\$14,000
146	Partners	SA_MAN	\$13,500
147	Errazuriz	SA_MAN	\$12,000
148	Cambrault	SA_MAN	\$11,000
149	Zlotkey	SA_MAN	\$10,500

SALARY列には、例4-4で指定した書式が保持されます。(例4-4の実行後、SQL*Plusを終了して再起動すると、列は元の書式に戻ります。)

CHANGEコマンドでの大文字と小文字の区別、およびCHANGEコマンドでワイルド・カードを使用してテキストのブロックを指定する方法については、「[CHANGE](#)」コマンドを参照してください。

例5-3 コマンド入力でのエラー

JOB_ID列を選択するつもりで誤ってJO_IDと入力したとします。1行目のJOB_IDを意図的に誤った綴りにして、次のようにコマンドを入力します。

```
SELECT EMPLOYEE_ID, LAST_NAME, JO_ID, SALARY
FROM EMP_DETAILS_VIEW
WHERE JOB_ID=' SA_MAN' ;
```

画面に次のメッセージが表示されます。

例5-4 エラーの修正

JO_IDをJOB_IDに変更するには、次のようにCHANGEコマンドを使用して行を変更します。

```
CHANGE /JO_ID/JOB_ID
```

修正後の行が、次のように表示されます。

5.2.3 行へのテキストの追加

バッファの行末にテキストを追加するには、次のようにAPPENDコマンドを使用します。

1. LISTコマンド(または行番号)を使用して、変更する行を表示します。
2. APPENDを入力し、その後に追加するテキストを続けて入力します。追加するテキストが空白で始まる場合は、APPENDという単語とテキストの1文字目を2つの空白で区切ります。1つはAPPENDとテキストの区切りで、もう1つはテキストとともにバッファ内に格納されます。

例5-5 行へのテキストの追加

現行の問合せの行4に空白およびDESC句を追加するには、最初に行4を次のように表示します。

```
LIST 4
```

```
4* ORDER BY SALARY
```

次のコマンドを入力します(APPENDとDESCの間には、必ず空白を2つ入力します)。

```
APPEND DESC
```

```
4* ORDER BY SALARY DESC
```

RUNを入力して、問合せを検証します。

```
1 SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, SALARY
2 FROM EMP_DETAILS_VIEW
3 WHERE JOB_ID=' SA_MAN'
4* ORDER BY SALARY DESC
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	MONTHLY SALARY
145	Russell	SA_MAN	\$14,000
146	Partners	SA_MAN	\$13,500
147	Errazuriz	SA_MAN	\$12,000
148	Cambrault	SA_MAN	\$11,000
149	Zlotkey	SA_MAN	\$10,500

5.2.4 新しい行の追加

カレント行の後に新しい行を挿入するには、INPUTコマンドを使用します。

行1の前に行を挿入するには、0(ゼロ)を入力し、その後にテキストを続けます。その行がバッファの先頭に挿入され、すべての行が再度1から番号付けされます。

```
0 SELECT EMPLOYEE_ID
```

```
4
```

新しい行を入力します。次に、[Return]を押します。

```
4 ORDER BY SALARY
```

次のように新しい行の入力を求めるプロンプトが再度表示されます。

```
5
```

再度[Return]を押して、これ以上入力しないことを示します。その後、RUNコマンドを使用して問合せを検証し再実行します。

```
1 SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, SALARY
2 FROM EMP_DETAILS_VIEW
3 WHERE JOB_ID='SA_MAN'
4* ORDER BY SALARY
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	MONTHLY SALARY
149	Zlotkey	SA_MAN	\$10,500
148	Cambrault	SA_MAN	\$11,000
147	Errazuriz	SA_MAN	\$12,000
146	Partners	SA_MAN	\$13,500
145	Russell	SA_MAN	\$14,000

例5-6 行の追加

[例5-4](#)で変更したSQLコマンドに4行目を追加するとします。すでに行3がカレント行なので、INPUTを入力し、[Return]を押します。

```
INPUT
```

次のように、新しい行の入力を求めるプロンプトが表示されます。

5.2.5 行の削除

バッファ内の行を削除するには、DELコマンドを使用します。DELを入力し、削除する行番号を指定します。

カレント行から最終行までを削除するとします。次のように、DELコマンドを使用します。

```
DEL * LAST
```

DELを実行すると、バッファの次の行がある場合は、その行がカレント行になります。

詳細は、「[DEL](#)」を参照してください。

5.3 スクリプトへのコメントの挿入について

スクリプトにコメントを入力するには、次の3つの方法があります。

- 単一行のコメントに対して、SQL*PlusのREMARKコマンドを使用する方法
 - 1行以上のコメントに対して、SQLのコメント・デリミタ(/* ...*/)を使用する方法
 - 単一行のコメントに対して、米国規格協会(ANSI)および国際標準化機構(ISO)のコメント「--」を使用する方法
- コマンドラインで入力したコメントは、SQLバッファには格納されません。

5.3.1 REMARKコマンドの使用

REMARKコマンドのみの行をスクリプト内に指定し、同じ行にコメントを続けます。コメントを次の行に継続するには、追加のREMARKコマンドを入力します。REMARKコマンドは、1つのSQLコマンドの行と行の間には挿入しないでください。

```
REMARK Commission Report;
REMARK to be run monthly.;
COLUMN LAST_NAME HEADING 'LAST_NAME';
COLUMN SALARY HEADING 'MONTHLY SALARY' FORMAT $99,999;
COLUMN COMMISSION_PCT HEADING 'COMMISSION %' FORMAT 90.90;
REMARK Includes only salesmen;
SELECT LAST_NAME, SALARY, COMMISSION_PCT
FROM EMP_DETAILS_VIEW
WHERE JOB_ID='SA_MAN';
```

5.3.2 /**/の使用

SQLのコメント・デリミタ(/* ...*/)は、スクリプト内の個別の行に入力するか、SQLコマンドと同じ行に入力するか、またはPL/SQLブロック内の行に入力します。

コメントの初めのスラッシュとアスタリスク(/*)の後に空白を入力する必要があります。

コメントは、次のように複数の行にわたっていてもかまいませんが、コメント内にコメントをネストさせることはできません。

```
/* Commission Report
   to be run monthly. */
COLUMN LAST_NAME HEADING 'LAST_NAME';
COLUMN SALARY HEADING 'MONTHLY SALARY' FORMAT $99,999;
COLUMN COMMISSION_PCT HEADING 'COMMISSION %' FORMAT 90.90;
REMARK Includes only salesmen;
SELECT LAST_NAME, SALARY, COMMISSION_PCT
FROM EMP_DETAILS_VIEW
/* Include only salesmen.*/
WHERE JOB_ID='SA_MAN';
```

5.3.3 --の使用

SQL文、PL/SQLブロックまたはSQL*Plusコマンドの中で、ANSI/ISOの「--」スタイルのコメントを使用できます。終了デリミタがないため、複数行にわたるコメントは入力できません。

PL/SQLおよびSQLの場合、次のように、ある行のコマンドの後にコメントを入力するか、コメントのみの行を入力します。

```
-- Commissions report to be run monthly
DECLARE --block for reporting monthly sales
```

SQL*Plusコマンドでは、行全体をコメントに指定する場合のみ、「--」を使用してコメントを挿入できます。たとえば、次のコメントは有効です。

```
-- set maximum width for LONG to 777
SET LONG 777
```


次のコメントは無効です。

```
SET LONG 777 -- set maximum width for LONG to 777
```

次のSQL*Plusコマンドを入力した場合、このコマンドはコメントとして解釈され、実行されません。

```
-- SET LONG 777
```

5.3.4 コメント挿入時のノート

SQL*Plusには、SQLコマンドまたはPL/SQLコマンドの解析機能はありません。新しい文のそれぞれについて、先頭から数個のキーワードがスキャンされ、SQL、PL/SQLまたはSQL*Plusのいずれのコマンドであるかが判断されます。コメントの挿入箇所によっては、SQL*Plusでコマンドが正しく認識されず、予期しない結果になる場合があります。次に、SQL*Plusのコメントを効果的に使用上のノートを示します。

1. 文の先頭から数個のキーワードには、コメントを挿入しないでください。次に例を示します。

```
CREATE OR REPLACE
2  /* HELLO */
3  PROCEDURE HELLO AS
4  BEGIN
5  DBMS_OUTPUT.PUT_LINE('HELLO');
6  END;
7  /
```

Warning: Procedure created with compilation errors.

例に示す位置にコメントを挿入すると、コマンドがコマンドとして認識されません。コメントの最初にあるスラッシュ(/)が文の終了記号と解釈され、SQL*PlusからサーバーにPL/SQLブロックが送信されます。コメントを移動して、このエラーを回避します。次に例を示します。

```
CREATE OR REPLACE PROCEDURE
2  /* HELLO */
3  HELLO AS
4  BEGIN
5  DBMS_OUTPUT.PUT_LINE('HELLO');
6  END;
7  /
```

Procedure created.

2. 文の終了記号(ピリオド、セミコロンまたはスラッシュ)の後に、コメントを挿入しないでください。たとえば、次のように入力したとします。

```
SELECT 'Y' FROM DUAL; -- TESTING
```

次のエラーが戻されます。

```
SELECT 'Y' FROM DUAL; -- TESTING
                        *
ERROR at line 1:
ORA-00911: invalid character
```

SQL*Plusでは、文の終了記号の後にテキストを記述することはできないため、コマンドは処理されません。

3. コメント行の終わり、またはSQL文またはPL/SQLブロック内のコメントの後に文の終了文字を挿入しないでください。たとえば、次のように入力したとします。

```
SELECT *  
-- COMMENT;
```

次のエラーが戻されます。

```
-- COMMENT  
      *  
ERROR at line 2:  
ORA-00923: FROM keyword not found where expected
```

セミコロンは文の終了記号として解釈されるため、不完全なSQLコマンドがSQL*Plusからサーバーに送信されて処理されるため、エラーになります。

4. SQL文またはPL/SQLブロックでは、コメントにアンパサンド(&)を使用しないでください。たとえば、次のようにスクリプトを入力したとします。

```
SELECT REGION_NAME, CITY  
/* THIS & THAT */  
FROM EMP_DETAILS_VIEW  
WHERE SALARY>12000;
```

SQL*Plusでは、アンパサンド(&)の後のテキストは置換変数として解釈されるため、この例では、変数&thatの値を求めるプロンプトが表示されます。

```
Enter value for that:  
old 2: /* THIS & THAT */  
new 2: /* THIS */
```

REGION_NAME	CITY
Americas	Seattle
Americas	Seattle
Americas	Seattle
Europe	Oxford
Europe	Oxford
Americas	Toronto

6 rows selected.

SET DEFINE OFFを使用して、置換文字をスキャンしないように設定できます。

置換文字および終了文字の詳細は、「[SET](#)」コマンドのDEFINE、SQLTERMINATORおよびSQLBLANKLINESを参照してください。

5.4 スクリプトの実行

STARTコマンドを使用すると、スクリプトが取得され、そこに含まれるコマンドが実行されます。SQLコマンド、PL/SQLブロックおよびSQL*Plusコマンドを含むスクリプトを実行するには、STARTを使用します。ファイル内に多くのコマンドを含むことができます。STARTコマンドの後に、ファイルの名前を入力します。

```
START file_name
```

デフォルトでは、ファイルに拡張子.SQLが割り当てられていると想定されています。

ノート:

Oracle Database リリース 19c、バージョン 19.3 以降、\$ (ドル)記号を含むスクリプトを実行すると、

Windows でエラーになります。これは、\$記号は Linux および Unix の環境変数を示しているためです。

次に例を示します。

```
SQL>@C:\¥User¥my$script.sql
```

LAST NAME	MONTHLY SALARY	COMMISSION %
-----------	----------------	--------------

-----	Russell	\$14,000	0.40	Partners	\$13,500	0.30		
Errazuriz	\$12,000	0.30	Cambrault	\$11,000	0.30	Zlotkey	\$10,500	0.20

@(アットマーク)コマンドを使用しても、スクリプトを実行できます。

```
@SALES
```

@および@@コマンドを使用すると、STARTコマンドと同様に、スクリプト内のコマンドが表示され実行されます。SET ECHOでの指定は、STARTコマンドと同様に、@および@@コマンドでも有効です。

SQL*Plusでコマンドが入力される様子を表示する場合は、SET ECHO ONに設定します。ECHOシステム変数は、START、@および@@コマンドで実行されるスクリプト内のコマンドの表示を制御します。ECHO変数をOFFに設定すると、コマンドは表示されません。

START、@および@@では、スクリプト内の最後のSQLコマンドまたはPL/SQLブロックがバッファ内に残ります。

例5-7 スクリプトの実行

SALES.SQLに格納されているコマンドを取得して実行するには、次のように入力します。

```
START SALES
```

SQL*PlusではファイルSALES内のコマンドが実行され、コマンドの結果が画面に表示されます。その際、ファイル内のSQL*Plusコマンドに従って、問合せ結果の書式が次のように設定されます。

5.4.1 SQL*Plus起動時のスクリプトの実行

SQL*Plus起動時にスクリプトを実行するには、次のいずれかのオプションを使用します。

- SQLPLUSコマンドの後にユーザー名、スラッシュ、空白、@およびファイル名を指定します。

```
SQLPLUS HR @SALES
```

SQL*Plusが起動し、パスワードの入力が求められ、スクリプトが実行されます。

- ファイルの1行目にユーザー名を指定します。SQLPLUSコマンドの後に@およびファイル名を指定します。SQL*Plusが起動し、パスワードの入力が求められ、ファイルが実行されます。

5.5 スクリプトのネスト

一連のスクリプトを順に実行するには、最初に、複数のSTARTコマンドを順に指定したスクリプトを作成します。それぞれのSTARTコマンドの後には、スクリプト名を続けます。次に、STARTコマンドが含まれているスクリプトを実行します。たとえば、SALESRPTというスクリプトには次のSTARTコマンドを挿入できます。

```
START Q1SALES
START Q2SALES
START Q3SALES
START Q4SALES
```

ノート:



この例では、@@コマンドが有効な場合があります。詳細は、「[@@\(二重アットマーク\)](#)」コマンドを参照してください。

5.6 リターン・コードを伴うスクリプトの終了について

EXITコマンドをスクリプトに含めると、スクリプトの終了時に値を戻すことができます。詳細は、「[EXIT](#)」コマンドを参照してください。

WHENEVER SQLERRORコマンドをスクリプトに含めると、スクリプトがSQLエラーを生成した場合に、自動的にリターン・コードを戻し、SQL*Plusを終了できます。同様に、WHENEVER OSERRORコマンドを含めると、オペレーティング・システム・エラーが発生した場合に、自動的に終了することができます。詳細は、「[WHENEVER SQLERROR](#)」コマンドおよび「[WHENEVER OSERROR](#)」コマンドを参照してください。

5.7 置換変数の定義

SQL*PlusのDEFINEコマンドを使用すると、置換変数という変数を定義して、1つのスクリプトの中で繰り返し使用できます。タイトル内で使用し、キーストロークを保存するための置換変数も(長い文字列を短い名前の変数の値として定義して)定義できます。

```
DEFINE L_NAME = "SMITH" (CHAR)
```

置換変数のすべての定義を表示するには、DEFINEのみを入力します。DEFINEを使用して明示的に定義する置換変数には、CHAR値のみ指定できます(ユーザーが変数に割り当てる値は、常にCHARデータ型として扱われます)。ACCEPTコマンドを使用すると、NUMBERデータ型の置換変数を暗黙的に定義できます。ACCEPTコマンドについては、後半でさらに学習します。

置換変数を削除するには、SQL*PlusコマンドのUNDEFINEの後に変数名を指定します。

例5-8 置換変数の定義

置換変数L_NAMEを定義し、それに値「SMITH」を指定するには、次のコマンドを入力します。

```
DEFINE L_NAME = SMITH
```

変数定義を確認するには、次のようにDEFINEの後に変数名を指定します。

```
DEFINE L_NAME
```

5.8 事前定義変数の使用について

SQL*Plusのインストール中に定義され、SQL*Plus情報を保持する9つの変数があります。これらの変数は他の変数と同様に、再定義、参照または削除できます。明示的に削除または再定義しない限り、これらの変数はどのセッションでも使用できます。

関連項目:

事前定義変数の表示およびその使用例については、「[事前定義変数](#)」を参照してください。

5.9 置換変数の使用

SALESと同様の問合せを作成し、職種がSA_MANだけでなく、様々な職種の従業員のリストを表示するとします。この場合、コマンドの実行ごとに異なる値を編集してWHERE句に挿入する方法もありますが、さらに簡単な方法があります。

WHERE句のテキストSA_MANのかわりに置換変数を使用すると、コマンド自体に値を書き込んだ場合と同じ結果が得られます。

置換変数は、前に1つまたは2つのアンパサンド(&)を付けたものです。SQL*Plusでは、コマンド内で置換変数が検出されると、そのコマンドに置換変数自体ではなく置換変数の値が含まれている場合と同様に、コマンドを実行します。

変数SORTCOLの値がJOB_IDで、変数MYTABLEの値がEMP_DETAILS_VIEWの場合のコマンド例を次に示します。

```
SELECT &SORTCOL, SALARY
FROM &MYTABLE
WHERE SALARY>12000;
```

このコマンドは、次のコマンドと同様に実行されます。

```
SELECT JOB_ID, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000;
```

5.9.1 置換変数を使用する位置および方法

置換変数は、SQLコマンドおよびSQL*Plusコマンド内の任意の位置で使用できますが、最初に入力する単語としては使用できません。SQL*Plusでは、コマンド内で未定義の置換変数が検出された場合、値の入力を求めるプロンプトが表示されます。

このプロンプトには、任意の文字列を入力できます。空白および句読点を含む文字列も入力できます。参照を含むSQLコマンドで、置換変数を引用符で囲む必要がある場所に引用符が挿入されていない場合は、プロンプトが表示されたときに引用符を挿入する必要があります。

SQL*Plusでは、キーボードからの応答が読み込まれます。これは、端末の入力または出力をファイルヘリダイレクトした場合でも同様です。端末が使用できない場合(たとえば、バッチ・モードでスクリプトを実行した場合)、SQL*Plusではリダイレクトされたファイルが使用されます。

プロンプトで値を入力すると、置換変数が含まれている行が2回表示されます。1回は入力した値に置換される前、もう1回は置換後です。この表示は、SETコマンドのVERIFY変数をOFFに設定すると非表示にできます。

```
Created file STATS
```

この時点で、次のようにスクリプトSTATSを実行します。

```
@STATS
```

値の入力を求めるプロンプトに次のように応答します。

```
Enter value for group_col: JOB_ID
old 1: SELECT  &GROUP_COL,
new 1: SELECT  JOB_ID,
Enter value for number_col: SALARY
old 2:          MAX(&NUMBER_COL) MAXIMUM
```

```

new 2:          MAX(SALARY) MAXIMUM
Enter value for table: EMP_DETAILS_VIEW
old 3: FROM      &TABLE
new 3: FROM      EMP_DETAILS_VIEW
Enter value for group_col: JOB_ID
old 4: GROUP BY &GROUP_COL
new 4: GROUP BY JOB_ID

```

次の出力が表示されます。

```

JOB_ID          MAXIMUM
-----
AC_ACCOUNT      8300
AC_MGR          12000
AD_ASST         4400
AD_PRES         24000
AD_VP           17000
FI_ACCOUNT      9000
FI_MGR          12000
HR_REP          6500
IT_PROG         9000
MK_MAN          13000
MK_REP          6000

JOB_ID          MAXIMUM
-----
PR_REP          10000
PU_CLERK        3100
PU_MAN          11000
SA_MAN          14000
SA_REP          11500
SH_CLERK        4200
ST_CLERK        3600
ST_MAN          8200

19 rows selected.

```

置換変数の直後に文字を追加する場合は、変数と文字の区切りにピリオドを使用します。次に例を示します。

```

SELECT SALARY FROM EMP_DETAILS_VIEW WHERE EMPLOYEE_ID=' &X.5' ;
Enter value for X: 20

```

これは、次のように解釈されます。

```

SELECT SALARY FROM EMP_DETAILS_VIEW WHERE EMPLOYEE_ID='205' ;

```

例5-9 置換変数の使用

数値列についてのサブグループ統計(最大値)の計算に使用するSTATSというスクリプトを、次のように作成します。

```

SELECT &GROUP_COL, MAX(&NUMBER_COL) MAXIMUM
FROM &TABLE
GROUP BY &GROUP_COL
.
SAVE STATS

```

5.9.2 値の入力を求める不要なプロンプトの回避

ファイルSTATSを拡張して、数値列の最小値、合計および平均値を含めるとします。[例5-9](#)では、値の入力を求める

SQL*PlusプロンプトがGROUP_COLに対して2回、NUMBER_COLに対して1回表示されます。それぞれのGROUP_COLまたはNUMBER_COLの前にはアンパサンドが1つ付いています。さらに3つのファンクションを、それぞれの前にアンパサンドを1つ付けてスクリプトに追加した場合は、合計で4回、数値列の値の入力を求めるプロンプトが表示されます。

グループ列および数値列の入力を求めるプロンプトが再度表示されないようにするには、STATS内でそれぞれのGROUP_COLおよびNUMBER_COLの前にもう1つのアンパサンドを追加します。SQL*Plusでは、前に2つのアンパサンドが付いた置換変数が自動的に定義(DEFINE)され、前に1つのみのアンパサンドが付いた置換変数は定義(DEFINE)されません。変数が定義済である場合、現行のセッションでは、変数の値の入力を求めるプロンプトは表示されません。

```
SELECT  &GROUP_COL,
MAX(&NUMBER_COL) MAXIMUM
FROM    &TABLE
GROUP BY &GROUP_COL

2

2* MAX(&NUMBER_COL) MAXIMUM

APPEND ,

2* MAX(&NUMBER_COL) MAXIMUM,

CHANGE/ &/&&

2* MAX(&&NUMBER_COL) MAXIMUM,

1

3 i

MIN (&&NUMBER_COL) MINIMUM,

4 i

SUM (&&NUMBER_COL) TOTAL,

5 i

AVG (&&NUMBER_COL) AVERAGE

6 i

1

1* SELECT  &GROUP_COL,

CHANGE/ &/&&

1* SELECT  &&GROUP_COL,

7

7* GROUP BY &GROUP_COL

CHANGE/ &/&&/

7* GROUP BY &&GROUP_COL
```

```
SAVE STATS2
```

```
Created file STATS2
```

最後に、スクリプトSTATS2を実行し、プロンプトに次のように応答します。

```
START STATS2
```

```
Enter value for group_col: JOB_ID
```

```
Enter value for number_col: SALARY
```

```
Enter value for table: EMP_DETAILS_VIEW
```

次の出力が表示されます。

JOB_ID	MAXIMUM	MINIMUM	TOTAL	AVERAGE
AC_ACCOUNT	8300	8300	8300	8300
AC_MGR	12000	12000	12000	12000
AD_ASST	4400	4400	4400	4400
AD PRES	24000	24000	24000	24000
AD_VP	17000	17000	34000	17000
FI_ACCOUNT	9000	6900	39600	7920
FI_MGR	12000	12000	12000	12000
HR_REP	6500	6500	6500	6500
IT_PROG	9000	4200	28800	5760
MK_MAN	13000	13000	13000	13000
MK_REP	6000	6000	6000	6000

JOB_ID	MAXIMUM	MINIMUM	TOTAL	AVERAGE
PR_REP	10000	10000	10000	10000
PU_CLERK	3100	2500	13900	2780
PU_MAN	11000	11000	11000	11000
SA_MAN	14000	10500	61000	12200
SA_REP	11500	6100	250500	8350
SH_CLERK	4200	2500	64300	3215
ST_CLERK	3600	2100	55700	2785
ST_MAN	8200	5800	36400	7280

```
19 rows selected.
```

NUMBER_COLおよびGROUP_COLの値の入力を求めるプロンプトは、1回のみ表示されます。現行セッションでSTATS2を再実行する場合、TABLEの入力を求めるプロンプトは表示されます(この変数は、名前に1つのアンパサンドが付いていて定義(DEFINE)されないため)が、GROUP_COLまたはNUMBER_COLの入力を求めるプロンプトは表示されません(これらの変数は、名前に二重アンパサンドが付いていて定義(DEFINE)されるため)。

先へ進む前に、次のように入力してシステム変数VERIFYの設定をONに戻します。

```
SET VERIFY ON
```

例5-10 二重アンパサンドの使用方法

二重アンパサンドを使用してスクリプトSTATSを拡張した後にそのスクリプトを実行するには、まず、次のように入力して置換の前後に各行が表示されないようにします。

```
SET VERIFY OFF
```

この時点で、次のコマンドを入力してSTATSを取得し、編集します。

```
GET STATS
```


5.9.3 制限

置換変数はバッファ編集コマンドのAPPEND、CHANGE、DELおよびINPUTには使用できません。また、置換が意味をなさない他のコマンドにも使用できません。バッファ編集コマンドのAPPEND、CHANGEおよびINPUTでは、「&」または「&&」で始まるテキストは、他のテキスト文字列と同様に単なる文字列として扱われます。

5.9.4 システム変数

SQL*PlusのSETコマンドとともに指定するシステム変数で、置換変数に影響するものを次に示します。

システム変数	置換変数への影響
SET CONCAT	置換変数またはパラメータの名前と、その置換変数またはパラメータの直後の文字とを区切る文字を定義します。デフォルトではピリオド(.)です。
SET DEFINE	置換文字(デフォルトではアンパサンド(&))を定義し、置換を ON/OFF にします。
SET ESCAPE	置換文字の前で使えるエスケープ文字を定義します。エスケープ文字が検出されると、置換文字は変数置換の要求ではなく、通常の文字として扱われます。デフォルトのエスケープ文字は円記号(¥)です。
SET NUMFORMAT	数値を表示するためのデフォルトの書式を設定します(数値置換変数による表示も含む)。
SET NUMWIDTH	数値を表示するためのデフォルトの幅を設定します(数値置換変数による表示も含む)。
SET VERIFY ON	置換の前後にスクリプトの各行を表示します。

システム変数の詳細は、「[SET](#)」を参照してください。

5.10 STARTコマンドを使用したパラメータの受渡し方法

置換変数に関連する値の入力を求めるプロンプトは、STARTコマンドを使用してスクリプト内のパラメータに値を渡すと回避できます。

この操作には、置換変数のかわりにスクリプト内でアンパサンド(&)およびその後につけて数値を指定します。このスクリプトを実行するたびに、STARTによって、ファイル内の各&1がSTARTコマンドのファイル名の後の最初の値(引数)に置換され、各&2が2番目の値に置換されます。

たとえば、MYFILEというスクリプトに次のコマンドを挿入できます。

```
SELECT * FROM EMP_DETAILS_VIEW
WHERE JOB_ID=' &1'
AND SALARY=' &2' ;
```

次のSTARTコマンドでは、スクリプトMYFILEの&1はPU_CLERKに置換され、&2は3100に置換されます。

```
START MYFILE PU_CLERK 3100
```

STARTコマンドに引数を使用した場合、SQL*Plusでは、スクリプト内の各パラメータが適切な引数の値でDEFINEされます。

```

1  COLUMN LAST_NAME HEADING 'LAST NAME'
2  COLUMN SALARY HEADING 'MONTHLY SALARY' FORMAT $99,999
3  COLUMN COMMISSION_PCT HEADING 'COMMISSION %' FORMAT 90.90
4  SELECT LAST_NAME, SALARY, COMMISSION_PCT
5  FROM EMP_DETAILS_VIEW
6* WHERE JOB_ID='SA_MAN'

6

6* WHERE JOB_ID='SA_MAN'

CHANGE /SA_MAN/&1

6* WHERE JOB_ID='&1'

SAVE ONEJOB

Created file ONEJOB

```

この時点で、次のようにパラメータSA_MANを使用してこのコマンドを実行します。

```
START ONEJOB SA_MAN
```

SQL*Plusでは、次のように、パラメータが含まれているSQLコマンドの行が、パラメータがその値に置換される前および置換された後に表示され、その後に結果が表示されます。

```
old 3: WHERE JOB_ID='&1'
new 3: WHERE JOB_ID='SA_MAN'
```

LAST NAME	MONTHLY SALARY	COMMISSION %
Russell	\$14,000	0.40
Partners	\$13,500	0.30
Errazuriz	\$12,000	0.30
Gambault	\$11,000	0.30
Zlotkey	\$10,500	0.20

スクリプト内で複数のパラメータを使用できます。また、1つのスクリプト内でそれぞれのパラメータを何回でも参照でき、複数のパラメータをどのような順序でも挿入できます。

パラメータを使用できない場合、RUNまたはスラッシュ(/)を使用してコマンドを実行すると、かわりに置換変数を使用できます。

先へ進む前に、次のコマンドを入力して、列を元のヘッダーに戻します。

```
CLEAR COLUMN
```

例5-11 STARTを使用したパラメータの受渡し方法

表示する職種をパラメータで指定する新しいスクリプトをSALESに基づいて作成するには、次のように入力します。

```
GET SALES
```

5.11 ユーザーとの対話について

PROMPT、ACCEPTおよびPAUSEという3つのSQL*Plusコマンドは、エンド・ユーザーとの対話に有効です。これらのコマンドを使用すると、画面へのメッセージの送信およびユーザーからの入力([Return]キーを押すなど)の受信ができます。PROMPTおよびACCEPTを使用して、SQL*Plusで置換変数用に自動生成される値の入力を求めるプロンプトのカスタマイズもできま

す。

5.11.1 置換変数値の受信

PROMPTおよびACCEPTを使用すると、エンド・ユーザーへのメッセージの送信およびエンド・ユーザーからの入力値の受信ができます。PROMPTは、指定したメッセージを画面に表示して、ユーザーに指示または情報を与えるコマンドです。ACCEPTは、ユーザーに対して値の入力を求めるプロンプトを表示し、入力された値を指定した置換変数に格納するコマンドです。値の入力を求めるプロンプトを複数行にわたって表示する場合は、PROMPTをACCEPTと組み合わせて使用します。

```
Created file PROMPT1.sql
```

TTITLEコマンドは、レポートの一番上のタイトルを設定します。TTITLEコマンドの詳細は、[ページおよびレポートのタイトルとサイズの定義について](#)を参照してください。

最後に、スクリプトを実行し、タイトルの入力を求めるプロンプトに次のように応答します。

```
START PROMPT1
```

```
Enter a title of up to 30 characters
```

```
Title: Department Report
```

Department	Report	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
145	John		Russell		14000
146	Karen		Partners		13500
147	Alberto		Errazuriz		12000
148	Gerald		Cambrault		11000
149	Eleni		Zlotkey		10500

先へ進む前に、次のように入力して、TTITLEコマンドをOFFにします。

```
TTITLE OFF
```

例5-12 入力のプロンプトおよびアクセプト

ユーザーにレポートのタイトルを入力するよう指示し、その入力値を変数MYTITLEに格納してその後の問合せで使用可能にするには、まず、次のように入力してバッファをクリアします。

```
CLEAR BUFFER
```

次に、スクリプトを次のとおり設定し、PROMPT1という名前で保存します。

```
PROMPT Enter a title of up to 30 characters
```

```
ACCEPT MYTITLE PROMPT 'Title: '
```

```
TTITLE LEFT MYTITLE SKIP 2
```

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY
```

```
FROM EMP_DETAILS_VIEW
```

```
WHERE JOB_ID='SA_MAN'
```

```
SAVE PROMPT1
```

5.11.2 置換変数の入力を求めるプロンプトのカスタマイズ

置換変数値の入力を求めるプロンプトをカスタマイズする場合は、次の例に示すように、PROMPTおよびACCEPTを置換変数と組み合わせて使用します。

```
Enter a valid employee ID
```

```
For Example 145, 206
```

```
Employee ID. :
```

```
205
```

```
old 3: WHERE EMPLOYEE_ID=&ENUMBER
```

```
new 3: WHERE EMPLOYEE_ID=      205
```

```
Department Report
```

FIRST_NAME	LAST_NAME	SALARY
Shelley	Higgins	12000

文字ではなく数字を入力する必要があります。ACCEPTコマンド内で変数名の後にNUMBERを指定したため、数値以外の値はSQL*Plusで受け入れられません。

「Employee ID.」の入力を求めるプロンプトに、数字のかわりに文字を入力してみてください。エラー・メッセージが表示され、正しい数字の再入力を求めるプロンプトが次のように表示されます。

```
START PROMPT2
```

SQL*PlusでEmployee IDの入力を求めるプロンプトが表示されると、次のように数字のかわりにoneという単語を入力します。

```
Enter a valid employee ID  
For Example 145, 206
```

```
Employee ID. :
```

```
one
```

```
SP2-0425: "one" is not a valid number
```

例5-13 PROMPTおよびACCEPTを置換変数と組み合わせて使用する方法

[例5-12](#)で示したように、SQL*Plusでは、置換変数を使用する場合、値の入力を求めるプロンプトが自動生成されます。このプロンプトを別のプロンプトに置き換えるには、置換変数を参照する問合せが含まれているスクリプトにPROMPTおよびACCEPTを挿入します。まず、次のコマンドを使用してバッファをクリアします。

```
CLEAR BUFFER
```

目的のファイルを作成するには、次のコマンドを入力します。

```
INPUT  
PROMPT Enter a valid employee ID  
PROMPT For Example 145, 206  
ACCEPT ENUMBER NUMBER PROMPT 'Employee ID. :'  
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMP_DETAILS_VIEW  
WHERE EMPLOYEE_ID=&ENUMBER;
```

PROMPT2という名前でこのファイルを保存します。次に、このスクリプトを実行します。PROMPTおよびACCEPTで指定されたテキストが使用され、ENUMBERの値の入力を求めるプロンプトが次のように表示されます。

```
START PROMPT2
```

Employee IDの入力を求めるプロンプトが、次のように表示されます。

5.11.3 メッセージの送信および入力としての[Return]のアクセプト

ユーザーの画面にメッセージを表示し、ユーザーがそのメッセージを読んだ後、[Return]を押すように指示するには、SQL*Plus コマンドのPAUSEを使用します。たとえば、スクリプトに次のような行を挿入します。

```
PROMPT Before continuing, make sure you have your account card.  
PAUSE Press RETURN to continue.
```

5.11.4 画面のクリア

レポートを表示する前に(または任意の時点で)画面を消去する場合は、スクリプト内の適切な場所に、次の書式でSQL*PlusのCLEARコマンドにSCREEN句を付けて挿入します。

```
CLEAR SCREEN
```

次の項へ進む前に、次のコマンドを入力して、すべての列を元の書式およびヘッダーにリセットします。

```
CLEAR COLUMNS
```

5.12 バインド変数の使用方法について

バインド変数は、SQL*Plusで作成し、PL/SQLまたはSQLで参照する変数です。SQL*Plusでバインド変数を作成した場合、その変数はPL/SQLサブプログラムの中で宣言した変数と同様に使用でき、SQL*Plusからもアクセスできます。データを保持する入力バインド変数としてバインド変数を使用し、後でPL/SQL文またはSQL文で使用してデータをデータベースに挿入できます。新規に定義した変数に値を割り当てることができます。この変数に割り当てられた値は、後で文に使用できます。

バインド変数はSQL*Plusから認識できるため、SQL*Plusでのバインド変数の値の表示、およびSQL*Plusで実行するPL/SQLサブプログラムでのバインド変数の参照が可能です。

5.12.1 バインド変数の作成

バインド変数は、SQL*PlusでVARIABLEコマンドを使用して作成します。次に例を示します。

```
VARIABLE ret_val NUMBER
```

このコマンドによって、データ型がNUMBERで、ret_valという名前のバインド変数が作成されます。詳細は、「[VARIABLE](#)」コマンドを参照してください。(あるセッションで作成したすべてのバインド変数を表示するには、引数を付けずにVARIABLEと入力します。)

5.12.2 バインド変数の参照

PL/SQLでバインド変数を参照するには、コロン(:)を入力し、その直後に変数の名前を指定します。次に例を示します。

```
:ret_val := 1;
```

SQL*Plusでこのバインド変数を変更するには、PL/SQLブロックを入力する必要があります。次に例を示します。

```
BEGIN  
  :ret_val:=4;  
END;  
/
```

```
PL/SQL procedure successfully completed.
```

このコマンドは、`ret_val`という名前のバインド変数に値を割り当てます。

5.12.3 バインド変数の表示

SQL*Plusでバインド変数の値を表示するには、SQL*PlusのPRINTコマンドを使用します。次に例を示します。

```
PRINT RET_VAL
```

```
RET_VAL
-----
4
```

このコマンドは、`ret_val`という名前のバインド変数を表示します。バインド変数の表示の詳細は、「[PRINT](#)」を参照してください。

5.12.4 入力バインドの実行

入力バインディングの変数に値を割り当てることができます。

```
SQL> variable abc number=123
```

```
SQL> select :abc from dual;
```

```
ABC
-----
123
```

```
SQL>
```

```
SQL> create table mytab (col1 number, col2 varchar2(10));
```

```
Table created.
```

```
SQL> var abc number=123
```

```
SQL> var xyz varchar2(10)='test'
```

```
SQL> insert into mytab values (:abc, :xyz);
```

```
1 row created.
```

```
SQL> select * from mytab;
```

```
COL1 COL2
-----
123 test
```

```
SQL>
```

詳細は、「[VARIABLE](#)」コマンドを参照してください。

5.13 REFCURSORバインド変数の使用方法

SQL*PlusのREFCURSORバインド変数を使用すると、PL/SQLブロックに含まれるSELECT文の結果をSQL*Plusからフェッチし、その書式を設定できます。

REFCURSORバインド変数は、ストアド・プロシージャ内のPL/SQLカーソル変数の参照にも使用できます。この機能を使用して、データベース内にSELECT文を格納し、それらの文をSQL*Plusから参照できます。

REFCURSORバインド変数は、ストアド・ファンクションから戻すこともできます。

```
PL/SQL procedure successfully completed.
```

この時点で、SELECT文の結果をSQL*PlusでPRINTコマンドを使用して表示できます。

```
PRINT employee_info
```

EMPLOYEE_ID	SALARY
145	14000
146	13500
147	12000
148	11000
149	10500

PRINT文は、カーソルもクローズします。結果を再出力するには、PRINTを使用する前にPL/SQLブロックを再実行する必要があります。

```
Package created.
```

次に、OPEN... FOR SELECT文が含まれたストアード・プロシージャを作成します。

```
CREATE OR REPLACE PACKAGE BODY EmpPack AS
  PROCEDURE EmpInfoRpt (emp_cv IN OUT EmpInfoTyp) AS
  BEGIN
    OPEN emp_cv FOR SELECT EMPLOYEE_ID, SALARY
    FROM EMP_DETAILS_VIEW
    WHERE JOB_ID='SA_MAN' ;
  END;
END;
/
```

```
Procedure created.
```

SQL*Plusバインド変数をパラメータとして、プロシージャを実行します。

```
VARIABLE cv REFCURSOR
EXECUTE EmpPack.EmpInfoRpt(:cv)
```

```
PL/SQL procedure successfully completed.
```

この時点でバインド変数を出力します。

```
PRINT cv
```

EMPLOYEE_ID	SALARY
145	14000
146	13500
147	12000
148	11000
149	10500

このプロシージャは、同じまたは別のREFCURSORバインド変数を使用して何度でも実行できます。

```
VARIABLE pcv REFCURSOR
EXECUTE EmpInfo_rpt(:pcv)
```

```
PL/SQL procedure successfully completed.
```

```
PRINT pcv
```

EMPLOYEE_ID	SALARY
-------------	--------

145	14000
146	13500
147	12000
148	11000
149	10500

Function created.

このファンクションを実行します。

```
VARIABLE rc REFCURSOR
EXECUTE :rc := EmpInfo_fn
```

PL/SQL procedure successfully completed.

この時点でバインド変数を出力します。

```
PRINT rc
```

EMPLOYEE_ID	SALARY
145	14000
146	13500
147	12000
148	11000
149	10500

このファンクションは、同じまたは別のREFCURSORバインド変数を使用して何度でも実行できます。

```
EXECUTE :rc := EmpInfo_fn
```

PL/SQL procedure successfully completed.

例5-14 REFCURSORバインド変数の作成、参照および表示

REFCURSORバインド変数を作成、参照および表示するには、最初にREFCURSORデータ型のローカル・バインド変数を宣言します。

```
create procedure p4 as
  c1 sys_refcursor;
  c2 sys_refcursor;
begin
  open c1 for SELECT * FROM DEPT;
  dbms_sql.return_result(c1);
  open c2 for SELECT * FROM EMP;
  dbms_sql.return_result(c2);
end;
/
```

次に、このバインド変数を使用するPL/SQLブロックをOPEN... FOR SELECT文に入力します。この文は、カーソル変数をオープンし、問合せを実行します。OPENコマンドおよびカーソル変数の詳細は、「[OPEN文](#)」を参照してください。

この例では、SQL*Plusの*employee_info*バインド変数をカーソル変数にバインドします。

```
BEGIN
OPEN :employee_info FOR SELECT EMPLOYEE_ID, SALARY
FROM   EMP_DETAILS_VIEW WHERE JOB_ID=' SA_MAN' ;
END;
/
```


例5-15 ストアド・プロシージャでのREFCURSOR変数の使用方法

REFCURSORバインド変数は、パラメータとしてプロシージャに渡されます。パラメータは、REF CURSOR型です。最初に、型を定義します。

```
CREATE OR REPLACE PACKAGE EmpPack AS
  TYPE EmpInfoTyp IS REF CURSOR;
  PROCEDURE EmpInfoRpt (emp_cv IN OUT EmpInfoTyp);
END EmpPack;
/
```

例5-16 ストアド・ファンクションでのREFCURSOR変数の使用方法

次の形式で、OPEN... FOR SELECT文を含むストアド・ファンクションを作成します。

```
CREATE OR REPLACE FUNCTION EmpInfo_fn RETURN -
cv_types.EmpInfo IS
resultset cv_types.EmpInfoTyp;
BEGIN
OPEN resultset FOR SELECT EMPLOYEE_ID, SALARY
FROM EMP_DETAILS_VIEW
WHERE JOB_ID='SA_MAN';
RETURN(resultset);
END;
/
```

5.14 PL/SQLブロックに含まれるSELECT文の結果の反復的なフェッチ

SQL*Plusでは、PL/SQLブロックまたはストアド・プロシージャに含まれるSELECT文の結果を反復的にフェッチし、その書式を設定できます。ローカルなREFCURSOR変数を定義する必要はありません。

SELECT文の結果が表示されます。

ResultSet #1

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

4 rows selected

ResultSet #2

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	05-APR-81	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	09-MAY-81	1100		

14 rows selected

例5-17 PL/SQLプロシージャの作成

2つの文をコールするPL/SQLプロシージャP4を作成します。

```
create procedure p4 as
  c1 sys_refcursor;
  c2 sys_refcursor;
begin
  open c1 for SELECT * FROM DEBT;
  dbms_sql.return_result(c1);
  open c2 for SELECT * FROM EMP;
  dbms_sql.return_result(c2);
end;
/
Procedure created.
```

次に、このプロシージャを実行して、プロシージャ内のSELECT文の結果を反復的に取得します。

```
exec p4
PL/SQL procedure successfully completed.
```

6 SQL*Plusレポートの書式設定

この章では、最終的なレポートを生成するために問合せ結果の書式を設定する方法について説明します。ここでは、HTML出力を除く次の項目について説明します。

- [列の書式設定について](#)
- [間隔の設定およびサマリー行によるレポートの明確化について](#)
- [ページおよびレポートのタイトルとサイズの定義について](#)
- [問合せ結果の格納および印刷について](#)

この章を読むときは、コンピュータで、示されている例を実際に試してみてください。始める前に、「[SQL*Plusのクイック・スタート](#)」で説明したHRサンプル・スキーマへのアクセス権限があることを確認してください。

6.1 列の書式設定について

SQL*PlusのCOLUMNコマンドを使用して、列ヘッダーを変更し、問合せ結果の列データの書式を再設定できます。

6.1.1 列ヘッダーの変更について

列ヘッダーを表示する場合は、デフォルトのヘッダーを使用するか、またはデフォルトのヘッダーをCOLUMNコマンドで変更できます。次の各項では、デフォルトのヘッダーの導出方法およびCOLUMNコマンドを使用してデフォルトのヘッダーを変更する方法について説明します。詳細は、「[COLUMN](#)」コマンドを参照してください。

6.1.1.1 デフォルトのヘッダー

SQL*Plusでは、問合せ結果を表示するときに、列または式の名前がデフォルトの列ヘッダーとして使用されます。ただし、多くの場合、列名は短く暗号のようで、式名は理解するのが困難です。

6.1.1.2 デフォルトのヘッダーの変更

次の形式でCOLUMNコマンドのHEADING句を使用すると、さらに有効な列ヘッダーを定義できます。

```
COLUMN column_name HEADING column_heading
```

LAST NAME	MONTHLY SALARY	COMMISSION
Russell	14000	.4
Partners	13500	.3
Errazuriz	12000	.3
Cambrault	11000	.3
Zlotkey	10500	.2

ノート:



新しいヘッダーは、別のヘッダーを入力するか、列の書式をリセットするか、または SQL*Plus を終了するまで有効です。

列ヘッダーを複数の単語に変更するには、COLUMNコマンドの入力時に、新しいヘッダーを一重引用符または二重引用符で

囲みます。列ヘッダーを複数の行に表示するには、新しい行を開始する位置に垂直バー(|)を使用します。(SETコマンドの HEADSEP変数の設定を変更すると、垂直バー以外の文字を使用できます。詳細は、「[SET](#)」コマンドを参照してください。)

LAST NAME	MONTHLY SALARY	COMMISSION
Russell	14000	.4
Partners	13500	.3
Errazuriz	12000	.3
Cambrault	11000	.3
Zlotkey	10500	.2

LAST NAME	MONTHLY SALARY	COMMISSION
=====	=====	=====
Russell	14000	.4
Partners	13500	.3
Errazuriz	12000	.3
Cambrault	11000	.3
Zlotkey	10500	.2

この時点で、次のように下線文字をダッシュに戻してください。

```
SET UNDERLINE '-'
```

ノート:



ダッシュは、引用符で囲む必要があります。囲まない場合、ダッシュは、次の行へコマンドを継続することを示すハイフンとして解釈されます。

例6-1 列ヘッダーの変更

LAST_NAME、SALARYおよびCOMMISSION_PCTに新しいヘッダーを指定してEMP_DETAILS_VIEWからレポートを生成するには、次のコマンドを入力します。

```
COLUMN LAST_NAME HEADING 'LAST NAME'
COLUMN SALARY HEADING 'MONTHLY SALARY'
COLUMN COMMISSION_PCT HEADING 'COMMISSION'
SELECT LAST_NAME, SALARY, COMMISSION_PCT
FROM EMP_DETAILS_VIEW
WHERE JOB_ID='SA_MAN';
```

例6-2 列ヘッダーの分割

SALARYおよびLAST_NAME列にそれぞれMONTHLY SALARYおよびLAST NAMEというヘッダーを付け、新しいヘッダーを2行に分割するには、次のように入力します。

```
COLUMN SALARY HEADING 'MONTHLY|SALARY'
COLUMN LAST_NAME HEADING 'LAST|NAME'
```

この時点で、次のように/(スラッシュ)コマンドで問合せを再実行します。

```
/
```

例6-3 下線文字の設定

ヘッダーに下線を付けるために使用する文字を等号に変更し、問合せを再実行するには、次のコマンドを入力します。

```
SET UNDERLINE =  
/
```

6.1.2 NUMBER列の書式設定について

NUMBER列を表示する場合は、SQL*Plusのデフォルトの表示幅を使用するか、またはCOLUMNコマンドを使用してデフォルトの表示幅を変更します。次の各項では、デフォルトの表示およびCOLUMNコマンドでデフォルトを変更する方法について説明します。書式モデルは、新しい書式モデルを入力するか、または次のコマンドを使用して列の書式をリセットするまで有効です。

```
COLUMN column_name CLEAR
```

または、SQL*Plusを終了するまで有効です。

6.1.2.1 デフォルトの表示

NUMBER列の幅は、ヘッダーの幅か、FORMATの幅に符号用の1つの空白を加えた幅のどちらか大きい方に等しくなります。明示的にFORMATを使用しない場合、列の幅は、常にSET NUMWIDTHの値以上になります。

通常、SQL*Plusでは、数値は、SETコマンドのNUMWIDTH変数の値で決まる標準表示幅(通常は10)を最大幅として、必要なだけの桁数で表示されます。可能な場合、SET NUMWIDTHの値より大きい数値は、許容最大文字数で切り上げられるか、または切り捨てられます。数値が大きすぎる場合は、ハッシュが表示されます。

COLUMNコマンドで書式モデルを使用すると、NUMBER列に別の書式を選択できます。書式モデルとは、列の中に数値を表示する方法を表したもので、複数の9を使用して数字の桁を表します。

6.1.2.2 デフォルト表示の変更

COLUMNコマンドを使用すると、次に示すように、書式を設定する列および使用するモデルを識別できます。

```
COLUMN column_name FORMAT model
```

列の中の数値にカンマ、ドル記号、山カッコ(負の値を囲む)、および先行0(ゼロ)を追加するには、書式モデルを使用します。値を一定の小数桁数まで丸め、負の値の右側(左側でなく)にマイナス記号を表示し、指数表記で値を表示することもできます。

単一の列に複数の書式モデルを使用するには、使用するモデルを結合して1つのCOLUMNコマンドにします([例6-4](#)を参照)。書式モデルのすべてのリストおよび詳細は、「[COLUMN](#)」を参照してください。

LAST NAME	MONTHLY SALARY	COMMISSION
Russell	\$14,000	.4
Partners	\$13,500	.3
Errazuriz	\$12,000	.3
Cambrault	\$11,000	.3
Zlotkey	\$10,500	.2

ドル記号などの他の書式を使用し、0(ゼロ)値に空白ではなく0(ゼロ)を表示する場合は、表示する値として書式モデル内で0(ゼロ)を使用します。

例6-4 NUMBER列の書式設定

SALARYの表示にドル記号、カンマ、および0(ゼロ)値に対して空白ではなく数字の0(ゼロ)を使用するには、次のコマンドを入力します。

```
COLUMN SALARY FORMAT $99,990
```

この時点で、現行の問合せを再実行します。

/

6.1.3 データ型の書式設定について

データ型を表示する場合は、SQL*Plusのデフォルトの表示幅を使用するか、またはCOLUMNコマンドを使用してデフォルトの表示幅を変更します。書式モデルは、新しい書式モデルを入力するか、または次のコマンドを使用して列の書式をリセットするまで有効です。

```
COLUMN column_name CLEAR
```

または、SQL*Plusを終了するまで有効です。このマニュアルでは、データ型に次の型が含まれます。

- CHAR
- NCHAR
- VARCHAR2 (VARCHAR)
- NVARCHAR2 (NCHAR VARYING)
- DATE
- LONG
- BLOB
- BFILE
- CLOB
- NCLOB
- XMLType

6.1.3.1 デフォルトの表示

データ型列のデフォルトの幅は、データベース内の列の幅です。LONG列、BLOB列、BFILE列、CLOB列、NCLOB列またはXMLType列のデフォルトの幅は、SET LONGCHUNKSIZEまたはSET LONGのいずれか小さい方の値です。

SQL*Plusでは、書式が設定されていないDATE列のデフォルトの幅と書式は、データベースのNLS_DATE_FORMATパラメータによって決定されます。それ以外の場合、デフォルトの書式の幅はA9です。DATE列の書式設定の詳細は、「[COLUMN](#)」コマンドのFORMAT句を参照してください。

データ型のデフォルトの文字位置は、左揃えです。

6.1.3.2 デフォルト表示の変更

データ型またはDATEの表示幅は、COLUMNコマンドと、文字A(英数字の意)の後に列幅を表す数値を続けた書式モデルを使用して変更できます。

COLUMNコマンド内で、書式を設定する列および使用するモデルを次のように指定します。

```
COLUMN column_name FORMAT model
```

列ヘッダーより短い幅を指定した場合は、ヘッダーが切り捨てられます。詳細は、「[COLUMN](#)」コマンドを参照してください。

```
LAST   MONTHLY  
NAME   SALARY  COMMISSION  
-----
```

```

Russ  $14,000      .4
ell

Part  $13,500     .3
ners

Erra  $12,000     .3
zuri
z

LAST  MONTHLY
NAME  SALARY  COMMISSION
-----
Camb  $11,000     .3
raul
t

Zlot  $10,500     .2
key

```

SETコマンドのWRAP変数をON (デフォルト値)に設定した場合、[例6-5](#)に示すように、従業員名は4文字目の後で次の行へ折り返されます。WRAPをOFFに設定した場合、名前は4文字目の後で切り捨てられます。

システム変数WRAPは、すべての列を制御します。列に対するWRAPの設定は、COLUMNコマンドのWRAPPED、WORD_WRAPPEDおよびTRUNCATED句を指定すると上書きできます。これらの句の詳細は、「[COLUMN](#)」コマンドを参照してください。この章の後半でCOLUMNコマンドのWORD_WRAPPEDを使用します。

NCLOB、BLOB、BFILEまたはマルチバイトCLOBの各列をWORD_WRAPPEDオプションで書式設定することはできません。NCLOB、BLOB、BFILEまたはマルチバイトCLOBの各列をCOLUMN WORD_WRAPPEDで書式設定した場合、列データはCOLUMN WRAPPEDが適用されたかのように動作します。

ノート:



列のヘッダーは、WRAP の設定、またはどの COLUMN コマンド句の設定にも関係なく切り捨てられます。

この時点で、次のように列を元の書式に戻します。

```

COLUMN LAST_NAME FORMAT A10

Building
-----
Owned

```

createXML、extract、text、getStringValの各ファンクション、およびXMLTypeデータの作成と操作の詳細は、『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』を参照してください。

例6-5 文字列の書式設定

LAST_NAME列の幅を4文字に設定し、現行の間合せを再実行するには、次のように入力します。

```

COLUMN LAST_NAME FORMAT A4
/

```

例6-6 XMLType列の書式設定

XMLType列の書式設定方法について説明する前に、XMLType列の定義を含む表を作成し、その表にデータを挿入する必要があります。XMLType列は、他のユーザー定義列と同様の方法で作成できます。XMLType列を含む表を作成するには、次のように入力します。

```
CREATE TABLE warehouses (  
  warehouse_id NUMBER(3),  
  warehouse_spec SYS.XMLTYPE,  
  warehouse_name VARCHAR2 (35),  
  location_id NUMBER(4));
```

warehouse_idおよびwarehouse_specの値を含む新しいレコードを新しいwarehouses表に挿入するには、次のように入力します。

```
INSERT into warehouses (warehouse_id, warehouse_spec)  
VALUES (100, sys.XMLTYPE.createXML(  
  '<Warehouse whNo="100">  
    <Building>Owned</Building>  
  </Warehouse>'));
```

XMLType列の幅を20文字に設定し、XMLType列を選択するには、次のように入力します。

```
COLUMN Building FORMAT A20  
SELECT  
  w. warehouse_spec. extract('/Warehouse/Building/text()').getStringVal()  
  "Building"  
FROM warehouses w;
```

6.1.4 列の表示属性のコピー

複数の列に同じ表示属性を設定する場合は、COLUMNコマンドのLIKE句を使用して入力すると、コマンドの長さを短くすることができます。LIKE句を使用すると、SQL*Plusで、前に定義した列の表示属性が新しい列にコピーされます。ただし、同じコマンドの別の句によって加えられた変更はコピーされません。

LAST NAME	MONTHLY SALARY	BONUS
Russell	\$14,000	\$0
Partners	\$13,500	\$0
Errazuriz	\$12,000	\$0
Cambrault	\$11,000	\$0
Zlotkey	\$10,500	\$0

例6-7 列の表示属性のコピー

COMMISSION_PCT列に、SALARYに指定した表示属性と同じ表示属性を指定し、別のヘッダーを指定するには、次のコマンドを入力します。

```
COLUMN COMMISSION_PCT LIKE SALARY HEADING BONUS
```

次のように問合せを再実行します。

```
/
```

6.1.5 列の表示属性の表示およびリセット

列の現行の表示属性を表示するには、次に示すように、COLUMNコマンドおよびその後列名のみを指定します。

```
COLUMN column_name
```


すべての列の現行の表示属性を表示するには、次のように、後ろに列名または句を指定せずにCOLUMNコマンドを入力します。

```
COLUMN
```

列の表示属性をデフォルト値にリセットするには、次に示すようにCOLUMNコマンドのCLEAR句を使用します。

```
COLUMN column_name CLEAR
```

```
columns cleared
```

例6-8 列の表示属性をデフォルトにリセットする方法

すべての列の表示属性をデフォルトにリセットするには、次のように入力します。

```
CLEAR COLUMNS
```

6.1.6 列の表示属性の非表示および表示について

特定の列に指定した表示属性を非表示および表示できます。列の表示属性を非表示にするには、次の形式でCOLUMNコマンドを入力します。

```
COLUMN column_name OFF
```

OFF句を使用すると、SQL*Plusで、列に対してデフォルトの表示属性が使用されますが、COLUMNコマンドで定義された属性は削除されません。COLUMNコマンドで定義した属性を表示するには、次のようにON句を使用します。

```
COLUMN column_name ON
```

6.1.7 折り返した列の値の後で文字の行を出力する方法

前述のとおり、SQL*Plusのデフォルトでは、列の値が列幅に収まらない場合、その値は追加行に折り返されます。折り返された出力行の後(または各行の後)にレコード・セパレータ文(字または空白の1行)を挿入する場合は、SETコマンドのRECSEPおよびRECSEPCHAR変数を使用します。

RECSEPを使用して、文字の行を出力するタイミングを決定できます。RECSEPをEACHに設定すると、各行の後に文字の行が出力されます。WRAPPEDに設定すると、折返し行の後に出力されます。OFFに設定すると、出力されません。RECSEPのデフォルトの設定は、WRAPPEDです。

RECSEPCHARは、各行に出力される文字を設定します。RECSEPCHARには、任意の文字を設定できます。

列の値が次の行に折り返される場合は、すべての単語を次の行に折り返すこともできます。この場合は、次に示すように、COLUMNコマンドのWORD_WRAPPED句を使用します。

```
COLUMN column_name WORD_WRAPPED
```

LAST_NAME	JOB_TITLE	CITY
King	President	Seattle
Kochhar	Administration Vice President	Seattle
De Haan	Administration Vice President	Seattle
Russell	Sales Manager	Oxford
Partners	Sales Manager	Oxford

```
Hartstein           Marketing Manager    Toronto
```

```
6 rows selected.
```

RECSEPをEACHに設定した場合、各行の後(この例では各部門の後)に文字の行が出力されます。

先へ進む前に、次のようにRECSEPをOFFに設定して、レコード・セパレータを出力しないようにします。

```
SET RECSEP OFF
```

例6-9 折り返した列の値の後に文字の行を出力する方法

列の値を折り返した後にダッシュの行を出力するには、次のコマンドを入力します。

```
SET RECSEP WRAPPED  
SET RECSEPCHAR "--"
```

最後に、次の問合せを入力します。

```
SELECT LAST_NAME, JOB_TITLE, CITY  
FROM EMP_DETAILS_VIEW  
WHERE SALARY>12000;
```

この時点で、次のようにJOB_TITLE列の幅を制限し、必要な場合は、SQL*Plusですべての単語を次の行に折り返します。

```
COLUMN JOB_TITLE FORMAT A20 WORD_WRAPPED
```

次のように問合せを実行します。

```
/
```

6.2 間隔の設定およびサマリー行によるレポートの明確化について

SQLのSELECTコマンドでORDER BY句を使用する場合、順序を指定された列(または式)に同じ値がある行は、同時に表示されます。SQL*PlusのBREAKおよびCOMPUTEコマンドを使用して、レコードのサブセットを作成し、各サブセットの後に空白またはサマリー行を追加すると、出力をユーザーにとってより有効なものにできます。

BREAKコマンド内に指定する列は、ブレイク列と呼ばれます。ORDER BY句にブレイク列を挿入すると、意味のあるレコードのサブセットを作成できます。その後、同じBREAKコマンド内でサブセットに書式設定を追加でき、COMPUTEコマンドにブレイク列を指定して(合計、平均などを含む)サマリー行を追加できます。

```
SELECT DEPARTMENT_ID, LAST_NAME, SALARY  
FROM EMP_DETAILS_VIEW  
WHERE SALARY > 12000  
ORDER BY DEPARTMENT_ID;
```

DEPARTMENT_ID	LAST_NAME	SALARY
20	Hartstein	13000
80	Russell	14000
80	Partners	13500
90	King	24000
90	Kochhar	17000
90	De Haan	17000

```
6 rows selected.
```

このレポートをさらに有効に使用するには、BREAKを使用してDEPARTMENT_IDをブレイク列として設定します。BREAKを

使用すると、DEPARTMENT_ID値の重複を抑止できます。また、部門と部門の間で、空白行の入力または改ページができます。BREAKをCOMPUTEと組み合わせて使用すると、各部門およびすべての部門について給与の合計を含むサマリー行を計算して出力できます。また、平均、最大、最小、標準偏差、平方偏差または行数を含むサマリー行も出力できます。

6.2.1 ブレーク列内の値の重複の抑止

BREAKコマンドをデフォルトで使用すると、指定された列または式の値の重複を抑止できます。したがって、ORDER BY句で指定した列の値が重複しないようにするには、次のようにBREAKコマンドを最も単純な形式で使用します。

```
BREAK ON break_column
```

ノート:



BREAK コマンドに列または式を指定する場合は、必ず、その列または式を指定した ORDER BY 句を使用してください。ORDER BY 句を使用しない場合は、列の値を変更すると常にブレークが発生します。

DEPARTMENT_ID	LAST_NAME	SALARY
---------------	-----------	--------

```
----- 20 Hartstein 13000 80 Russell 14000 Partners 13500
90 King 24000 Kochhar 17000 De Haan 17000 6 rows selected.
```

例6-10 ブレーク列内の値の重複の抑止

表示する問合せ結果で重複した部門番号の表示を抑止するには、次のコマンドを入力します。

```
BREAK ON DEPARTMENT_ID;
```

次に、バッファに格納されている現行の問合せの例を示します。

```
SELECT DEPARTMENT_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000
ORDER BY DEPARTMENT_ID;
```

6.2.2 ブレーク列の値が変更されたときの空白の挿入

ブレーク列内で値が変更されると、常に、空白行の挿入または改ページができます。*n*行の空白行を挿入するには、次の形式でBREAKコマンドを使用します。

```
BREAK ON break_column SKIP n
```

ページをスキップするには、次の形式でBREAKコマンドを入力します。

```
BREAK ON break_column SKIP PAGE
```

DEPARTMENT_ID	LAST_NAME	SALARY
20	Hartstein	13000
80	Russell	14000
	Partners	13500
90	King	24000
	Kochhar	17000
	De Haan	17000

6 rows selected.

例6-11 ブレーク列の値が変更されたときの空白の挿入

部門と部門の間に1行の空白行を挿入するには、次のコマンドを入力します。

```
BREAK ON DEPARTMENT_ID SKIP 1
```

この時点で、次のように問合せを再実行します。

/

6.2.3 各行の後に空白を挿入する方法

各行の後に空白行または空白ページを挿入できます。各行の後にn行ずつスキップするには、次の形式でBREAKコマンドを使用します。

```
BREAK ON ROW SKIP n
```

各行の後に1ページずつスキップするには、次の形式でBREAKコマンドを使用します。

```
BREAK ON ROW SKIP PAGE
```

ノート:



SKIP PAGE を指定しても、NEWPAGE 0 を指定しないかぎり、物理的なページ・ブレーク文字は生成されません。

6.2.4 複数の間隔設定技法の使用方法

ORDER BY句に複数の列が存在し、各列の値の変更時に空白を挿入するとします。BREAKコマンドは、入力すると常に前のBREAKコマンドと置き換えられます。したがって、1つのレポート内で様々な間隔設定技法を使用する場合、または順序を指定された複数の列で値が変更されたときに空白を挿入する場合は、1つのBREAKコマンドに複数の列およびアクションを設定する必要があります。

Page: 1			
DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
20	MK_MAN	Hartstein	13000
Page: 2			
DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
80	SA_MAN	Russell	14000
		Partners	13500
Page: 3			
DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
90	AD_PRES	King	24000
	AD_VP	Kochhar	17000

6 rows selected.

例6-12 間隔設定技法の統合

次のように入力します。

```
SELECT DEPARTMENT_ID, JOB_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000
ORDER BY DEPARTMENT_ID, JOB_ID;
```

この時点で、DEPARTMENT_IDの値が変更されたときに1ページをスキップし、JOB_IDの値が変更されたときに1行スキップするには、次のコマンドを入力します。

```
BREAK ON DEPARTMENT_ID SKIP PAGE ON JOB_ID SKIP 1
```

SKIP PAGEが有効になったことを示すには、次のように入力して、ページ番号付きのTTITLEを作成します。

```
TTITLE COL 35 FORMAT 9 'Page:' SQL.PNO
```

この新しい問合せの実行結果は、次のように表示されます。

6.2.5 ブレーク定義の表示および取消し

先へ進む前に、次のように入力して、定義を変更せずに上部タイトルを非表示にします。

```
TTITLE OFF
```

句を指定せずにBREAKコマンドを入力すると、現行のブレーク定義を表示できます。

```
BREAK
```

CLEARコマンドにBREAKS句を指定して入力すると、現行のブレーク定義を取り消せます。

```
CLEAR BREAKS
```

すべてのスクリプトの先頭にCLEAR BREAKSコマンドを指定して、ファイル内で実行する問合せに、前回のBREAKコマンドが影響しないように設定できます。

6.2.6 ブレーク列の値が変更されたときにサマリー行を計算する方法

BREAKコマンドでレポートの行をサブセットに編成した場合、各サブセット内の行に対して様々な計算を実行できます。この計算は、SQL*PlusのCOMPUTEコマンドの関数を使用して実行します。BREAKコマンドおよびCOMPUTEコマンドを同時に指定し、次の形式で入力します。

```
BREAK ON break_column
COMPUTE function LABEL label_name OF column column column
... ON break_column
```

BREAKコマンドには、複数のブレーク列、および行のスキップなどのアクションを挿入できます。ただし、COMPUTEコマンドでONの後に指定した列は、BREAKコマンドのONの後も指定する必要があります。COMPUTEコマンドと同時に使用しているときに、BREAKコマンドに複数のブレーク列およびアクションを挿入するには、これらのコマンドを次の形式で使用します。

```
BREAK ON break_column_1 SKIP PAGE ON break_column_2 SKIP 1
COMPUTE function LABEL label_name OF column column column
... ON break_column_2
```

COMPUTEコマンドは、対応するBREAKコマンドがない場合は機能しません。

COMPUTEは、NUMBER列および(場合によっては)すべてのタイプの列に対して実行できます。詳細は、「[COMPUTE](#)」コマンドを参照してください。

次の表に、COMPUTE関数およびその機能を示します。

表6-1 COMPUTE関数

関数	機能
SUM	列内の値の合計を計算する
MINIMUM	列内の最小値を計算する
MAXIMUM	列内の最大値を計算する
AVG	列内の値の平均を計算する
STD	列内の値の標準偏差を計算する
VARIANCE	列内の値の平方偏差を計算する
COUNT	列内の値の NULL 以外の値の数を計算する
NUMBER	列内の行数を計算する

COMPUTEコマンド内で指定した関数は、OFの後およびONの前に入力したすべての列に適用されます。計算後の値は、順序が指定された列の値が変更されると、別々の行に出力されます。

ON REPORT計算およびON ROW計算のラベルは、1列目に表示されます。それ以外の場合、ラベルはON句で指定した列に表示されます。

COMPUTE LABELを使用すると、計算ラベルを変更できます。計算後の値のラベルを定義しない場合は、略称化されていないファンクション・キーワードが出力されます。

計算ラベルを非表示にするには、ブレーク列で、COLUMNコマンドのNOPRINTオプションを使用します。詳細は、「[COMPUTE](#)」コマンドを参照してください。COMPUTEが実行されている列に対してNOPRINTオプションを使用すると、COMPUTEの実行結果も非表示になります。

```
break on DEPARTMENT_ID page nodup
      on JOB_ID skip 1 nodup
```

この時点で、次のCOMPUTEコマンドを入力し、現行の問合せを実行します。

```
COMPUTE SUM OF SALARY ON DEPARTMENT_ID
/
```

DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
20	MK_MAN	Hartstein	13000

```

sum                                     13000

DEPARTMENT_ID JOB_ID      LAST_NAME      SALARY
-----
          80 SA_MAN      Russell      14000
                        Partners      13500

*****
sum                                     27500

DEPARTMENT_ID JOB_ID      LAST_NAME      SALARY
-----
          90 AD_PRE      King          24000
                        AD_VP      Kochhar      17000
                        De Haan      17000

*****
sum                                     58000

6 rows selected.

```

計算ラベルを出力せずに部門10および20の給与の合計を計算するには:

```

COLUMN DUMMY NOPRINT;
COMPUTE SUM OF SALARY ON DUMMY;
BREAK ON DUMMY SKIP 1;
SELECT DEPARTMENT_ID DUMMY, DEPARTMENT_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000
ORDER BY DEPARTMENT_ID;

```

```

DEPARTMENT_ID LAST_NAME      SALARY
-----
          20 Hartstein      13000
                        -----
                        13000

          80 Russell      14000
          80 Partners      13500
                        -----
                        27500

          90 King          24000
          90 Kochhar      17000
          90 De Haan      17000
                        -----
                        58000

```

6 rows selected.

レポートの終わりで給与を計算するには:

```

COLUMN DUMMY NOPRINT;
COMPUTE SUM OF SALARY ON DUMMY;
BREAK ON DUMMY;
SELECT NULL DUMMY, DEPARTMENT_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000
ORDER BY DEPARTMENT_ID;

```

DEPARTMENT_ID	LAST_NAME	SALARY
20	Hartstein	13000
80	Russell	14000
80	Partners	13500
90	King	24000
90	Kochhar	17000
90	De Haan	17000
		98500

6 rows selected.

NUMBER列の書式を設定する場合は、レポートに挿入する合計欄のサイズも考慮する必要があります。

例6-13 小計の計算および出力

SALARYの合計を部門別に計算するには、最初に、次のように入力して現行のBREAK定義を表示します。

```
BREAK
```

次のように現行のBREAK定義が表示されます。

6.2.7 レポートの終わりでのサマリー行の計算

BREAKコマンドおよびCOMPUTEコマンドを次の形式で使用すると、列内のすべての値に基づいてサマリー行を計算し出力できます。

```
BREAK ON REPORT
COMPUTE function LABEL label_name OF column column column
... ON REPORT
```

LAST_NAME	SALARY
Russell	14000
Partners	13500
Errazuriz	12000
Cambrault	11000
Zlotkey	10500
TOTAL	61000

総合計(または、総平均、総最大値など)を小計(または小平均など)とともに出力するには、BREAKコマンドにブレイク列およびON REPORT句を挿入します。その後、次のように、ブレイク列用に1つのCOMPUTEコマンドを入力し、ON REPORTを計算するためにもう1つのCOMPUTEコマンドを入力します。

```
BREAK ON break_column ON REPORT
COMPUTE function LABEL label_name OF column ON break_column
COMPUTE function LABEL label_name OF column ON REPORT
```

例6-14 総合計の計算および出力

すべての販売担当者の給与の総合計を計算して出力し、計算ラベルを変更するには、最初に、次のようにBREAKコマンドおよびCOMPUTEコマンドを入力します。

```
BREAK ON REPORT
COMPUTE SUM LABEL TOTAL OF SALARY ON REPORT
```

次に、新しい問合せを入力して実行します。


```
SELECT LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE JOB_ID=' SA_MAN' ;
```

6.2.8 複数のサマリー値およびサマリー行の計算

異なる列について同じタイプのサマリー値を計算して出力できます。そのためには、列ごとに別々のCOMPUTEコマンドを入力します。

```
1* SELECT LAST_NAME, SALARY
APPEND , COMMISSION_PCT;
```

最後に、修正した問合せを次のように実行し、結果を表示します。

```
/
```

LAST_NAME	SALARY	COMMISSION_PCT
Russell	14000	.4
Partners	13500	.3
Errazuriz	12000	.3
Cambrault	11000	.3
Zlotkey	10500	.2
sum	61000	1.5

同じブレイク列について、複数のサマリー行を出力することもできます。そのためには、次のようにCOMPUTEコマンドに各サマリーについての関数を挿入します。

```
COMPUTE function LABEL label_name function
LABEL label_name function LABEL label_name ...
OF column ON break_column
```

OFの後およびONの前に複数の列を挿入した場合、COMPUTEによって、指定したそれぞれの列についての値が計算され、出力されます。

DEPARTMENT_ID	LAST_NAME	SALARY
30	Colmenares	2500
	Himuro	2600
	Tobias	2800
	Baida	2900
	Khoo	3100
	Raphaely	11000

	avg	4150
	sum	24900

6 rows selected.

例6-15 異なる列についての同じタイプのサマリー値の計算

すべての販売担当者の給与およびコミッションの合計を出力するには、最初に次のCOMPUTEコマンドを入力します。

```
COMPUTE SUM OF SALARY COMMISSION_PCT ON REPORT
```

BREAKコマンドを入力する必要はありません。[例6-14](#)で入力したBREAKコマンドは、まだ有効です。この時点で、次のように

選択した問合せの最初の行を変更し、COMMISSION_PCTを追加します。

1

例6-16 同じブレーク列についての複数のサマリー行の計算

販売部門の給与の平均および合計を計算するには、最初に、次のBREAKコマンドおよびCOMPUTEコマンドを入力します。

```
BREAK ON DEPARTMENT_ID  
COMPUTE AVG SUM OF SALARY ON DEPARTMENT_ID
```

この時点で、次の問合せを入力して実行します。

```
SELECT DEPARTMENT_ID, LAST_NAME, SALARY  
FROM EMP_DETAILS_VIEW  
WHERE DEPARTMENT_ID = 30  
ORDER BY DEPARTMENT_ID, SALARY;
```

6.2.9 COMPUTE定義の表示および取消し

次のように、句を指定せずにCOMPUTEコマンドを入力すると、現行のCOMPUTE定義を表示できます。

```
COMPUTE  
  
breaks cleared  
  
CLEAR COMPUTES  
  
computes cleared
```

すべてのスクリプトの先頭にCLEAR BREAKコマンドおよびCLEAR COMPUTESコマンドを指定すると、前に入力したBREAKコマンドおよびCOMPUTEコマンドが、ファイル内で実行する問合せに影響しないようにすることができます。

例6-17 COMPUTE定義の取消し

すべてのCOMPUTE定義およびそれに伴ったBREAK定義を取り消すには、次のコマンドを入力します。

```
CLEAR BREAKS
```

6.3 ページおよびレポートのタイトルとサイズの定義について

ページとは、画面に表示される1画面全体の情報、またはスプールされる(印刷される)レポートの1ページを指します。各ページの上部タイトルおよび下部タイトルの配置、1ページごとの行数の設定、および各行の幅の設定ができます。

レポートとは、問合せの結果全体を指します。それぞれのレポートにヘッダーおよびフッターを配置し、それらの書式をページの上部タイトルおよび下部タイトルと同じ方法で設定できます。

6.3.1 上部タイトルと下部タイトルおよびヘッダーとフッターの設定

前述のとおり、レポートの各ページの一番上に表示するタイトルを設定できます。また、各ページの一番下に表示するタイトルも設定できます。TTITLEコマンドで上部タイトルを定義し、BTITLEコマンドで下部タイトルを定義します。

それぞれのレポートにヘッダーおよびフッターを設定することもできます。REPHEADERコマンドでレポート・ヘッダーを定義し、REPFOOTERコマンドでレポート・フッターを定義します。

TTITLE、BTITLE、REPHEADERまたはREPFOOTERコマンドでは、コマンド名の後に、位置または書式を指定する1つ以上の句を指定し、その後、その位置に指定するCHAR値またはその書式を設定するCHAR値を指定します。次のように、複

数の句およびCHAR値のセットを挿入できます。

```
TTITLE position_clause(s) char_value position_clause(s) char_value ...
BTITLE position_clause(s) char_value position_clause(s) char_value ...
REPHEADER position_clause(s) char_value position_clause(s) char_value ...
REPFOOTER position_clause(s) char_value position_clause(s) char_value ...
```

すべてのTTITLE、BTITLE、REPHEADERおよびREPFOOTER句の詳細は、「[TTITLE](#)」コマンドおよび「[REPHEADER](#)」コマンドを参照してください。

```

ACME SALES DEPARTMENT PERSONNEL REPORT
DEPARTMENT_ID LAST_NAME                SALARY
-----
          30 Colmenares                2500
          30 Himuro                    2600
          30 Tobias                     2800
          30 Baida                     2900
          30 Khoo                      3100
          30 Raphaely                  11000
```

COMPANY CONFIDENTIAL

6 rows selected.

```

ACME SALES DEPARTMENT PERSONNEL REPORT
PERFECT WIDGETS
```

COMPANY CONFIDENTIAL

```

ACME SALES DEPARTMENT PERSONNEL REPORT
DEPARTMENT_ID LAST_NAME                SALARY
-----
          30 Colmenares                2500
          30 Himuro                    2600
          30 Tobias                     2800
          30 Baida                     2900
          30 Khoo                      3100
          30 Raphaely                  11000
```

COMPANY CONFIDENTIAL

6 rows selected.

レポート・ヘッダーの定義を変更せずに非表示にするには、次のように入力します。

```
REPHEADER OFF
```

例6-18 ページの上部タイトルおよび下部タイトルの配置

レポートの各ページの一番上および一番下にタイトルを挿入するには、次のように入力します。

```
TTITLE CENTER -
"ACME SALES DEPARTMENT PERSONNEL REPORT"
BTITLE CENTER "COMPANY CONFIDENTIAL"
```

この時点で、次のように現行の問合せを実行します。

/

例6-19 レポートのヘッダーの配置

個々のページにレポート・ヘッダーを中央揃えで配置するには、次のように入力します。

```
REPHEADER PAGE CENTER ' PERFECT WIDGETS'
```

この時点で、次のように現行の間合せを実行します。

```
/
```

次に示す2ページの出力が表示され、最初のページには新しく設定したREPHEADERが表示されます。

6.3.1.1 タイトル要素の位置決定

前の例に示したレポートは、会社名をさらに強調し、個別の行の両端にレポートの種類および部門名を配置するとさらに体裁が良くなります。また、行サイズを小さくして、タイトルをデータの近くに中央揃えに配置するのも効果的です。

これらの変更を加えるには、次の例に示すように、TTITLEコマンドにいくつかの句を追加し、システム変数LINESIZEをリセットします。

レポートのヘッダーおよびフッターは、REPHEADERコマンドおよびREPFOOTERコマンドを使用してBTITLEおよびTTITLEと同じ方法で書式を設定できます。

```

              A C M E W I D G E T
              =====
PERSONNEL REPORT              SALES DEPARTMENT

DEPARTMENT_ID LAST_NAME              SALARY
-----
          30 Colmenares              2500
          30 Himuro                  2600
          30 Tobias                   2800
          30 Baida                   2900
          30 Khoo                    3100
          30 Raphaely                11000
              COMPANY CONFIDENTIAL

6 rows selected.
```

LEFT、RIGHTおよびCENTER句を使用して、後続の値を行の始め、終わりおよび中央に配置します。1行以上、下へ移動するには、SKIP句を使用します。

結果の最後の行と下部タイトルの間に空白がなくなっていることに注意してください。下部タイトルの最終行は、ページの最終行に表示されます。レポートの最終行から下部タイトルまでの空白の量は、ページ全体のサイズ、上部タイトルが占める行数、そのページ内の行数によって決まります。この例の上部タイトルは、その前の例で示した上部タイトルより3行多く行数をとります。1ページごとの行数の設定方法は、この章の後半で学習します。

下部タイトルの前にn行の空白行を常に出力するには、BTITLEコマンドの先頭にSKIP n句を使用します。たとえば、この例で下部タイトルの前で1行スキップするには、次のコマンドを入力します。

```
BTITLE SKIP 1 CENTER ' COMPANY CONFIDENTIAL'
```

例6-20 タイトル要素の位置決定

上部タイトルの位置を設定しなおして人事レポートを再表示するには、次のコマンドを入力します。

```
TTITLE CENTER ' A C M E W I D G E T' SKIP 1 -
CENTER ===== SKIP 1 LEFT ' PERSONNEL REPORT' -
RIGHT ' SALES DEPARTMENT' SKIP 2
```

```
SET LINESIZE 60
```

```
/
```

6.3.1.2 タイトル要素のインデント

特定の空白の個数分だけタイトル要素をインデントするには、TTITLEコマンドまたはBTITLEコマンドにCOL句を使用します。たとえば、COL 1を指定すると、後続の値が先頭文字の位置に置かれます。したがって、これはLEFT、つまり0個のインデントに相当します。COL 15を指定すると、タイトル要素が15文字目に配置され、14個分の空白でインデントされます。

```
ACME WIDGET
```

```
SALES DEPARTMENT PERSONNEL REPORT
```

DEPARTMENT_ID	LAST_NAME	SALARY
30	Colmenares	2500
30	Himuro	2600
30	Tobias	2800
30	Baida	2900
30	Khoo	3100
30	Raphaely	11000

```
COMPANY CONFIDENTIAL
```

```
6 rows selected.
```

例6-21 タイトル要素のインデント

会社名を左揃えで出力し、レポート名を次の行に空白5個分のみインデントして出力するには、次のように入力します。

```
TTITLE LEFT 'ACME WIDGET' SKIP 1 -
```

```
COL 6 'SALES DEPARTMENT PERSONNEL REPORT' SKIP 2
```

この時点で現行の問合せを実行し、結果を表示します。

```
/
```

6.3.1.3 長いタイトルの入力

長さが500文字を超えるタイトルを入力する必要がある場合は、SQL*PlusコマンドのDEFINEを使用して、タイトルの各行のテキストを別々の置換変数に格納します。

```
DEFINE LINE1 = 'This is the first line...'
```

```
DEFINE LINE2 = 'This is the second line...'
```

```
DEFINE LINE3 = 'This is the third line...'
```

その後、次のようにTTITLEコマンドまたはBTITLEコマンド内でそれらの変数を参照します。

```
TTITLE CENTER LINE1 SKIP 1 CENTER LINE2 SKIP 1 -
```

```
CENTER LINE3
```

6.3.2 タイトル内にシステム管理値を表示する方法

現行ページ番号およびその他のシステム管理値をタイトル内に表示するには、システム値の名前をタイトル要素として入力します。たとえば、次のように入力します。

```
TTITLE LEFT system-maintained_value_name
```

タイトル内に表示できるシステム管理値は5つあり、そのうち最も一般的なものはSQL.PNO(現行のページ番号)です。タイトル

内に表示できるシステム管理値のリストについては、「[TTITLE](#)」コマンドを参照してください。

```
ACMEWIDGET                                PAGE:           1
```

DEPARTMENT_ID	LAST_NAME	SALARY
30	Colmenares	2500
30	Himuro	2600
30	Tobias	2800
30	Baida	2900
30	Khoo	3100
30	Raphaely	11000

COMPANY CONFIDENTIAL

6 rows selected.

SQL.PNOの書式が空白10個分の幅であることに注意してください。この書式は、TTITLE(またはBTITLE)のFORMAT句を使用して変更できます。

```
ACME WIDGET                                ' PAGE:'         1
```

DEPARTMENT_ID	LAST_NAME	SALARY
30	Colmenares	2500
30	Himuro	2600
30	Tobias	2800
30	Baida	2900
30	Khoo	3100
30	Raphaely	11000

COMPANY CONFIDENTIAL

6 rows selected.

例6-22 タイトル内での現行ページ番号の表示

現行ページ番号を会社名とともに各ページの一番上に表示するには、次のコマンドを入力します。

```
TTITLE LEFT 'ACME WIDGET' RIGHT 'PAGE:' SQL.PNO SKIP 2
```

この時点で、現行の問合せを再実行します。

/

例6-23 タイトル内でのシステム管理値の書式設定

PAGE:という単語とページ番号の間隔を狭くするには、次のようにTTITLEコマンドを再入力します。

```
TTITLE LEFT 'ACME WIDGET' RIGHT 'PAGE:' FORMAT 999 -  
SQL.PNO SKIP 2
```

この時点で、次のように問合せを再実行します。

/

6.3.3 ページ・タイトル定義の表示と非表示の切替え

ページ・タイトルの定義を表示するには、句を指定せずに該当するタイトル・コマンドを入力します。

```
TTITLE  
BTITLE
```

タイトル定義を非表示にするには、次のように入力します。

```
TTITLE OFF  
BTITLE OFF
```

これらのコマンドを実行すると、タイトルはレポートに表示されなくなりますが、それらのタイトルの現行の定義は消去されません。現行の定義は、次のように入力すると再度表示できます。

```
TTITLE ON  
BTITLE ON
```

6.3.4 タイトル内に列の値を表示する方法

マスター/ディテール・レポートを作成し、各ページの一番上に変更されるマスター列の値を表示し、その下にマスター列の値に対する詳細な問合せ結果を表示できます。上部タイトル内の列値を参照するには、求める値を変数に格納し、TTITLEコマンド内でその変数を参照します。変数を定義するには、次の書式のCOLUMNコマンドを使用します。

```
COLUMN column_name NEW_VALUE variable_name
```

SKIP PAGE句を使用して、ORDER BY句およびBREAKコマンド内にマスター列を挿入する必要があります。

```
Manager:      101  
  
DEPARTMENT_ID LAST_NAME          SALARY  
-----  
          10 Whalen                4400  
          40 Mavris                 6500  
          70 Baer                   10000  
         100 Greenberg              12000  
         110 Higgins                12000  
  
Manager:      201  
  
DEPARTMENT_ID LAST_NAME          SALARY  
-----  
          20 Fay                    6000  
  
6 rows selected.
```

ページの一番下に列の値を出力する場合は、次の形式でCOLUMNコマンドを使用します。

```
COLUMN column_name OLD_VALUE variable_name
```

マスター列の新しい値の検出後、新しいページにブレークするプロセスの一部として下部タイトルが出力されます。したがって、単にマスター列のNEW_VALUEを参照した場合、次のディテール・セットの値が得られます。OLD_VALUEには、ページ・ブレークが開始される前に有効であったマスター列の値が記録されます。

例6-24 マスター/ディテール・レポートの作成

2人のマネージャの従業員番号を別々のページの一番上にそれぞれ表示し、そのマネージャの部下をマネージャの従業員番号

と同じページに表示するレポートを作成するとします。最初に、次のように入力して、現行のマネージャの従業員番号を保持する変数MGRVARを作成します。

```
COLUMN MANAGER_ID NEW_VALUE MGRVAR NOPRINT
```

マネージャの従業員番号は、タイトル内に表示されるのみなので、ディテールの列としては出力しません。この例で入力したNOPRINT句を使用すると、MANAGER_ID列は出力されません。

次に、ページ・タイトルにラベルおよび値を挿入し、適切なBREAKコマンドを入力して、最後の例の下部タイトルを非表示にします。

```
TTITLE LEFT 'Manager: ' MGRVAR SKIP 2  
BREAK ON MANAGER_ID SKIP PAGE  
BTITLE OFF
```

最後に、次の問合せを入力して実行します。

```
SELECT MANAGER_ID, DEPARTMENT_ID, LAST_NAME, SALARY  
FROM EMP_DETAILS_VIEW  
WHERE MANAGER_ID IN (101, 201)  
ORDER BY MANAGER_ID, DEPARTMENT_ID;
```

6.3.5 タイトル内に現在の日付を表示する方法について

タイトルに値を入力するだけで、レポートに日付を付けることができます。非定型のレポートの場合はこの方法でも十分ですが、同じレポートを繰り返し実行する場合は、レポートの実行時に日付が自動的に付くようにするとより効果的です。この機能は、現在の日付を保持する変数を作成して使用します。

事前に定義した置換変数_DATEを参照して、他の変数の場合と同様に、現在の日付をタイトルに表示できます。

LOGINファイルまたはSELECT文に指定されている日付書式モデルによって、SQL*Plusでの日付の表示書式が決まります。日付書式モデルの詳細は、『[Oracle Database SQL言語リファレンス](#)』を参照してください。LOGINファイルの詳細は、『[LOGINファイルの変更](#)』を参照してください。

これらのコマンドは、対話形式でも入力できます。詳細は、『[COLUMN](#)』を参照してください。

6.3.6 ページ・サイズの設定

通常、レポートの1ページには、SETコマンドのNEWPAGE変数で設定された行数分の空白行、上部タイトル、列ヘッダー、問合せ結果および下部タイトルが含まれています。SQL*Plusでは、長すぎて1ページに収まらないレポートは複数の連続したページに表示され、各ページに独自のタイトルおよび列ヘッダーが付けられます。SQL*Plusで各ページに表示されるデータの量は、現行のページ・サイズによって決まります。

SQL*Plusで使用されるデフォルトのページ・サイズを次に示します。

- 上部タイトルより前の行数: 1
- 上部タイトルからページ一番下までの1ページ当たりの行数: 14
- 1行当たりの文字数: 80

これらの設定は、ご使用のコンピュータ画面のサイズ、または用紙サイズ(印刷の場合)に合うように変更できます。

ページの長さはシステム変数PAGESIZEを使用して変更できます。ページの長さの変更は、レポートの印刷時などに必要な場合があります。

各ページの始めから上部タイトルまでの行数を設定するには、SETコマンドのNEWPAGE変数を使用します。


```
SET NEWPAGE number_of_lines
```

NEWPAGEを0(ゼロ)に設定した場合、SQL*Plusでは、新しいページを開始するために0(ゼロ)行がスキップされ、改ページ文字が表示および印刷されます。ほとんどのタイプのコンピュータ画面では、改ページ文字を使用すると画面が消去され、カーソルが1行目の先頭に移動されます。レポートの印刷時に改ページ文字が検出されると、その検出位置がページの最後ではない場合でも、次のページの一番上の印字位置までプリンタが移動します。NEWPAGEをNONEに設定した場合、レポートのページ間で空白行または改ページ文字は出力されません。

1ページの行数を設定するには、次のようにSETコマンドのPAGESIZE変数を使用します。

```
SET PAGESIZE number_of_lines
```

出力でタイトルを正しく中央揃えするために行サイズを小さくできます。また、幅の広い用紙に出力するために行サイズを大きくすることもできます。行の幅は、次のようにSETコマンドのLINE SIZE変数を使用して変更できます。

```
SET LINE SIZE number_of_characters
```

```

                ACME WIDGET PERSONNEL REPORT
                01-JAN-2001

DEPARTMENT_ID FIRST          LAST          MONTHLY
              NAME          NAME          SALARY
-----
          90 Steven          King          $24,000
          90 Neena          Kochhar          $17,000
          90 Lex            De Haan          $17,000
          80 John          Russell          $14,000
          80 Karen          Partners          $13,500
          20 Michael        Hartstein          $13,000

6 rows selected.
```

この時点で、次のように入力して、PAGESIZE、NEWPAGEおよびLINE SIZEをデフォルト値にリセットします。

```
SET PAGESIZE 14
SET NEWPAGE 1
SET LINE SIZE 80
```

これらの変数の現在の設定値を表示するには、次のようにSHOWコマンドを使用します。

```
SHOW PAGESIZE
SHOW NEWPAGE
SHOW LINE SIZE
```

SQL*PlusコマンドのSPOOLを使用して、問合せ結果をファイルに格納できます。また、ご使用のコンピュータのデフォルト・プリンタでの印刷もできます。

例6-25 ページ・サイズの設定

ページ・サイズを66行に設定し、各ページの始まりで画面をクリア(またはプリンタを改ページ)し、行サイズを70に設定するには、次のコマンドを入力します。

```
SET PAGESIZE 66
SET NEWPAGE 0
SET LINE SIZE 70
```

この時点で、次のコマンドを入力し実行して、結果を表示します。

```
TTITLE CENTER ' ACME WIDGET PERSONNEL REPORT' SKIP 1 -
```

この時点で、次のように問合せを実行します。

```
COLUMN FIRST_NAME HEADING 'FIRST|NAME';
COLUMN LAST_NAME HEADING 'LAST|NAME';
COLUMN SALARY HEADING 'MONTHLY|SALARY' FORMAT $99,999;
SELECT DEPARTMENT_ID, FIRST_NAME, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000;
```

6.4 問合せ結果の格納および印刷について

問合せ結果は、印刷する前にワード・プロセッサで編集する場合、あるいは書状、電子メールまたはその他の文書に挿入する場合、ファイルに格納します。

画面に表示したまま問合せ結果をファイルに格納するには、次の形式でSPOOLコマンドを入力します。

```
SPOOL file_name
```

ファイル名の後にピリオドおよび拡張子を指定しない場合、SPOOLを使用すると、ファイル名にデフォルトのファイル拡張子が追加され、そのファイルは出力ファイルとして認識されます。このデフォルトは、オペレーティング・システムによって異なりますが、ほとんどのホストではLSTまたはLISです。/dev/nullや/dev/stderrなどのシステム生成ファイルにスプールする場合、拡張子は追加されません。詳細は、ご使用のオペレーティング・システムのプラットフォーム固有のOracleのマニュアルを参照してください。

次の形式でSPOOLコマンドを入力してスプーリングをOFFにするまで、SQL*Plusでは情報がファイルにスプールされ続けます。

```
SPOOL OFF
```

6.4.1 フラット・ファイルの作成

異なるソフトウェア製品間でのデータの移動時に、フラット・ファイル(エスケープ文字、ヘッダーまたは余分な埋込み文字を含まないオペレーティング・システム・ファイル)を使用する必要がある場合があります。たとえば、Oracle Netを使用していない場合、Oracle9iからOracle Database 10gへデータを移行するには、SQL*Loaderでできるようにフラット・ファイルを作成する必要があります。

SQL*Plusでフラット・ファイルを作成するには、最初に次のSETコマンドを入力します。

```
SET NEWPAGE 0
SET SPACE 0
SET LINESIZE 80
SET PAGESIZE 0
SET ECHO OFF
SET FEEDBACK OFF
SET VERIFY OFF
SET HEADING OFF
SET MARKUP HTML OFF SPOOL OFF
```

これらのコマンドの入力後、前述のSPOOLコマンドを使用してフラット・ファイルを作成します。

SET COLSEPコマンドは、列に線を付ける場合に有効です。詳細は、「[SET](#)」コマンドを参照してください。

6.4.2 ファイルへの結果の格納

画面に表示したまま問合せ結果をファイルに格納するには、次の形式でSPOOLコマンドを入力します。

```
SPOOL file_name
```

SPOOLコマンドの入力後に画面に表示されたすべての情報が、指定したファイルに格納されます。

6.4.3 プリンタへの結果の出力

問合せ結果を印刷するには、前述のとおり、結果をファイルにスプールします。その後、SPOOL OFFではなく、次のようにコマンドを入力します。

```
SPOOL OUT
```

スプールが停止し、スプールされたファイルの内容がコンピュータの標準(デフォルト)プリンタにコピーされます。SPOOL OUTを使用すると、スプール・ファイルは印刷後に削除されません。

```

      A C M E W I D G E T

EMPLOYEE REPORT                                PAGE: 1

DEPARTMENT LAST NAME                MONTHLY SALARY
-----
      20 Hartstein                      $13,000
*****
sum                                  $13,000

      80 Russell
      Partners                      $13,500
*****
sum                                  $27,500

      90 King
      Kochhar                      $17,000
      De Haan                      $17,000
*****
sum                                  $58,000

sum                                  $98,500
                                COMPANY CONFIDENTIAL

6 rows selected.
```

例6-26 プリンタへの問合せ結果の出力

最終レポートを生成し、結果をスプールして印刷するには、次のコマンドを含むスクリプトEMPRPTを作成します。

最初に、EDITを使用して、オペレーティング・システムのテキスト・エディタでスクリプトを作成します。

```
EDIT EMPRPT
```

次に、テキスト・エディタを使用して次のコマンドをファイルに入力します。

```
SPOOL TEMP
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES

COLUMN DEPARTMENT_ID HEADING DEPARTMENT
COLUMN LAST_NAME HEADING 'LAST NAME'
COLUMN SALARY HEADING 'MONTHLY SALARY' FORMAT $99,999

BREAK ON DEPARTMENT_ID SKIP 1 ON REPORT
COMPUTE SUM OF SALARY ON DEPARTMENT_ID
```

```

COMPUTE SUM OF SALARY ON REPORT

SET PAGESIZE 24
SET NEWPAGE 0
SET LINESIZE 70

TTITLE CENTER 'A C M E W I D G E T' SKIP 2 -
LEFT 'EMPLOYEE REPORT' RIGHT 'PAGE:' -
FORMAT 999 SQL.PNO SKIP 2
BTITLE CENTER 'COMPANY CONFIDENTIAL'

SELECT DEPARTMENT_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000
ORDER BY DEPARTMENT_ID;

SPOOL OFF

```

出力を画面に表示しない場合は、ファイルの始めにSET TERMOUT OFFを追加し、ファイルの終わりにSET TERMOUT ONを追加します。テキスト・エディタでファイルを保存し、終了します(自動的にSQL*Plusに戻ります)。この時点で、次のようにEMPRPTスクリプトを実行します。

```
@EMPRPT
```

画面には、次のように出力され(TERMOUTをOFFに設定した場合を除く)、ファイルTEMPにスプールされます。

7 SQL*Plusからのレポートの生成

この章では、問合せ結果を含むHTMLおよびCSVレポートの生成方法について説明します。この章の構成は、次のとおりです。

- [SQL*Plusコマンドラインを使用したレポートの作成について](#)

7.1 SQL*Plusコマンドラインを使用したレポートの作成について

プレーン・テキスト出力およびSQL*Plusコマンドライン・インタフェースを使用すると、完全なWebページ、Webページに埋め込まれるHTML出力またはCSV形式のデータを生成できます。SQLPLUS -MARKUP HTML ONまたはSET MARKUP HTML ON SPOOL ONを使用して、<HTML>および<BODY>タグが自動的に埋め込まれる完全なHTMLページを生成できます。SQLPLUS -MARKUP CSV ONまたはSET MARKUP CSV ONを使用して、レポートをCSV形式で生成できます。

デフォルトでは、MARKUP HTML ONを使用して取得されたデータはHTML形式で出力されます。ただし、オプションで、HTMLの<PRE>タグに直接出力して、SQL*Plusで表示される形式と同様にWebブラウザで表示できます。これらのコマンドの詳細は、「SQLPLUS [MARKUPオプション](#)」および「[SET MARKUPコマンド](#)」を参照してください。

SQLPLUS -MARKUP HTML ONは、プログラム・スクリプトにSQL*Plusを埋め込む場合に有効です。起動時、コマンドの実行前に、HTMLおよびBODYタグが出力されます。その後のすべての出力は、SQL*Plusが終了するまでHTML形式です。

-SILENTおよび-RESTRICTコマンドライン・オプションを-MARKUPとともに使用すると、SQL*Plusプロンプトとバナー情報を非表示にし、一部のコマンドの使用を制限できます。

SET MARKUP HTML ON SPOOL ONを使用すると、その後にスプールされたそれぞれのファイルに対するHTMLページを生成できます。SPOOL OFFを実行するか、またはSQL*Plusを終了すると、スプール・ファイルのHTMLタグが閉じます。

SET MARKUP HTML ON SPOOL OFFを使用すると、既存のWebページへの埋込みに適したHTML出力を生成できます。この方法で生成されたHTML出力には、<HTML>または<BODY>タグは含まれません。

SQL*Plusコマンドラインで-M[ARKUP] CSV ONオプションを使用することにより、ユーザー・セッションにログイン中にCSVマークアップを有効にできます。詳細は、[SQL*Plusプログラムの構文](#)を参照してください。ユーザー・セッションにログイン中は、SET MARKUP CSV ONコマンドを使用してCSVマークアップを有効にできます。

DELIMITERオプションを使用して、デリミタ文字を指定できます。QUOTE OFFを使用して、引用符なしでテキストを出力することもできます。

[SET FEEDBACK](#)コマンドのONLYオプションを使用することで、問合せにより返されたデータを非表示にすることができます。選択され問合せにより返された行の数が表示されます。

7.1.1 HTMLレポートの作成

SQL*Plusセッション中、SET MARKUPコマンドを対話形式で使用して、HTMLをスプール・ファイルに書き出します。出力結果はWebブラウザで表示できます。

SET MARKUP HTML ON SPOOL ONでは、SQL*Plusの出力をHTML形式にエンコードすることを指定するのみで、出力ファイルの作成またはファイルへの書出しの開始は実行しません。スプール・ファイルの生成を開始するには、SQL*PlusのSPOOLコマンドを使用する必要があります。このファイルには、<HTML>タグ、</HTML>タグなどのHTMLタグが含まれます。

HTMLファイルを作成する場合は、HTMLファイルの標準ファイル拡張子である.htmlまたは.htmを指定します。これによって、出力ファイルの種類を簡単に識別でき、また、WebブラウザでHTMLファイルを認識し、正しく表示することができます。拡張子を指定しない場合は、SQL*Plusのデフォルトのファイル拡張子が使用されます。

スプール・ファイルに最終的なHTMLタグを追加してファイルを閉じるには、SPOOL OFFまたはEXITを使用します。SPOOL filenameコマンドを入力すると、SPOOL OFFまたはEXITでは、現行のスプール・ファイルが閉じられ、指定した名前の新しいHTMLのスプール・ファイルが作成されます。

SET MARKUPコマンドを使用すると、必要に応じてHTML出力をONまたはOFFにできます。

```
SQL> SELECT '<A HREF="http://oracle.com/||DEPARTMENT_NAME||.html">||DEPARTMENT_NAME||'</A>'
DEPARTMENT_NAME, CITY
2 FROM EMP_DETAILS_VIEW
3 WHERE SALARY>12000;
```

DEPARTMENT	CITY
Executive	Seattle
Executive	Seattle
Executive	Seattle
Sales	Oxford
Sales	Oxford
Marketing	Toronto

6 rows selected.

```
SQL> SPOOL OFF
```

この例では、プロンプトおよび問合せSQL文は非表示にされていません。スクリプトの起動方法によって、SET ECHO OFFまたはコマンドライン -SILENTオプションを使用すると、非表示にできます。

この例のSQL*Plusコマンドには、使用方法に関する項目が含まれています。

- 長いSQL*Plusコマンドの行を続けるには、ハイフンを使用します。
- 表のWIDTHおよびBORDER属性を設定するには、TABLEオプションを使用します。
- DEPARTMENT_NAME列にENTMAP OFFを設定し、HTMLのハイパーリンクを正しく構成するには、COLUMNコマンドを使用します。引用符、山カッコなどのHTMLの特殊文字は、対応するエンティティである「"」、「&」、「<」および「>」には置き換えられません。
- 文字列と変数要素を連結してハイパーリンクを作成するには、SELECT文に引用符および連結文字を使用します。

Webブラウザまたはテキスト・エディタでreport.htmlソースを表示すると、次のように、Department列の表セルに完全な形式のハイパーリンクが設定されていることがわかります。

```
<html>
<head>
<TITLE>Department Report</TITLE> <STYLE type="text/css">
<!-- BODY {background: #FFFC6} --> </STYLE>
<meta name="generator" content="SQL*Plus 10.2.0.1">
</head>
<body TEXT="#FF00ff">
SQL&gt; SELECT '&lt;A HREF=&quot;http://oracle.com/
||DEPARTMENT_NAME||.html&quot;&gt;' ||DEPARTMENT_NAME
||'&lt;/A&gt;' DEPARTMENT_NAME, CITY
<br>
2 FROM EMP_DETAILS_VIEW
<br>
3* WHERE SALARY&gt;12000
<br>
<p>
<table WIDTH="90%" BORDER="5">
<tr><th>DEPARTMENT</th><th>CITY</th></tr>
<tr><td><A HREF="http://oracle.com/Executive.html">Executive</A></td>
<td>Seattle</td></tr>
```

```

<tr><td><A HREF="http://oracle.com/Executive.html">Executive</A></td>
<td>Seattle</td></tr>
<tr><td><A HREF="http://oracle.com/Executive.html">Executive</A></td>
<td>Seattle</td></tr>
<tr><td><A HREF="http://oracle.com/Sales.html">Sales</A></td>
<td>Oxford</td></tr>
<tr><td><A HREF="http://oracle.com/Sales.html">Sales</A></td>
<td>Oxford</td></tr>
<tr><td><A HREF="http://oracle.com/Marketing.html">Marketing</A></td>
<td>Toronto</td></tr>
</table>
<p>
6 rows selected.<br>
SQL> spool off
<br>
</body>
</html>

```

DEPARTMENT_NAME	CITY
Executive	Seattle
Executive	Seattle
Executive	Seattle
Sales	Oxford
Sales	Oxford
Marketing	Toronto

6 rows selected.

この例のSQLPLUSコマンドには、ネストした引用符による3つのレイヤーが含まれています。レイヤーには、内側から順に次のものがあります。

- "2"は、BORDERに対するHTML属性値を引用符で囲んだものです。
- BORDER="2"は、TABLEオプションに対するテキスト引数を引用符で囲んだものです。
- "HTML ON TABLE 'BORDER="2" ' "は、-MARKUPオプションに対する引数を引用符で囲んだものです。

引用符のネスト方法は、オペレーティング・システムやプログラム・スクリプト言語によって異なります。

例7-1 対話型レポートの作成

SET MARKUPコマンドを使用すると、対話型のSQL*PlusセッションでHTML出力を作成できます。HTMLの<HEAD>タグには、埋込み型のスタイル・シートまたはその他の有効なテキストを挿入できます。SQL*Plusセッションを開いて、次のように入力します。

```

SET MARKUP HTML ON SPOOL ON PREFORMAT OFF ENTMAP ON -
HEAD "<TITLE>Department Report</TITLE> " -
<STYLE type=' text/css' > -
<!-- BODY {background: #FFFC6} --> -
</STYLE>" -
BODY "TEXT=' #FF00ff' " -
TABLE "WIDTH=' 90%' BORDER=' 5' "

```

COLUMNコマンドを使用して、列の出力を制御します。次のCOLUMNコマンドでは、SQL問合せ出力に対する新しいヘッダー名を作成できます。最初のコマンドを使用すると、DEPARTMENT_NAME列にマップするエンティティをOFFにして、出力

データのDEPARTMENT_NAME列にHTMLのハイパーリンクを正しく作成できます。

```
COLUMN DEPARTMENT_NAME HEADING 'DEPARTMENT' ENTMAP OFF
COLUMN CITY HEADING 'CITY'
```

SET MARKUP HTML SPOOL ONを使用すると、HTMLがスプール・ファイルに書き出されます。次のSPOOLコマンドでは、指定したファイルに<HTML>および<BODY>タグの書き込みをトリガーします。

```
SPOOL report.html
```

SPOOLコマンドの後の標準出力への入力または表示は、スプール・ファイルreport.htmlに書き出されます。

次のようにSQL問合せを入力します。

```
SELECT ' <A HREF="http://oracle.com/' || DEPARTMENT_NAME || '.html">' || DEPARTMENT_NAME || ' </A>'
DEPARTMENT_NAME, CITY
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000;
```

SPOOL OFFコマンドを入力します。

```
SPOOL OFF
```

スプール・ファイルreport.htmlを閉じる前に、</BODY>および</HTML>タグが追加されます。

report.sqlからの出力はファイルreport.htmlで、Webブラウザにロードできます。Webブラウザでreport.htmlを開きます。次のように表示されるはずです。

例7-2 SQLPLUSコマンドを使用したレポートの作成

オペレーティング・システムのプロンプトで次のコマンドを入力します。

```
SQLPLUS -S -M "HTML ON TABLE 'BORDER=2'" "HR@ora10g @depart.sql">depart.html
```

depart.sqlには、次の内容が含まれています。

```
SELECT DEPARTMENT_NAME, CITY
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000;
EXIT
```

この例では、ユーザーHRでSQL*Plusを起動し、HRパスワードの入力を求めるプロンプトを表示させ、HTMLをONに設定し、TABLEにBORDER属性を設定して、スクリプトdepart.sqlを実行します。depart.sqlからの出力は、この場合では、オペレーティング・システム・コマンド「>」(オペレーティング・システムによって異なる場合があります)を使用してdepart.htmlファイルにリダイレクトされたWebページです。SQL*PlusがWebサーバーのCGIスクリプトでコールされた場合は、Webブラウザに送信されます。CGIスクリプトからのSQL*Plusのコールについては、[レポートでのSQL*Plusコマンドの非表示について](#)を参照してください。

Webブラウザを起動し、適切なURLを入力してdepart.htmlを開きます。

7.1.1.1 HTMLエンティティ

一部の文字(「<」、「>」、「"」および「&」)は、HTMLで意味が事前定義されています。前述の例では、SET MARKUP HTML ONコマンドが入力されると、すぐに「>」の文字が「>」で置き換えられています。これらの文字をWebブラウザで表示できるようにするために、HTMLには代替文字のエンティティがあります。

表7-1 対応するHTMLエンティティ

文字	HTMLエンティティ	意味
<	<	HTML タグ・ラベルの開始
>	>	HTML タグ・ラベルの終了
"	"	二重引用符
&	&	アンパサンド

Webブラウザには「>」が表示されますが、エンコードされたHTMLファイルでの実際のテキストは、HTMLエンティティの「>」です。SET MARKUPオプションのENTMAPは、HTMLエンティティの置換を指定します。ENTMAPは、デフォルトでONに設定されています。デフォルトでは、「<」、「>」、「"」および「&」は、常にその文字を表すHTMLエンティティに置換されます。これによって、SQL*Plusコマンドまたは問合せ結果のデータにこれらの文字が存在した場合でも、Webブラウザが誤った解釈をしなくなります。

グローバル・レベルでSET MARKUP HTML ENTMAP ONを使用するか、または列レベルでCOLUMN *column_name* ENTMAP ONを使用して、ENTMAPを設定できます。

7.1.2 CSVレポートの作成

SQL*Plusコマンドラインで-M[ARKUP] CSV ONオプションを使用することにより、ユーザー・セッションにログイン中にCSVマークアップを有効にできます。詳細は、[SQL*Plusプログラムの構文](#)を参照してください。ユーザー・セッションにログイン中は、SET MARKUP CSV ONコマンドを使用してCSVマークアップを有効にできます。

DELIMITERオプションを使用して、デリミタ文字を指定できます。QUOTE OFFを使用して、引用符なしでテキストを出力することもできます。

CSVレポートの作成の詳細は、[SET MARKUP CSV ON](#)を参照してください。

7.1.3 レポートでのSQL*Plusコマンドの非表示について

CGIスクリプトまたはオペレーティング・システム・スクリプトを使用して、埋込み型のSQL*Plusレポートを生成する場合、SQLPLUSの-SILENTオプションを-MARKUPと組み合わせて使用すると特に有効です。これによって、SQL*PlusコマンドおよびSQL*Plusバナーを非表示にできます。HTML出力には、SQL問合せの結果データのみが表示されます。

STARTコマンドで実行されたスクリプト内の各コマンドを非表示にするには、SET ECHO OFFも使用できます。

8 SQL*Plusのチューニング

この章では、パフォーマンスを向上させるためにSQL*Plusをチューニングする方法を説明します。内容は次のとおりです。

- [文のトレースについて](#)
- [タイミング統計の収集について](#)
- [パラレル問合せおよび分散問合せのトレース](#)
- [以前のデータベースでの実行計画の出力](#)
- [SQL*Plusスクリプトのチューニングについて](#)

Oracle Databaseのチューニングについては、『[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。

8.1 文のトレースについて

SQLオプティマイザが使用する実行パス、および文の実行統計に関するレポートを自動的に取得できます。このレポートは、SQLのDML(SELECT、DELETE、UPDATEおよびINSERT)文が正常に実行された後に生成されます。このレポートは、これらの文のパフォーマンスを監視し、チューニングする場合に有効です。

動的サンプリングが有効な場合、DMLに関するSQL*Plusのレポート出力が異なる場合があります。

8.1.1 自動トレース・レポートの制御

AUTOTRACEシステム変数を設定することによって、レポートを制御できます。

自動トレースの設定	結果
SET AUTOTRACE OFF	AUTOTRACE レポートは生成されません。これはデフォルトです。
SET AUTOTRACE ON EXPLAIN	AUTOTRACE レポートには、オプティマイザの実行パスのみ示されます。
SET AUTOTRACE ON STATISTICS	AUTOTRACE レポートには、SQL 文の実行統計のみ示されます。
SET AUTOTRACE ON	AUTOTRACE レポートには、オプティマイザの実行パスと SQL 文の実行統計の両方が含まれます。
SET AUTOTRACE TRACEONLY	SET AUTOTRACE ON に似ていますが、ユーザーの問合せの出力は行われません。STATISTICS が使用可能な場合、問合せデータはフェッチされますが、出力は行われません。

この機能を使用するには、自分のスキーマにPLAN_TABLE表を作成する必要があります。また、PLUSTRACEロールが必要です。DBA権限に、PLUSTRACEロールを付与する必要があります。ロールの付与方法およびPLAN_TABLE表の作成方法については、『[Oracle Database SQL言語リファレンス](#)』を参照してください。

ロールおよびPLAN_TABLE表の詳細は、『[Oracle Database SQL言語リファレンス](#)』および「[SET](#)」コマンドのAUTOTRACE変数を参照してください。

ノート:



SQL*Plus AUTOTRACE では、ALTER SESSION SET CONTAINER オプションを使用したコンテナの切替はサポートしていません。この場合に収集された統計データには一貫性がない可能性があります。

例8-1 PLAN_TABLEの作成

SQL*Plusセッションから次のコマンドを実行して、HRスキーマにPLAN_TABLEを作成します。

```
CONNECT HR
@$ORACLE_HOME/rdbms/admin/utlxplan.sql

Table created.
```

例8-2 PLUSTRACEロールの作成

SQL*Plusセッションから次のコマンドを実行して、PLUSTRACEロールを作成し、DBAにそのロールを付与します。

```
CONNECT / AS SYSDBA
@$ORACLE_HOME/sqlplus/admin/plustrce.sql

drop role plustrace;

Role dropped.

create role plustrace;

Role created.

grant plustrace to dba with admin option;

Grant succeeded.
```

例8-3 PLUSTRACEロールの付与

SQL*Plusセッションから次のコマンドを実行して、HRユーザーにPLUSTRACEロールを付与します。

```
CONNECT / AS SYSDBA
GRANT PLUSTRACE TO HR;

Grant succeeded.
```

8.1.2 実行計画

実行計画は、SQLオプティマイザの問合せ実行パスを示します。実行計画の出力は、EXPLAIN PLANおよびDBMS_XPLANを使用して生成します。

8.1.3 統計

統計は、文を実行したときにサーバーによって記録され、文の実行に必要とされたシステム・リソースを示します。結果には、次の統計が含まれます。

データベース統計名	説明
recursive calls	ユーザーおよびシステム・レベルの両方で生成された再帰コールの数。Oracle Database では、内部処理に使用した表を保持します。これらの表を変更する必要が

データベース統計名	説明
	ある場合、Oracle Database では内部 SQL 文を内部的に生成し、この内部 SQL 文がかわりに再帰的コールを生成します。
db block gets	CURRENT ブロックが要求された回数。
consistent gets	ブロックに対して読取り一貫性が要求された回数。
physical reads	ディスクから読み込まれたデータ・ブロックの合計数。「physical reads direct」とバッファ・キャッシュへの読み込み合計数を足した数と同じ値になります。
redo size	生成された REDO の合計バイト数
bytes sent through Oracle Net Services to client	フォアグラウンド・プロセスからクライアントへ送信された合計バイト数。
bytes received through Oracle Net Services from client	Oracle Net によってクライアントから受信した合計バイト数。
Oracle Net Services round-trips to/from client	クライアントとの間で送受信された Oracle Net メッセージの合計数。
sorts (memory)	メモリー内で完全に実行され、ディスク書込みを必要としないソート操作数
sorts (disk)	1 回以上のディスク書込みを必要としたソート操作数
rows processed	操作中に処理された行の数。

統計の中で参照されるクライアントは、SQL*Plusです。「Oracle Net」は、Oracle Netがインストールされているかどうかに関係なく、SQL*Plusとサーバー間の汎用プロセス通信を意味します。統計レポートのデフォルト書式は変更できません。

データベース統計の完全なリストについては、「[統計の説明](#)」を参照してください。統計およびその解釈方法の詳細は、「[データベース統計の収集](#)」を参照してください。

LAST_NAME	SALARY	JOB_TITLE
King	24000	President
De Haan	17000	Administration Vice President
Kochhar	17000	Administration Vice President
Partners	13500	Sales Manager
Russell	14000	Sales Manager
Hartstein	13000	Marketing Manager
6 rows selected.		
Execution Plan		

Plan hash value: 2988506077

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	360	6 (17)	00:00:01
* 1	HASH JOIN		6	360	6 (17)	00:00:01
* 2	TABLE ACCESS FULL	EMPLOYEES	6	204	3 (0)	00:00:01
3	TABLE ACCESS FULL	JOBS	19	494	2 (0)	00:00:01

Predicate Information (identified by operation id):

- 1 - access("E"."JOB_ID"="J"."JOB_ID")
- 2 - filter("E"."SALARY">12000)

Note

- dynamic sampling used for this statement

Statistics

0 recursive calls
0 db block gets
10 consistent gets
0 physical reads
0 redo size
706 bytes sent via Oracle Net Services to client
496 bytes received via Oracle Net Services from client
2 Oracle Net Services roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
6 rows processed

6 rows selected.

Execution Plan

Plan hash value: 2988506077

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	360	6 (17)	00:00:01
* 1	HASH JOIN		6	360	6 (17)	00:00:01
* 2	TABLE ACCESS FULL	EMPLOYEES	6	204	3 (0)	00:00:01
3	TABLE ACCESS FULL	JOBS	19	494	2 (0)	00:00:01

Predicate Information (identified by operation id):

- 1 - access("E"."JOB_ID"="J"."JOB_ID")
- 2 - filter("E"."SALARY">12000)

Note

- dynamic sampling used for this statement

Statistics

0 recursive calls

```

0 db block gets
10 consistent gets
0 physical reads
0 redo size
706 bytes sent via Oracle Net Services to client
496 bytes received via Oracle Net Services from client
2 Oracle Net Services roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
6 rows processed

```

このオプションは、大規模な問合せのチューニング中に、問合せレポートを表示しない場合に有効です。



ノート:

出力は、サーバーのバージョンおよび構成によって異なります。

例8-4 パフォーマンス統計および問合せ実行パス用の文のトレース

SQLバッファに次の文が含まれているとします。

```

SELECT E. LAST_NAME, E. SALARY, J. JOB_TITLE
FROM EMPLOYEES E, JOBS J
WHERE E. JOB_ID=J. JOB_ID AND E. SALARY>12000;

```

この文を実行すると、次のように文のトレースが自動的に行われます。

```

SET AUTOTRACE ON
/

```

例8-5 問合せデータを表示せずに文をトレースする方法

問合せデータを表示せずに同じ文をトレースするには、次のように入力します。

```

SET AUTOTRACE TRACEONLY
/

```

8.2 タイミング統計の収集について

SQL*PlusのTIMINGコマンドを使用すると、1つ以上のコマンドまたはブロックを実行するために使用されたコンピュータ・リソースの量に関するデータを収集して表示できます。TIMINGは、経過時間に関するデータを収集し、タイマーの時間内に実行されたコマンドに関するデータを保存します。

統計を収集するAUTOTRACEの使用については、[TIMING](#)コマンドおよび[文のトレースについて](#)を参照してください。

すべてのタイマーを削除するには、CLEAR TIMINGを入力します。

8.3 パラレル問合せおよび分散問合せのトレース

パラレル問合せまたは分散問合せ内で文をトレースした場合は、使用する文によって実行計画の出力が異なります。

例8-6 パラレル問合せオプションによる文のトレース

パラレル問合せオプションを実行してパラレル問合せをトレースするには:

```

create table D2_t1 (unique1 number) parallel -
(degree 6);

```

Table created.

```
create table D2_t2 (unique1 number) parallel -  
(degree 6);
```

Table created.

```
create unique index d2_i_unique1 on d2_t1(unique1);
```

Index created.

```
set long 500 longchunksiz 500  
SET AUTOTRACE ON EXPLAIN  
SELECT /*+ INDEX(B, D2_I_UNIQUE1) USE_NL(B) ORDERED -  
*/ COUNT (A.UNIQUE1)  
FROM D2_T2 A, D2_T1 B  
WHERE A.UNIQUE1 = B.UNIQUE1;
```

Execution Plan

Plan hash value: 107954098

Id	Operation	Name	Rows	Bytes	Cost(%CPU)	Time	TQ	IN-OUT	PQ
Distrib									
0	SELECT STATEMENT		1	26	1 (0)	00:00:01			
1	SORT AGGREGATE		1	26					
2	PX COORDINATOR								
3	PX SEND QC (RANDOM)	:TQ10001	1	26			Q1,01	P->S	QC
4	SORT AGGREGATE		1	26			Q1,01	PCWP	
5	NESTED LOOPS		1	26	1 (0)	00:00:01	Q1,01	PCWP	
6	PX RECEIVE						Q1,01	PCWP	
7	PX SEND BROADCAST	:TQ10000					Q1,00	P->P	
8	PX BLOCK ITERATOR		1	13	0 (0)	00:00:01	Q1,00	PCWC	
9	TABLE ACCESS FULL	D2_T2	1	13	0 (0)	00:00:01	Q1,00	PCWP	
10	PX BLOCK ITERATOR		1	13	2 (0)	00:00:01	Q1,01	PCWC	
* 11	TABLE ACCESS FULL	D2_T1	1	13	2 (0)	00:00:01	Q1,01	PCWP	

Predicate Information (identified by operation id):

11 - filter("A"."UNIQUE1"="B"."UNIQUE1")

Note

- dynamic sampling used for this statement

例8-7 ディスク読み取りおよびバッファ取得の監視

```
SET AUTOTRACE TRACEONLY STATISTICS
```

結果は次のようになります。

Statistics

```
-----
      467 recursive calls
       27 db block gets
      147 consistent gets
       20 physical reads
     4548 redo size
      502 bytes sent via Oracle Net Services to client
      496 bytes received via Oracle Net Services from client
         2 Oracle Net Services roundtrips to/from client
       14 sorts (memory)
         0 sorts (disk)
         1 rows processed
```

consistent getsまたはphysical readsの値が、戻されるデータ量に比例して高い場合、問合せにはコストがかかるため、再確認して最適化する必要があります。たとえば、戻される行が1,000行未満で、consistent getsが1,000,000、physical readsが10,000の場合、さらに最適化が必要です。

ノート:



V\$SQL または TKPROF を使用して、ディスク読み取りおよびバッファ取得を監視することもできます。

8.4 以前のデータベースでの実行計画の出力

Oracle Database 9リリース2(9.2)以前では、実行計画の出力が異なります。

実行計画の各行には、連続した行番号が付きます。SQL*Plusは、親操作の行番号も表示します。

実行計画は、次の順序で表示される4つの列で構成されます。

列名	説明
ID_PLUS_EXP	各実行ステップの行番号を表示します。
PARENT_ID_PLUS_EXP	各ステップとその親の関係を表示します。この列は、大きなレポートに有効です。
PLAN_PLUS_EXP	レポートの各ステップを表示します。
OBJECT_NODE_PLUS_EXP	使用されたデータベース・リンク、またはパラレル問合せサーバーを表示します。

列の書式は、COLUMNコマンドで変更できます。たとえば、PARENT_ID_PLUS_EXP列を表示しないようにするには、次の

ように入力します。

COLUMN PARENT_ID_PLUS_EXP NOPRINT

実行計画の出力は、EXPLAIN PLANコマンドを使用して生成します。

パラレル問合せまたは分散問合せ内で文をトレースする場合、実行計画には、コストベースのオプティマイザの行数見積り(カーディナリティ)が表示されます。一般に、各ノードでのコスト、カーディナリティおよびバイト数は、累積結果を表しています。たとえば、結合ノードのコストは結合操作を実行するためのコストのみを考慮したものではなく、その結合でリレーションにアクセスするコスト全体を含んでいます。

アスタリスク(*)が付いた行は、パラレル操作またはリモート操作を表しています。それぞれの操作は、レポートの2番目の部分で説明されています。パラレル操作および分散操作の詳細は、「[分散トランザクションの概念](#)」を参照してください。

このレポートの2番目のセクションは、次の順序で表示される3つの列で構成されます。

列名	説明
ID_PLUS_EXP	各実行ステップの行番号を表示します。
OTHER_TAG_PLUS_EXP	OTHER_PLUS_EXP 列内の SQL 文の機能を説明します。
OTHER_PLUS_EXP	パラレル・サーバーまたはリモート・データベースについての問合せのテキストを表示します。

列の書式は、COLUMNコマンドで変更できます。

8.5 SQL*Plusスクリプトのチューニングについて

スクリプトで実行されるSQL問合せをチューニングすることによって、ほとんどの場合、パフォーマンス上の利点が得られます。このチューニングには、SQL*PlusのAUTOTRACEコマンドのようなツールを使用します。Oracle DatabaseのSQLオプティマイザを最大限に活用することには、問合せの再構成も含まれます。SQL文のチューニングについては、『[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。

SQL*Plus固有のコマンドをチューニングすることで得られるパフォーマンスの向上はわずかですが、その向上が重要となるアプリケーションもあります。次のシステム変数およびコマンドは、SQL*Plusのパフォーマンスに影響を与える場合があります。

8.5.1 COLUMN NOPRINT

COLUMN NOPRINTを設定すると、列ヘッダーおよび列で選択されるすべての値の画面表示および印刷は実行されません。

SELECTから必要のない列を削除する方が、その列を表示しないようにCOLUMN NOPRINTを使用するよりも効率的です。問合せから列を削除すると、SQLエンジンによるその列の処理もSQL*Plusへの列データの送信も必要ありません。

8.5.2 SET APPINFO OFF

DBMS_APPLICATION_INFOパッケージを使用して、スクリプトの自動登録を設定します。APPINFO OFFを設定すると、管理者はスクリプトのパフォーマンスおよびリソースの使用率を監視できなくなります。

多数のSQLスクリプトがコールされている場合、APPINFO OFFをチューニングすると、内部SQL*PlusによるデータベースのDBMS_APPLICATION_INFOパッケージへのコールが停止します。

8.5.3 SET ARRAYSIZE

SQL*Plusがデータベースから一度にフェッチする行数を設定します。有効な値は1から5000です。

ARRAYSIZEの設定による効果は、Oracle Databaseによるネットワーク・パケットへの書き込み効率、およびネットワークの待ち時間とスループットに基づきます。SQL*PlusおよびOracle Databaseの最近のバージョンでは、ARRAYSIZEによる効果はほとんどありません。サイズが非常に大きくなると、より多くのSQL*Plusメモリーが使用され、全体のパフォーマンスが低下する場合があります。

8.5.4 SET DEFINE OFF

SET DEFINE OFFを設定すると、置換変数をそれらの値に置換するコマンドの解析は実行されません。

8.5.5 SET FLUSH OFF

SET FLUSH OFFを設定すると、オペレーティング・システムは出力をバッファに格納できます。ONを設定すると、バッファを使用せず、画面への出力をフラッシュします。FLUSHをONまたはOFFに設定して得られる効果は、オペレーティング・システムおよびデータによって異なります。また、その効果はわずかである場合があります。

ユーザーとの対話が不要で、完了するまで出力結果を確認する必要のないスクリプトを実行する場合のみ、OFFを使用します。

8.5.6 SET LINESIZE

SET LINESIZEでは、SQL*Plusが新しい行を開始する前に、1行に表示する文字の合計数を設定します。

LINESIZEをできるだけ小さくして、余分なメモリー割当ておよびメモリーのコピーを回避します。

ただし、LINESIZEを小さくしすぎると、隣りの列とサイズが合わない列には別の線が表示されます。これによって、パフォーマンスが大幅に低下する場合があります。

8.5.7 SET LONGCHUNKSIZE

SET LONGCHUNKSIZEでは、SQL*PlusがBLOB、BFILE、CLOB、LONG、NCLOBまたはXMLType値を取得するときの増分値のサイズを設定します。

LONGまたはLOBがフェッチされている場合は、別のサイズで試してください。

8.5.8 SET PAGESIZE

各出力ページの行数を設定します。

PAGESIZEを大きくしてヘッダーを頻繁に表示しないようにするか、0に設定してヘッダーを表示しないようにします。

8.5.9 SET SERVEROUTPUT

SET SERVEROUTPUT OFFを設定すると、SQL*Plusのストアド・プロシージャまたはPL/SQLブロックの出力(DBMS_OUTPUT.PUT_LINE)は表示されません。

SERVEROUTPUT OFFを設定すると、内部SQL*Plusによる、ユーザーSQL文の後に実行されたDBMS_OUTPUTパッケージへのコールが停止します。

8.5.10 SET SQLPROMPT

SQL*Plusコマンド・プロンプトを設定します。

デフォルトのプロンプト「SQL>」を使用して、変数の置換がプロンプトの表示ごとに発生するのを停止します。

8.5.11 SET TAB

SQL*Plusが空白をどのように端末に出力するかを指定します。

TAB ONを設定すると、複数の空白が端末の出力で圧縮されます。この設定では、書き込まれているデータが大幅に縮小しないかぎり、得られる効果より必要となる処理の方がわずかに上回ります。

8.5.12 SET TERMOUT

SET TERMOUT OFFを指定すると非表示になるため、スクリプトからの出力を画面に表示せずに、出力をスプールできます。

ファイルへのスプールおよび端末への書き込みが両方とも不要な場合は、SQLスクリプトでSET TERMOUT OFFを使用して端末の出力を無効にします。

8.5.13 SET TRIMOUT ON SET TRIMSPOOL ON

SET TRIMOUT ONまたはSET TRIMSPOOL ONを指定すると、表示またはスプールされる各行の末尾で後続の空白を削除します。

これらの変数をONに設定すると、書き込まれるデータ量を減らすことができます。ただし、LINESIZEが最適な場合は、変数をOFFに設定した方がより速い場合があります。問合せがルーチン进行处理している間、SQL*Plusの出力行に空白が書き込まれるため、空白の削除には余計な手間がかかります。

8.5.14 UNDEFINE

明示的に(DEFINEコマンドを使用して)定義したか、暗黙的に(STARTコマンドまたはCOLUMN NEW_VAL|OLD_VALへの引数によって)定義した置換変数を削除します。

UNDEFINEを使用すると、不要な置換変数を削除できます。これによって、&変数、new_value変数またはold_value変数を使用する操作に費やす時間を短縮できます。

9 SQL*Plusのセキュリティ

この章では、データベース表およびSQL*Plusコマンドへのアクセスの制御に使用可能な方法について説明します。次のトピックについて説明します。

- [SQL*Plus、SQLおよびPL/SQLコマンドの使用禁止](#)
- [ロールの作成および制御について](#)
- [SQLPLUS -RESTRICTでのコマンドの使用禁止について](#)
- [プログラム引数のセキュリティについて](#)

9.1 SQL*Plus、SQLおよびPL/SQLコマンドの使用禁止

ノート:



Oracle Database 19c 以降、SQL*Plus 表 PRODUCT_USER_PROFILE (PUP 表)はサポート対象外です。Oracle Database の設定を使用してデータを保護し、すべてのクライアント・アプリケーション間で一貫したセキュリティを確保することをお勧めします。

特定のユーザーに対してSQLまたはSQL*Plusコマンドを使用禁止にするには、Userid列にユーザーのユーザー名、Attribute列にコマンド名およびChar_Value列にDISABLEDを入力した行を挿入します。Scope、Numeric_ValueおよびDate_Value列にはNULLを入力します。次に例を示します。

PRODUCT	USERID	ATTRIBUTE	SCOPE	NUMERIC VALUE	CHAR VALUE	DATE VALUE	LONG VALUE
SQL*Plus	HR	HOST			DISABLED		
SQL*Plus	%	INSERT			DISABLED		
SQL*Plus	%	UPDATE			DISABLED		
SQL*Plus	%	DELETE			DISABLED		

コマンドを再度使用可能にするには、制限を含む行を削除します。

使用禁止にできるSQL*Plusコマンド:

- ACCEPT
- DEFINE
- PASSWORD
- SHUTDOWN
- APPEND
- DEL
- PAUSE
- SPOOL
- ARCHIVE LOG

- DESCRIBE
- PRINT
- START (@, @@)
- ATTRIBUTE
- DISCONNECT
- PROMPT
- STARTUP
- BREAK
- EDIT
- RECOVER
- STORE
- BTITLE
- EXECUTE
- REMARK
- TIMING
- CHANGE
- EXIT/QUIT
- REPFOOTER
- TTITLE
- CLEAR
- GET
- REPHEADER
- UNDEFINE
- COLUMN
- HELP (?)
- RUN
- VARIABLE
- COMPUTE
- HOST
- SAVE
- WHENEVER OSERROR
- CONNECT
- INPUT

- SET
- WHENEVER SQLERROR
- COPY
- LIST (;)
- SHOW
- XQUERY

使用禁止にできるSQLコマンド:

- ALTER
- ANALYZE
- ASSOCIATE
- AUDIT
- CALL
- COMMENT
- COMMIT
- CREATE
- DELETE
- DISASSOCIATE
- DROP
- EXPLAIN
- FLASHBACK
- GRANT
- INSERT
- LOCK
- MERGE
- NOAUDIT
- PURGE
- RENAME
- REVOKE
- ROLLBACK
- SAVEPOINT
- SELECT
- SET CONSTRAINTS
- SET ROLE

- SET TRANSACTION
- TRUNCATE
- UPDATE
- VALIDATE

次に、使用禁止にできるPL/SQLコマンドを示します。

- BEGIN
- DECLARE

ノート:



- HOST を使用禁止にすると、ご使用のオペレーティング・システムの HOST の別名(Windows 環境では \$、UNIX 環境では!など)も使用禁止になります。
- LIST を使用禁止にすると、;および番号(スクリプト内の行に移動するために入力した番号)も使用禁止になります。
- コマンドライン・ヘルプへのアクセスを禁止するには、HELP および?を別々に使用禁止にする必要があります。
- SQL*Plus の SET コマンドを使用禁止にすると、SQL の SET CONSTRAINTS、SET ROLE および SET TRANSACTION コマンドも使用禁止になります。
- SQL*Plus の START を使用禁止にすると、@および@@も使用禁止になります。
- BEGIN および DECLARE コマンドを使用禁止にしても、PL/SQL を実行する SQL*Plus の EXECUTE コマンドは使用可能です。EXECUTE を使用禁止にする場合は、別々に行ってください。
- EXIT/QUIT を使用禁止にすることはお薦めしません。使用禁止にする場合は、EOF 文字(UNIX の場合は[Ctrl]+[D]、Windows の場合は[Ctrl]+[Z]など)を送信してコマンドライン・セッションを終了してください。または、SQL*Plus の処理を終了してセッションを終了してください。また、使用禁止にすると、WHENEVER OSERROR および WHENEVER SQLERROR での EXIT 操作も禁止になります。

1. 次のコマンドを使用して、SYSTEMとしてログインします。

```
SQLPLUS SYSTEM
```

2. 次のコマンドを使用して、PUP表に行を挿入します。

```
INSERT INTO PRODUCT_USER_PROFILE
VALUES ('SQL*Plus', 'HR', 'SELECT', NULL, NULL, 'DISABLED', NULL, NULL);
```

3. 次のようにHRで接続して、SELECTを実行します。

```
CONNECT HR
SELECT * FROM EMP_DETAILS_VIEW;
```

このコマンドによって、次のエラー・メッセージが表示されます。

```
SP2-0544: Command SELECT disabled in Product User Profile
```

4. この行を削除し、ユーザーHRから制限を削除するには、SYSTEMで再接続し、次のように入力します。

```
DELETE FROM PRODUCT_USER_PROFILE WHERE USERID = 'HR';
```

9.2 ロールの作成および制御について

SQLコマンドを使用して、データベース表に対するセキュリティを確保するために、ロールへのアクセスを作成および制御できます。ロールを作成し、そのロールにアクセスするユーザーを制御することによって、特定のデータベース権限に特定のユーザーのみがアクセスできるようにします。

ロールは、SQLのCREATE、GRANTおよびSETコマンドで作成され、使用されます。

- ロールを作成するには、CREATEコマンドを使用します。パスワードなしのロールでもパスワード付きのロールでも作成できます。
- ロールにアクセス権限を付与するには、GRANTコマンドを使用します。この方法で、ロールに関連する権限にアクセスするユーザーを制御します。
- ロールにアクセスするには、SET ROLEコマンドを使用します。パスワード付きのロールを作成した場合、ユーザーがそのロールにアクセスするには、そのパスワードを知っている必要があります。

9.2.1 SET ROLEの使用禁止について

ユーザーは、SQL*Plusから任意のSQLコマンドを送ることができます。特定の状況下では、これはセキュリティ上の問題の原因になります。適切な予防手段を取っておかないと、ユーザーがSET ROLEを使用して、アプリケーション・ロールで取得した権限にアクセスするおそれがあります。これらの権限を使用して、ユーザーがSQL*PlusからSQL文を発行し、その結果データベース表が誤って変更されてしまう可能性があります。

アプリケーション・ユーザーによるSQL*Plus内のアプリケーション・ロールへのアクセスを防止するには、PUP表を使用して、SET ROLEコマンドを使用禁止にします。BEGINおよびSQL*PlusのEXECUTEコマンドも使用禁止にして、アプリケーション・ユーザーがPL/SQLブロックを使用してアプリケーション・ロールを設定できないようにする必要があります。これによって、SQL*Plusユーザーは、SQL*Plusの起動時に使用可能なロールに関連した権限のみを使用できるようになります。ユーザー・ロールの作成および使用方法の詳細は、[『Oracle Database SQL言語リファレンス』](#)および[『Oracle Database管理者ガイド』](#)を参照してください。

9.2.2 ユーザー・ロールの使用禁止について

特定のユーザーに対して特定のロールを使用禁止にするには、Userid列にユーザーのユーザー名、Attribute列にROLESおよびChar_Value列にロール名を入れた行をPUP表に挿入します。

ノート:



Userid 列に PUBLIC または%を入力する場合は、すべてのユーザーに対してロールを使用禁止にします。PUBLIC に付与されるロールに対して%または PUBLIC のみを使用するようにします。ユーザーに付与されていないロールを使用禁止にしようとする場合、そのユーザーのロールはすべて使用禁止にされません。

Scope、Numeric_ValueおよびDate_Value列にはNULLを入力します。次に例を示します。

PRODUCT	USERID	ATTRIBUTE	SCOPE	NUMERIC VALUE	CHAR VALUE	DATE VALUE	LONG VALUE
SQL*Plus	HR	ROLES			ROLE1		
SQL*Plus	PUBLIC	ROLES			ROLE2		

ログイン時には、前述の表の行は次のコマンドに変換されます。

```
SET ROLE ALL EXCEPT ROLE1, ROLE2
```

ユーザーがログイン後にSET ROLEコマンドを使用してロールを変更できないようにするには、SET ROLEコマンドを使用禁止にします。

ロールを再度使用可能にするには、制限を含む行を削除します。

詳細は、[SET ROLEの使用禁止について](#)を参照してください。

9.3 SQLPLUS -RESTRICTでのコマンドの使用禁止について

RESTRICTオプションを使用すると、オペレーティング・システムと対話する特定のコマンドを使用禁止にできます。ただし、-RESTRICTオプションで使用禁止にされたコマンドは、サーバーと接続していないときにも使用できず、SQL*Plusを終了するまで使用禁止のままです。

次の表に、各制限レベルでの使用禁止コマンドを示します。

コマンド	レベル1	レベル2	レベル3
EDIT	使用禁止	使用禁止	使用禁止
GET			使用禁止
HOST	使用禁止	使用禁止	使用禁止
SAVE		使用禁止	使用禁止
SPOOL		使用禁止	使用禁止
START			使用禁止
STORE		使用禁止	使用禁止

ノート:



- HOST を使用禁止にすると、ご使用のオペレーティング・システムの HOST の別名(Windows 環境では\$、UNIX 環境では!)なども使用禁止になります。
- SQL*Plus の START コマンドを使用禁止にすると、SQL*Plus の@および@@コマンドも使用禁止

になります。

RESTRICTオプションの詳細は、SQLPLUSの「[RESTRICTオプション](#)」を参照してください。

9.4 プログラム引数のセキュリティについて

オペレーティング・システムによっては、実行中のプログラムをすべてのユーザーが参照することができます。コマンドライン引数も示される場合は、SQL*Plusユーザーのユーザー名およびパスワードも表示できます。

たとえば、ほとんどのUNIXまたはLinuxシステムでは、psコマンドはプログラム引数を示します。パスワードが表示されないようにするかどうかは、SQL*Plusの使用方法によって異なります。

- SQL*Plusを対話形式で実行する場合は、常に、SQL*Plusに接続情報(特にパスワード)の入力を求めるプロンプトが表示されるまで待機します。
- UNIXのシェル・スクリプトからSQLスクリプトのバッチを実行する場合は、環境変数MYUSERNAMEおよびMYPASSWORDに適切な値を設定します。次の文字列を含むシェル・スクリプトを実行します。

```
sqlplus /nolog <<EOF
connect $MYUSERNAME/$MYPASSWORD
select ...
EOF
```

- バッチのSQLスクリプトを実行する場合は、ユーザー名とパスワードをSQLスクリプトの最初の行としてハード・コードします。その後、次のコマンドでスクリプトをコールします。

```
sqlplus @myscript.sql
```

このようにSQL*Plusを起動した場合、SQL*Plusはusername/password@connection_identifier文字列としてスクリプトの最初の行を使用します。

ユーザー名およびパスワードは、ファイルまたはスクリプトに格納しないようにします。ユーザー名およびパスワードをファイルまたはスクリプトに格納する場合は、そのファイルまたはスクリプトが不正なアクセスから保護されていることを確認してください。

10 SQL*Plusによるデータベース管理

この章では、SQL*Plusで使用可能なデータベース管理ツールの概要を説明します。ここでは、次の項目について説明します。

- [概要](#)
- [データベースの起動および停止の概要](#)
- [REDOログ・ファイル](#)
- [データベースのリカバリ](#)

この章は、データベース管理者向けです。この機能を使用するには、データベース管理者の権限が必要です。

データベース管理の詳細は、『[Oracle Database概要](#)』を参照してください。

10.1 概要

データベースの起動、停止などの特別な操作は、データベース管理者(DBA)が行います。DBAは、一般ユーザーには割り当てられていない権限を所有しています。この章で概要を説明するコマンドは、通常、DBAが使用します。

SQL*Plusでのセキュリティおよびロールの詳細は、「[SQL*Plusのセキュリティ](#)」を参照してください。

10.2 データベースの起動および停止の概要

すべてのユーザーが、Oracle Databaseを利用できるとはかぎりません。データベースをオープンまたはクローズするか、あるいはインスタンスを起動または停止するには、DBA権限を持っているか、SYSOPERまたはSYSDBAで接続する必要があります。それ以外のユーザーは、Oracle Databaseの現在の状態を制御できません。

10.2.1 データベースの起動

データベースを起動するには:

1. インスタンスの起動

インスタンスは、バックグラウンド・プロセスを制御し、Oracle Databaseに接続するためのメモリー領域を割り当てます。

2. データベースのマウント

データベースをマウントすることによって、データベースがすでに起動されているインスタンスと対応付けられます。

3. データベースのオープン

データベースをオープンすると、通常のデータベース操作が可能になります。

例10-1 インスタンスの起動

データベースをマウントせずにOracle Databaseインスタンスを起動するには、次のように入力します。

```
STARTUP NOMOUNT
```

例10-2 データベースのマウント

インスタンスを起動し、データベースをクローズしたままマウントするには、次のように入力します。

```
STARTUP MOUNT
```

例10-3 データベースのオープン

Oracle Databaseサーバーのパラメータ・ファイルINITSALES.ORAを使用してインスタンスを起動し、SALESというデータベースをマウントおよびオープンして、データベース管理者に対してアクセスを制限するには、次のように入力します。

```
STARTUP OPEN sales PFILE=INITSALES.ORA RESTRICT
```

SALESは、INITSALES.ORAパラメータ・ファイルのDB_NAMEパラメータで指定したデータベース名です。

データベース起動の詳細は、「[起動および停止](#)」を参照してください。また、「[STARTUP](#)」コマンドも参照してください。

10.2.2 PDBの起動

プラガブル・データベース(PDB)は、統合データベース内に存在するスキーマおよびスキーマ・オブジェクトの自己完結型コレクションです。

プラガブル・データベースを起動するには:

1. /NOLOG引数を使用してSQL*Plusを起動します。

```
sqlplus /nolog
```

2. 簡易接続またはネット・サービス名を使用してCONNECTコマンドを発行し、PDBに接続します。
3. STARTUPコマンドを発行します。

データベース起動の詳細は、「[起動および停止](#)」を参照してください。また、「[STARTUP](#)」コマンドも参照してください。

10.2.3 データベースの停止

データベースの停止には、次の3つのステップが含まれます。

1. データベースのクローズ

データベースがクローズされると、SGA内のすべてのデータベースおよびリカバリ・データがデータ・ファイルおよびREDOログ・ファイルに書き込まれ、すべてのオンライン・データ・ファイルがクローズされます。

2. データベースのデスマウント

データベースをデスマウントすることによって、データベースがインスタンスから切り離され、データベースの制御ファイルがクローズされます。

3. インスタンスの停止

インスタンスの停止によって、SGAがメモリーから解放され、Oracle Databaseインスタンスを構成するバックグラウンドのOracle Databaseプロセスが終了します。

例10-4 データベースの停止

オープンおよびマウントされたデータベースを正常に停止するには、次のように入力します。

```
SHUTDOWN
```

データベース停止の詳細は、「[データベースの停止](#)」を参照してください。また、「[SHUTDOWN](#)」コマンドも参照してください。

```
Database closed.  
Database dismounted.  
ORACLE instance shut down.
```

10.2.4 PDBの停止

プラガブル・データベース(PDB)を停止するには:

1. 必要な権限でPDBに接続します。
2. SHUTDOWNコマンドを実行します。

ノート:



- 現在のコンテナが PDB の場合、SHUTDOWN コマンドは、CDB インスタンスではなく PDB のみをクローズします。
- PDB には、非 CDB の SHUTDOWN TRANSACTIONAL または SHUTDOWN ABORT コマンドと同等の SHUTDOWN コマンドはありません。

データベース起動の詳細は、『[Oracle Database管理者ガイド](#)』を参照してください。また、『[STARTUP](#)』コマンドも参照してください。

10.3 REDOログ・ファイル

各Oracle Databaseには、2つ以上のREDOログ・ファイルが用意されています。1つのデータベースに対するREDOログ・ファイルのセットは、データベースのREDOログと呼ばれます。

REDOログは、データに加えられた変更を記録するために使用されます。たとえば、データベースに障害が発生した場合、REDOログを使用してデータベースをリカバリします。REDOログ自体の障害を防ぐため、Oracle Databaseでは、REDOログのミラー化によって、2つ以上のREDOログのコピーを異なるディスクに保持できます。

10.3.1 ARCHIVELOGモード

データベースをARCHIVELOGモードでオープンすると、オンラインREDOログをアーカイブできます。

アーカイブREDOログには、データベースに加えられたすべての変更が永久に保存されているため、SQL ALTER SYSTEMコマンドを使用すると、インスタンス障害のみでなくディスク障害も完全にリカバリできます。

REDOログ・ファイルおよびデータベースのアーカイブ・モードの詳細は、『[ARCHIVE LOG](#)』コマンドを参照してください。

アーカイブされた現行のログ・ファイルの詳細を表示するには、次のように入力します。

```
ARCHIVE LOG LIST
```

Database log mode	Archive Mode
Automatic archival	Enabled
Archive destination	/vobs/oracle/dbs/arch
Oldest online log sequence	221
Next log sequence to archive	222
Current log sequence	222

10.4 データベースのリカバリ

破損したデータベースがARCHIVELOGモードの場合、完全メディア・リカバリまたは不完全メディア・リカバリの操作の候補となります。メディア・リカバリ操作を開始するには、RECOVERコマンドを使用します。データのリカバリの詳細は、『[RECOVER](#)』コマンドを参照してください。

リカバリ操作を開始するには、DBA権限が必要です。

制御ファイルのバックアップ・ファイルを使用し指定時間までにデータベースをリカバリするには、次のように入力します。

```
RECOVER DATABASE UNTIL TIME '1998-11-23:12:47:30' -  
USING BACKUP CONTROLFILE
```

2つのオフライン表領域をリカバリするには、次のように入力します。

```
RECOVER TABLESPACE ts1, ts2
```

表領域のリカバリを続行する前に、対象の表領域がオフラインになっていることを確認してください。

11 SQL*Plusのグローバル化・サポート

グローバル化・サポートによって、ネイティブ言語によるデータの格納、処理および取得が可能です。Oracle Databaseに格納できる言語は、Oracle Databaseでサポートされている文字セットでエンコードされます。グローバル化・サポートによって、データベース・ユーティリティ、エラー・メッセージ、ソート順序および日付、時間、通貨、数値およびカレンダーの表記規則に、ネイティブ言語とロケールが適用されます。

トピック:

- [SQL*Plusコマンドラインでのグローバル化・サポートの構成について](#)
- [環境変数NLS_LANG](#)

グローバル化・サポートの詳細は、Oracle Technology Network上のグローバル化に関するノート (<http://www.oracle.com/technetwork/products/globalization/>)を参照してください。

また、「[グローバル化・サポートの概要](#)」を参照してください。

11.1 SQL*Plusコマンドラインでのグローバル化・サポートの構成について

SQL*Plusでは、環境変数NLS_LANGを使用すると、複数言語がサポートされます。SQL*Plusに別の言語を表示するには、次の項目を構成してからSQL*Plusを起動する必要があります。

- SQL*Plusクライアント環境のNLS_LANG
- インストール中のOracle Database

11.1.1 SQL*Plusクライアント

SQL*Plusクライアント環境の構成は、起動時にSQL*Plusによって読み取られる環境変数NLS_LANGを設定して行います。

11.1.2 Oracle Database

Oracle Database環境の構成は、必要な文字セットを指定してデータベースを作成して行います。

11.2 NLS_LANG環境変数

環境変数NLS_LANGには3つの要素があり、各要素は、グローバル化機能のサブセットを制御します。

選択した文字セットが、オペレーティング・システムおよびキーボードでサポートされている必要があります。追加のサポート・ソフトウェアをインストールする必要がある場合もあります。NLS_LANGおよびソフトウェア・サポートの詳細は、「[グローバル化のサポート環境の設定](#)」を参照してください。

SQL*Plusクライアントのロケール固有の動作を設定するには、NLSパラメータを使用します。これらのパラメータは、サーバーの初期化パラメータとしてなど、いくつかの方法で指定できます。サーバーの動作を制御する設定の詳細は、「[NLSデータベースのパラメータ](#)」を参照してください。

NLS_LANGの構文は、次のとおりです。

```
NLS_LANG = language_territory.charset
```

*language*には、Oracle Databaseのメッセージ、ソート順序および日と月の名前で使用される表記規則を指定します。たとえば、日本語のメッセージを受信するには、*language*にJAPANESEを設定します。*language*を指定しない場合は、デフォルトでAMERICANが設定されます。

*territory*には、デフォルトの日付、通貨および数値の表記規則を指定します。たとえば、日本語地域の書式を使用するには、*territory*にJAPANを設定します。*territory*を設定しない場合は、*language*の値からデフォルト値が導出され、AMERICAに設定されます。

SQL*Plusコマンドラインでは、*charset*に、SQL*Plusがデータ処理で使用する文字セット・コードを指定します。一般的に、ユーザー端末の文字セット・コードに適したコードを指定します。矛盾する組合せを設定できますが、動作はしません。たとえば、次のように西ヨーロッパの文字セットを使用しながら、日本語をサポートすることはできません。

```
NLS_LANG=JAPANESE_JAPAN. WE8DEC
```

ただし、日本語はUnicode文字セットを使用してサポートできます。次に例を示します。

```
NLS_LANG=JAPANESE_JAPAN. UTF8
```

11.2.1 NLS_LANGの設定表示

SELECTコマンドを使用して、NLS_LANGの設定を表示できます。

```
SELECT * FROM NLS_SESSION_PARAMETERS;
```

NLS_LANGUAGEおよびNLS_TERRITORYの値は、NLS_LANG変数の言語および地域のコンポーネントに対応しています。

また、NLSの動的パフォーマンスのビュー表のV\$NLS_VALID_VALUESを問い合わせると、NLS_SORT、NLS_LANGUAGE、NLS_TERRITORYおよびNLS_CHARACTERSETの有効な値のリストを表示できます。

11.3 NLS_LANGの設定

グローバル化の機能を制御するNLS_LANG環境変数を設定できます。

例11-1 SQL*Plusの日本語サポートの構成(Windows)

1. 現行のSQL*Plusセッションが終了していることを確認します。
2. 「スタート」→「設定」→「コントロール パネル」から「システム」を開きます。
3. 「詳細設定」タブを選択してから、「環境変数」を選択します。
4. 新規の環境変数NLS_LANGを、Japanese_Japan.JA16SJISという値を指定して作成します。
5. ここでの設定を有効にするために、Windowsを再起動する必要があります。

例11-2 SQL*Plusの日本語サポートの構成(UNIX)

1. 現行のSQL*Plusセッションが終了していることを確認します。
2. 使用しているUNIXのシェルによってsetまたはsetenvのいずれかを使用して、NLS_LANG変数を設定します。たとえばCシェルでは、次のように入力します。

```
setenv NLS_LANG Japanese_Japan. UTF8
```

または

```
setenv NLS_LANG Japanese_Japan. JA16SJIS
```


または

```
setenv NLS_LANG Japanese_Japan. JA16EUC
```

NLS_LANGパラメータの正確な値は、UNIX端末のロケール設定によって決まります。NLS_LANG設定の詳細は、「[NLS_LANGの値の指定](#)」を参照してください。

例11-3 Oracle Databaseの日本語サポートの構成

UTF-8文字コードを使用して、日本語文字セットでデータを格納するには、Oracle DatabaseがAL32UTF8文字セットで作成されていることを確認します。US7ASCII以外の文字セットでデータベースを作成する方法については、Oracle Databaseのインストレーション・ガイドを参照してください。

第III部 SQL*Plusリファレンス

第III部では、SQL*Plusのコマンド・リファレンスおよびSQL*Plusのエラー・メッセージについて説明します。

第III部は、次の章で構成されています。

- [SQL*Plusコマンド・リファレンス](#)
- [SQL*Plusのエラー・メッセージ](#)

12 SQL*Plusコマンド・リファレンス

この章では、SQL*Plusコマンドをアルファベット順に説明します。それぞれのコマンドについて、次の内容を説明します。

セクション	説明
構文	コマンドの入力方法を示します。また、コマンドの基本的な使用方法を簡単に説明します。
用語	構文の中に含まれるそれぞれの項または句の働きについて説明します。
用途	コマンドの使用方法およびコマンドの働きに関する追加情報を示します。
例	コマンドの例を示します。

長いSQL*Plusコマンドは、行末にハイフン(-)を付けて[Return]を押すと、次の行に続けて入力できます。必要な場合は、ハイフンの前に空白も入力できます。SQL*Plusでは、右山カッコ(>)が次の行のプロンプトとして表示されます。

SQL*Plusコマンドは、必ずしもセミコロンで終了する必要はありません。コマンドの入力が終了したら、[Return]を押します。SQL*Plusコマンドの末尾にセミコロンを入力してもかまいません。

12.1 SQL*Plusコマンド一覧

コマンド	説明
@ (at sign)	指定したスクリプトの SQL*Plus 文を実行します。スクリプトは、ローカル・ファイル・システムまたは Web サーバーからコールされます。
@@ (double at sign)	スクリプトを実行します。このコマンドは、@ (アットマーク) コマンドと同じです。スクリプトをコールしているパスと同じパス内に指定されているスクリプトを検索するため、ネストしたスクリプトの実行に便利です。
/ (slash)	SQL コマンドまたは PL/SQL ブロックを実行します。
ACCEPT	入力された行を読み込み、指定された置換変数に行の内容を格納します。
APPEND	指定したテキストをバッファ内のカレント行の終わりに追加します。
ARCHIVE LOG	REDO ログ・ファイルに関する情報を表示します。
ATTRIBUTE	オブジェクト型列の特定の属性の表示特性を指定し、1 つまたはすべての属性の現行の表示特性を表示します。
BREAK	レポート内で書式設定を変更する場合の場所と方法を指定したり、現行のブレイク定義を表示します。
BTITLE	各レポート・ページの下部に、指定したタイトルを配置して書式設定したり、現行の BTITLE 定義を表示します。
CHANGE	バッファ内のカレント行のテキストを変更します。
CLEAR	指定したオプション(たとえば、BREAKS や COLUMNS など)の現行の値または設定を、リセットまたは消去します。
COLUMN	特定の列の表示特性を指定したり、1 つまたはすべての列の現行の表示特性を表示します。
COMPUTE	選択した行のサブセットについて、各種の標準計算方法を使用してサマリー行を計算して出力したり、すべての COMPUTE 定義を表示します。
CONNECT	指定したユーザーで Oracle Database に接続します。

コマンド	説明
COPY	問合せから、同じまたは別のデータベース内の表に結果をコピーします。
DEFINE	置換変数を指定して、その変数に CHAR 値を割り当てたり、1 つまたはすべての変数の値および変数型を表示します。
DEL	バッファの 1 つ以上の行を削除します。
DESCRIBE	指定した表、ビューまたはシノニムの列定義を表示したり、指定したファンクションまたはプロシージャの仕様を表示します。
DISCONNECT	データベースに対する保留中の変更をコミットし、現行のユーザーを Oracle Database からログオフします。ただし、SQL*Plus は終了しません。
EDIT	指定したファイルの内容またはバッファの内容を対象として、オペレーティング・システムのテキスト・エディタを起動します。
EXECUTE	1 つの PL/SQL 文を実行します。
EXIT	SQL*Plus を終了し、オペレーティング・システムに制御を戻します。
GET	オペレーティング・システム・ファイルをバッファにロードします。
HELP	SQL*Plus コマンドラインのヘルプにアクセスします。
HISTORY	現在の SQL*Plus セッションで発行されたコマンド、SQL*Plus コマンドおよび SQL 文または PL/SQL 文の履歴をリコールします。
HOST	SQL*Plus を終了せずに、オペレーティング・システムのコマンドを実行します。
INPUT	バッファ内のカレント行の後に、1 つ以上の新しい行を追加します。
LIST	バッファ内の 1 つ以上の行を表示します。
PASSWORD	パスワードを入力デバイスにエコー表示せずに変更できるようにします。
PAUSE	指定したテキストを表示し、その後ユーザーが[Return]を押すまで待機します。
PRINT	バインド変数の現行の値を表示します。

コマンド	説明
PROMPT	指定したメッセージをユーザーの画面に送信します。
EXIT	SQL*Plus を終了し、オペレーティング・システムに制御を戻します。EXIT と同じ機能です。
RECOVER	1 つ以上の表領域またはデータ・ファイル、あるいはデータベース全体のメディア・リカバリを実行します。
REMARK	スクリプト内でコメントを開始します。
REPFOOTER	各レポートの下部に、指定したレポート・フッターを配置して書式設定したり、現行の REPFOOTER 定義を表示します。
REPHEADER	各レポートの上部に、指定したレポート・ヘッダーを配置して書式設定したり、現行の REPHEADER 定義を表示します。
RUN	現在、SQL バッファに格納されている SQL コマンドまたは PL/SQL ブロックを表示し、実行します。
SAVE	バッファの内容を、オペレーティング・システム・ファイル(スクリプト)に保存します。
SET	現行のセッションに対する SQL*Plus 環境を変更するシステム変数を設定します。
SHOW	SQL*Plus システム変数の値または現行の SQL*Plus 環境を表示します。
SHUTDOWN	現在実行中の Oracle Database インスタンスを停止します。
SPOOL	問合せの結果を、オペレーティング・システム・ファイルに格納し、必要に応じてそのファイルをプリンタに送信します。
START	指定したスクリプトの SQL 文を実行します。スクリプトは、ローカル・ファイル・システムまたは Web サーバーから SQL*Plus コマンドラインでコールされます。
STARTUP	Oracle Database インスタンスを起動し、データベースをマウントおよびオープンします。
STORE	現行の SQL*Plus 環境の属性を、オペレーティング・システムのスクリプトに保存します。
TIMING	経過時間に関するタイミング・データの記録、現行のタイマーのタイトルとタイミング・データの表示、またはアクティブなタイマー数の表示を行います。

コマンド	説明
<u>TTITLE</u>	各レポート・ページの上部に、指定したタイトルを配置して書式設定したり、現行の TTITLE 定義を表示します。
<u>UNDEFINE</u>	明示的に(DEFINE コマンドを使用して)定義したか、または暗黙的に(START コマンドでの引数を使用して)定義した 1 つ以上の置換変数を削除します。
<u>VARIABLE</u>	PL/SQL で参照可能なバインド変数を宣言するか、1 つまたはすべての変数の現在の表示特性を表示します。
<u>WHENEVER OSERROR</u>	オペレーティング・システム・コマンドによってエラーが発生した場合は、SQL*Plus を終了します。
<u>WHENEVER SQLERROR</u>	SQL コマンドまたは PL/SQL ブロックによってエラーが発生した場合は、SQL*Plus を終了します。
<u>XQUERY</u>	XQuery 1.0 文を実行します。

12.2 @ (アットマーク)

構文

```
@{url | file_name[. ext] } [arg...]
```

指定したスクリプトのSQL*Plus文を実行します。スクリプトは、ローカル・ファイル・システムまたはWebサーバーからコールされます。@コマンドは、@@およびSTARTと同様の働きをします。

項

url

指定したWebサーバーで実行するスクリプトのURLを指定します。SQL*Plusは、HTTPおよびFTPプロトコルをサポートしますが、HTTPSはサポートしません。今回のリリースでは、`http://username:password@machine_name.domain...`形式のHTTP認証はサポートされていません。

file_name[. ext]

実行するスクリプトを指定します。extを指定しない場合、SQL*Plusは、デフォルトのコマンド・ファイル拡張子(通常はSQL)が指定されたものとみなします。デフォルトの拡張子を変更する方法については、「[SET SUF\[FIX\] {SQL | text}](#)」を参照してください。

@*file_name.ext*を入力すると、SQL*Plusは、現行のデフォルト・ディレクトリで、そのファイル名および拡張子を持つファイルを検索します。現行のディレクトリで該当するファイルが見つからない場合、SQL*Plusは、そのファイルを見つけるためにシステム依存パスを検索します。オペレーティング・システムによっては、このパス検索がサポートされていない場合もあります。オペレーティング・システム固有の情報については、ご使用のオペレーティング・システムのプラットフォーム固有のOracleマニュアルを参照してください。

arg...

スクリプト内のパラメータに渡すデータ項目を指定します。1つ以上の引数を入力すると、SQL*Plusは、該当する値をスクリプトのパラメータ(&1、&2など)に代入します。最初の引数によって、出現したすべての&1が置換され、2番目の引数によって、出現したすべての&2が置換されます。

@コマンドは、引数に値を指定してパラメータを定義します。このセッションで同じスクリプトを再度実行する場合は、新しい引数を入力することも、引数を省略して現行の値を使用することもできます。パラメータの使用の詳細は、「[置換変数の使用](#)」を参照してください。

使用方法

スクリプト起動時は、COLUMNコマンドなどで事前に設定されたすべての内容が有効です。スクリプトで設定が変更される場合、新しい値が有効になるのは、スクリプト終了後です。

スクリプトには、通常、対話形式で入力するコマンド(主に、SQLコマンド、SQL*PlusコマンドまたはPL/SQLブロック)であれば、どのコマンドでも組み込むことができます。

STARTコマンドが使用禁止にされると(「[SQL*Plus、SQLおよびPL/SQLコマンドの使用禁止](#)」を参照)、@コマンドも使用禁止になります。STARTコマンドについては、「[START](#)」を参照してください。

SQL*Plusは、@コマンドを実行する前にSQLTERMINATOR(デフォルトではセミコロン)を削除します。コマンドにセミコロンが必要な場合、もう1つのSQLTERMINATORを追加します。詳細は、「[SET SQLT\[ERMINATOR\] {; | c | ON | OFF}](#)」を参照してください。

例

名前がPRINTRPTで拡張子がSQLであるスクリプトを実行するには、次のように入力します。

```
@PRINTRPT
```

名前がWKRPTで拡張子がQRYであるスクリプトを実行するには、次のように入力します。

```
@WKRPT.QRY
```

YEARENDという名前のスクリプトをURLで指定して実行し、値を通常どおりの方法でYEARENDで参照する変数に渡すには、次のように入力します。

```
@HTTP://machine_name.domain:port/YEAREND.SQL VAL1 VAL2
```

```
@FTP://machine_name.domain:port/YEAREND.SQL VAL1 VAL2
```

SQLレポート用に構成されたWebサーバー上で、SQL*Plusを要求して動的スクリプトを実行するには、次のように入力します。

```
@HTTP://machine_name.domain:port/SCRIPTSERVER?ENDOFYEAR VAL1 VAL2
```

12.3 @@ (二重アットマーク)

構文

```
@@{url | file_name[. ext] } [arg...]
```

スクリプトを実行します。このコマンドは、@ (アットマーク) コマンドとほぼ同じです。ネストしたスクリプトを実行すると、このコマンドはコールされたスクリプトと同じパス内またはurlでネストしたスクリプトを検索します。@@コマンドは、@およびSTARTと同様の働きをします。

項

url

指定したWebサーバーで実行するスクリプトのURLを指定します。SQL*Plusは、HTTPおよびFTPプロトコルをサポートしますが、HTTPSはサポートしません。今回のリリースでは、http://username:password@machine_name.domain...形式のHTTP認証はサポートされていません。

file_name[. ext]

実行するネストしたスクリプトを指定します。extを指定しない場合、SQL*Plusは、デフォルトのコマンド・ファイル拡張子(通常はSQL)が指定されたものとみなします。デフォルトの拡張子を変更する方法については、「[SET SUF\[FIX\] {SQL | text}](#)」を参照してください。

あるスクリプトから、@@file_name.extを入力すると、SQL*Plusは、そのスクリプトと同じディレクトリからfile_name.extを実行します。

@@file_name.extを対話形式で入力すると、SQL*Plusは、現行の作業ディレクトリ、またはこのコマンドがコールされたスクリプトと同じurlからfile_name.extを実行します。該当するファイルが見つからない場合、SQL*Plusは、そのファイルを見つけるためにシステム依存パスを検索します。オペレーティング・システムによっては、このパス検索がサポートされていない場合があります。オペレーティング・システム固有の情報については、ご使用のオペレーティング・システムのプラットフォーム固有のOracleマニュアルを参照してください。

arg...

スクリプト内のパラメータに渡すデータ項目を指定します。1つ以上の引数を入力すると、SQL*Plusは、該当する値をスクリプトのパラメータ(&1、&2など)に代入します。最初の引数によって、出現したすべての&1が置換され、2番目の引数によって、出現したすべての&2が置換されます。

@@コマンドは、引数に値を指定してパラメータを定義します。このセッションで同じスクリプトを再度実行する場合は、新しい引数を入力することも、引数を省略して現行の値を使用することもできます。パラメータの使用の詳細は、「[置換変数の使用](#)」を参照してください。

使用方法

スクリプト起動時は、COLUMNコマンドなどで事前に設定されたすべての内容が有効です。スクリプトで設定が変更される場合、新しい値が有効になるのは、スクリプト終了後です。

スクリプトには、通常、対話形式で入力するコマンド(主に、SQLコマンドまたはSQL*Plusコマンド)であれば、どのコマンドでも組み込むことができます。

STARTコマンドが使用禁止にされると(「[SQL*Plus、SQLおよびPL/SQLコマンドの使用禁止](#)」を参照)、@@コマンドも使用禁止になります。詳細は、「[SPOOL](#)」を参照してください。

SQL*Plusは、@@コマンドを実行する前にSQLTERMINATOR(デフォルトではセミコロン)を削除します。これに対処するには、SQLTERMINATORをもう1つ追加します。詳細は、[SET SQLT\[ERMINATOR\] {; | c | ON | OFF}](#)を参照してください。

例

PRINTRPTという次のようなスクリプトがあるとします。

```
SELECT DEPARTMENT_ID, CITY FROM EMP_DETAILS_VIEW WHERE SALARY>12000;  
@EMPRPT.SQL  
@@ WKRPT.SQL
```

PRINTRPTを起動し、@コマンドの部分になると、現行の作業ディレクトリ内でEMPRPTというスクリプトが検索され、実行されます。PRINTRPTの@@コマンドの部分では、PRINTRPTと同じパスの中でWKRPTというスクリプトが検索され、実行されます。

同じスクリプトPRINTRPTがWebサーバー上にあり、START `HTTP://machine_name.domain:port/PRINTRPT`で実行するとします。@コマンドの部分になると、現行の作業ディレクトリ内でEMPRPTというスクリプトが検索され、実行されます。PRINTRPTの@@コマンドの部分では、PRINTRPTと同じ
`url(HTTP://machine_name.domain:port/WKRPT.SQL)`でWKRPTというスクリプトが検索され、実行されます。

12.4 / (スラッシュ)

構文

/ (スラッシュ)

最後に実行されたSQLコマンドまたはSQLバッファに格納されているPL/SQLブロックを実行します。

使用方法

/(スラッシュ)コマンドは、コマンド・プロンプトまたは複数行コマンドの行番号プロンプトに入力できます。

スラッシュ・コマンドは、RUNと同様の働きをしますが、コマンドを表示しません。

スラッシュ・コマンドを使用してSQLコマンドまたはPL/SQLブロックを実行しても、SQLバッファ内のコマンドにエラーがないかぎり、バッファ内のカレント行番号は変更されません。エラーがある場合、SQL*Plusは、カレント行番号をエラーがある行番号に変更します。

例

次のSQLスクリプトを入力したとします。

```
SELECT CITY, COUNTRY_NAME FROM EMP_DETAILS_VIEW WHERE SALARY=12000;
```

バッファ内のコマンドを再実行するには、スラッシュ(/)を入力します。

/

CITY	COUNTRY_NAME
Seattle	United States of America
Oxford	United Kingdom
Seattle	United States of America

12.5 ACCEPT

構文

```
ACC[EPT] variable [NUM[BER] | CHAR | DATE | BINARY_FLOAT | BINARY_DOUBLE] [FOR[MAT] format] [DEF[AULT] default] [PROMPT text|NOPR[OMPT]] [HIDE]
```

1行の入力を読み取り、所定の置換変数に格納します。

項

variable

値を格納する変数の名前を指定します。*variable*を指定しない場合、SQL*Plusによって変数が作成されます。

NUM[BER]

*variable*がNUMBERデータ型に設定されます。応答がデータ型と一致しない場合、ACCEPTはエラー・メッセージを表示し、再度プロンプトを表示します。

CHAR

*variable*がCHARデータ型に設定されます。CHARの最大長は240バイトです。マルチバイト文字セットを使用する場合は、1つのCHARのサイズが1バイトを超えることがあります。

DATE

応答が、有効なDATE書式に設定されます。応答が、有効なDATE書式でない場合、ACCEPTはエラー・メッセージを表示し、再度プロンプトを表示します。データ型はCHARです。

BINARY_FLOAT

*variable*がBINARY_FLOATデータ型に設定されます。BINARY_FLOATは、米国電気電子技術者協会(IEEE)の2進浮動小数点演算についてのIEEE規格(IEEE規格754-1985)に実質的に準拠する浮動小数点数です。

BINARY_DOUBLE

*variable*がBINARY_DOUBLEデータ型に設定されます。BINARY_DOUBLEは、米国電気電子技術者協会(IEEE)の2進浮動小数点演算についてのIEEE規格(IEEE規格754-1985)に実質的に準拠する浮動小数点数です。

FOR[MAT]

応答の入力書式を指定します。応答が指定された書式と一致しない場合、ACCEPTはエラー・メッセージを表示し、再度プロンプトを表示します。文字書式で指定した文字長よりも長い文字を入力すると、エラー・メッセージが表示され、値を再入力する必要があります。数値書式で指定した精度を超える数値を入力すると、エラー・メッセージが表示され、値を再入力する必要があります。書式要素は、A10または9.999などのテキスト定数である必要があります。書式要素の完全なリストは、[「COLUMN」](#)を参照してください。

データ型がDATEである場合は、dd/mm/yyなどのOracle Databaseの日付書式が有効です。DATEに書式を指定しない場合は、デフォルトで現行セッションのNLS_DATE_FORMATになります。Oracle Database日付書式の詳細は、[「書式モデル」](#)を参照してください。

DEF[AULT]

応答がない場合のデフォルト値を設定します。応答が定義されている場合は、指定した書式に従っている必要があります。

```
PROMPT text
```

ユーザーが指定した`variable`の値を受け入れる前に、画面にテキストを表示します。

```
NOPR[OMPT]
```

1行スキップし、プロンプトを表示せずに入力を待ちます。

```
HIDE
```

入力した応答が表示されません。

変数を表示または参照するには、`DEFINE`コマンドを使用します。詳細は、「[DEFINE](#)」コマンドを参照してください。

例

「Password:」というプロンプトを表示し、`pswd`という名前の`CHAR`変数に応答を入れて、表示しないようにするには、次のように入力します。

```
ACCEPT pswd CHAR PROMPT 'Password: ' HIDE
```

「Enter weekly salary:」というプロンプトを表示し、デフォルト値が000.0の`salary`という名前の`NUMBER`変数に応答を入れるには、次のように入力します。

```
ACCEPT salary NUMBER FORMAT '999.99' DEFAULT '000.0' -  
PROMPT 'Enter weekly salary: '
```

「Enter date hired:」というプロンプトを表示し、書式が`dd/mm/yyyy`でデフォルト値が01/01/2003である`DATE`変数`hired`に応答を入れるには、次のように入力します。

```
ACCEPT hired DATE FORMAT 'dd/mm/yyyy' DEFAULT '01/01/2003' -  
PROMPT 'Enter date hired: '
```

「Enter employee lastname:」というプロンプトを表示し、`lastname`という名前の`CHAR`変数に応答を入れるには、次のように入力します。

```
ACCEPT lastname CHAR FORMAT 'A20' -  
PROMPT 'Enter employee lastname: '
```

12.6 APPEND

構文

```
A[PPEND] text
```

*text*は追加するテキストを示します。

指定したテキストをSQLバッファ内のカレント行の終わりに追加します。

*text*とその前の文字を空白で区切るには、APPENDと*text*の間に空白を2つ入力します。

セミコロンで終わるテキストを追加するには、コマンドを2つのセミコロンで終了します(セミコロンが1つのみの場合、SQL*Plusはそれをオプションのコマンド終了記号と解釈します)。

例

バッファの最初の行に、カンマ・デリミタ、空白および列名CITYを追加するには、次のようにその行を表示してその行をカレント行にします。

```
1
1* SELECT DEPARTMENT_ID
```

次に、APPENDを入力します。

```
APPEND , CITY
1
1* SELECT DEPARTMENT_ID, CITY
```

この行にセミコロンを追加するには、次のように入力します。

```
APPEND ;;
```

SQL*Plusは、最初のセミコロンをこの行に追加し、2番目のセミコロンをAPPENDコマンドの終了記号として解釈します。

12.7 ARCHIVE LOG

構文

ARCHIVE LOG LIST

REDOログ・ファイルに関する情報を表示します。

項

LIST

アーカイブ予定のREDOログ・ファイルの範囲、現行のログ・ファイル・グループの順序番号および現行のアーカイブ先(オプションのコマンド・テキストまたは初期化パラメータLOG_ARCHIVE_DESTで指定します)を表示するよう要求します。

ARCHIVELOGモードおよび自動アーカイブを使用すると、次のように表示されます。

ARCHIVE LOG LIST

Database log mode	Archive Mode
Automatic archival	Enabled
Archive destination	/vobs/oracle/dbs/arch
Oldest online log sequence	221
Next log sequence to archive	222
Current log sequence	222

現行のログ・グループと次にアーカイブするログ・グループのログ順序番号は同一なので、自動アーカイブによって、現行のログ・グループまでのすべてのログ・グループがアーカイブされています。

ARCHIVELOGモードで、自動アーカイブを無効にする場合、最後の3行を次のように指定します。

Oldest online log sequence	222
Next log sequence to archive	222
Current log sequence	225

NOARCHIVELOGモードの場合は、「アーカイブする次のログ順序 %lu」の行が表示されません。

ログ順序番号は、ログ・ライターが他のREDOログ・ファイル・グループに書込みを開始するたびに増加します。ログ順序番号は、使用したログの数を示すものではありません。オンラインREDOログ・ファイル・グループが再利用されると、毎回その内容に新しいログ順序番号が割り当てられます。

使用方法

SYSOPERまたはSYSDBAでデータベースに接続している必要があります。データベース接続については、「[CONNECT](#)」コマンドを参照してください。

アーカイブ先の指定方法については、ご使用のプラットフォーム固有のOracle Databaseのマニュアルを参照してください。

ノート:



ARCHIVE LOG LIST を使用できるのは、現行のインスタンスのみです。アーカイブを開始(START)および停止(STOP)するには、SQL コマンド ALTER SYSTEM を使用します。SQL コマンドの使用の詳細は、[『Oracle Database SQL 言語リファレンス』](#)を参照してください。

12.8 ATTRIBUTE

構文

```
ATTR[IBUTE] [type_name.attribute_name [option ...]]
```

*option*は、次のいずれかの句を表します。

```
ALI[AS] alias CLE[AR] FOR[MAT] format LIKE {type_name.attribute_name | alias} ON | OFF
```

オブジェクト型列の特定の表示特性(NUMBERデータの書式など)を指定します。列および属性は、共通のネームスペースを共有するため、同じ名前は指定できません。

さらに、1つまたはすべての属性の現行の表示特性を表示します。

指定した属性についてのみ現行の表示特性を表示するには、ATTRIBUTEの後に、*type_name.attribute_name*のみを入力し、他の句は省略します。現行の属性すべてについて表示特性を表示するには、句を指定せずにATTRIBUTEを入力します。

項

```
type_name.attribute_name
```

オブジェクト型、*type_name*の特定のオブジェクトに関する属性セットのデータ項目(属性の名前)を指定します。

同じオブジェクト型のオブジェクトを選択した場合は、*type_name.attribute_name*に対するATTRIBUTEコマンドは、このセッション内で参照する該当オブジェクトすべてに適用されます。

```
ALI[AS] alias
```

指定した別名が*type_name.attribute_name*に割り当てられます。これは、他のATTRIBUTEコマンドで*type_name.attribute_name*を参照するために使用できます。

```
CLE[AR]
```

*attribute_name*の表示特性をリセットします。指定する書式は、変数ではなく、A10や\$9,999などのテキスト定数である必要があります。

```
FOR[MAT] format
```

列の表示書式を指定します。指定する書式は、変数ではなく、A10や\$9,999などのテキスト定数である必要があります。

```
LIKE {type_name.attribute_name | alias}
```

別の属性の表示特性がコピーされます。LIKEによってコピーされるのは、現行のATTRIBUTEコマンド内の他の句で定義されていない特性のみです。

```
ON | OFF
```

列の表示特性の状態を制御します。OFFを指定すると、特性の定義に影響することなく、属性の特性が非表示になります。ONを指定すると、特性が表示されます。

使用方法

1つ以上の属性について、ATTRIBUTEコマンドをいつでも入力できます。属性をOFFに切り替えるか、CLEAR COLUMNコマンドを使用しないかぎり、各属性について設定されているすべての属性特性がセッションの終わりまで有効になります。したがって、入力したATTRIBUTEコマンドによって、複数のSQL SELECTコマンドについて特定の属性の表示特性を制御できます。

同じ属性について複数のATTRIBUTEコマンドを入力すると、SQL*Plusは、それらのコマンドのすべての句をまとめて適用します。複数のATTRIBUTEコマンドで、同じ属性に対して同じ句が適用される場合は、最後に入力した句が出力に適用されます。

例

オブジェクト型EMPLOYEE_TYPEのLAST_NAME属性の幅を20文字に設定するには、次のように入力します。

```
ATTRIBUTE EMPLOYEE_TYPE. LAST_NAME FORMAT A20
```

オブジェクト型EMPLOYEE_TYPEのSALARY属性の書式を、100万ドルの位まで表示し、セントの単位を四捨五入し、3桁区切りにカンマを使用し、値が0(ゼロ)のときは\$0.00を表示するように設定するには、次のように入力します。

```
ATTRIBUTE EMPLOYEE_TYPE. SALARY FORMAT $9,999,990.99
```

12.9 BREAK

構文

```
BRE[AK] [ON report_element [action [action]]] ...
```

*report_element*の構文は、{*column*|*expr*|ROW|REPORT}です。

*action*の構文は、[SKI[P] *n*][SKI[P]] PAGE] [NODUP[LICATES]]|DUP[LICATES]]です。

次に示すような、レポート内での変更位置および実行する書式設定のアクションを指定します。

- 特定の列について重複する値の非表示
- 特定の列値が変更されるたびに1行スキップ
- 特定の列値が変更されるたび、またはレポートの終わりにCOMPUTEによって計算された数値を出力

「[COMPUTE](#)」コマンドを参照してください。

現在のBREAKの定義を表示するには、句を指定せずにBREAKと入力します。

項

```
ON column [action [action]]
```

アクションを指定する場合は、指定した列(ブレイク列)でブレイクが発生するたびにSQL*Plusがとるアクションを指定します。(columnには、表もビューも追加できません。これを行うために、SQL文の中でその列に別名を付けることができます。)ブレイクとは、列または式の値の変更、行の出力またはレポートの終わりの3つのイベントのいずれかです。

アクションを省略した場合、BREAK ON列は、columnの中の重複する値を出力せず、対応するCOMPUTEコマンドに指定された計算をSQL*Plusが実行するレポート内の位置にマークが設定されます。

ON columnは、1回以上指定できます。次の例のように、複数のON句を指定したとします。

```
BREAK ON DEPARTMENT_ID SKIP PAGE ON JOB_ID -  
SKIP 1 ON SALARY SKIP 1
```

最初のON句は、最も外側のブレイク(この場合はON DEPARTMENT_ID)を表し、最後のON句は、最も内側のブレイク(この場合はON SALARY)を表します。SQL*Plusは、それぞれの出力行から指定されたブレイクを検索します。このとき、最も外側のブレイクから始めて、最も内側のブレイクに到達するまで、入力された句の順序で処理が進められます。例では、SQL*Plusは、DEPARTMENT_ID、JOB_ID、SALARYの順で値の変更を検索します。

次にSQL*Plusは、最も内側のブレイクについて指定されているアクションから始めて、最も外側のブレイクまで、逆の順で各アクションを実行します(この例では、ON SALARYの場合のSKIP 1から始めて、ON DEPARTMENT_IDの場合のSKIP PAGEまで進みます)。SQL*Plusは、初期検索で見つかった最初のブレイクについて指定されているアクションまでの各アクションを実行します。

たとえば、ある行でJOB_IDの値が変更され、DEPARTMENT_IDおよびSALARYの値には変更がなかった場合、SQL*Plusは、2行スキップしてからその行を出力します(1行はON SALARY句のSKIP1の結果で、もう1行はON JOB_ID句のSKIP1の結果です)。

ON columnを使用する場合は必ず、SQL SELECTコマンドの中でORDER BY句も使用する必要があります。通常、BREAKコマンドの中で使用する列は、ORDER BY句の中にも同じ順序で使用します(ただし、ORDER BY句で指定したすべての列をBREAKコマンドの中で指定しなくてもかまいません)。これは、レポート上の意味のない位置でのブレイクの発生を回避するためです。

この項の前半で指定したBREAKコマンドが使用されている場合、次のSELECTコマンドによって、有効な結果が得られます。

```
SELECT DEPARTMENT_ID, JOB_ID, SALARY, LAST_NAME
FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000
ORDER BY DEPARTMENT_ID, JOB_ID, SALARY, LAST_NAME;
```

DEPARTMENT_IDが同じであるすべての行がまとめて1ページに出力され、そのページの中では、JOB_IDが同じであるすべての行が1グループにまとめられます。それぞれの職種(JOB)グループの中では、SALARYが同じである職種がグループにまとめられます。このBREAKコマンドにはLAST_NAMEは指定されていないので、LAST_NAME内にブレイクがあってもアクションは実行されません。

BREAKコマンドでは、NULL同士は等しいとみなされますが、その他の値とは等しくありません。これは、WHERE句でのNULLの扱いとは異なります。

```
ON expr [action [action]]
```

アクションを挿入する場合は、式の値が変更されたときにSQL*Plusがとるアクションを指定します。

アクションを省略した場合、BREAK ON *expr*は、*expr*の中で重複する値を出力せず、対応するCOMPUTEコマンドに指定された計算をSQL*Plusが実行する位置にマークを設定します。

1つ以上の表列を含む式、あるいはSQL SELECTまたはSQL*PlusのCOLUMNコマンドでレポート列に割り当てられている別名を含む式を使用できます。BREAKコマンドの中で式を使用する場合は、SELECTコマンドの中での順序と同じ順序で*expr*を入力する必要があります。たとえば、SELECTコマンドの中の式がa+bである場合、BREAKコマンドの中でSELECTコマンドの中の式を参照するときに、b+aまたは(a+b)はいずれも使用できません。

ON *column*に指定した情報は、ON *expr*にも適用できます。

```
ON ROW [action [action]]
```

アクションを挿入する場合は、SQL SELECTコマンドが行を戻したときにSQL*Plusがとるアクションを指定します。ROWブレイクは、BREAKコマンド内のどこに指定してあっても、最も内側のブレイクになります。特定の行についてBREAKを指定する場合は、必ずアクションを指定します。

```
ON REPORT [action]
```

対応するCOMPUTEコマンドに指定された計算を、SQL*Plusが実行するレポート内の位置にマークを設定します。総合計またはその他の総計値を出力するには、BREAK ON REPORTとCOMPUTEを併用します。

REPORTブレイクは、BREAKコマンド内のどこに指定してあっても、最も外側のブレイクになります。

SQL*Plusはレポートの終わりに1ページをスキップしないので、BREAK ON REPORT SKIP PAGEは使用できません。

```
SKI [P] n
```

ブレイクが発生した行を出力する前に、*n*行スキップします。BREAK SKIP *n*は、PREFORMATがSET ONでないかぎり、SET MARKUP HTML ONモードでは動作しません。

```
[SKI [P]] PAGE
```

ブレイクが発生した行を出力する前に、1ページ分として定義されている行数のみスキップします。1ページの行数は、SETコマンドのPAGESIZE句を使用して設定できます。PAGESIZEは、SQL*Plusが1ページとみなす行数を変更するのみです。したがって、SKIP PAGEを指定しても、NEWPAGE 0を指定しないかぎり、物理的なページ・ブレイク文字は生成されません。また、レポートに印刷されるデータの最終行の後にブレイクがある場合、SQL*Plusはこのページをスキップしません。

```
NODUP[LICATES]
```

あるブレーク列の値が前の行の列値と同じ場合は、その値ではなく空白が出力されます。

DUP[LICATES]

選択した各行のブレーク列の値が出力されます。

現行のブレーク定義を表示するには、句を指定せずに、BREAKを入力します。

使用方法

新しいBREAKコマンドを入力すると、前のBREAKコマンドは無効になります。

BREAKコマンドを取り消すには、CLEAR BREAKSを使用します。

例

職種の重複値、SALARYの平均および追加でSALARYの合計を出力するレポートを作成するには、次のコマンドを入力します。(この例では、部門50および80から、SH_CLERKおよびSA_MANの職種のみが選択されます)。

```
BREAK ON DEPARTMENT_ID ON JOB_ID DUPLICATES
COMPUTE SUM OF SALARY ON DEPARTMENT_ID
COMPUTE AVG OF SALARY ON JOB_ID
SELECT DEPARTMENT_ID, JOB_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE JOB_ID IN ('SH_CLERK', 'SA_MAN')
AND DEPARTMENT_ID IN (50, 80)
ORDER BY DEPARTMENT_ID, JOB_ID;
```

DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
	50 SH_CLERK	Taylor	3200
	SH_CLERK	Fleaur	3100
	.		
	.		
	SH_CLERK	Gates	2900

DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
	50 SH_CLERK	Perkins	2500
	SH_CLERK	Bell	4000
	.		
	.		
	SH_CLERK	Grant	2600

	avg		3215

DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY

	sum		64300
	80 SA_MAN	Russell	14000
	SA_MAN	Partners	13500
	SA_MAN	Errazuriz	12000
	SA_MAN	Cambrault	11000
	SA_MAN	Zlotkey	10500

	avg		12200

DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY

sum			61000

25 rows selected.

12.10 BTITLE

構文

```
BTI[TLE] [printspec [text | variable] ...] | [ON | OFF]
```

*printspec*には、テキストの配置および書式設定に使用される次の句のうちの1つ以上を指定します。

BOLD

CE[NTER]

COL *n*

FORMAT *text*

LE[FT]

R[IGHT]

S[KIP] [*n*]

TAB *n*

各レポート・ページの下部に、指定したタイトルを配置して書式設定したり、現行のBTITLE定義を表示します。

現行のBTITLE定義を表示するには、句を指定せずに、BTITLEを入力します。旧形式のBTITLEの詳細は、「[BTI\[TLE\] text\(旧形式は廃止\)](#)」を参照してください。

項

BTITLEコマンド構文の項および句については、「[TTITLE](#)」コマンドを参照してください。

使用方法

最初に現れる*text*の前に*printspec*句を入力しない場合、BTITLEはテキストを左揃えにします。有効な*printspec*句(LEFT、SKIP、COLなど)がコマンド名のすぐ後に続いている場合、SQL*Plusは新形式でBTITLEを解釈します。

SQL*Plusの置換文字列(& variables)は、BTITLEの実行前に展開されます。生成された文字列はBTITLEテキストとして格納されます。後続の各結果ページの生成中に予期しない結果が発生して、展開された変数の値自身が変数として解釈される場合があります。

BTITLEコマンドでのこの二重置換を回避するには、各結果ページで置換される変数に&接頭辞を使用しないでください。置換変数を使用してBTITLEに未変更のテキストを挿入する場合は、1回のみ置換されるようにテキストを引用符で囲みます。

例

左側にはCORPORATE PLANNING DEPARTMENTという下部タイトル、右側には日付が示されるように設定するには、次のように入力します。

```
BTITLE LEFT 'CORPORATE PLANNING DEPARTMENT' -  
RIGHT '1 JAN 2001'
```

50桁目からCONFIDENTIALという下部タイトルが表示され、その後に6つの空白を置いて日付が続くように設定するには、次のように入力します。

```
BTITLE COL 50 'CONFIDENTIAL'  
TAB 6  
'1 JAN 2001'
```

12.11 CHANGE

構文

```
C[CHANGE] sepchar old [sepchar [new [sepchar]]]
```

バッファ内のカレント行で最初に一致した文字列を変更します。

項

sepchar

「/」や「!」など英数字以外の文字を指定します。*sepchar*には、*old*または*new*に使用されていない文字を使用します。

old

変更するテキストを指定します。CHANGEでは、*old*を検索する際に大文字と小文字の区別は無視されます。次に例を示します。

```
CHANGE /aq/aw
```

「aq」、「AQ」、「aQ」、「Aq」と最初に一致した文字列が検索され、それが「aw」に変更されます。SQL*Plusは、*new*テキストをユーザーが指定したとおりに挿入します。

*old*の前に「...」を付けた場合、それは*old*と最初に一致した文字列に達するまでのすべてのもの(最初に一致した文字列を含む)を意味します。*old*の後に「...」を付けた場合は、*old*と最初に一致した文字列とそこから行末までのすべてのものを意味します。途中に「...」を含めた場合は、*old*より前の部分から*old*から後の部分までのすべてのものが該当します。

new

*old*と置換するテキストを指定します。*new*と、オプションの2番目および3番目のsepcharsを省略した場合、CHANGEは、バッファのカレント行から*old*を削除します。

使用方法

CHANGEは、バッファのカレント行の上で、指定された既存の最初に一致した文字列を、指定された新しいテキストに変更します。LIST出力では、カレント行にはアスタリスク(*)によってマークが設定されています。

CHANGEを使用して、Oracle Databaseエラーの原因となったバッファ内の行を変更することもできます。SQL*Plusは、エラーがある行をバッファのカレント行に設定するので、それを変更できます。

1つの行全体を入力しなおすには、行番号を入力し、続いてその行の新しい内容を入力します。バッファ内の行数より大きい行番号を指定し、その後にテキストを続けた場合、SQL*Plusは、バッファの終わりに新しい行を追加して、そこにそのテキストを入れます。行番号として0(ゼロ)を指定し、その後にテキストを入力した場合、SQL*Plusは、バッファの先頭にその行を挿入します(その行が行1になります)。

例

3を入力し、バッファのカレント行に次のテキストが含まれていることを確認します。

```
3
```

```
3* WHERE JOB_ID IS IN ('CLERK', 'SA_MAN')
```

次のコマンドを入力します。

```
CHANGE /CLERK/SH_CLERK/
```


バッファ内のテキストは、次のように変更されます。

```
3* WHERE JOB_ID IS IN ('SH_CLERK', 'SA_MAN')
```

または、次のコマンドを入力します。

```
CHANGE /'CLERK',... /'SH_CLERK' /
```

元の行は、次のように変更されます。

```
3* WHERE JOB_ID IS IN ('SH_CLERK')
```

または、次のコマンドを入力します。

```
CHANGE /(...)/('SA_MAN')/
```

元の行は、次のように変更されます。

```
3* WHERE JOB_ID IS IN ('SA_MAN')
```

行番号を使用して、行全体の内容を置換できます。たとえば、次のように入力したとします。

```
3 WHERE JOB_ID IS IN ('SH_CLERK')
```

バッファの2番目の行が次のように置換されます。

```
WHERE JOB_ID IS IN ('SH_CLERK')
```

行番号に続けて文字列を入力すると、その行番号の後にどのようなテキストがあってもすべて置換されます。次に例を示します。

```
2 CHANGE/OLD/NEW/
```

バッファの2番目の行が次のように変更されます。

```
2* C/OLD/NEW/
```

12.12 CLEAR

構文

```
CL[EAR] option ...
```

*option*は、次のいずれかの句を表します。

```
BRE[AKS] BUFF[ER] COL[UMNS] COMP[UTES] SCR[EEN] SQL TIMI[NG]
```

指定したオプションの現行の値または設定をリセットまたは消去します。

項

```
BRE[AKS]
```

BREAKコマンドで設定したブレイク定義が取り消されます。

```
BUFF[ER]
```

バッファからテキストがクリアされます。複数のバッファを使用している場合以外は、CLEAR BUFFERの働きはCLEAR SQLと同じです。

このコマンドの廃止された形式の詳細は、「[SET BUF\[FER\] {bufferSQL}\(廃止\)](#)」を参照してください。

```
COL[UMNS]
```

すべての列について、COLUMNコマンドによって設定した列表示属性がデフォルト設定にリセットされます。1つの列のみについて表示属性をリセットするには、COLUMNコマンドのCLEAR句を使用します。CLEAR COLUMNSでは、該当する列のATTRIBUTEも消去されます。

```
COMP[UTES]
```

COMPUTEコマンドで設定したすべてのCOMPUTE定義が取り消されます。

```
SCR[EEN]
```

画面がクリアされます。

```
SQL
```

SQLバッファからテキストがクリアされます。複数のバッファを使用している場合以外は、CLEAR SQLの働きはCLEAR BUFFERと同じです。

このコマンドの廃止された形式の詳細は、「[SET BUF\[FER\] {bufferSQL}\(廃止\)](#)」を参照してください。

```
TIMI[NG]
```

TIMINGコマンドにより作成されたすべてのタイマーが削除されます。

例

ブレイクをクリアするには、次のように入力します。

```
CLEAR BREAKS
```

列定義をクリアするには、次のように入力します。

```
CLEAR COLUMNS
```

12.13 COLUMN

構文

```
COL[UMN] [{column | expr} [option ...]]
```

*option*は、次のいずれかの句を表します。

```
ALI[AS] alias CLE[AR] ENTMAP {ON | OFF} FOLD_A[FTER] FOLD_B[EFORE] FOR[MAT] format HEA[DING] text  
JUS[TIFY] {L[EFT] | C[ENTER] | R[IGHT]} LIKE {expr | alias} NEWL[INE] NEW_V[ALUE] variable NOPRI[NT] |  
PRI[NT] NUL[L] text OLD_V[ALUE] variable ON | OFF WRA[PPED] | WOR[D_WRAPPED] | TRU[NCATED]
```

特定の列について次のような表示属性を指定します。

- 列ヘッダーのテキスト
- 列ヘッダーの位置
- NUMBERデータの書式
- 列データの折返し

また、1つまたはすべての列の現行の表示属性も表示します。

指定した列または式についてのみ現行の表示属性を表示するには、*column*または*expr*のみを指定してCOLUMNを入力します(その他の句は使用しません)。現行のすべての列表示属性を表示するには、句を指定せずに、COLUMNを入力します。

項

```
{column | expr}
```

COLUMNコマンドで、SQL SELECT文の中のどのデータ項目を参照するかを指示します(通常は列の名前)。COLUMNコマンドの中で式を使用する場合は、SELECT文の中での順序と同じ順序で*expr*を入力する必要があります。たとえば、SELECT文の中の式が $a+b$ である場合、COLUMNコマンドの中でSELECT文の中の式を参照するときに、 $b+a$ または $(a+b)$ はいずれも使用できません。

異なる表から同じ名前の列を選択してある場合は、その列名を指定したCOLUMNコマンドは両方の列に適用されます。LAST_NAME列を対象とするCOLUMNコマンドは、このセッションで参照するLAST_NAMEという名前のすべての列に適用されます。COLUMNは、SELECTコマンドの表名接頭辞を無視します。また、名前が二重引用符で囲まれている場合を除き、空白も無視されます。

列を別々に書式設定するには、SELECTコマンド自体の中で各列にそれぞれ一意の別名を割り当て(COLUMNコマンドのALIAS句は使用しないでください)、各列の別名を指定してCOLUMNコマンドを入力します。

```
ALI[AS] alias
```

指定した別名が列に割り当てられます。この別名は、BREAKおよびCOMPUTE、他のCOLUMNコマンドの中でその列を参照するために使用できます。

```
CLE[AR]
```

列の表示属性がデフォルト値にリセットされます。

すべての列の属性をリセットするには、CLEAR COLUMNSコマンドを使用します。CLEAR COLUMNSでは、該当する列のATTRIBUTEも消去されます。

```
ENTMAP {ON | OFF}
```

HTML出力で選択された列に対して、エンティティのマッピングのONまたはOFFを指定します。この機能を使用すると、たとえば、同じレポートの別の列にエンティティをマッピングしていても、データの列にHTMLのハイパーリンクを含むことができます。HTMLのハイパーリンクを含む列に対してエンティティのマッピングをOFFにすると、HTMLのアンカー・タグのデリミタ、「<」、「>」、「"」および「&」が、レポートで正しく解釈されます。ONにすると、それぞれのエンティティである「<」、「>」、「"」および「&」に置き換えられ、WebブラウザでHTMLが正しく解釈されません。

列ヘッダーおよびCOMPUTEラベルのエンティティ、または列に表示される出力は、列のENTMAPの値によってマップされるかが決まります。

COLUMN ENTMAPのデフォルト設定は、MARKUP HTML ENTMAPオプションの現行の設定です。

MARKUP HTML ENTMAPオプションの詳細は、「SET [MARKUPオプション](#)」を参照してください。

FOLD_A[FTER]

列ヘッダーの後と列の各行の後に改行を挿入します。SQL*Plusは、SELECTリストの最後の列の後に余分な改行を挿入しません。FOLD_A[FTER]は、PREFORMATにONが設定されている場合を除き、SET MARKUP HTML ONモードでは動作しません。

FOLD_B[EFORE]

列ヘッダーの前と列の各行の前に改行を挿入します。SQL*Plusは、SELECTリストの最初の列の前に余分な改行を挿入しません。FOLD_A[FTER]は、PREFORMATにONが設定されている場合を除き、SET MARKUP HTML ONモードでは動作しません。

FOR[MAT] *format*

列の表示書式を指定します。指定する書式は、A10または\$9,999などのテキスト定数である必要があります。

文字列

CHAR、NCHAR、VARCHAR2(VARCHAR)およびNVARCHAR2(NCHAR VARYING)列のデフォルトの幅は、データベース内の列の幅です。SQL*Plusは、これらのデータ型を左揃えで書式設定します。値が列幅に収まらない場合、SQL*Plusは、SET WRAPの設定に応じて、文字列を折り返すか切り捨てます。

LONG、BLOB、BFILE、CLOB、NCLOBまたはXMLType列のデフォルトの幅は、SET LONGCHUNKSIZEまたはSET LONGのいずれか小さい方の値です。

データ型の幅を、FORMAT *An*を使用して*n*に変更します。(Aはアルファベットを表します。)列ヘッダーより短い幅を指定した場合は、SQL*Plusではヘッダーが切り捨てられます。

SQL*Plusは、2000バイトの後のXMLType列を切り捨てるか折り返します。これを回避するには、XMLType列にCOLUMNの書式を明示的に設定する必要があります。COLUMNの書式は、1行当たり最大60000にすることができます。

DATE列

SQL*Plusでは、書式が設定されていないDATE列のデフォルトの幅と書式は、NLS_DATE_FORMATパラメータから導出されます。NLS_DATE_FORMATの設定は、NLS_TERRITORYパラメータによって決定されます。たとえば、アメリカ合衆国の場合、NLS_TERRITORYのデフォルト書式はDD-Mon-RR、デフォルトの幅はA9です。NLSパラメータは、データベースのパラメータ・ファイル内で設定される場合と、環境変数やそれに相当するプラットフォーム固有メカニズムで設定される場合があります。また、NLSパラメータは、ALTER SESSIONコマンドによってセッションごとに指定することもできます。DATE書式とNLSパラメータの詳細は、『[Oracle Database SQL言語リファレンス](#)』を参照してください。

SQL SELECT文の中でSQL関数TO_CHARを使用して、任意のDATE列の書式を変更できます。明示的なCOLUMN FORMATコマンドを使用して、列幅を調整することもできます。

Oracle Databaseでは、TO_CHARのようなSQL関数を使用するときに、非常に幅の広い列が自動的に使用できるようになります。デフォルトの列幅は、SQL*Plusおよびデータベースで使用されている文字セットによって異なります。複数の文字セットを使用している場合にスクリプトの移植性を最大限にするには、Oracle Databaseで、選択した列ごとにCOLUMN FORMATを使用することをお勧めします。

DATE列の幅を*n*に変更するには、FORMAT *An*を指定したCOLUMNコマンドを使用します。列ヘッダーより短い幅を指定した場合は、ヘッダーが切り捨てられます。

NUMBER列

数値列では、SET NUMFORMAT設定(SET NUMWIDTH設定より優先)よりCOLUMN FORMAT設定が優先されます。

[\[SET NUMF\[ORMAT\] format\]](#)および[\[SET NUM\[WIDTH\] {10 | n}\]](#)を参照してください。

NUMBER列の幅を変更するには、FORMATの後に、[表12-1](#)に指定された要素を使用します。

表12-1 数値書式

要素	例	説明
,	9,999	指定した位置にカンマが表示されます。
.	99.99	数値の整数部と小数部を区切るピリオド(小数点)が表示されます。
\$	\$9999	先行ドル記号が表示されます。
0	0999 9990	先行ゼロが表示されます。後続ゼロが表示されます。
9	9999	9 の数で指定した桁数の値が表示されます。値には、正の値の場合は先行空白、負の値の場合は先頭に先行マイナス記号が表示されます。先行 0(ゼロ)には空白が表示されます。0(ゼロ)の値には 0(ゼロ)が表示されます。
B	B9999	書式モデル内の 0(ゼロ)にかかわらず、整数部が 0(ゼロ)の場合、固定小数点数の整数部に空白が表示されます。
C	C999	指定した位置に ISO 通貨記号が表示されます。
D	99D99	数値の整数部と小数部を区切る小数点文字が表示されます。
EEEE	9.999EEEE	値が科学表記法で表示されます(書式には確実に 4 つの「E」を含める必要があります)。
G	9G999	数値の整数部の指定した位置に桁グループ・セパレータが表示されます。
L	L999	指定した位置に各国通貨記号が表示されます。
MI	9999MI	負の値の後に後続マイナス記号が表示されます。正の値の後に後続空白が表示されます。

要素	例	説明
PR	9999PR	負の値は<山カッコ>で囲まれて表示されます。正の値の場合は、先行空白および後続空白が表示されます。
RN rn	RN rn	大文字のローマ数字が表示されます。小文字のローマ数字が表示されます。値は 1 から 3999 の整数となります。
S	S9999 9999S	先行マイナス記号またはプラス記号が表示されます。後続マイナス記号またはプラス記号が表示されます。
TM	TM	小数点文字の最小の数が表示されます。デフォルトは TM9 です。出力に使用される固定表記法の場合は最大 64 文字、科学表記法の場合は 65 文字以上です。TM の前に他の要素を指定することはできません。TM の後には単一の 9 または E のみを指定できます。
U	U9999	指定した位置に、第 2 通貨記号が表示されます。
V	999V99	10 n を掛けた値が表示されます。 n は、「V」の後の「9」の数です。
X	XXXX xxxx	指定した桁数の四捨五入された値が 16 進数値で表示されます。

MIおよびPR書式要素が使用できるのは、数値書式モデルの最後の桁のみです。S書式要素が使用できるのは、最初または最後の桁のみです。

数値書式モデルにMI、SまたはPR書式が含まれていない場合、負の戻り値には自動的に先行負記号が含まれ、正の値には自動的に先行空白が含まれます。

1つの数値書式モデルに挿入できる小数点文字(D)またはピリオド(.)は1つのみです。ただし、桁グループ・セパレータ(G)またはカンマ(,)は、2つ以上挿入できます。数値書式モデル内で、小数点文字またはピリオドより右側に、グループ・セパレータまたはカンマを使用することはできません。

SQL*Plusは、NUMBERデータを右揃えで書式設定します。NUMBER列の幅は、ヘッダーの幅か、FORMATの幅に符号用の1つの空白を加えた幅のどちらか大きい方に等しくなります。明示的にCOLUMN FORMATまたはSET NUMFORMATを使用しない場合、常に、列の幅はSET NUMWIDTHの値以上になります。

SQL*Plusは、NUMBERデータを書式設定またはフィールド幅にあわせて四捨五入する場合があります。

値が列に収まらない場合、SQL*Plusは数字のかわりにシャープ記号(#)を表示します。

正の値が極端に大きく、数値の四捨五入の際に数値オーバーフローが発生する場合は、値のかわりに無限大記号(~)が表示されます。同様に、負の値が極端に小さく、数値の四捨五入の際に数値オーバーフローが発生する場合は、値のかわりに負の無限大記号(-~)が表示されます。

HEA[DING] *text*

列ヘッダーを定義します。HEADING句を使用しない場合の、デフォルトの列ヘッダーは、*column*または*expr*です。*text*に空白または句読点文字が含まれている場合は、一重または二重引用符でそのテキストを囲む必要があります。HEADSEP文字(デフォルトでは「|」)が発生するたびに、新しい行が始まります。

次に例を示します。

COLUMN LAST_NAME HEADING 'Employee |Name'

この場合は、2行の列ヘッダーが生成されます。

HEADSEP文字の変更については、「[SET HEADS\[EP\] { | c | ON | OFF }](#)」を参照してください。

JUS[TIFY] [L[EFT] | C[ENTER] | R[IGHT]]

ヘッダーを整列させます。JUSTIFY句を使用しない場合、NUMBER列のヘッダーのデフォルトはRIGHTで、その他の列型のヘッダーのデフォルトはLEFTです。

LIKE {*expr* | *alias*}

他の列または式(他のCOLUMNコマンドですでに属性を定義してあるもの)の表示属性がコピーされます。LIKEによってコピーされるのは、現行のCOLUMNコマンド内の他の句で定義されていない属性のみです。

NEWL[INE]

列の値を表示する前に新しい行を開始します。NEWLINEは、FOLD_BEFOREと同じ働きをします。NEWL[INE]は、PREFORMATがSET ONでないかぎり、SET MARKUP HTML ONモードでは動作しません。

NEW_V[ALUE] *variable*

列値を保持する変数を指定します。この変数は、TTITLEコマンドの中で参照できます。NEW_VALUEは、列値または日付を上部タイトルに表示するために使用します。この列は、SKIP PAGEアクションを伴うBREAKコマンドに挿入しておく必要があります。変数名にシャープ記号(#)は挿入できません。

NEW_VALUEは、ページごとに新しいマスター・レコードが使用されるマスター/ディテール・レポートに使用すると便利です。マスター/ディテール・レポートを作成するには、ORDER BY句にもこの列を挿入しておく必要があります。この項の終わりに記載されている例を参照してください。

NEW_V[ALUE]で指定された変数は、TTITLEを実行する前に展開されます。生成された文字列はTTITLEテキストとして格納されます。後続の各レポート・ページの生成中に予期しない結果が発生して、展開された変数の値自身が変数として解釈される場合があります。

TTITLEコマンドでのこの二重置換を回避するには、各レポート・ページで置換されるNEW_V[ALUE]変数に&接頭辞を使用しないでください。置換変数を使用してTTITLEに未変更のテキストを挿入する場合は、1回のみ置換されるようにテキストを引用符で囲みます。

下部タイトルに列値を表示する方法については、次のOLD_V[ALUE]変数を参照してください。タイトル内の変数の参照については、「[TTITLE](#)」コマンドを参照してください。書式設定および有効な書式モデルについては、前述のFOR[MAT]書式を参照してください。

NOPRI[NT] | PRI[NT]

列(列ヘッダーおよび選択したすべての値)の印刷を制御します。NOPRINTを指定すると、列の画面表示および印刷が行われません。PRINTを指定すると、列が印刷されます。

NUL[L] *text*

SQL*Plusによって特定の列にNULL値として表示されるテキストを制御します。デフォルトは空白です。SET NULLは、すべての列のすべてのNULL値のかわりに表示されるテキストを制御します。ただし、COLUMNコマンドのNULL句に指定した列には適用されません。NULL値を選択すると、変数の型は常にCHARになるので、SET NULLテキストをその変数に格納できます。

OLD_V[ALUE] *variable*

列値を保持する変数を指定します。この変数は、BTITLEコマンドの中で参照できます。OLD_VALUEは、下部タイトルに列値を表示するために使用します。この列は、SKIP PAGEアクションを伴うBREAKコマンドに挿入しておく必要があります。

OLD_VALUEは、それぞれのページごとに新しいマスター・レコードが使用されるマスター/ディテール・レポートに使用すると便利です。マスター/ディテール・レポートを作成するには、ORDER BY句にもこの列を挿入しておく必要があります。

OLD_V[ALUE]で指定された変数は、BTITLEを実行する前に展開されます。生成された文字列はBTITLEテキストとして格納されます。後続の各レポート・ページの生成中に予期しない結果が発生して、展開された変数の値自身が変数として解釈される場合があります。

BTITLEコマンドでのこの二重置換を回避するには、各レポート・ページで置換されるOLD_V[ALUE]変数に&接頭辞を使用しないでください。置換変数を使用してBTITLEに未変更のテキストを挿入する場合は、1回のみ置換されるようにテキストを引用符で囲みます。

上部タイトルに列値を表示する方法については、NEW_V[ALUE]変数を参照してください。タイトル内の変数の参照については、「[TTITLE](#)」コマンドを参照してください。

ON | OFF

列の表示特性の状態を制御します。OFFを指定すると、属性の定義に影響を与えることなく列の属性が非表示にされます。ONを指定すると、属性が表示されます。

WRA[PPED] | WOR[D_WRAPPED] | TRU[NCATED]

長すぎて列に収まらないデータ型またはDATE列をSQL*Plusでどのように取り扱うかを指定します。WRAPPEDを指定すると、列の境界内で文字列が折り返され、必要に応じて新しい行が開始されます。WORD_WRAPが使用可能な場合、SQL*Plusは、組込みの改行文字を含むすべての先行空白(リターン、改行文字、タブ、空白など)をスキップして、各行を左揃えにします。行の境界上にない組込み空白はスキップされません。TRUNCATEDを指定すると、最初の表示行の終わりの文字列が切り捨てられます。

NCLOB、BLOB、BFILEまたはマルチバイトCLOBの各列をWORD_WRAPPEDオプションで書式設定することはできません。NCLOB、BLOB、BFILEまたはマルチバイトCLOBの各列をCOLUMN WORD_WRAPPEDで書式設定した場合、列データはCOLUMN WRAPPEDが適用されたかのように動作します。

使用方法

入力したCOLUMNコマンドによって、複数のSQL SELECTコマンドについて特定の列の表示属性を制御できます。

1つ以上の列について、COLUMNコマンドをいくつでも入力できます。列をOFFに切り替えるか、CLEAR COLUMNコマンドを使用しないかぎり、各列について設定されているすべての列属性がセッションの終わりまで有効になります。

同じ列について複数のCOLUMNコマンドを入力すると、SQL*Plusは、それらのコマンドのすべての句をまとめて適用します。複数のCOLUMNコマンドで、同じ列に対して同じ句が適用される場合は、最後に入力した句によって出力が制御されます。

例

LAST_NAMEの幅を20文字にし、EMPLOYEE NAMEを列ヘッダーとして2行に表示するには、次のように入力します。

```
COLUMN LAST_NAME FORMAT A20 HEADING 'EMPLOYEE|NAME'
```

SALARY列の書式を次のように入力すると、100万ドルの位まで表示します。セントの単位で四捨五入して、3桁区切りにカンマを使用します。値が0(ゼロ)の場合は、\$0.00と表示します。

```
COLUMN SALARY FORMAT $9,999,990.99
```

長い式を含む列に別名NETを割り当て、結果をドル書式で表示し、NULL値には<NULL> を表示するには、次のように入

力します。

```
COLUMN SALARY+COMMISSION_PCT+BONUS-EXPENSES-INS-TAX ALIAS NET  
COLUMN NET FORMAT $9,999,999.99 NULL '<NULL>'
```

この例では、この列指定を2つのコマンドに分けています。最初のコマンドは別名NETを定義し、2番目のコマンドはNETを使用して書式を定義しています。

また、最初のコマンドでは、SELECT文で入力したとおり、正確に式を入力する必要があります。そうしないと、SQL*Plusは、COLUMNコマンドを適正な列に対応付けることができません。

REMARKSという名前の列の中の長い値を折り返すには、次のように入力します。

```
COLUMN REMARKS FORMAT A20 WRAP
```

CUSTOMER	DATE	QUANTITY	REMARKS
123	25-AUG-2001	144	This order must be shipped by air freight to ORD

WRAPをWORD_WRAPで置換した場合、REMARKSは次のようになります。

CUSTOMER	DATE	QUANTITY	REMARKS
123	25-AUG-2001	144	This order must be shipped by air freight to ORD

TRUNCATEを指定した場合、REMARKSは次のようになります。

CUSTOMER	DATE	QUANTITY	REMARKS
123	25-AUG-2001	144	This order must be s

現在の日付および各職種の名前を上部タイトルに出力するには、次のように入力します。EMP_DETAILS_VIEWのかわりに、HRスキーマのEMPLOYEES表を使用します。

日付変数の作成方法の詳細は、[タイトル内に現在の日付を表示する方法について](#)を参照してください。

2ページにわたるレポートは次のようになり、「Job Report」が現行サイズの行の中央に配置されています。

```
COLUMN JOB_ID NOPRINT NEW_VALUE JOBVAR  
COLUMN TODAY NOPRINT NEW_VALUE DATEVAR  
BREAK ON JOB_ID SKIP PAGE ON TODAY  
TTITLE CENTER 'Job Report' RIGHT DATEVAR SKIP 2 -  
LEFT 'Job: ' JOBVAR SKIP 2  
SELECT TO_CHAR(SYSDATE, 'MM/DD/YYYY') TODAY,  
LAST_NAME, JOB_ID, MANAGER_ID, HIRE_DATE, SALARY, DEPARTMENT_ID  
FROM EMPLOYEES WHERE JOB_ID IN ('MK_MAN', 'SA_MAN')  
ORDER BY JOB_ID, LAST_NAME;
```

DATE列のデフォルト書式を「YYYY-MM-DD」に変更するには、次のように入力します。

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD';
```

Session altered.

変更内容を表示するには、次のようなSELECT文を入力します。

```
SELECT HIRE_DATE FROM EMPLOYEES WHERE EMPLOYEE_ID = 206;
```

Job: SA_MAN

HIRE_DATE

1994-06-07

ALTER SESSIONコマンドの詳細は、[ALTER SESSION](#)を参照してください。

12.14 COMPUTE

構文

```
COMP[UTE] [function [LAB[EL] text] ...] OF {expr | column | alias} ... ON {expr | column | alias | REPORT | ROW} ...]
```

BREAKコマンドと組み合わせて、選択した行のサブセットについて、各種の標準計算方法を使用してサマリー行を計算して出力します。すべてのCOMPUTE定義もリスト表示します。サマリーの作成方法の詳細は、[間隔の設定およびサマリー行によるレポートの明確化について](#)を参照してください。

項

function ...

[表12-2](#)に示す関数の1つを指定します。複数の関数を指定する場合は、関数と関数の間を空白で区切ります。

COMPUTEコマンドの関数は、COMPUTEコマンド内の順序にかかわらず、常にAVG、COUNT、MINIMUM、MAXIMUM、NUMBER、SUM、STD、VARIANCEの順序で実行されます。

表12-2 COMPUTE関数

関数	計算	適用されるデータ型
AVG	NULL 以外の値の平均	NUMBER
COUNT	NULL 以外の値の数	すべての型
MINIMUM	最小値	NUMBER、CHAR、NCHAR、VARCHAR2(VARCHAR)、NVARCHAR2(NCHAR VARYING)
MAXIMUM	最大値	NUMBER、CHAR、NCHAR、VARCHAR2(VARCHAR)、NVARCHAR2(NCHAR VARYING)
NUMBER	行の数	すべての型
SUM	NULL 以外の値の合計	NUMBER
STD	NULL 以外の値の標準偏差	NUMBER
VARIANCE	NULL 以外の値の平方偏差	NUMBER

LAB[EL] *text*

計算値に対応して出力されるラベルを定義します。LABEL句を使用しない場合のデフォルトの*text*は、省略されていないファンクション・キーワードです。空白または句読点を含む場合、*text*を一重引用符で囲む必要があります。ラベルは左揃えで出力さ

れ、列幅または行サイズのいずれか小さい方に合せて切り捨てられます。ラベルの最大長は500文字です。

計算された値のラベルは、指定したブレイク列に表示されます。ラベルを非表示にするには、ブレイク列に対して、COLUMNコマンドのNOPRINTオプションを使用します。

COMPUTEコマンドの中で同じ関数を繰り返し使用すると、SQL*Plusはアラートを発行し、そのうちの最初の関数のみを使用します。

ON REPORTおよびON ROW計算では、SELECT構文のリストに指定された最初の列にラベルが表示されます。SELECT文の中で最初にNOPRINT列を使用することによって、ラベルを非表示にできます。SELECT文ON REPORTまたはON ROWの最初の列の関数を計算する場合は、計算済の値が最初の列に表示され、ラベルは表示されません。ラベルを見るには、SELECT構文のリストで最初にダミー列を選択します。

```
OF {expr | column | alias} ...
```

OF句では、式または関数の参照を二重引用符で囲むことによって、SELECT文の中の式または関数の参照を指定できます。列の名前および別名に引用符は必要ありません。

```
ON {expr | column | alias | REPORT | ROW} ...
```

複数のCOMPUTEコマンドのON句で同じ列が参照されている場合は、最後のCOMPUTEコマンドのみが適用されます。

ON句でSQL SELECT式または関数の参照を指定するには、その式または関数の参照を引用符で囲みます。列の名前および別名に引用符は必要ありません。

すべてのCOMPUTE定義を表示するには、句を指定せずにCOMPUTEを入力します。

使用方法

計算が実行されるためには、次の条件がすべて満たされている必要があります。

- OF句で参照している1つ以上の式、列または列の別名が、SELECT文の中でも指定されている。
- ON句で参照している式、列または列の別名が、SELECT文の中および最新のBREAKコマンドの中にも指定されている。
- ON句でROWまたはREPORTを参照する場合は、最新のBREAKコマンドの中でもROWまたはREPORTを参照する。

すべてのCOMPUTE定義を取り消すには、CLEAR COMPUTESコマンドを使用します。

COMPUTEが実行されている列に対してNOPRINTオプションを使用すると、COMPUTEの実行結果も非表示になります。

例

AC_MGR(アカウント・マネージャ)およびSA_MAN(セールスマン)の各職種別に給与の小計を計算し、TOTALという計算ラベルを付けるには、次のように入力します。

```
BREAK ON JOB_ID SKIP 1;
COMPUTE SUM LABEL 'TOTAL' OF SALARY ON JOB_ID;
SELECT JOB_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE JOB_ID IN ('AC_MGR', 'SA_MAN')
ORDER BY JOB_ID, SALARY;
```

JOB_ID	LAST_NAME	SALARY
AC_MGR	Higgins	12000

TOTAL		12000

SA_MAN	Zlotkey	10500
	Cambrault	11000
	Errazuriz	12000
	Partners	13500
	Russell	14000

TOTAL		61000
-------	--	-------

6 rows selected.

レポート上の12,000を超える給与の合計を計算するには、次のように入力します。

```

COMPUTE SUM OF SALARY ON REPORT
BREAK ON REPORT
COLUMN DUMMY HEADING ''
SELECT ' ' DUMMY, SALARY, EMPLOYEE_ID
FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000
ORDER BY SALARY;

```

SALARY	EMPLOYEE_ID
13000	201
13500	146
14000	145
17000	101
17000	102
24000	100
sum	98500

6 rows selected.

管理部門および経理部門の中での給与の平均と最高額を計算するには、次のように入力します。

```

BREAK ON DEPARTMENT_NAME SKIP 1
COMPUTE AVG LABEL 'Dept Average' -
      MAX LABEL 'Dept Maximum' -
      OF SALARY ON DEPARTMENT_NAME
SELECT DEPARTMENT_NAME, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE DEPARTMENT_NAME IN ('Executive', 'Accounting')
ORDER BY DEPARTMENT_NAME;

```

DEPARTMENT_NAME	LAST_NAME	SALARY
Accounting	Higgins	12000
	Gietz	8300

Dept Average		10150
Dept Maximum		12000
Executive	King	24000
	Kochhar	17000
	De Haan	17000

Dept Average		19333.3333
Dept Maximum		24000

部門20以下の給与を合計し、計算ラベルを出力しないようにするには、次のようにします。

```

COLUMN DUMMY NOPRINT
COMPUTE SUM OF SALARY ON DUMMY
BREAK ON DUMMY SKIP 1
SELECT DEPARTMENT_ID DUMMY, DEPARTMENT_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE DEPARTMENT_ID <= 20
ORDER BY DEPARTMENT_ID;

```

DEPARTMENT_ID	LAST_NAME	SALARY
10	Whalen	4400

		4400
20	Hartstein	13000
20	Fay	6000

		19000

計算ラベルを出力せずに、レポートの終わりに給与合計額が表示されるようにするには、次のように入力します。

```

COLUMN DUMMY NOPRINT
COMPUTE SUM OF SALARY ON DUMMY
BREAK ON DUMMY
SELECT NULL DUMMY, DEPARTMENT_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE DEPARTMENT_ID <= 30
ORDER BY DEPARTMENT_ID;

```

DEPARTMENT_ID	LAST_NAME	SALARY
10	Whalen	4400
20	Hartstein	13000
20	Fay	6000
30	Raphaely	11000
30	Khoo	3100
30	Baida	2900
30	Tobias	2800
30	Himuro	2600
30	Colmenares	2500

		48300

9 rows selected.

12.15 CONNECT

構文

```
CONN[ECT] [{ /logon | / | proxy} [AS {SYSASM | SYSBACKUP | SYSDBA | SYSDG | SYSOPER | SYSRAC | SYSKM}]  
[edition=va/ue]]
```

*logon*の構文は、次のとおりです。

```
username[/password] [@connect_identifier]
```

*proxy*の構文は、次のとおりです。

```
proxyuser[username] [/password] [@connect_identifier]
```

ノート:



proxy の構文で *username* を囲んでいる大カッコは、オプションの項を示しているのではなく、構文に必要です。たとえば、プロキシ・ユーザー *hr* を介してパスワード *welcome1* で *scott* に接続するには、次のように入力します。

```
CONNECT hr [scott] /welcome1
```

指定したユーザー名でOracle Databaseに接続します。CONNECTコマンドを実行すると、サイト・プロファイルglogin.sqlおよびユーザー・プロファイルlogin.sqlが実行されます。

CONNECTコマンドでは、初期接続に失敗した場合にユーザー名またはパスワードを求めるプロンプトは表示されません。

警告:



パスワードをプレーン・テキストで指定すると、セキュリティ上の危険があります。パスワードを省略し、パスワードの入力を求めるプロンプトが表示された場合にのみ入力することで、この危険を回避できます。

エンタープライズ・ユーザー・プロキシを使用してデータベースに接続するには、最初にプロキシを設定する必要があります。エンタープライズ・ユーザー・プロキシの設定については、[『Oracle Database Enterprise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。

項

```
username[/password]
```

Oracle Databaseへの接続に使用するユーザー名およびパスワードを指定します。*username*および*password*を省略すると、SQL*Plusは入力を求めるプロンプトを表示します。*username*の入力を求めるプロンプトが表示された場合にスラッシュ (/)を入力するか、[Return]を押すか、または「実行」をクリックすると、SQL*Plusはデフォルトのログオンを使用してユーザーをログインさせます。詳細は、「[/\(スラッシュ\)](#)」を参照してください。

*password*のみを省略すると、SQL*Plusでは、*password*の入力を求めるプロンプトが表示されます。プロンプトを表示する場合、SQL*Plusが端末画面に*password*を表示することはありません。

SQL*Plusでのパスワードの変更方法については、[「PASSWORD」](#)コマンドを参照してください。

```
connect_identifier
```

Oracle Net接続識別子を指定します。正確な構文はOracle Netの構成によって異なります。詳細は、Oracle Netのマニュアルを参照するか、またはデータベース管理者に問い合わせてください。SQL*Plusは、サービス名についてはプロンプトを表示せず、接続識別子の指定がない場合は、デフォルトのデータベースを使用します。

connect_identifierは、プラグブル・データベース(PDB)への接続にも使用されます。[『Oracle Database管理者ガイド』](#)を参照してください。

```
edition=va/ue
```

Oracleセッション・エディションの値です。1つのエディションによって、データベース内に1つのオブジェクトの2つ以上のバージョンが存在できます。これにより、変更したオブジェクトをデータベースにロードし、コンパイルして稼働時間中に実行するためのステージング領域が提供されます。このことは、特に、アプリケーションへのパッチ適用に関連する停止時間を短縮する場合に有用です。eedition=valueは、ORA_EDITION環境変数で指定されたエディション値よりも優先されます。詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。

```
/ (slash)
```

オペレーティング・システム認証を使用するデフォルト・ログオンを表します。デフォルト・ログオンを使用する場合、connect_identifierは指定できません。デフォルト・ログオンでは通常、SQL*Plusはユーザー名OPS\$nameを使用してユーザーをログインさせようとします。nameは、ご使用のオペレーティング・システムのユーザー名です。オペレーティング・システム認証の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。

SQL*Plusコマンドラインでは、アプリケーションでデータベースへの接続にパスワード資格証明が使用されます。このコマンドラインでは、資格証明をクライアント・サイドのOracle Walletに格納できます。Oracle Walletを使用するようにクライアントを構成すると、アプリケーションで、次の構文を使用してパスワード認証を使用するデータベースに接続できます。

```
CONNECT /@database_alias
```

セキュアな外部パスワード・ストアを使用するようにクライアントを構成する方法およびそのパスワード・ストア内の資格証明を管理する方法については、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

```
AS {SYSASM | SYSBACKUP | SYSDBA | SYSDG | SYSOPER | SYSRAC | SYSKM}
```

AS句を使用すると、システム権限が付与されているユーザーによる特権付き接続が可能です。また、デフォルト・ログオンである/(スラッシュ)でもこれらの特権付きの接続を使用できます。

システム権限については、[『Oracle Database管理者ガイド』](#)を参照してください。

使用方法

CONNECTは、現行のトランザクションをデータベースに対してコミットし、現行のユーザー名をOracle Databaseから切断し、指定されたユーザー名で再接続します。

データベースへの接続の際にアカウントの有効期限が切れたユーザーとしてログオンまたは接続した場合、SQL*Plusは、接続の前にパスワードを変更するように求めるプロンプトを表示します。

データベースへの接続の際にアカウントがロックされている場合は、メッセージが表示され、DBAがロックを解除するまで、そのアカウントへの(そのユーザーとしての)接続は許可されません。

アカウント管理の詳細は、[『Oracle Database SQL言語リファレンス』](#)の「CREATE USER」、「ALTER USER」および「CREATE PROFILE」コマンドを参照してください。

例

Oracle Netを介し、ユーザー名HRを使用して、FLEETDBというOracle Net別名を持つデータベースに接続するには、次のように入力します。

```
CONNECT HR@FLEETDB
```

パスワード・ファイルの設定の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。

現行ノードのインスタンスにユーザーHRの権限で接続するには、次のように入力します。

```
CONNECT HR AS SYSDBA
```

現行ノードのインスタンスにデフォルト・ユーザーの権限で接続するには、次のように入力します。

```
CONNECT / AS SYSDBA
```

簡単な接続またはネット・サービス名を使用して、CONNECTコマンドによりCDBまたは非CDBに接続できます。

次の文では、**hrapp**サービスを使用して**hr**ユーザーに接続します。**hrapp**サービスには、**hrpdb** PDBのPDBプロパティがあります。この例では、クライアントが**hrapp**サービスの**ネット・サービス名**を持つように構成されているものとします。

```
CONNECT hr@hrapp
```

12.16 COPY

COPYコマンドでは、Oracle8i以上に導入されたデータ型または機能処理するための機能拡張は行われていません。COPYコマンドは、今後のリリースで非推奨となる予定です。

COPYコマンドの詳細および構文については、「[SQL*PlusのCOPYコマンド](#)」を参照してください。

12.17 DEFINE

構文

```
DEF [INE] [variable] | [variable = text]
```

ユーザー変数または事前定義変数を指定し、その変数にCHAR値を割り当てたり、1つまたはすべての変数の値および変数型を表示します。

項

variable

どのユーザー変数または事前定義変数の値を割り当てるか(あるいは表示するか)を指定します。

text

*variable*に割り当てるCHAR値を指定します。*text*に句読点または空白が含まれている場合は、テキストを一重引用符で囲みます。

```
variable = text
```

置換変数を定義(命名)し、それにCHAR値を割り当てます。

*variable*の値と型を表示するには、DEFINEに続けて*variable*を入力します。すべての置換変数の値と型を表示するには、句を指定せずに、DEFINEを入力します。

使用方法

事前定義変数は、次の操作を行うまでそれぞれの値を保持します。

- その変数を参照する新しいDEFINEコマンドの入力
- その変数を参照するUNDEFINEコマンドの入力
- その変数を参照するACCEPTコマンドの入力
- COLUMNコマンドのNEW_VALUE句またはOLD_VALUE句の中でその変数が参照され、次にSQL SELECTコマンドでその列が参照される
- EXIT SQL*Plus

ストアド・クエリーまたはスクリプトを実行するたびに、SQL*Plusは、*variable*を参照するそれぞれの置換変数(&*variable*または&&*variable*の書式)を、*variable*の値で置換します。SQL*Plusは、ユーザーが*variable*をUNDEFINEするまで、そのセッションでの*variable*値の入力を求めるプロンプトを表示しません。

事前定義変数の値を複数の行に展開した場合(SQL*Plusのコマンド継続文字を使用)、SQL*Plusは各継続文字および改行を空白で置換します。たとえば、次のように入力したとします。

```
DEFINE TEXT = ' ONE-  
TWO-  
THREE'
```

SQL*Plusはこれを次のように解釈します。

```
DEFINE TEXT = ' ONE TWO THREE'
```

予期しない結果を回避するために、変数に渡される値と同じ名前が付いている変数の定義を使用しないでください。定義した

変数に指定した値が変数名と一致する場合、指定された値のかわりに、一致する変数の内容が使用されます。

SQL*Plus起動時に事前定義される変数があります。DEFINEと入力すると、それらの定義が表示されます。

例

値MANAGERを変数POSに割り当てるには、次のように入力します。

```
DEFINE POS = MANAGER
```

&POSへの参照を含むコマンドを実行すると、SQL*Plusは、&POSを値MANAGERで置換し、POS値の入力を求めるプロンプトは表示しません。

CHAR値20を変数DEPARTMENT_IDに割り当てるには、次のように入力します。

```
DEFINE DEPARTMENT_ID = 20
```

20という数値を入力しても、SQL*Plusは、2および0という2つの文字で構成されるCHAR値をDEPARTMENT_IDに割り当てます。

DEPARTMENT_IDの定義を表示するには、次のように入力します。

```
DEFINE DEPARTMENT_ID
```

```
DEFINE DEPARTMENT_ID = "20" (CHAR)
```

この結果は、DEPARTMENT_IDの値が20であることを示しています。

12.17.1 事前定義変数

次の9つの変数がSQL*Plusのインストール中に定義されます。これらの変数とユーザー定義変数の違いは、事前定義値を持つ点のみです。

表12-3 SQL*Plusのインストール時に事前定義される変数

変数名	次を含む
__CONNECT_IDENTIFIER	可能な宛先への接続に使用される接続識別子。
__DATE	現在の日付、またはユーザー定義の固定文字列。
__EDITOR	EDIT コマンドで使用されるエディタを指定します。
__O_VERSION	インストールされた Oracle Database の現行のバージョン。
__O_RELEASE	インストールされた Oracle Database の完全なリリース番号。
__PRIVILEGE	現在の接続での権限レベル。
__SQLPLUS_RELEASE	インストールされた SQL*Plus コンポーネントの完全なリリース番号。

変数名	次を含む
_USER	接続に使用されるユーザー名。
_SQL_ID	実行された SQL 文の sql_id。
_CONNECT_IDENTIFIER	

接続識別子の INSTANCE_NAME、SERVICE_NAME または ORACLE_SID が含まれます。接続の際にユーザーが接続識別子を指定しない場合、_CONNECT_IDENTIFIER には ORACLE_SID が含まれます。

_DATE

動変数としての現在の日付、または固定文字列が含まれます。現行の日付はデフォルトで、NLS_DATE_FORMAT の値を使用して書式設定されます。

_DATE は標準の置換変数として使用できるため、ユーザーはこれを TTITLE に含めることができます。_DATE が動的で、TTITLE で使用される場合は、すべての標準の変数のセマンティクスを持ちます。_DATE が値ではなくアンパサンドとともに使用される場合、TTITLE コマンドが実行される時間に設定されます。アンパサンドを接頭辞に使用しない場合は、ページごとに再度評価されます。TTITLE での _DATE 付きの詳細なレポート、または &_DATE への複数の参照を持つ詳細なレポートでは、変数の出現ごとに異なる時間が表示されます。

TTITLE で _DATE を使用するユーザーは、ほとんどが、&_DATE のようにアンパサンドを使用します。そのため、レポートの各ページは同一のタイムスタンプを持ちます。これは、現在の日付の書式に秒のコンポーネントが含まれる場合、特に正確です。

DEFINE (引数なし)、または &_DATE を使用する間接参照では、現在の日付が指定されます。

_DATE の値は UNDEFINED にすることも、または明示的に DEFINE _DATE を使用して固定文字列に設定することもできます。

デフォルトの動的な日付動作を再度有効にするには、次のように指定します。

```
DEFINE _DATE = "" (an empty string)
```

_DATE を使用すると、時刻の値をご使用の SQL*Plus プロンプトに含めることができます。

_EDITOR

EDIT コマンドで使用されるデフォルトのエディタを指定します。

Windows オペレーティング・システムへの SQL*Plus のインストールの場合は、インストール時にメモ帳に設定されます。UNIX オペレーティング・システムの場合は、UNIX 環境変数の値、EDITOR (存在する場合) に設定されます。存在しない場合は Ed に設定されます。

デフォルトのテキスト・エディタの名前を保持するには、DEFINE コマンドを使用して _EDITOR を再定義します。たとえば、EDIT で使用するエディタを vi として定義するには、次のコマンドを入力します。

```
DEFINE _EDITOR = vi
```

_O_VERSION

インストールされた Oracle Database の現行のバージョンが、次のような書式で含まれます。

Oracle Database 19c Enterprise Edition リリース 19.0.0.0.0 - Production

バージョン 19.1.0.0.0

_O_RELEASE

インストールされたOracle Databaseの完全なリリース番号が、次のような書式で含まれます。

1801000000

_PRIVILEGE

現在の接続での権限レベルを示す値が含まれます。次のいずれかの値です。

- AS SYSASM
- AS SYSBACKUP
- AS SYSDBA
- AS SYSDG
- AS SYSOPER
- AS SYSRAC
- 空の文字列(通常のユーザー接続または接続しない場合)

AS SYSASM、AS SYSBACKUP、AS SYSDBA、AS SYSDG、AS SYSOPERおよびAS SYSRACは、データベース管理者レベルの権限です。

関連項目:

AS SYSDBAおよびAS SYSOPER権限の詳細は、「[GRANT](#)」を参照してください。

_SQLPLUS_RELEASE

インストールされたSQL*Plusコンポーネントの完全なリリース番号が、次のような書式で含まれます。

1801000000

_USER

現在接続しているユーザーの名前が含まれます。

_SQL_ID

現在実行されているSQL文またはPL/SQL文のsql_idが含まれています。

DEFINEコマンドを使用すると、これらの変数の各値を参照できます。

これらの変数は、他の置換変数と同様にアクセスおよび再定義できます。これらの変数はTTITLE、「&」置換変数、またはSQL*Plusコマンドライン・プロンプトで使用できます。

DEFINEコマンドを使用すると、他のDEFINE定義を参照する場合と同じ方法で、これらの9つの事前定義変数の定義を参照できます。また、DEFINEコマンドを使用してこれらの値を再定義したり、UNDEFINEコマンドを使用してこれらの定義を削除し、使用不可にすることもできます。

特定の変数の定義を表示するには、次のように入力します。

```
DEFINE variable
```

*variable*には、定義を表示する置換変数の名前を指定します。

すべての事前定義変数およびユーザー定義変数の定義を表示するには、次のように入力します。

DEFINE

すべての事前定義変数およびユーザー定義変数の定義が表示されます。

UNDEFINEコマンドを使用して、置換変数の定義を削除し、使用不可にできます。

事前定義変数の使用例

接続識別子を表示するSQL*Plusプロンプトを変更するには、次のように入力します。

```
SET SQLPROMPT '_CONNECT_IDENTIFIER > '
```

事前定義された_SQLPLUS_RELEASE置換変数の値を表示するには、次のように入力します。

```
DEFINE _SQLPLUS_RELEASE
```

事前定義された変数_SQLPLUS_RELEASEの値が表示されます。

```
DEFINE _SQLPLUS_RELEASE = "1801000000" (CHAR)
```

12.18 DEL

構文

```
DEL [ n | n m | n * | n LAST | * | * n | * LAST | LAST ]
```

バッファの1つ以上の行を削除します。

項

項	説明
<i>n</i>	行 <i>n</i> が削除されます。
<i>n m</i>	行 <i>n</i> から <i>m</i> までが削除されます。
<i>n *</i>	行 <i>n</i> からカレント行までが削除されます。
<i>n LAST</i>	行 <i>n</i> から最終行までが削除されます。
<i>*</i>	カレント行が削除されます。
<i>* n</i>	カレント行から行 <i>n</i> までが削除されます。
<i>* LAST</i>	カレント行から最終行までが削除されます。
<i>LAST</i>	最終行が削除されます。

バッファ内のカレント行を削除するには、句を指定せずに、DELを入力します。

使用方法

DELを実行すると、バッファの次の行がある場合は、その行がカレント行になります。DELを繰り返し入力することによって、複数の連続した行を削除できます。

ノート:



DEL は SQL*Plus コマンド、DELETE は SQL コマンドです。SQL DELETE コマンドの詳細は、「[DELETE](#)」を参照してください。

例

SQLバッファに次の問合せが含まれているとします。

```
SELECT LAST_NAME, DEPARTMENT_ID
FROM EMP_DETAILS_VIEW
WHERE JOB_ID = 'SA_MAN'
```



```
ORDER BY DEPARTMENT_ID;
```

WHERE句を含む行をカレント行にするには、次のように入力します。

```
LIST 3
```

```
3* WHERE JOB_ID = 'SA_MAN'
```

続いて次のように入力します。

```
DEL
```

現在、SQLバッファには次の行が含まれています。

```
SELECT LAST_NAME, DEPARTMENT_ID  
FROM EMP_DETAILS_VIEW  
ORDER BY DEPARTMENT_ID
```

バッファの3番目の行を削除するには、次のように入力します。

```
DEL 3
```

現在、SQLバッファには次の行が含まれています。

```
SELECT LAST_NAME, DEPARTMENT_ID  
FROM EMP_DETAILS_VIEW
```

12.19 DESCRIBE

構文

```
DESC[RIBE] { [schema.] object[@db_link] }
```

指定した表、ビューまたはシノニムの列定義を表示したり、指定したファンクションまたはプロシージャの仕様を表示します。

項

schema

オブジェクトまたはオブジェクトを表示するアクセス権が存在するスキーマを指定します。*schema*を省略した場合、そのオブジェクトが*public synonym*でない場合は、コマンド実行者自身のオブジェクトとみなされます。

object

定義を表示する表、ビュー、型、プロシージャ、ファンクション、パッケージまたはシノニムを指定します。

@*db_link*

*object*が存在しているデータベースに対応するデータベース・リンク名です。異なるスキーマ内の別の表にアクセスする権限の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。

使用方法

表、ビュー、型およびシノニムの定義の表示には、次の情報が含まれます。

- それぞれの列の名前
- それぞれの列についてNULL値が許可されるかどうか(NULLまたはNOT NULL)。
- 列のデータ型(CHAR、DATE、LONG、LONGRAW、NUMBER、RAW、ROWID、VARCHAR2(VARCHAR)、XMLTypeなど)
- 列の精度(および、数値列の場合は位取り(ある場合))。

DESCRIBEを実行すると、VARCHAR列がVARCHAR2の型とともに戻されます。

DESCRIBEコマンドを使用すると、SET DESCRIBEコマンドで設定した深さレベルまで、オブジェクトの定義を繰り返し表示できます。1つのオブジェクトに複数のオブジェクト型が含まれている場合は、行番号および属性や列のインデントを表示することもできます。詳細は、「SET」コマンドを参照してください。

表示されるデータの幅を制御するには、SET LINESIZEを使用します。

DESCRIBEコマンドで出力される列には、通常、現在指定されている行サイズの割合が割り当てられます。SET LINESIZEコマンドを使用して行サイズを増減させると、通常、各列の割合が増減します。これによって、表示上、テキストの折返しが意図しない位置で行われる場合があります。詳細は、「SET」コマンドを参照してください。

ファンクションおよびプロシージャの定義には、次の情報が含まれます。

- PL/SQLオブジェクトの型(ファンクションまたはプロシージャ)
- ファンクションまたはプロシージャの名前
- 戻される値の型(ファンクションの場合)
- 引数名、型、入力または出力、デフォルト値(ある場合)

- 列内のデータが暗号化されるかどうかを示すENCRYPTキーワード

例

EMP_DETAILS_VIEWの定義を表示するには、次のように入力します。

```
DESCRIBE EMP_DETAILS_VIEW
```

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER (6)
JOB_ID	NOT NULL	VARCHAR2 (10)
MANAGER_ID		NUMBER (6)
DEPARTMENT_ID		NUMBER (4)
LOCATION_ID		NUMBER (4)
COUNTRY_ID		CHAR (2)
FIRST_NAME		VARCHAR2 (20)
LAST_NAME	NOT NULL	VARCHAR2 (25)
SALARY		NUMBER (8, 2)
COMMISSION_PCT		NUMBER (2, 2)
DEPARTMENT_NAME	NOT NULL	VARCHAR2 (30)
JOB_TITLE	NOT NULL	VARCHAR2 (35)
CITY	NOT NULL	VARCHAR2 (30)
STATE_PROVINCE		VARCHAR2 (25)
COUNTRY_NAME		VARCHAR2 (40)
REGION_NAME		VARCHAR2 (25)

CUSTOMER_LOOKUPという名前のプロシージャの定義を表示するには、次のように入力します。

```
DESCRIBE customer_lookup
```

```
PROCEDURE customer_lookup
```

Argument Name	Type	In/Out	Default?
CUST_ID	NUMBER	IN	
CUST_NAME	VARCHAR2	OUT	

プロシージャaprocおよびbprocを含むパッケージAPACKを作成し、この定義を表示するには、次のように入力します。

```
CREATE PACKAGE apack AS
PROCEDURE aproc(P1 CHAR, P2 NUMBER);
PROCEDURE bproc(P1 CHAR, P2 NUMBER);
END apack;
/
```

Package created.

```
DESCRIBE apack
```

```
PROCEDURE APROC
```

Argument Name	Type	In/Out	Default?
P1	CHAR	IN	
P2	NUMBER	IN	

```
PROCEDURE BPROC
```

Argument Name	Type	In/Out	Default?
P1	CHAR	IN	
P2	NUMBER	IN	

STREET属性およびCITY属性を含むオブジェクト型ADDRESSを作成し、この定義を表示するには、次のように入力します。

```
CREATE TYPE ADDRESS AS OBJECT
( STREET VARCHAR2(20),
  CITY VARCHAR2(20)
);
/
```

Type created.

DESCRIBE address

Name	Null?	Type
STREET		VARCHAR2(20)
CITY		VARCHAR2(20)

属性LAST_NAME、EMPADDR、JOB_IDおよびSALARYを含むオブジェクト型EMPLOYEEを作成し、この定義を表示するには、次のように入力します。

```
CREATE TYPE EMPLOYEE AS OBJECT
(LAST_NAME VARCHAR2(30),
EMPADDR ADDRESS,
JOB_ID VARCHAR2(20),
SALARY NUMBER(7,2)
);
/
```

Type created.

DESCRIBE employee

Name	Null?	Type
LAST_NAME		VARCHAR2(30)
EMPADDR		ADDRESS
JOB_ID		VARCHAR2(20)
SALARY		NUMBER(7,2)

オブジェクト型addr_typeをオブジェクト型ADDRESSの表として作成し、この定義を表示するには、次のように入力します。

```
CREATE TYPE addr_type IS TABLE OF ADDRESS;
/
```

Type created.

DESCRIBE addr_type

addr_type TABLE OF ADDRESS		
Name	Null?	Type
STREET		VARCHAR2(20)
CITY		VARCHAR2(20)

オブジェクト型addr_varrayをオブジェクト型ADDRESSのVARRAYとして作成し、この定義を表示するには、次のように入力します。

```
CREATE TYPE addr_varray AS VARRAY(10) OF ADDRESS;
/
```

Type created.

DESCRIBE addr_varray

addr_varray VARRAY(10) OF ADDRESS

Name	Null?	Type
STREET		VARCHAR2 (20)
CITY		VARCHAR2 (20)

DEPARTMENT_ID、PERSONおよびLOC列を含む表departmentを作成し、この定義を表示するには、次のように入力します。

```
CREATE TABLE department
(DEPARTMENT_ID NUMBER,
PERSON EMPLOYEE,
LOC NUMBER
);
/
```

Table created.

DESCRIBE department

Name	Null?	Type
DEPARTMENT_ID		NUMBER
PERSON		EMPLOYEE
LOC		NUMBER

属性NUMERATOR、DENOMINATORおよびMETHOD rational_orderを含むオブジェクト型rationalを作成し、この定義を表示するには、次のように入力します。

```
CREATE OR REPLACE TYPE rational AS OBJECT
(NUMERATOR NUMBER,
DENOMINATOR NUMBER,
MAP MEMBER FUNCTION rational_order -
RETURN DOUBLE PRECISION,
PRAGMA RESTRICT_REFERENCES
(rational_order, RNDS, WNDS, RNPS, WNPS) );
/
```

```
CREATE OR REPLACE TYPE BODY rational AS OBJECT
MAP MEMBER FUNCTION rational_order -
RETURN DOUBLE PRECISION IS
BEGIN
RETURN NUMERATOR/DENOMINATOR;
END;
END;
/
```

DESCRIBE rational

Name	Null?	Type
NUMERATOR		NUMBER
DENOMINATOR		NUMBER
METHOD		

XMLTypeの列を含む表を作成し、この定義を表示するには、次のように入力します。

```
CREATE TABLE PROPERTY (Price NUMBER, Description SYS.XMLTYPE);
```

Table created

```
DESCRIBE PROPERTY;
```

Name	Null?	Type
PRICE		NUMBER
DESCRIPTION		SYS.XMLTYPE

SETコマンドを使用しDESCRIBEの出力を書式設定するには、次のように入力します。

```
SET LINESIZE 80
SET DESCRIBE DEPTH 2
SET DESCRIBE INDENT ON
SET DESCRIBE LINE OFF
```

オブジェクトの設定を表示するには、SHOWコマンドを使用して次のように入力します。

```
SHOW DESCRIBE
```

```
DESCRIBE DEPTH 2 LINENUM OFF INDENT ON
```

```
DESCRIBE employee
```

Name	Null?	Type
FIRST_NAME		VARCHAR2 (30)
EMPADDR		ADDRESS
STREET		VARCHAR2 (20)
CITY		VARCHAR2 (20)
JOB_ID		VARCHAR2 (20)
SALARY		NUMBER (7, 2)

暗号化された列col2を含む表des2_tableを作成し、定義するには、次のように入力します。

```
CREATE TABLE des2_table (
col1 VARCHAR2(10),
col2 VARCHAR2(15) ENCRYPT,
col3 CHAR(5),
col4 CHAR(20));
```

Table created

```
DESCRIBE des2_table;
```

Name	Null?	Type
COL1		VARCHAR2 (10)
COL2		VARCHAR2 (15) ENCRYPT
COL3		CHAR (5)
COL4		CHAR (20)

CREATE TYPEコマンドの使用方法の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。

SET DESCRIBEコマンドおよびSHOW DESCRIBEコマンドの使用方法的詳細は、[「SET」](#)コマンドおよび[「SHOW」](#)コマンドを参照してください。

12.20 DISCONNECT

構文

```
DISC[ONNECT]
```

データベースに対する保留中の変更をコミットし、現行のユーザー名をOracle Databaseからログアウトします。ただし、SQL*Plusは終了しません。

使用方法

ユーザーがOracle Databaseからログアウトしても、SQL*Plusは実行したままにしておく場合に、データベースへのユーザー・アクセスを防止するには、スクリプト内でDISCONNECTを使用します。SQL*Plusコマンドラインでは、EXITまたはQUITを使用してOracle Databaseからログアウトし、コンピュータのオペレーティング・システムに制御を戻します。

例

次に示すとおり、スクリプトをCONNECTコマンドで開始し、DISCONNECTコマンドで終了できます。

```
CONNECT HR
SELECT LAST_NAME, DEPARTMENT_NAME FROM EMP_DETAILS_VIEW;
DISCONNECT
SET INSTANCE FIN2
CONNECT HR2
```


12.21 EDIT

構文

```
ED[IT] [file_name [. ext]]
```

file_name [. *ext*]には、編集するファイル(通常はスクリプト)を指定します。

指定したファイルの内容またはバッファの内容を対象として、オペレーティング・システムのテキスト・エディタを起動します。

SQLバッファの内容をオペレーティング・システムのテキスト・エディタで編集するには、*filename*を指定せずにEDITを入力します。

使用方法

拡張子を省略すると、SQL*Plusは、デフォルトのコマンド・ファイル拡張子(通常はSQL)が指定されたものとみなします。デフォルトの拡張子の変更については、「SET」コマンドのSUFFIX変数を参照してください。

*filename*を指定すると、SQL*Plusは、そのディレクトリを見つけるためにORACLE_PATHで指定されたディレクトリを検索します。SQL*PlusがORACLE_PATHでスクリプト・ファイルを検出できない場合、またはORACLE_PATHが設定されていない場合は、現行の作業ディレクトリが検索されます。いずれのディレクトリにもファイルが見つからない場合、SQL*Plusは、指定した名前のファイルを作成します。

置換変数_EDITORには、EDITによって起動されるテキスト・エディタの名前が含まれます。_EDITORの値を変更することによって、テキスト・エディタを変更できます。置換変数の値の変更については、「[DEFINE](#)」を参照してください。_EDITORが未定義の場合、EDITは、デフォルトのオペレーティング・システム・エディタの起動を試みます。

EDITを指定すると、SQLバッファの内容を入れたAFIEDT.BUFというデフォルト名のファイルが(現行の作業ディレクトリ内に)作成され、テキスト・エディタが起動されて、そのファイルの内容が表示されます。ファイルAFIEDT.BUFがすでに存在している場合は、そのファイルが、バッファの内容で上書きされます。SET EDITFILEコマンドを使用して、デフォルトのファイル名を変更できます。EDITコマンドでデフォルトのファイル名の設定を変更する方法については、「SET」コマンドのEDITFILE変数を参照してください。

ノート:



デフォルトのファイル名 AFIEDT.BUF は、一部のオペレーティング・システムでは名前が異なる場合があります。

ファイル名の指定がなく、バッファが空の場合は、EDITからエラー・メッセージが戻されます。

WindowsでのSQL*Plus 9.0以下のバージョンでは、コマンドEDIT *は、無効なファイル名の警告を表示した後で空のファイルを開きました。WindowsでのSQL*Plus 10以上のバージョンでは、EDIT *は、無効なファイル名の警告を表示しますが、空のファイルは開かれません。SQL*Plus 9.0の動作を保持する場合は、次のようにSETコマンドを入力します。

```
SET SQLPLUSCOMPATIBILITY 9.0
```

編集セッションを終了し、SQL*Plusに戻るには、テキスト・エディタの通常の終了方法に従って編集セッションを終了します。エディタの終了時、SQL*Plusはファイルの内容をバッファにロードします。

ノート:



Windows で、ワードパッドをエディタとして使用する(`_EDITOR=write.exe`)場合、ワードパッドを終了するとバッファは再ロードされません。この場合、GET を使用してバッファを再ロードします。

例

オペレーティング・システムのテキスト・エディタを使用して、拡張子SQLが付いたファイルREPORTを編集するには、次のように入力します。

```
EDIT REPORT
```

12.22 EXECUTE

構文

```
EXEC[UTE] statement
```

*statement*にはPL/SQL文を指定します。

1つのPL/SQL文を実行します。EXECUTEコマンドは、ストアド・プロシージャを参照するPL/SQL文を実行する場合に便利です。PL/SQLの詳細は、[『Oracle Database PL/SQL言語リファレンス』](#)を参照してください。

使用方法

PL/SQL文があるためにEXECUTEコマンドが1行に収まらない場合は、SQL*Plusの継続文字(ハイフン)を使用します。

コマンドおよびPL/SQL文の長さが、SET LINESIZEで定義された長さを超えないようにする必要があります。

SET FEEDBACK OFFに設定すると、メッセージ「PL/SQLプロシージャが正常に完了しました。」を出力しないようにできます。

例

変数:*n*が次のように定義されている場合

```
VARIABLE n NUMBER
```

次のEXECUTEコマンドは、バインド変数*n*に値を割り当てます。

```
EXECUTE :n := 1
```

```
PL/SQL procedure successfully completed.
```

バインド変数の作成方法については、「[VARIABLE](#)」コマンドを参照してください。

12.23 EXIT

構文

```
[EXIT | QUIT] [SUCCESS | FAILURE | WARNING | n | variable | :BindVariable] [COMMIT | ROLLBACK]
```

保留中のすべての変更をコミットまたはロールバックし、Oracle DatabaseをログアウトしてSQL*Plusを終了し、オペレーティング・システムに制御を戻します。

項

```
[EXIT | QUIT]
```

どちらでも同じように使用できます(QUITはEXITのシノニムです)。

SUCCESS

正常に終了します。

FAILURE

障害を示すリターン・コードが戻され、処理が終了します。

WARNING

アラートを示すリターン・コードが戻され、処理が終了します。

COMMIT

終了する前に、保留中の変更がデータベースに保存されます。

n

リターン・コードとして指定する整数を指定します。

variable

SQL.SQLCODEのような、ユーザー定義変数またはシステム変数(バインド変数ではありません)を指定します。EXIT *variable*では、*variable*の値がリターン・コードとして戻され、処理が終了します。

:BindVariable

VARIABLEコマンドを使用してSQL*Plusで作成され、PL/SQLまたはその他のサブプログラムの中で参照される変数を指定します。*:BindVariable*を使用すると、サブプログラムを終了し、SQL*Plusの画面に戻ることができます。

ROLLBACK

ROLLBACK文が実行され、終了の前に、データベースに対する保留中の変更が破棄されます。

句を指定せずにEXITを入力すると、コミットが実行され、値SUCCESSが戻され、処理が終了します。

使用方法

EXITを使用すると、オペレーティング・システムのリターン・コードを指定できます。そのため、SQL*Plusのスクリプトをバッチ・モードで実行できます。また、予期しないイベントの発生をプログラムによって検出できます。検出の方法は、オペレーティング・システムによって異なります。

キーワードSUCCESS、WARNINGおよびFAILUREは、オペレーティング・システム依存値を表します。システムによっては、

WARNINGとFAILUREの区別がない場合もあります。

オペレーティング・システムによっては、オペレーティング・システム・リターン・コードの範囲が制限されていることもあります。その場合は、プラットフォーム間のEXIT *n*およびEXIT *variable* の移植性が制限されます。たとえば、UNIXでは、リターン・コード用の記憶域は1バイトのみです。したがって、リターン・コードの範囲は0から255に制限されます。

EXITオプションに構文エラーがあった場合または非数値変数を使用した場合は、SQL*PlusによってEXIT FAILURE COMMITが実行されます。

条件付きの終了については、「[WHENEVER SQLERROR](#)」および「[WHENEVER OSERROR](#)」コマンドを参照してください。

例

次の例では、まだコミットされていないすべてのトランザクションをコミットし、最後に実行されたSQLコマンドまたはPL/SQLブロックのエラー・コードを戻します。

```
EXIT SQL.SQLCODE
```

12.24 GET

構文

```
GET [FILE] file_name[. ext] [LIST | NOLIST]
```

オペレーティング・システム・ファイルをSQLバッファにロードします。

項

FILE

後続の引数によって、ロードするスクリプト名を指定するキーワードです。オプションのキーワードは、通常、省略されます。

fileはコマンド・キーワードであるため、fileというスクリプトをロードする場合は、一重引用符で囲む必要があります。

```
file_name[. ext]
```

ロードするファイル(通常はスクリプト)を指定します。

LIST

ロード後、ファイルの内容が表示されます。これはデフォルトです。

NOLIST

リストが非表示になります。

使用方法

ファイル拡張子を指定しない場合、SQL*Plusは、デフォルトのコマンド・ファイル拡張子(通常はSQL)が指定されたものとみなします。デフォルトの拡張子を変更する方法については、「[SET SUF\[IX\] {SQL | text}](#)」を参照してください。

指定するファイル名にlistまたはfileという単語が含まれている場合は、ファイル名を二重引用符で囲む必要があります。

SQL*Plusは、現行の作業ディレクトリでファイルを検索します。

オペレーティング・システム・ファイルには、単一のSQL文またはPL/SQLブロックが含まれている必要があります。文は、セミコロンで終了しないでください。SQL*Plusコマンド、または複数のSQL文かPL/SQLブロックが、オペレーティング・システム・ファイルからSQLバッファにロードされた場合、RUNコマンドまたは/(スラッシュ)コマンドを使用してバッファを実行すると、エラーになります。

SAVEコマンドで作成されたファイルをロードするには、GETコマンドを使用します。詳細は、「[SAVE](#)」を参照してください。

例

名前がYEARENDRPTで拡張子がSQLのファイルをバッファにロードするには、次のようにします。

```
GET YEARENDRPT
```

12.25 HELP

構文

```
HELP | ? [topic]
```

*topic*にはCOLUMNなどのSQL*Plusのヘルプ・トピックを指定します。

SQL*Plusコマンドラインのヘルプにアクセスします。HELP INDEXまたは?INDEXを入力して、項目のリストを表示します。Oracle Databaseライブラリは<http://www.oracle.com/technology/documentation/>で参照できます。

ヘルプ・システムに関するヘルプを表示するには、*topic*を指定しないでHELPまたは?を入力します。

使用方法

HELPの後には、1つのトピックのみ入力できます。トピックを略称(たとえば、COLUMNをCOL)で指定できます。ただし、略称で指定したトピックのみを入力して、その略称が一意でない場合、SQL*Plusは略称に一致するすべてのトピックのヘルプを表示します。たとえば、次のように入力したとします。

```
HELP EX
```

SQL*Plusは、EXECUTEコマンドの構文に続いてEXITコマンドの構文也表示します。

ヘルプが使用不可能という旨のメッセージが表示された場合は、データベース管理者に連絡してください。

例

SQL*Plusコマンドのリストを表示するには、次のように入力します。

```
HELP INDEX
```

または

```
? INDEX
```

SQL*Plusコマンドのリストを1列に表示するには、次のように入力します。

```
HELP TOPICS
```

12.26 HISTORY

構文

```
HIST[ORY] [n RUN | EDIT | DEL[ETE]] | [CLEAR | LIST]
```

ユーザーは以前使用したSQL*Plus、SQLまたはPL/SQLのコマンドを現在のセッションの履歴リストから実行、編集または削除できます。SET HISTORYコマンドを使用することで、現在のSQL*PlusセッションでHISTORYコマンドを有効または無効にできます。

HISTORYコマンドで、次の操作を実行できます。

- コマンド履歴リストのすべてのエントリをリストします。
- コマンド履歴リストのエントリを実行します。
- コマンド履歴リストのエントリを編集します。
- コマンド履歴リストからエントリを削除します。
- コマンド履歴リストのすべてのエントリをクリアします。

項

```
HIST[ORY]
```

コマンド履歴リストのすべてのエントリをリストします。

```
n
```

コマンド履歴リストのエントリを表します。アスタリスク(*)は、コマンド履歴リストで最後に使用されたコマンドを示します。

```
RUN
```

コマンド履歴リストからエントリ*n*を実行できます。

```
EDIT
```

デフォルトのテキスト・エディタを使用して、コマンド履歴リストのエントリ*n*を編集できます。コマンド履歴リストのエントリ*n*を編集して変更を保存すると、リストの末尾に新規エントリが作成されます。コマンド履歴リストのエントリ数が最大限度に到達すると、新規エントリに対応するためリスト内で最も古いエントリがクリアされます。

```
DEL [ETE]
```

コマンド履歴リストからエントリ*n*を削除できます。履歴リストからエントリを削除すると、リストが並べ替えられ、最新の変更が反映されます。

```
CLEAR
```

履歴リストのすべてのエントリをクリアできます。クリアすると、履歴リストはリカバリできません。

```
LIST
```

履歴リストのすべてのエントリをリストします。これはHIST[ORY]コマンドを単独で使用するのと同じです。

使用方法

デフォルトのテキスト・エディタの名前を保持するには、SQL*PlusのDEFINEコマンドを使用して変数_EDITORを定義します。たとえば、EDITで使用するエディタをviとして定義するには、次のコマンドを入力します。


```
DEFINE _EDITOR = vi
```

_EDITORが未定義の場合、EDITは、デフォルトのオペレーティング・システム・エディタの起動を試みます。詳細は、[「DEFINE」コマンド](#)を参照してください。

例12-1 例

次の例では、履歴リストの5番目のエントリを実行します。

```
HIST[ORY] 5 RUN
```

次の例では、履歴リストの3番目のエントリを編集できます。

```
HIST[ORY] 3 EDIT
```

次の例では、履歴リストから2番目のエントリを削除できます。

```
HIST[ORY] 2 DEL [ETE]
```

次の例では、履歴リストからすべてのエントリを削除できます。

```
HIST[ORY] CLEAR
```

次の例は、コマンド履歴を有効または無効にする方法およびコマンド履歴ステータスを確認する方法を示しています。

```
SQL> set history on
SQL> show history
History is ON and set to "100"
SQL> set history off
SQL> show history
History is OFF
SQL> set history 1000
SQL> show history
History is ON and set to "1000"
```

次の例は、履歴リストのすべてのエントリをリストする方法を示しています。

```
SQL> show history
History is ON and set to "100"
SQL> show user
USER is "SYSTEM"
SQL> desc dual
  Name                               Null?   Type
-----
PROD                                VARCHAR2(1)

SQL> select * from dual;

D
-
X

SQL> history 4 run

D
-
X

SQL> history
  1 show history
  2 show user
```

```
3 desc dual
* 4 select * from dual;
```

アスタリスク(*)は、コマンド履歴リストで最後に使用されたコマンドを示します。

次の例は、履歴リストのすべてのエントリをリストしてから、2番目のエントリを実行する方法を示しています。

```
SQL> history
1 show history
2 show user
3 desc dual
* 4 select * from dual;
SQL> history 2 run
USER is "SYSTEM"
SQL> history
1 show hist
* 2 show user
3 desc dual
4 select * from dual;
```

12.27 HOST

構文

```
HO[ST] [command]
```

*command*は、オペレーティング・システム・コマンドを表します。

SQL*Plusを終了せずに、オペレーティング・システム・コマンドを実行します。

オペレーティング・システムのプロンプトを表示するには、*command*を指定しないでHOSTのみを入力します。そのプロンプトでは、複数のオペレーティング・システム・コマンドを入力できます。SQL*Plusに戻る方法については、ご使用のオペレーティング・システムのプラットフォーム固有のOracleマニュアルを参照してください。

ノート:



HOST コマンドを使用して SQL*Plus セッションから入力したオペレーティング・システムのコマンドは、現行の SQL*Plus セッションに影響を与えません。たとえば、オペレーティング・システムの環境変数を設定しても、影響を受けるのは引き続き起動する SQL*Plus セッションのみです。

HOST を使用禁止にできます。HOST を使用禁止にする方法の詳細は、「[SQL*Plus のセキュリティ](#)」を参照してください。

使用方法

一部のオペレーティング・システムでは、HOSTのかわりに\$(Windows)、!(UNIX)などの文字が使用可能で、HOSTコマンドにアクセスできない場合があります。詳細は、ご使用のオペレーティング・システムのプラットフォーム固有のOracleのマニュアルを参照するか、またはDBAに問い合わせてください。

一部のプラットフォームでは、オペレーティング・システムによって異なるHOSTの戻り値によって_RC置換変数が作成されます。この置換変数は移植不可能なため、スクリプトで_RC置換変数を使用しないことをお勧めします。

SQL*Plusは、HOSTコマンドを実行する前にSQLTERMINATOR(デフォルトではセミコロン)を削除します。これに対処するには、SQLTERMINATORをもう1つ追加します。詳細は、[SET SQLT\[ERMINATOR\] {; | c | ON | OFF}](#)を参照してください。

例

UNIXオペレーティング・システムのコマンドls *.sqlを実行するには、次のように入力します。

```
HOST ls *.sql
```

Windowsオペレーティング・システムのコマンドdir *.sqlを実行するには、次のように入力します。

```
HOST dir *.sql
```

12.28 INPUT

構文

```
I [INPUT] [ text ]
```

*text*には、追加するテキストを指定します。

バッファ内のカレント行の後に、1行以上の新規テキストを追加します。

1行を追加するには、コマンドINPUTの後にテキスト行を入力します。コマンドとテキストの間は1つの空白で区切ります。1つ以上の空白で行を始めるには、INPUTおよびテキストの最初の空白以外の文字の間に複数の空白を入力します。

複数の行を追加するには、テキストを指定しないでINPUTを入力します。INPUTは、それぞれの行に対するプロンプトを表示します。INPUTを終了するには、NULL(空の)行またはピリオドを入力します。

使用方法

バッファ内の行数より大きい行番号をコマンド・プロンプトで入力して、その番号に続けてテキストを入力すると、SQL*Plusは、そのテキストをバッファの最後の新規行に追加します。その行番号に0(ゼロ)を指定して、0に続けてテキストを入力すると、SQL*Plusは、バッファの最初に行を挿入します(その行が行1になります)。

例

SQLバッファに次のコマンドが含まれているとします。

```
SELECT LAST_NAME, DEPARTMENT_ID, SALARY, COMMISSION_PCT
FROM EMP_DETAILS_VIEW
```

問合せにORDER BY句を追加するには、次のように入力します。

```
LIST 2
```

```
2* FROM EMP_DETAILS_VIEW
```

```
INPUT ORDER BY LAST_NAME
```

LIST 2によって、行2がカレント行になります。INPUTは、カレント行の後にORDER BY句を含む新規行を追加します。現在、SQLバッファには次の行が含まれています。

```
1 SELECT LAST_NAME, DEPARTMENT_ID, SALARY, COMMISSION_PCT
2 FROM EMP_DETAILS_VIEW
3* ORDER BY LAST_NAME
```

2行のWHERE句を追加するには、次のように入力します。

```
LIST 2
```

```
2* FROM EMP_DETAILS_VIEW
```

```
INPUT
```

```
3 WHERE JOB_ID = 'SA_MAN'
4 AND COMMISSION_PCT= .25
5
```

INPUTは、空白行またはピリオドが入力されるまで新規行の入力を求めるプロンプトを表示します。現在、SQLバッファには次の行が含まれています。

```
SELECT LAST_NAME, DEPARTMENT_ID, SALARY, COMMISSION_PCT
FROM EMP_DETAILS_VIEW
WHERE JOB_ID = 'SA_MAN'
AND COMMISSION_PCT = .25
ORDER BY LAST_NAME
```

12.29 LIST

構文

```
L[IST] [ n | n m | n * | n LAST | * | * n | * LAST | LAST ]
```

SQLバッファの1つ以上の行を表示します。

SQL*Plusコマンドラインでは、SQLバッファのすべての行を表示するために「;」を使用することもできます。

項

項	説明
<i>n</i>	行 <i>n</i> が表示されます。
<i>n m</i>	行 <i>n</i> から <i>m</i> が表示されます。
<i>n *</i>	行 <i>n</i> からカレント行までが表示されます。
<i>n LAST</i>	行 <i>n</i> から最終行までが表示されます。
<i>*</i>	カレント行が表示されます。
<i>* n</i>	カレント行から行 <i>n</i> までが表示されます。
<i>* LAST</i>	カレント行から最終行までが表示されます。
<i>LAST</i>	最終行が表示されます。

すべての行を表示するには、句を指定しないでLISTのみを入力するか、「;」を入力します。表示された最終行が新しいカレント行になります(アスタリスクが付きます)。

例

バッファの内容の表示するには、次のように入力します。

```
LIST
```

または、次のように入力します。

```
;  
  
1 SELECT LAST_NAME, DEPARTMENT_ID, JOB_ID  
2 FROM EMP_DETAILS_VIEW  
3 WHERE JOB_ID = 'SH_CLERK'  
4* ORDER BY DEPARTMENT_ID
```

アスタリスクは、行4がカレント行であることを示しています。

2行目のみを表示するには、次のように入力します。

LIST 2

2行目は、次のように表示されます。

```
2* FROM EMP_DETAILS_VIEW
```

カレント行(この時点では行2)から最終行までを表示するには、次のように入力します。

```
LIST * LAST
```

次のように表示されます。

```
2 FROM EMP_DETAILS_VIEW  
3 WHERE JOB_ID = 'SH_CLERK'  
4* ORDER BY DEPARTMENT_ID
```

12.30 PASSWORD

構文

```
PASSW[ORD] [username]
```

*username*にはユーザーを指定します。ユーザー名を省略すると、*username*はデフォルトで現行のユーザーになります。

パスワードを入力デバイスにエコー表示せずに変更できるようにします。

ユーザーのパスワード・プロファイルにPASSWORD_ROLLOVER_TIMEパラメータのゼロ以外の値が含まれている場合、パスワードを変更すると、ユーザー・アカウントのパスワード・ロールオーバー期間が開始されます。

段階的なパスワード・ロールオーバー(古いパスワードと新しいパスワードの両方が有効な期間)を有効にするプロファイルに関連付けられているデータベース・ユーザーの場合、SQL*Plusを使用してパスワードを変更し、新しく作成されたパスワードと古いパスワードの両方をユーザー認証中に機能させることができます。

詳細は、Oracle Databaseセキュリティ・ガイドの[アプリケーションの段階的データベース・パスワード・ロールオーバー](#)を参照してください。

パスワード・ロールオーバー期間中は、古いパスワード(*password1*)または新しいパスワード(*password2*)のいずれかを使用してログインできます。

ロールオーバー期間中にパスワードを再度変更するには、古いパスワードとして*password1*または*password2*のいずれかを使用します。たとえば、パスワードを*password3*に変更するには、次のいずれかの文を使用します。

```
ALTER USER jones IDENTIFIED BY password3 REPLACE password1
```

```
ALTER USER jones IDENTIFIED BY password3 REPLACE password2
```

パスワード・ロールオーバー期間中にパスワードを変更できる回数に制限はありません。ただし、パスワード・ロールオーバー期間中は、パスワードを最初に変更したときから固定されることに注意してください。

使用方法

他のユーザーのパスワードを変更するには、適切な権限が付与されている必要があります。パスワードの変更方法の詳細は、「[CONNECT](#)」コマンドを参照してください。

例

現行のパスワードを変更するには、次のように入力します。

```
PASSWORD
Changing password for your_password
Old password: your_password
New password: new_password
Retype new password: new_password
Password changed
```

DBAとしてログインしていて、ユーザーjohnwの現在のパスワードjohnwpassをjohnwnewpassに変更するには、次のようにします。

```
PASSWORD johnw
Changing password for johnw
New password: johnwnewpass
Retype new password: johnwnewpass
Password changed
```


パスワードは、画面でエコー表示されません。ここでは、参考のためにパスワードを表示しています。

12.31 PAUSE

構文

```
PAU[SE] [text]
```

*text*には、表示するテキストを指定します。

指定したテキストを表示し、その後ユーザーが[Return]を押すまで待ちます。

2行の空白行を表示するには、テキストを指定しないでPAUSEのみを入力します。

使用方法

PAUSEは、常にユーザーの応答を待つので、ユーザーが確実に[Return]を押すように通知するメッセージを使用してください。

コマンド入力のソースをファイルとして指定した場合でも、PAUSEは端末から入力を読み取ります(端末が使用できる場合)。

レポートのページ間での一時停止については、「[SET PAU\[SE\] {ON | OFF | *text*}](#)」を参照してください。

例

「Adjust paper and press RETURN to continue.」と出力し、ユーザーが[Return]を押すまでSQL*Plusが待つようにするには、スクリプトに次のPAUSEコマンドを含める必要があります。

```
SET PAUSE OFF
PAUSE Adjust paper and press RETURN to continue.
SELECT ...
```

12.32 PRINT

構文

```
PRINT [variable ...]
```

variable ...には、値を表示するバインド変数の名前を指定します。

バインド変数の現行の値を表示します。

すべてのバインド変数を出力するには、変数を指定しないでPRINTのみを入力します。

使用方法

バインド変数は、VARIABLEコマンドを使用して作成します。詳細と例は、「[VARIABLE](#)」を参照してください。

出力の問合せと同じように、PRINT出力の書式を制御できます。詳細は、「[SQL*Plusレポートの書式設定](#)」で説明されている書式設定の技法を参照してください。

成功したPL/SQLブロックで参照されたバインド変数、またはEXECUTEコマンドで使用されたバインド変数を自動的に表示するには、「SET」コマンドのAUTOPRINT句を使用します。詳細は、「[SET](#)」を参照してください。

例

次に、PRINTコマンドの例を示します。

```
VARIABLE n NUMBER
BEGIN
:n := 1;
END;
/

PL/SQL procedure successfully completed.

PRINT n

N
-----
1
```

12.33 PROMPT

構文

```
PROMPT [text]
```

*text*には、表示するメッセージのテキストを指定します。

指定したメッセージまたは空白行を、ユーザーの画面に送信します。*text*を省略すると、PROMPTはユーザーの画面に空白行を表示します。

使用方法

このコマンドをスクリプトで使用して、ユーザーに情報を提示することができます。

例

次に、ASKFORDEPT.SQLというスクリプトで、ACCEPTとともにPROMPTを使用する例を示します。ASKFORDEPT.SQLには、次のSQL*PlusコマンドおよびSQLコマンドが含まれています。

```
PROMPT PROMPT Please enter a valid department PROMPT For example: 10
SELECT DEPARTMENT_NAME FROM
EMP_DETAILS_VIEW WHERE DEPARTMENT_ID = &NEWDEPT
```

STARTまたは@を使用して、このファイルを実行するとします。

```
@ASKFORDEPT.SQL VAL1
@HTTP://machine_name.domain:port/ASKFORDEPT.SQL VAL1
```

```
Please enter a valid department
For example: 10
Department ID?>
```

プロンプトの「Department ID?>」で部門番号を入力できます。デフォルトでは、SQL*Plusにより、置換の前および後に &NEWDEPTを含む行がリストされ、「Department ID?>」プロンプトに入力された数値に対応する部門名が表示されます。この動作を回避するには、SET VERIFY OFFを使用します。

12.34 RECOVER

構文

```
RECOVER { general | managed | BEGIN BACKUP | END BACKUP }
```

*general*句の構文は次のとおりです。

```
[AUTOMATIC] [FROM location]  
{ { full_database_recovery | partial_database_recovery | LOGFILE filename }  
[ {TEST | ALLOW integer CORRUPTION | parallel_clause } [TEST  
| ALLOW integer CORRUPTION | parallel_clause ]... ] CONTINUE [DEFAULT] | CANCEL }
```

*full_database_recovery*句の構文は次のとおりです。

```
[STANDBY] DATABASE  
[ {UNTIL {CANCEL | TIME date | CHANGE integer } | USING BACKUP CONTROLFILE  
| SNAPSHOT TIME date}... ]
```

*partial_database_recovery*句の構文は次のとおりです。

```
{ TABLESPACE tablespace [, tablespace]...  
| DATAFILE { filename | filenumber } [, filename | filenumber]...  
| STANDBY {TABLESPACE tablespace [, tablespace]...  
| DATAFILE { filename | filenumber } [, filename | filenumber]...}  
UNTIL [CONSISTENT WITH] CONTROLFILE }
```

*parallel*句の構文は次のとおりです。

```
{ NOPARALLEL | PARALLEL [ integer ] }
```

*managed*句の構文は次のとおりです。

```
MANAGED STANDBY DATABASE recover_clause | cancel_clause | finish_clause
```

where the *recover_clause* has the following syntax:

```
{ { DISCONNECT [ FROM SESSION ] | { TIMEOUT integer | NOTIMEOUT } }  
| { NODELAY | DEFAULT DELAY | DELAY integer }  
| NEXT integer | { EXPIRE integer | NO EXPIRE }  
| parallel_clause | USING CURRENT LOGFILE | UNTIL CHANGE integer  
| THROUGH { [ THREAD integer ] SEQUENCE integer  
| ALL ARCHIVELOG | { ALL | LAST | NEXT } SWITCHOVER} } ...
```

*cancel_clause*の構文は次のとおりです。

```
CANCEL [IMMEDIATE] [WAIT | NOWAIT]
```

*finish_clause*の構文は次のとおりです。

```
[ DISCONNECT [ FROM SESSION ] ] [ parallel_clause ]  
FINISH [ SKIP [ STANDBY LOGFILE ] ] [ WAIT | NOWAIT ]
```

*parallel_clause*の構文は次のとおりです。

```
{ NOPARALLEL | PARALLEL [ integer ] }
```

1つ以上の表領域、1つ以上のデータ・ファイル、またはデータベース全体に対してメディア・リカバリを実行します。RECOVERコ

マンドの詳細は、『[Oracle Database管理者ガイド](#)』、『[Oracle Database SQL言語リファレンス](#)』のALTER DATABASE RECOVERコマンドに関する項、および『[Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド](#)』を参照してください。

項

AUTOMATIC

リカバリ操作を続ける必要のある次のアーカイブREDOログ・ファイルの名前を自動的に生成します。Oracle Databaseは、ターゲットのREDOログ・ファイル名を生成するために、LOG_ARCHIVE_DEST(またはLOG_ARCHIVE_DEST_1)パラメータおよびLOG_ARCHIVE_FORMATパラメータ(またはそのデフォルト)を使用します。そのファイルが存在する場合、REDOはそのファイルが適用されるように含めます。ファイルが存在しない場合、SQL*Plusは、提案として生成したファイル名をプロンプトに表示します。

AUTOMATICまたはLOGFILEのどちらも指定しない場合、SQL*Plusで自動的に生成したファイル名をプロンプトに表示します。生成されたファイル名を指定するか、または完全に修飾されたファイル名に置き換えます。アーカイブされたファイル名が、Oracle Databaseが生成したものと異なっていると判明している場合は、LOGFILE句を使用することによってファイル名を指定するための時間を削減できます。

FROM *location*

アーカイブREDOログ・ファイル・グループが読み取られた位置を指定します。*location*の値は、完全に指定されたファイル位置である必要があります。このパラメータを省略すると、SQL*Plusは、アーカイブREDOログ・ファイル・グループが初期化パラメータLOG_ARCHIVE_DESTまたはLOG_ARCHIVE_DEST_1で指定された位置に存在するとみなします。ファイルをSET LOGSOURCEで設定した場合、FROMは指定できません。

full_database_recovery

データベース全体のリカバリを指定できます。

partial_database_recovery

個々の表領域およびデータ・ファイルのリカバリを指定できます。

LOGFILE

指定したREDOログ・ファイルを適用して、メディア・リカバリを続行します。対話形式のリカバリ・モード(AUTORECOVERY OFF)では、無効なログ・ファイル名を入力すると、無効なログ・ファイル名に対するエラーが表示され、新しいログ・ファイル名の入力を求めるプロンプトが表示されます。

TEST

発生する可能性がある問題を検知するために試行リカバリを指定します。REDOは正常に適用されますが、変更はディスクに書き込まれません。また、変更は試行リカバリの最後にロールバックされます。バックアップをリストアした場合は、試行リカバリでTEST句のみを使用できます。ログ・ファイルが破損している場合、リカバリ処理中に許容可能な破損ブロックの数を指定します。通常のリカバリ処理では、*integer*に2以上は指定できません。

ALLOW *integer* CORRUPTION

ログ・ファイルが破損している場合、リカバリ処理中に許容可能な破損ブロックの数を指定します。

parallel _clause

リカバリ操作中に使用する平行処理の並列度を指定できます。

CONTINUE

スレッドを無効にするために中断した後、複数インスタンス・リカバリを続行します。

CONTINUE DEFAULT

ログ・ファイル名を指定しない場合、Oracle Databaseが自動的に生成したREDOログ・ファイル名を使用してリカバリを続行します。これは、Oracle Databaseがファイル名の入力を求めるプロンプトを表示しないことを除いて、AUTOMATICを指定するのと同じです。

CANCEL

取消しベースのリカバリを中断します。

SNAPSHOT TIME date

ストレージ・スナップショット最適化を使用して、ストレージ・スナップショットでデータベースをリカバリします。

STANDBY DATABASE

制御ファイルおよびプライマリ・データベースからコピーしたアーカイブREDOログ・ファイルを使用して、スタンバイ・データベースをリカバリします。スタンバイ・データベースは、マウントされているがオープンされていない状態である必要があります。

DATABASE

データベース全体をリカバリします。

UNTIL CANCEL

取消しベースの不完全リカバリを指定します。アーカイブREDOログ・ファイルとして生成されたファイル名をプロンプトに表示することによって、リカバリが進められます。また、ファイル名のかわりにCANCELを指定すると、リカバリは完了します。

UNTIL TIME

時間ベースの不完全リカバリを指定します。一重引用符を使用して、次の形式で指定します。

'YYYY-MM-DD:HH24:MI:SS'

UNTIL CHANGE

変更ベースの不完全リカバリを指定します。*integer*は、リカバリする最後のシステム変更番号(SCN)の次の番号です。たとえば、データベースを9番のSCNのトランザクションまでリストアする場合は、UNTIL CHANGE 10と指定します。

USING BACKUP CONTROLFILE

現在の制御ファイルのかわりに使用される制御ファイルのバックアップを指定します。

TABLESPACE

特定の表領域をリカバリします。*tablespace*は、現行データベースの表領域の名前です。1文で、16の表領域をリカバリできます。

DATAFILE

特定のデータ・ファイルをリカバリします。データ・ファイルのすべての番号を指定できます。

STANDBY TABLESPACE

スタンバイ・データベース内の損失または破損した表領域を、プライマリ・データベースおよび制御ファイルからコピーしたアーカイブREDOログ・ファイルを使用して再構築します。

STANDBY DATAFILE

スタンバイ・データベース内の損失または破損したデータ・ファイルを、プライマリ・データベースおよび制御ファイルからコピーしたアーカイブREDOログ・ファイルを使用して再構築します。

UNTIL CONSISTENT WITH CONTROLFILE

古いスタンバイ・データ・ファイルまたは表領域のリカバリは、現行のスタンバイ・データベースの制御ファイルを使用するように指定します。

PARALLEL [*integer*]

これはデフォルトです。SQL*Plusは並列度を選択します。並列度とは、すべての関係するインスタンスで使用可能なCPUの数に、PARALLEL_THREADS_PER_CPU初期化パラメータの値を掛けたものです。

RECOVERY_PARALLELISM初期化パラメータは、PARALLELキーワードによって上書きされます。PARALLELキーワードの詳細は、『[Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)』を参照してください。

*integer*を使用して、パラレル操作で使用するパラレル・スレッド数である並列度を指定します。各パラレル・スレッドは、1つまたは2つのパラレル実行プロセスを使用します。

NOPARALLEL

シリアル・リカバリ処理を指定します。

MANAGED STANDBY DATABASE

継続スタンバイ・リカバリ・モードを指定します。このモードでは、スタンバイ・データベースは、スタンバイ・データベースのアーキテクチャ全体の、アクティブ・コンポーネントとみなされます。プライマリ・データベースは、REDOログ・ファイルをスタンバイ・サイトに積極的にアーカイブします。これらのアーカイブREDOログがスタンバイ・サイトにアーカイブされると、管理スタンバイ・リカバリ操作で使用可能になります。継続スタンバイ・リカバリは、メディア・リカバリに制限されています。

この句のパラメータの詳細は、『[Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド](#)』を参照してください。

DISCONNECT

管理REDO処理(MRP)が分離バックグラウンド・プロセスとしてアーカイブREDOファイルを適用する必要があることを示します。現行のセッションは使用可能のままとなります。

TIMEOUT

継続リカバリ操作までの待機時間を分単位で指定します。リカバリ・プロセスは*integer*に指定された時間(分)、要求されたアーカイブREDOログがスタンバイ・データベースへ書込み可能になるまで待機します。REDOログ・ファイルがその時間内に書込み可能にならない場合、リカバリ・プロセスは、エラー・メッセージを表示して終了します。この場合、継続スタンバイ・リカバリ・モードに戻るための文を再発行できます。

この句を指定しない場合、またはNOTIMEOUTを指定する場合は、RECOVER CANCEL句とともに文を再発行するか、インスタンスを停止するか、またはインスタンス障害が発生するまで、データベースは継続スタンバイ・リカバリ・モードのままです。

NODELAY

プライマリ・データベースのLOG_ARCHIVE_DEST_nパラメータで設定されたDELAYを無視して、スタンバイ・データベースに遅延されているアーカイブ・ログをすぐに適用します。この句を省略すると、アーカイブ・ログの適用は、パラメータの設定に従って遅延します。DELAYがパラメータで指定されていない場合、アーカイブ・ログはすぐに適用されます。

DEFAULT DELAY

LOG_ARCHIVE_DEST_n初期化パラメータで指定したデフォルトの時間(分)待機してから、アーカイブREDOログを適用します。

DELAY *integer*

*integer*に指定した時間(分)待機してから、アーカイブREDOログを適用します。

NEXT *integer*

指定された番号のアーカイブREDOログを、アーカイブ後できるだけ早く適用します。このパラメータは、プライマリ・データベースのLOG_ARCHIVE_DEST_nパラメータのDELAY設定、および以前のSQL*Plus RECOVERコマンドとALTER DATABASE RECOVERコマンドに指定された遅延値よりも一時的に優先されます。

EXPIRE *integer*

管理リカバリ操作が自動終了してから現在の時刻までの時間を分単位で指定します。

NO EXPIRE

以前指定したEXPIRE *integer*オプションを無効にします。

USING CURRENT LOGFILE

REDOログ・ファイルが一杯になった場合、スタンバイ・オンラインREDOログ・ファイルからREDOログ・ファイルをリカバリします。その際、スタンバイ・データベースでREDOログ・ファイルをアーカイブしておく必要はありません。

UNTIL CHANGE *integer*

管理リカバリを指定したシステム変更番号(SCN)まで処理します。ただし、指定した番号は含まれません。

THROUGH THREAD *integer* SEQUENCE *integer*

アーカイブ・ログのスレッド番号および順序番号に基づいて管理リカバリを終了します。対応するアーカイブ・ログが適用されると、管理リカバリは終了します。指定しない場合は、THREADのデフォルト値は1になります。

THROUGH ALL ARCHIVELOG

すべてのアーカイブ・ログがリカバリされるまで、管理スタンバイを継続します。この文を使用すると、以前の文で発行されたTHROUGH THREAD *integer* SEQUENCE *integer*句を上書きできます。THROUGH句を省略した場合、これがデフォルト値となります。

THROUGH ALL SWITCHOVER

スイッチオーバー操作が実行される間、管理スタンバイ・リカバリを実行できるようにします。

THROUGH LAST SWITCHOVER

最終のEND-OF-REDOアーカイブ・インジケータの後、管理スタンバイ・リカバリを終了します。

THROUGH NEXT SWITCHOVER

次のEND-OF-REDOアーカイブ・インジケータのリカバリ後、管理スタンバイ・リカバリを終了します。

CANCEL (*managed clause*)

現在のアーカイブREDOファイルの適用後、管理スタンバイ・リカバリを終了します。リカバリ処理が終了すると、セッションに制御が戻ります。

CANCEL IMMEDIATE

現行のアーカイブREDOファイルを適用した後、または次のREDOログ・ファイルが読み取られた後(いずれか早い方の処理の後)で、管理スタンバイ・リカバリを終了します。リカバリ処理が終了すると、セッションに制御が戻ります。

CANCEL IMMEDIATE WAIT

現行のアーカイブREDOファイルを適用した後、または次のREDOログ・ファイルが読み取られた後(いずれか早い方の処理の後)で、管理スタンバイ・リカバリを終了します。管理スタンバイ・リカバリが終了すると、セッションに制御が戻ります。

CANCEL IMMEDIATEは、RECOVER MANAGED STANDBY DATABASE文を発行した同じセッションから発行できません。

CANCEL IMMEDIATE NOWAIT

現行のアーカイブREDOファイルを適用した後、または次のREDOログ・ファイルが読み取られた後(いずれか早い方の処理の後)で、管理スタンバイ・リカバリを終了します。すぐにセッションに制御が戻ります。

CANCEL NOWAIT

次のREDOログ・ファイルの読み取り後に管理スタンバイ・リカバリを終了し、すぐに制御をセッションに戻します。

FINISH

スタンバイ・データベースの現行のスタンバイ・オンライン・ログ・ファイルをリカバリします。プライマリ・データベースに障害が発生した場合に、この句は有効です。この句はアーカイブ・ログに指定された遅延を無視します。そのため、ログがすぐに適用されます。

TIMEOUT、DELAY、EXPIREまたはNEXT句を指定している場合は、FINISHを発行できません。

使用方法

OSDBAロールを使用可能にする必要があります。マルチスレッド・サーバーを経由して接続した場合、RECOVERコマンドを使用できません。

データベース全体(すべての表領域)でメディア・リカバリを実行するには、マウントされていてクローズ状態である必要があります。また、リカバリに必要なすべての表領域がオンラインである必要があります。

表領域でメディア・リカバリを実行するには、データベースはマウントされているか、またはオープン状態である必要があります。また、表領域はオフラインである必要があります。

データ・ファイルでメディア・リカバリを実行するには、データベースがオープンしていて破損したデータ・ファイルとオフライン(そのファイルがSYSTEM表領域の一部である場合を除く)でマウントされた状態である必要があります。

RECOVERコマンドを使用する前に、以前のバックアップから破損したデータ・ファイルのコピーをリストアしておく必要があります。バックアップが作成されたときまでさかのぼって、すべてのアーカイブされたREDOログ・ファイルおよびオンラインのREDOログ・ファイルにアクセスできることを確認してください。

リカバリ時に他のログ・ファイルが必要な場合、必要とされるファイルの名前が提案として生成されます。名前は、初期化パラメータLOG_ARCHIVE_DESTおよびLOG_ARCHIVE_FORMATで指定された値から導出されます。必要に応じて、LOG_ARCHIVE_DESTで指定した宛先に、リカバリに必要なアーカイブREDOログ・ファイルのコピーをリストアします。SET LOGSOURCEコマンドでLOGSOURCE変数を設定することによって初期化パラメータを上書きできます。

リカバリ時に、[Return]を押すことによって提案されたログ名を指定できます。ログ名のかわりにCANCELを入力することによってリカバリをキャンセルできます。また、プロンプトを進めないでファイルを自動選択するにはプロンプトでAUTOを入力します。

自動リカバリ(SET AUTORECOVERY ON)を使用可能にする場合は、ファイル名をプロンプトに表示させないでリカバリを進めます。状態を示すメッセージは、それぞれのログ・ファイルが適用されているときに表示されます。通常のメディア・リカバリが実行されている場合、完了した状態で戻ってきます。

例

データベース全体をリカバリするには、次のように入力します。

```
RECOVER DATABASE
```

指定した時間までにデータベースをリカバリするには、次のように入力します。

```
RECOVER DATABASE UNTIL TIME 01-JAN-2001:04:32:00
```

データベースから2つの表領域ts_oneおよびts_twoをリカバリするには、次のように入力します。

```
RECOVER TABLESPACE ts_one, ts_two
```

データベースからデータ・ファイルdata1.dbをリカバリするには、次のように入力します。

```
RECOVER DATAFILE 'data1.db'
```

12.35 REMARK

構文

REM[ARK]

スクリプト内でコメントを開始します。SQL*Plusはコメントをコマンドとして解釈しません。

使用方法

REMARKコマンドは、行の先頭に指定する必要があります。コメントは、その行の終わりで終了します。コメントおよびコマンドは同じ行に指定できません。

REMARK行の終わりにある「-」は、行継続文字とみなされます。

スクリプトに、SQLコメント・デリミタ/* ... */またはANSI/ISOコメント・デリミタ--を使用してコメントを入力する方法の詳細は、[スクリプトへのコメントの挿入について](#)を参照してください。

例

次のスクリプトには、代表的なコメントがいくつか含まれています。

```
REM COMPUTE uses BREAK ON REPORT to break on end of table
BREAK ON REPORT
COMPUTE SUM OF "DEPARTMENT 10" "DEPARTMENT 20" -
"DEPARTMENT 30" "TOTAL BY JOB_ID" ON REPORT
REM Each column displays the sums of salaries by job for
REM one of the departments 10, 20, 30.
SELECT JOB_ID,
SUM(DECODE( DEPARTMENT_ID, 10, SALARY, 0)) "DEPARTMENT 10",
SUM(DECODE( DEPARTMENT_ID, 20, SALARY, 0)) "DEPARTMENT 20",
SUM(DECODE( DEPARTMENT_ID, 30, SALARY, 0)) "DEPARTMENT 30",
SUM(SALARY) "TOTAL BY JOB_ID"
FROM EMP_DETAILS_VIEW
GROUP BY JOB_ID;
```

12.36 REPFOOTER

構文

```
REP[FOOTER] [PAGE] [printspec [text | variable] ...] | [ON | OFF]
```

*printspec*には、テキストの配置および書式設定に使用される次の句のうちの1つ以上を指定します。

```
COL n S[KIP] [n] TAB n LE[FT] CE[NTER] R[IGHT] BOLD FORMAT text
```

各レポートの下部に、指定したレポート・フッターを配置して書式設定したり、現行のREPFOOTER定義を表示します。

現行のREPFOOTER定義を表示するには、句を指定しないでREPFOOTERのみを入力します。

項

REPFOOTERコマンド構文の項および句については、「[REPHEADER](#)」コマンドを参照してください。

使用方法

テキストまたは変数の前に*printspec*句を入力しない場合、REPFOOTERは、テキストまたは変数を左揃えにします。

*printspec*には、定数および変数をいくつでも指定できます。SQL*Plusは、定数および変数を指定した順序で表示し、それぞれの定数または変数をその直前の*printspec*句で指定されている位置および書式に設定します。

ノート:



SET EMBEDDED が ON の場合、レポート・フッターは非表示になります。

例

「END EMPLOYEE LISTING REPORT」を独立ページのレポート・フッターとして定義して、中央揃えにするには、次のように入力します。

```
REPFOOTER PAGE CENTER 'END EMPLOYEE LISTING REPORT'
TTITLE RIGHT 'Page: ' FORMAT 999 SQL.PNO
SELECT LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000;
```

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Russell	14000
Partners	13500
Hartstein	13000
sum	98500

END EMPLOYEE LISTING REPORT

Page: 2

6 rows selected.

レポート・フッターの定義を変更しないで、レポート・フッターを非表示にするには、次のように入力します。

REPFOOTER OFF

12.37 REPHEADER

構文

```
REPH[EADER] [PAGE] [printspec [text | variable] ...] | [ON | OFF]
```

*printspec*には、テキストの配置および書式設定に使用される次の句のうちの1つ以上を指定します。

```
COL n S[KIP] [n] TAB n LE[FT] CE[NTER] R[IGHT] BOLD FORMAT text
```

各レポートの上部に、指定したレポート・ヘッダーを配置して書式設定したり、現行のREPHEADER定義を表示します。

現行のREPHEADER定義を表示するには、句を指定しないでREPHEADERのみを入力します。

項

これらの項および句は、REPFOOTERコマンドにも適用されます。

PAGE

指定したレポート・ヘッダーを印刷した後または印刷する前に、新規ページが開始されます。

text

レポート・ヘッダーまたはレポート・フッターのテキストを指定します。1行に2つ以上の語を入れるには、*text*を一重引用符で囲んで入力します。デフォルトはNULLです。

variable

置換変数、または次のシステム管理値のいずれかを指定します。SQL.LNOはカレント行の番号です。SQL.PNOは現行のページ番号です。SQL.CODEは現行のエラー・コードです。SQL.RELEASEは現行のOracle Databaseのリリース番号です。SQL.USERは現行のユーザー名です。

これらの値の1つを出力するには、適切な変数をレポート・ヘッダーまたはレポート・フッターの中で参照します。FORMAT句を使用して、*variable*の形式を設定できます。

OFF

定義に影響を与えずに、レポート・ヘッダーまたはレポート・フッターをオフ(非表示)にします。

COL *n*

カレント行の列*n*までインデントします(列*n*を過ぎている場合は、後退します)。ここでいう「列」とは、印刷位置のことで、表の列ではありません。

S[KIP] [*n*]

新規行の先頭まで*n*回スキップします。*n*を省略すると、1回スキップします。*n*に0を入力すると、カレント行の先頭まで戻ります。

TAB *n*

*n*列分前方に(*n*に負の値を入力した場合は後方に)スキップします。ここでいう「列」とは、印刷位置のことで、表の列ではありません。

LE[FT] CE[NTER] R[IGHT]

カレント行のデータを、それぞれ左揃え、中央揃えおよび右揃えにします。SQL*Plusは、*printspec*の終わりまで、あるいは次

のLEFT、CENTER、RIGHTまたはCOLコマンドまでのデータ項目を、1グループとして整列させます。CENTERおよびRIGHTでは、SET LINESIZE値を使用し、後続のデータ項目の位置が計算されます。

BOLD

データを太字で印刷します。SQL*Plusは、端末上で同じデータを3回続けて出力することによって、太字印刷を行います。一部のオペレーティング・システムでは、SQL*Plusがプリンタにテキストを太字でなく3回連続で印刷するように指示することがあります。

FORMAT *text*

次のFORMAT句まで、またはこのコマンドの終わりまでの、データ項目の書式を決定する書式モデルを指定します。書式モデルは、A10や\$999などのテキスト定数である必要があります。書式設定および有効な書式モデルの詳細は、「[COLUMN](#)」を参照してください。

書式モデルのデータ型が、指定したデータ項目のデータ型と一致しない場合、FORMAT句はその項目には影響しません。

指定したデータ項目の前に適切な書式モデルがない場合、SQL*Plusは、SET NUMFORMATで指定された書式に従って、NUMBER値を出力します。また、SET NUMFORMATが指定されていない場合はデフォルトの書式に従って、NUMBER値を出力します。SQL*Plusは、デフォルトの書式を使用してDATE値を出力します。

使用方法

テキストまたは変数の前に*printspec*句を入力しない場合、REPHEADERは、テキストまたは変数を左揃えにします。

*printspec*には、定数および変数をいくつでも指定できます。SQL*Plusは、定数および変数を指定した順序で表示し、それぞれの定数または変数をその直前の*printspec*句で指定されている位置および書式に設定します。

例

「EMPLOYEE LISTING REPORT」を独立ページのレポート・ヘッダーとして定義して、中央揃えにするには、次のように入力します。

```
REPHEADER PAGE CENTER 'EMPLOYEE LISTING REPORT'
TTITLE RIGHT 'Page: ' FORMAT 999 SQL.PNO
SELECT LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000;
```

		Page: 1
	EMPLOYEE LISTING REPORT	
		Page: 2
LAST_NAME	SALARY	
-----	-----	
King	24000	
Kochhar	17000	
De Haan	17000	
Russell	14000	
Partners	13500	
Hartstein	13000	

sum	98500	

6 rows selected.

レポート・ヘッダーの定義を変更せずに非表示にするには、次のように入力します。

```
REPHEADER OFF
```


12.38 RUN

構文

```
R[UN]
```

現在、SQLバッファに格納されているSQLコマンドまたはPL/SQLブロックを表示して実行します。

使用方法

RUNを実行すると、SQLバッファの最終行がカレント行になります。

/(スラッシュ)コマンドは、RUNと同様の動作をしますが、SQLバッファ内のコマンドを画面に表示しません。SQLバッファには、常に最新のSQL文またはPL/SQLブロックが含まれます。

例

SQLバッファに次のスクリプトが含まれているとします。

```
SELECT DEPARTMENT_ID  
FROM EMP_DETAILS_VIEW  
WHERE SALARY>12000
```

RUNを使用してスクリプトを実行するには、次のように入力します。

RUN

```
1 SELECT DEPARTMENT_ID  
2 FROM EMP_DETAILS_VIEW  
3 WHERE SALARY>12000
```

DEPARTMENT_ID

```
-----  
90  
90  
90  
80  
80  
20
```

6 rows selected.

12.39 SAVE

構文

```
SAV[E] [FILE] file_name[. ext] [CRE[ATE] | REP[LACE] | APP[END]]
```

SQLバッファの内容を、オペレーティング・システムのスクリプトに保存します。

項

FILE

後続の引数によって、保存したスクリプトに付ける名前を指定するキーワードです。オプションのキーワードは、通常、省略されます。

fileはコマンド・キーワードであるため、fileというスクリプトを保存する場合は、一重引用符で囲む必要があります。

```
file_name[. ext]
```

バッファの内容を保存するスクリプトを指定します。

CREATE

指定した名前で、ファイルを新規作成します。これはデフォルトの動作です。

REP[LACE]

既存のファイルの内容を置換します。ファイルが存在しない場合、ファイルが作成されます。

APP[END]

指定したファイルの終わりに、バッファの内容を追加します。

使用方法

拡張子を指定しない場合、SQL*Plusは、デフォルトのコマンド・ファイル拡張子(通常はSQL)が指定されたものとみなします。デフォルトの拡張子を変更する方法については、「[SET SUF\[IX\] {SQL | text}](#)」を参照してください。

SAVEコマンド句(CREATE、REPLACEまたはAPPEND)と同じ名前でファイルを保存するには、ファイル拡張子を指定する必要があります。

SAVEを使用してSQLバッファの内容を保存すると、スラッシュ(/)を含む1行がファイルの終わりに追加されます。

例

名前がDEPTSALRPTで、拡張子がSQLのファイルにバッファ内容を保存するには、次のように入力します。

```
SAVE DEPTSALRPT
```

名前がDEPTSALRPTで、拡張子がOLDのファイルにバッファ内容を保存するには、次のように入力します。

```
SAVE DEPTSALRPT. OLD
```

12.40 SET

現行のセッションに対するSQL*Plus環境を変更するシステム変数を設定します。たとえば、この操作は、次の目的のために行います。

- HTML書式設定のカスタマイズ
- 列ヘッダーの出力の有効化または無効化
- 1ページごとの行数の設定
- データの表示幅の設定

構文

```
SET system_variable value
```

*system_variable*および*value*は、次の[SETシステム変数の一覧](#)表に示すいずれかの句を表します。

使用方法

SQL*Plusには、SQL*Plusセッション用の特定の環境を設定可能にするシステム変数(SETコマンド変数)が保持されています。これらのシステム変数は、SETコマンドを使用して変更し、SHOWコマンドを使用して表示できます。各システム変数のデフォルト値は、次のセクションで下線が付けられます。

SET ROLEおよびSET TRANSACTIONはSQLコマンドです(詳細は、「[SQL文: MERGEからUPDATE](#)」を参照してください)。キーワードTRANSACTIONまたはROLEが後に指定されていない場合、SETはSQL*Plusコマンドとみなされます。

12.41 SETシステム変数の一覧

システム変数	説明
<u>SET APPEND[INFO]{ON OFF text}</u>	DBMS_APPLICATION_INFO パッケージを使用して、スクリプトの自動登録を設定します。
<u>SET ARRAY[SIZE] {15 n}</u>	SQL*Plus がデータベースから一度にフェッチする行の数(バッチ)を設定します。
<u>SET AUTO[COMMIT]{ON OFF IMMEDIATE} n}</u>	データベースに対する保留中の変更を、Oracle Database がコミットする時点を制御します。
<u>SET AUTOP[RINT] {ON OFF}</u>	バインド変数の自動出力を設定します。
<u>SET AUTORECOVERY [ON OFF]</u>	ON を指定すると、リカバリ時に必要なアーカイブ REDO ログ・ファイルのデフォルトのファイル名を自動的に適用するために RECOVER コマンドが設定されます。
<u>SET AUTOT[RACE] {ON OFF TRACE[ONLY]} [EXP[LAIN]] [STAT[ISTICS]]</u>	正常に実行された SQL DML 文(SELECT、INSERT、UPDATE、DELETE または MERGE)のレポートを表示します。
<u>SET BLO[CKTERMINATOR] {. c ON OFF}</u>	PL/SQL ブロックを終了するために使用する英数字以外の文字を、c に設定します。
<u>SET CMDS[EP] {; c ON OFF}</u>	1 行に入力された複数の SQL*Plus コマンドを区切るために使用する英数字以外の文字を、c に設定します。
<u>SET COLINVI[SIBLE] [ON OFF]</u>	ON にすると、DESCRIBE コマンドで非表示列の列情報が表示されるよう設定されます。
<u>SET COLSEP { text}</u>	選択された列の間に出力するテキストを設定します。
<u>SET CON[CAT] {. c ON OFF}</u>	通常は SQL*Plus が置換変数名の一部として解釈する文字を、変数の直後に続けて指定する場合に、置換変数参照を終了させるために使用できる文字を設定します。
<u>SET COPYC[OMMIT] {0 n}</u>	データベースに対する変更を、COPY コマンドによってコミットした後のバッチの数を制御します。
<u>SET COPYTYPECHECK {ON OFF}</u>	COPY コマンドを使用して表へ挿入または追加しているときは、データ型

システム変数	説明
	の比較を行わないように設定します。
<u>SET DEF[INE] {& c ON OFF}</u>	変数の接頭辞として使用する文字を、c に設定します。
<u>SET DESCRIBE [DEPTH {1 n ALL}]</u> <u>[LINENUM {ON OFF}] [INDENT</u> <u>{ON OFF}]</u>	オブジェクトの定義を繰り返し表示する深さレベルを設定します。
<u>SET ECHO {ON OFF}</u>	コマンドが実行されるときに、START コマンドによりスクリプト内の各コマンドを表示するかどうかを制御します。
<u>SET EDITF[ILE] file_name[.ext]</u>	EDIT コマンド用のデフォルトのファイル名を設定します。
<u>SET EMB[EDDED] {ON OFF}</u>	ページのどこから各レポートが始まるかを制御します。
<u>SET ERRORL[OGGING] {ON OFF}</u> <u>[TABLE [schema.]tablename]</u> <u>[TRUNCATE] [IDENTIFIER identifier]</u>	SQL、PL/SQL および SQL*Plus の各エラーをエラー・ログ表に記録した後で問合せできるようにします。
<u>SET ESC[APE] {¥ c ON OFF}</u>	エスケープ文字として使用する文字を定義します。
<u>SET ESCCHAR {@ ? % OFF}</u>	ファイル名でエスケープする特殊文字を指定します。エラーの原因となる文字変換を防止します。
<u>SET EXITC[OMMIT] {ON OFF}</u>	デフォルトの EXIT 動作を COMMIT にするか ROLLBACK にするかを指定します。
<u>SET FEED[BACK] {6 n ON OFF </u> <u>ONLY}] [SQL_ID]</u>	問合せによって n 個以上のレコードを選択した場合に、問合せから戻されるレコード数を表示します。
<u>SET FLAGGER {OFF ENTRY </u> <u>INTERMED[IATE] FULL}</u>	SQL 文が ANSI/ISO SQL92 規格に準拠しているかどうかを調べます。
<u>SET FLU[SH] {ON OFF}</u>	出力をユーザーの表示デバイスに、いつ送るかを制御します。
<u>SET HEA[DING] {ON OFF}</u>	レポートへの列ヘッダーの出力を制御します。
<u>SET HEADS[EP] { c ON OFF}</u>	ヘッダーのセパレータ文字として入力する文字を定義します。
<u>SET HIST[ORY] {ON OFF n}</u>	現在の SQL*Plus セッションで発行されたコマンドおよび SQL 文または PL/SQL 文の履歴を有効または無効にします。

システム変数	説明
<u>SET INSTANCE [instance_path LOCAL]</u>	セッションのデフォルトのインスタンスを、指定したインスタンス・パスに変更します。
<u>SET LIN[ESIZE] {80 n WINDOW}</u>	SQL*Plus が新しい行を開始する前に、1 行に表示する文字の合計数を設定します。
<u>SET LOBOF[FSET] {1 n}</u>	BLOB、BFILE、CLOB および NCLOB データを取得し、表示する開始位置を設定します。
<u>SET LOBPREFETCH {0 n}</u>	SQL*Plus がデータベースから一度にプリフェッチする LOB データの量を設定します。
<u>SET LOGSOURCE [pathname]</u>	リカバリ中に取得されるアーカイブ・ログの位置を指定します。
<u>SET LONG {80 n}</u>	LONG、BLOB、BFILE、CLOB、NCLOB および XMLType 値を表示するため、および LONG 値をコピーするための最大幅をバイト単位で設定します。
<u>SET LONGC[HUNKSIZE] {80 n}</u>	SQL*Plus が LONG、BLOB、BFILE、CLOB、NCLOB または XMLType 値を取得するときの増分値のサイズ(バイト単位)を設定します。
<u>SET MARK[UP]</u>	CSV 形式のデータまたは HTML マークアップ・テキストを出力します。
<u>SET NEWP[AGE] {1 n NONE}</u>	各ページの最上部から上部タイトルまでの間に入れる空白行の数を設定します。
<u>SET NULL text</u>	SQL SELECT コマンドの実行結果の中で NULL 値を表すテキストを設定します。
<u>SET NUMF[ORMAT] format</u>	数値を表示するためのデフォルトの書式を設定します。
<u>SET NUM[WIDTH] {10 n}</u>	数値を表示するためのデフォルトの幅を設定します。
<u>SET PAGES[IZE] {14 n}</u>	各ページの行数を設定します。
<u>SET PAU[SE] {ON OFF text}</u>	レポートの実行中に端末のスクロール制御を実現します。
<u>SET RECSEP {WR[APPED] EA[CH] OFF}</u>	RECSEP には、SQL*Plus がどこでレコードを分離するかを指定します。

システム変数	説明
<u>SET RECSEPCHAR { c }</u>	レコード・セパレータを表示または出力します。
<u>SET ROWLIMIT {n OFF}</u>	問合せに対して表示する行数の制限を設定します。
<u>SET ROWPREFETCH {15 n}</u>	SQL*Plus がデータベースから一度にプリフェッチする行数を設定します。
<u>SET SECURED COL {OFF ON}</u> <u>[UNAUTH[ORIZED] text]</u> <u>[UNK[NOWN] text]</u>	セキュアな列値を、列を表示する権限のないユーザーおよびセキュリティが不明の列に対して表示する方法を設定します。
<u>SET SERVEROUT[PUT] {ON OFF}</u> <u>[SIZE {n UNL[IMITED]}]</u> <u>[FOR[MAT] {WRA[PPED] </u> <u>WOR[D_WAPPED] </u> <u>TRU[NCATED]}]</u>	ストアド・プロシージャの出力(DBMS_OUTPUT.PUT_LINE)または PL/SQL ブロックの出力を、SQL*Plus で表示するかどうかを制御します。
<u>SET SHIFT[INOUT] {VIS[IBLE] </u> <u>INV[ISIBLE]}]</u>	シフト文字を表示する端末に適した位置合せを実現します。
<u>SET SHOW[MODE] {ON OFF}</u>	SET を使用して、SQL*Plus システム変数の設定を変更したときに、新旧の設定値を SQL*Plus で表示するかどうかを制御します。
<u>SET SQLBL[ANKLINES] {ON OFF}</u>	SQL*Plus が SQL コマンドまたはスクリプト内に空白行を入れるかどうかを制御します。
<u>SET SQLC[ASE] {MIX[ED] LO[WER]</u> <u> UP[PER]}]</u>	SQL コマンドおよび PL/SQL ブロックの大/小文字を実行直前に変換します。
<u>SET SQLCO[NTINUE] {> text}</u>	1 つの SQL*Plus コマンドをハイフン(-)を使用して次の行に続けた後で、プロンプトとして SQL*Plus によって表示する文字列順序を設定します。
<u>SET SQLN[UMBER] {ON OFF}</u>	SQL コマンドまたは PL/SQL ブロックの 2 行目以降の行のためのプロンプトを設定します。
<u>SET SQLPLUSCOMPAT[IBILITY]</u> <u>{x.y[.z]}</u>	VARIABLE の動作または出力書式を x.y[.z] で指定されたリリースまたはバージョン互換に設定します。
<u>SET SQLPRE[FIX] {# c}</u>	SQL*Plus の接頭文字を設定します。
<u>SET SQLPRE[FIX] {# c}</u> <u>SET</u> <u>SQLP[ROMPT] {SQL> text}</u>	SQL*Plus コマンド・プロンプトを設定します。

システム変数	説明
<u>SET SQLT[ERMINATOR] {; c ON OFF}</u>	SQL コマンドを終了および実行するために使用する文字を、c に設定します。
<u>SET STATEMENTC[ACHE] {0 n}</u>	文キャッシュ・サイズを設定します。
<u>SET SUF[IX] {SQL text}</u>	スクリプトを参照するコマンドの中で SQL*Plus が使用するデフォルト・ファイルを設定します。
<u>SET TAB {ON OFF}</u>	SQL*Plus が空白をどのように端末に出力するかを指定します。
<u>SET TERM[OUT] {ON OFF}</u>	スクリプトから実行するコマンドによる出力の表示を制御します。
<u>SET TI[ME] {ON OFF}</u>	現在の時刻表示を制御します。
<u>SET TIMI[NG] {ON OFF}</u>	タイミング統計の表示を制御します。
<u>SET TRIM[OUT] {ON OFF}</u>	SQL*Plus で、それぞれの表示行の終わりに後続の空白を入れるかどうかを指定します。
<u>SET TRIMS[POOL] {ON OFF}</u>	SQL*Plus で、それぞれのスプール行の終わりに後続の空白を入れるかどうかを指定します。
<u>SET UND[ERLINE] {- c ON OFF}</u>	SQL*Plus で、レポートの列ヘッダーに下線を付けるために使用する文字を、c に設定します。
<u>SET VER[IFY] {ON OFF}</u>	置換変数を値に置き換える前後で、SQL 文または PL/SQL コマンドのテキストを表示するかどうかを制御します。
<u>SET WRA[P] {ON OFF}</u>	カレント行の幅に対して、選択した行が長すぎる場合に、切り捨てるかどうかを制御します。
<u>SET XMLOPT[IMIZATIONCHECK] [ON OFF]</u>	完全に最適化された XML 問合せおよび DML 操作のみを実行するように指定します。開発およびデバッグでのみ使用してください。本番環境では使用しないでください。
<u>SET XQUERY BASEURI {text}</u>	使用するベース URI を定義します。これは、汎用 XQuery 式の作成時にアクセスするファイルの接頭辞を変更する場合に有効です。
<u>SET XQUERY ORDERING {UNORDERED ORDERED </u>	XQuery の結果の順序付けを制御します。

システム変数	説明
<u>DEFAULT}</u>	
<u>SET XQUERY NODE {BYVALUE BYREFERENCE DEFAULT}</u>	作成されるノードまたは戻されるノードの保存モードを設定します。
<u>SET XQUERY CONTEXT {text}</u>	XQuery コンテキストの項目を指定します。ノードまたは値のいずれかを指定できます。

12.41.1 SET APPINFO{ON | OFF | テキスト}

DBMS_APPLICATION_INFOパッケージを使用して、スクリプトの自動登録を設定します。

この設定によって、各スクリプトのパフォーマンスおよびリソースの使用状況をDBAが監視できます。V\$SESSIONおよびV\$SQLAREA仮想表のMODULE列には、登録名が表示されます。DBMS_APPLICATION_INFO.READ_MODULEプロシージャを使用して、登録名を読み取ることもできます。

ONを指定すると、@コマンド、@@コマンドまたはSTARTコマンドによって起動されたスクリプトが登録されます。OFFを指定すると、スクリプトの登録が無効になります。そのかわりに、テキストの現在の設定値が登録されます。*text*には、スクリプトが実行されていない場合、またはAPPINFOがOFFの場合(デフォルト)に登録するテキストを指定します。*text*のデフォルトは「SQL*Plus」です。*text*に複数の語を入力する場合、それらの語全体を引用符で囲む必要があります。*text*の最大長は、DBMS_APPLICATION_INFOパッケージによって制限されます。

登録名の書式は、*nn@xfilename*です。*nn*は、スクリプトの深さレベルです。*x*は、スクリプト名を切り捨てる場合は「<」、そうでない場合は空白です。*filename*はスクリプト名で、DBMS_APPLICATION_INFOパッケージ・インタフェースで許可されている長さまで切り捨てられる場合があります。

例

APPINFOの値(デフォルトはOFF)を表示するには、次のように入力します。

```
SET APPINFO ON
SHOW APPINFO
```

```
APPINFO is ON and set to "SQL*Plus"
```

デフォルト・テキストを変更するには、次のように入力します。

```
SET APPINFO 'This is SQL*Plus'
```

登録が行われたことを確認するには、次のように入力します。

```
VARIABLE MOD VARCHAR2(50)
VARIABLE ACT VARCHAR2(40)
EXECUTE DBMS_APPLICATION_INFO.READ_MODULE(:MOD, :ACT);
```

```
PL/SQL procedure successfully completed.
```

```
PRINT MOD
```

```
MOD
```

```
-----
This is SQL*Plus
```

APPINFOをデフォルト設定に戻すには、次のように入力します。

```
SET APPINFO OFF
```

12.41.2 SET ARRAY[SIZE] {15 | *n*}

SQL*Plusがデータベースから一度にフェッチする行数を設定します。

有効な値は1から5000です。値を大きくすると、多数の行をフェッチする問合せおよび副問合せの効率は向上しますが、必要なメモリーも多くなります。100より大きい値を使用しても、パフォーマンスはほとんど向上しません。ARRAYSIZEを指定しても、効率を上げる以外にSQL*Plus操作の結果に対して影響はありません。

12.41.3 SET AUTO[COMMIT]{ON | OFF | IMM[EDIATE] | *n*}

SQLまたはPL/SQLコマンドの後、データベースに対する保留中の変更がOracle Databaseによってコミットされる時点を制御します。

ONを指定すると、Oracle DatabaseによってINSERT、UPDATE、DELETEまたはPL/SQLブロックがそれぞれ正常に実行された後で、データベースに対する保留中の変更がコミットされます。OFFを指定すると、自動コミットされないため、手動で(たとえば、SQLコマンドのCOMMITを使用して)変更をコミットする必要があります。IMMEDIATEを指定すると、ONと同じ機能が得られます。*n*を指定すると、Oracle Databaseによって*n*個のSQL INSERT、UPDATE、DELETEの各コマンドまたはPL/SQLブロックが正常に実行された後で、データベースに対する保留中の変更がコミットされます。*n*は、0(ゼロ)以上2,000,000,000未満である必要があります。文カウンタは、*n*個のINSERT、UPDATE、DELETEコマンドまたはPL/SQLブロック、コミット、ロールバックまたはSET AUTOCOMMITコマンドの完了後、0(ゼロ)にリセットされます。

SET AUTOCOMMITでは、コミットの動作はSQL*Plusの終了時には変更されません。コミットされていないデータは、デフォルトでコミットされます。

ノート:



この機能の場合は、実際に含まれているSQLコマンドの数に関係なく、1つのPL/SQLブロックが1つのトランザクションとみなされます。

12.41.4 SET AUTOP[RINT] {ON | OFF}

バインド変数の自動出力を設定します。

ONまたはOFFによって、SQL*Plusがバインド変数(正常に実行されたPL/SQLブロックで参照されているか、またはEXECUTEコマンドで使用されているもの)を自動的に表示するかどうかを制御します。

バインド変数の表示の詳細は、「[PRINT](#)」を参照してください。

12.41.5 SET AUTORECOVERY [ON | OFF]

ONを指定すると、リカバリ時に必要なアーカイブREDOログ・ファイルのデフォルトのファイル名を自動的に適用するためにRECOVERコマンドが設定されます。

予期できる名前で予期できる位置に必要なファイルが提供されるので、対話は必要ありません。使用されるファイル名は、初期化パラメータLOG_ARCHIVE_DESTおよびLOG_ARCHIVE_FORMATの値から導出されます。

OFFを指定すると、デフォルト・オプションとして、手動でファイル名を入力するか、または自動生成されたデフォルトのファイル名を指定するかを選択できます。データベース・リカバリの詳細は、[「RECOVER」](#)を参照してください。

例

AUTOMATICにリカバリ・モードを設定するには、次のように入力します。

```
SET AUTORECOVERY ON
RECOVER DATABASE
```

12.41.6 SET AUTOT[RACE] {ON | OFF | TRACE[ONLY]} [EXP[LAIN]] [STAT[ISTICS]]

正常に実行されたSQL DML文(SELECT、INSERT、UPDATE、DELETEまたはMERGEなど)のレポートを表示します。

レポートには、実行統計および問合せ実行パスを含めることができます。

動的サンプリングが有効な場合、DMLに関するSQL*Plusのレポート出力が異なる場合があります。

OFFを指定すると、トレース・レポートは表示されません。ONを指定すると、トレース・レポートが表示されます。TRACEONLYを指定すると、トレース・レポートは表示されますが、問合せデータは、存在しても出力されません。EXPLAINを指定すると、実行計画が実行されて、問合せ実行パスが表示されます。STATISTICSを指定すると、SQL文統計が表示されます。

EXPLAIN PLANの詳細は、[「EXPLAIN PLAN」](#)を参照してください。

明示オプションを指定しないで、ONまたはTRACEONLYを使用すると、デフォルトでEXPLAIN STATISTICSが使用されます。

TRACEONLYオプションは、大きい問合せデータを抑止する場合に有効です。STATISTICSを指定すると、SQL*Plusは、サーバーから問合せデータをフェッチしますが、データは表示しません。

AUTOTRACEレポートは、文が正常に完了した後で出力されます。

SQL*PlusがSTATISTICSレポートを作成すると、データベースへの2番目の接続が自動的に確立されます。STATISTICSオプションがOFFに設定されたか、またはSQL*Plusからログアウトした場合に、この接続はクローズされます。

AUTOTRACEレポートの書式は、接続されているサーバーのバージョンおよびサーバーの構成によって異なります。

AUTOTRACE PLANの追加情報および表形式の出力は、Oracle Database 10gリリース1(10.1)以上に接続している場合にサポートされます。それより前のデータベースに接続する場合は、旧形式またはAUTOTRACEレポートが使用されます。

FIPSフラグ付けが使用可能な状態のときは、AUTOTRACEは使用できません。

AUTOTRACEの詳細は、[文のトレースについて](#)を参照してください。

12.41.7 SET BLO[CKTERMINATOR] {. | c | ON | OFF}

PL/SQLブロックを終了するために使用する文字を、cに設定します。

英数字または空白は使用できません。ブロックを実行するには、RUNまたは/(スラッシュ)コマンドを実行する必要があります。

OFFを設定すると、SQL*Plusは、PL/SQLブロック・ターミネータを認識しません。ONを設定すると、cの値を最近使用された文字ではなくデフォルトのピリオド(.)に戻します。

12.41.8 SET CMD[EP] {; | c | ON | OFF}

1行に入力された複数のSQL*Plusコマンドを区切るために使用する英数字以外の文字を、cに設定します。

ONまたはOFFは、1行に複数のコマンドを入力できるかどうかを制御します。ONを設定すると、自動的にコマンド・セパレータ文字がセミicolon(;)に設定されます。

例

TTITLEでタイトルを指定して、同一行にCOLUMNで列を書式設定するには、次のように入力します。

```
SET CMDSEP +
TTITLE LEFT 'SALARIES' + COLUMN SALARY FORMAT $99,999
SELECT LAST_NAME, SALARY FROM EMP_DETAILS_VIEW
WHERE JOB_ID = 'SH_CLERK';
```

SALARIES LAST_NAME	SALARY
Taylor	\$3,200
Fleaur	\$3,100
Sullivan	\$2,500
Geoni	\$2,800
Sarchand	\$4,200
Bull	\$4,100
Dellinger	\$3,400
Cabrio	\$3,000
Chung	\$3,800
Dilly	\$3,600
Gates	\$2,900
Perkins	\$2,500
Bell	\$4,000
Everett	\$3,900
McCain	\$3,200
Jones	\$2,800

SALARIES LAST_NAME	SALARY
Walsh	\$3,100
Feeney	\$3,000
OConnell	\$2,600
Grant	\$2,600

20 rows selected.

12.41.9 SET COLINVI[SIBLE] [ON | OFF]

ONを指定すると、DESCRIBEコマンドで非表示列の情報を表示できるようになります。

SET COLINVISIBLEを指定しても、非表示列を含む問合せ文に対して影響はありません。非表示列のデータを取得するには、問合せでその列を明示的に指定します。

例

非表示列の情報をDESCRIBEコマンドで表示するには、次のようにします。

非表示列を含む表を作成します。

```
create table test_invisible_cols (emp_id number, emp_info char(20),
                                emp_acc_no number invisible);
```

Table created.

DESCRIBEコマンドを使用して表の列を表示します。

```
describe test_invisible_cols
```

Name	Null?	Type
EMP_ID		NUMBER
EMP_INFO		CHAR (20)

デフォルトのSET COLINVISIBLE OFFでは、非表示列は結果に表示されません。SET COLINVISIBLEのデフォルト設定をONに変更します。

```
SET COLINVISIBLE ON
```

```
colinvisible ON
```

再度、DESCRIBEコマンドを使用して表の列を表示します。今度は非表示列が出力されます。

```
describe test_invisible_cols
```

Name	Null?	Type
EMP_ID		NUMBER
EMP_INFO		CHAR (20)
EMP_ACC_NO (INVISIBLE)		NUMBER

12.41.10 SET COLSEP { | テキスト }

列出力と列出力の間に出力される列セパレータ文字を設定します。

COLSEP変数に空白または句読記号が含まれている場合は、一重引用符でテキストを囲む必要があります。*text*のデフォルト値は単一の空白です。

複数行を含む行の場合、異なる行を開始する列の間には列セパレータは出力されません。列セパレータは、BREAK...SKIP *n*によって生成される空白行には表示されず、レコード・セパレータが列セパレータで上書きされることもありません。詳細は、[SET RECSEP {WR\[APPED\] | EA\[CH\] | OFF}](#)を参照してください。

例

列セパレータを「|」に設定するには、次のように入力します。

```
SET MARKUP HTML PREFORMAT ON
SET COLSEP ' | '
SELECT LAST_NAME, JOB_ID, DEPARTMENT_ID
FROM EMP_DETAILS_VIEW
WHERE DEPARTMENT_ID = 20;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID
Hartstein	MK_MAN	20
Fay	MK_REP	20

12.41.11 SET CON[CAT] { . | c | ON | OFF }

通常は、SQL*Plusが置換変数の一部として解釈する文字を、変数の直後に続けて指定する場合に、置換変数参照を終了するために使用できる文字を設定します。

CONCATをオンに切り替えると、SQL*PlusはCONCATの値をピリオドにリセットします。

12.41.12 SET COPYC[OMMIT] {0 | n}

データベースに対する変更を、COPYコマンドによってコミットした後の行数を制御します。

COPYでは、*n*個の行バッチがコピーされるたびに、宛先データベースへの行がコミットされます。有効な値は0から5000です。ARRAYSIZE変数を使用して、バッチのサイズを設定できます。COPYCOMMITを0に設定すると、COPYではコピー操作が終わったときにのみコミットを実行します。

12.41.13 SET COPYTYPECHECK {ON | OFF}

COPYコマンドを使用して表へ挿入または追加しているときは、データ型の比較を行わないように設定します。

このため、CHARをDB2 DATEにコピーする必要がある場合、DB2へのコピーが容易になります。

12.41.14 SET DEF[INE] {& | c | ON | OFF}

置換変数の接頭辞として使用する文字を、*c*に設定します。

ONまたはOFFによって、SQL*Plusがコマンドをスキャンして置換変数を検索し、その値に置換するかどうかを指定します。ONを設定すると、*c*の値を最近使用された文字ではなくデフォルトの「&」に戻します。OFFを設定すると、SCAN変数の設定がオーバーライドされます。

SCAN変数の詳細は、「[SET SCAN {ON|OFF}\(廃止\)](#)」を参照してください。

12.41.15 SET DESCRIBE [DEPTH {1 | n | ALL}] [LINENUM {ON | OFF}] [INDENT {ON | OFF}]

オブジェクトの定義を繰り返し表示する深さレベルを設定します。

DEPTH句の有効範囲は、1から50です。SET DESCRIBE DEPTH ALLを設定する場合、深さは最大レベルの50に設定します。1つのオブジェクトに複数のオブジェクト型が含まれている場合は、行番号および属性や列のインデントを表示することもできます。SET LINESIZEコマンドを使用して、表示されるデータ幅を制御します。

オブジェクトの詳細は、「[DESCRIBE](#)」を参照してください。

例

オブジェクト型ADDRESSを作成するには、次のように入力します。

```
CREATE TYPE ADDRESS AS OBJECT
( STREET  VARCHAR2(20),
  CITY    VARCHAR2(20)
);
/
```

Type created

オブジェクト型ADDRESSのネストされたオブジェクトEMPADDRを含む表EMPLOYEEを作成するには、次のように入力します。

```
CREATE TABLE EMPLOYEE
(LAST_NAME VARCHAR2(30),
 EMPADDR ADDRESS,
```

```

JOB_ID VARCHAR2 (20),
SALARY NUMBER (7, 2)
);
/

```

Table created

深さレベル2まで表EMPLOYEEを表示して、行数も表示しながら出力をインデントするには、次のように入力します。

```

SET DESCRIBE DEPTH 2 LINENUM ON INDENT ON
DESCRIBE employee

```

	Name	Null?	Type
1	LAST_NAME		VARCHAR2 (30)
2	EMPADDR		ADDRESS
3	2 STREET		VARCHAR2 (20)
4	2 CITY		VARCHAR2 (20)
5	JOB_ID		VARCHAR2 (20)
6	SALARY		NUMBER (7, 2)

12.41.16 SET ECHO {ON | OFF}

@、@@またはSTARTを使用して実行するスクリプトでコマンドをエコー表示するかどうかを制御します。ONを指定すると、画面にコマンドが表示されます。OFFを指定すると、非表示になります。ECHOは、対話方式で入力するコマンドまたはオペレーティング・システムからSQL*Plusにリダイレクトするコマンドの表示には影響しません。

12.41.17 SET EDITF[ILE] *file_name* [.ext]

EDITコマンド用のデフォルトのファイル名を設定します。EDITコマンドの詳細は、「[EDIT](#)」を参照してください。EDITコマンド用のデフォルトのファイル名は、SQLバッファであるafiedt.bufです。

パスまたはファイル拡張子(あるいはその両方)を挿入できます。デフォルトの拡張子を変更する方法については、「[SET SUF\[IX\] {SQL | text}](#)」を参照してください。デフォルトのファイル名およびファイル名の最大長は、オペレーティング・システムによって異なります。

12.41.18 SET EMB[EDDED] {ON | OFF}

ページのどこから各レポートが始まるかを制御します。

OFFを指定すると、各レポートは新しいページの一番上から始まります。ONを指定すると、新しいページのどこからでもレポートを開始できます。直前のレポートが終了した直後に次のレポートを出力するには、EMBEDDEDをONに設定します。

12.41.19 SET ERRORL[OGGING] {ON | OFF} [TABLE [*schema.*]*tablename*] [TRUNCATE] [IDENTIFIER 識別子]

SQL*Plusエラー・ロギングをONまたはOFFに切り替えます。エラー・ロギングでは、SQL、PL/SQLおよびSQL*Plusのエラーと関連パラメータをエラー・ログ表に記録します。このログ表を問い合せて、問合せによって発生するエラーを確認できます。エラー・ロギングがONの場合は、問合せが対話型またはスクリプトのどちらで実行されるかに関係なく、エラーが記録されます。これは、長時間実行される問合せによって生成されるエラーの取得に特に有効で、SPOOLコマンドの使用によるすべての出力の取得、または実行中の監視を回避できます。

デフォルトで、エラーは各スキーマの表SPERRORLOGに書き込まれます。この表が存在しない場合は、自動的に作成されます。TABLE *schema.tablename* オプションを使用して、他の表を使用するように指定することもできます。SPERRORLOG 以外の表を使用する場合は、その表がすでに存在していて、アクセス可能である必要があります。「[ユーザー定義エラー・ログ表の作成](#)」を参照してください。

内部エラーが発生した場合は、それ自体をコールするエラー・ログによって発生する再帰エラーを回避するため、エラー・ロギングは自動的にOFFに設定されます。

エラー・ロギングは、デフォルトでOFFに設定されます。

ON

ORA、PLSおよびSP2エラーをデフォルト表のSPERRORLOGに書き込みます。

OFF

エラーを無効にします。

TABLE [*schema.*] *tablename*

デフォルトのSPERRORLOGのかわりにユーザー定義表を使用するように指定します。*schema.*を省略すると、現行のスキーマに表が作成されます。指定する表は存在している必要があり、それに対するアクセス権限が必要です。

指定した表が存在しないか、またはその表に対するアクセス権がない場合は、エラー・メッセージが表示されて、デフォルト表のSPERRORLOGが使用されます。

TRUNCATE

エラー・ログ表の既存の行すべてがクリアされて、現行セッションからのエラーの記録が始まります。

IDENTIFIER *identifier*

エラーを識別するためのユーザー定義文字列。特定のセッションから、または特定の問合せバージョンからエラーを識別するために使用できます。

ユーザー定義エラー・ログ表の作成

1つ以上のエラー・ログ表を作成して、デフォルトのかわりに使用できます。TABLE *schema.tablename* オプションを使用してユーザー定義エラー・ログ表を指定する前に、この表を作成して、表に対するアクセス権限を確保する必要があります。エラー・ログ表には、次の列定義があります。

表12-4 SQL*Plusエラー・ログの列定義

列	タイプ	説明
username	VARCHAR(256)	Oracle アカウント名。
timestamp	TIMESTAMP	エラー発生時間。
script	VARCHAR(1024)	発生元のスクリプトがある場合は、その名前。
identifier	VARCHAR(256)	ユーザー定義識別子文字列。

列	タイプ	説明
message	CLOB	ORA、PLA または SP2 のエラー・メッセージ。フィードバック・メッセージは含まれていません。たとえば、「PL/SQL ブロックが作成されました。」は記録されません。
statement	CLOB	エラーを発生させた文。

ユーザー定義エラー・ログ表の使用

ユーザー定義ログ表を使用するには、その表へのアクセス権限が必要であり、SET ERRORLOGGINGコマンドにTABLE *schema.tablename* オプションを付けて発行して、(必要に応じて)エラー・ログ表とスキーマを識別する必要があります。

エラー・ログ表の問合せ

エラー・ログ表に記録されたレコードを表示するには、任意の表で表示する必要がある列を問い合わせます。使用可能な列を[表 12-4](#)に示しています。

例

デフォルトのエラー・ログ表を使用して、次の内容を含むスクリプト *myfile.sql* による問合せエラーを記録する場合:

```
VARIABLE U REFCURSOR
BEGIN
  OPEN :U FOR SELECT * FROM DEPT;
END;
/

SHOW ERRORS PROCEDURE 'SSS'

SET GARBAGE

SELECT *
FROM
GARBAGE
;
```

次のように入力します。

```
SET ERRORLOGGING ON
@myfile
```

これによって、次の出力が表示されます。

```
open :u for select * from dept;
                                *
ERROR at line 2:
ORA-6550: line 2, column 29:
PLS-00201: ORA-00942: table or view does not exist
ORA-6550: line 2, column 16:
PL/SQL: SQL Statement ignored

ERROR:
ORA-00907: missing right parenthesis

SP2-0158: unknown SET option "garbage"

garbage
*
```

```
ERROR at line 3:
ORA-00942: table or view does not exist
```

デフォルトのエラー・ログ表SPERRORLOGに書き込まれたエラー・ログを表示するには、次のように入力します。

```
SELECT TIMESTAMP, USERNAME, SCRIPT, IDENTIFIER, STATEMENT, MESSAGE
FROM SPERRORLOG;
```

これによって、次の出力が表示されます。

TIMESTAMP	ユーザー名	スクリプト	識別子	文	メッセージ
Mon May 08 21:30:03 2006	SYSTEM	d:¥myfile.sql		open :u for select * from dept;	ORA-06550: 「行 2、列 27:」
Mon May 08 21:30:05 2006	SYSTEM	d:¥myfile.sql		open :u for select * from dept;	PL/SQL: ORA-00942: 「表またはビューが存在しま せん。」
Mon May 08 21:30:05 2006	SYSTEM	d:¥myfile.sql		open :u for select * from dept;	ORA-06550: 「行 2、列 13:」
Mon May 08 21:30:05 2006	SYSTEM	d:¥myfile.sql		open :u for select * from dept;	PL/SQL: 「SQL 文が無視 されました。」
Mon May 08 21:30:06 2006	SYSTEM	d:¥myfile.sql		show errors procedure "sss"	ORA-00907: 「右カッコが ありません。」
Mon May 08 21:30:09 2006	SYSTEM	d:¥myfile.sql		set garbage	SP2-0158: 「不明な SET オプションです "garbage"」
Mon May 08 21:30:10 2006	SYSTEM	d:¥myfile.sql		garbage	ORA-00942: 「表または ビューが存在しません。」

例2

ユーザー定義エラー・ログ表を使用して、次の内容を含むスクリプト*myfile.sql*による問合せエラーを記録する場合:

```
VARIABLE U REFCURSOR
BEGIN
  OPEN :U FOR SELECT * FROM DEPT;
END;
/
```

```
SHOW ERRORS PROCEDURE 'SSS'
```

```
SET GARBAGE
```

```
SELECT *  
FROM  
GARBAGE  
;
```

次のように入力します。

```
SET ERRORLOGGING ON  
@MYFILE
```

これによって、次の出力が表示されます。

```
open :u for select * from dept;  
                                *  
ERROR at line 2:  
ORA-6550: line 2, column 29:  
PLS-00201: ORA-00942: table or view does not exist  
ORA-6550: line 2, column 16:  
PL/SQL: SQL Statement ignored  
  
ERROR:  
ORA-00907: missing right parenthesis  
  
SP2-0158: unknown SET option "garbage"  
  
garbage  
*  
ERROR at line 3:  
ORA-00942: table or view does not exist
```

デフォルトのエラー・ログ表SPERRORLOGに書き込まれたエラー・ログを表示するには、次のように入力します。

```
SELECT TIMESTAMP, USERNAME, SCRIPT, IDENTIFIER, STATEMENT, MESSAGE  
FROM SPERRORLOG;
```

これによって、次の出力が表示されます。

TIMESTAMP	ユーザー名	スクリプト	識別子	文	メッセージ
Mon May 08 21:30:03 2006	SYSTEM	d:¥myfile.sql		open :u for select * from dept;	ORA-06550: 「行 2、列 27:」
Mon May 08 21:30:05 2006	SYSTEM	d:¥myfile.sql		open :u for select * from dept;	PL/SQL: ORA-00942: 「表またはビューが存在しませ ん。」
Mon May 08 21:30:05	SYSTEM	d:¥myfile.sql		open :u for select * from	ORA-06550: 「行 2、列 13:」

TIMESTAMP	ユーザー名	スクリプト	識別子	文	メッセージ
2006				dept;	
Mon May 08 21:30:05 2006	SYSTEM	d:¥myfile.sql		open :u for select * from dept;	PL/SQL: 「SQL 文が無視されました。」
Mon May 08 21:30:06 2006	SYSTEM	d:¥myfile.sql		show errors procedure "sss"	ORA-00907: 「右カッコがありません。」
Mon May 08 21:30:09 2006	SYSTEM	d:¥myfile.sql		set garbage	SP2-0158: 「不明な SET オプションです "garbage"」
Mon May 08 21:30:10 2006	SYSTEM	d:¥myfile.sql		garbage	ORA-00942: 「表またはビューが存在しません。」

例3

デフォルト以外のエラー・ログ表を使用するには:

- 使用するエラー・ログ表を作成します。
- SET ERRORLOGGING ONコマンドのTABLEオプションで、使用する表を指定します。

エラー・ログ表の列定義は[表12-4](#)で定義されている必要があります。

Johnがエラー・ログ表`john_sperrorlog`を使用する場合を想定します。Johnは、次のSQL文を実行して新しいエラー・ログ表を作成します。

```
DROP TABLE john_sperrorlog;
CREATE TABLE john_sperrorlog (username VARCHAR(256),
timestamp TIMESTAMP,
script VARCHAR(1024),
identifier VARCHAR(256),
message CLOB,
statement CLOB);
```

ユーザー・コメント`giridhar123`から前の例のコミットを削除しました(14Feb08)

次にJohnは、次のSETコマンドを発行して、新しく作成したエラー・ログ表を使用したエラー・ロギングを有効にします。

```
SET ERRORLOGGING ON TABLE john_sperrorlog
```

これで、Johnのエラー・ロギングはすべて、デフォルトのエラー・ログ表SPERRORLOGではなく、`john_sperrorlog`に記録されます。

エラー・ログ表へのアクセス権限は、任意のユーザー表へのアクセス権限と同様に処理されます。

12.41.20 SET ESC[APE] {¥ | c | ON | OFF}

エスケープ文字として使用する文字を定義します。

OFFを指定すると、エスケープ文字の定義が解除されます。ONを指定すると、エスケープ文字が使用可能になります。ONを指定すると、cの値がデフォルトの「¥」に戻ります。

置換文字(SET DEFINEで指定された文字)の前にエスケープ文字を指定することによって、SQL*Plusにその置換文字を変数置換のための要求としてでなく、通常の文字として処理するように指示できます。

例

感嘆符(!)をエスケープ文字として定義し、次のように入力したとします。

```
SET ESCAPE !  
ACCEPT v1 PROMPT 'Enter !&1:'
```

次のプロンプトが表示されます。

```
Enter &1:
```

エスケープ文字をデフォルト値の¥(円記号)に戻すには、次のように入力します。

```
SET ESCAPE ON
```

12.41.21 SET ESCCHAR

構文

```
SET ESCCHAR {@ | ? | % | OFF}
```

SPOOL、START、@、RUNおよびEDITコマンドのファイル名で使用了した場合に、エスケープして解釈しない文字を指定します。これらの特殊文字は、次のように変換されます。

- ファイル名の@はOracle SIDに変換されます
- ?はUNIXでOracleホームに変換されます
- %はWindowsでOracleホームに変換されます

これらの文字をファイル名で使用しないでください。これらの文字を使用するレガシー・ファイルがある場合は、SET ESCCHAR コマンドをGLoginファイルに含めてサイト全体に実装すると便利です。

エスケープしない場合、@、?および%の文字は解釈の際に重要となり、SPOOL、START、@、RUNおよびEDITコマンドでエラーの原因となります。

SET ESCCHARは、デフォルトでOFFに設定されます。

ノート:



Oracle Database リリース 19c バージョン 19.3 以降では、\$文字を含むファイル名は Windows で実行されなくなります。

12.41.22 SET EXITC[OMMIT] {ON | OFF}

デフォルトのEXIT動作をCOMMITにするかROLLBACKにするかを指定します。

デフォルト設定はONです。つまり、作業はコミットするかどうかの意図にかかわらず、終了時にコミットされます。EXITCOMMIT OFFを設定すると、作業は終了時にロールバックされます。

[表12-5](#)に、SETコマンド(AUTOCOMMITとEXITCOMMIT)およびEXITコマンドの組合せに関連付けられた終了アクションを示します。

表12-5 終了動作: AUTOCOMMIT、EXITCOMMIT、EXIT

AUTOCOMMIT	EXITCOMMIT	EXIT	終了動作
ON	ON	-	COMMIT
ON	OFF	-	COMMIT
OFF	ON	-	COMMIT
OFF	OFF	-	ROLLBACK
ON	ON	COMMIT	COMMIT
ON	ON	ROLLBACK	COMMIT
ON	OFF	COMMIT	COMMIT
ON	OFF	ROLLBACK	COMMIT
OFF	ON	COMMIT	COMMIT
OFF	ON	ROLLBACK	ROLLBACK
OFF	OFF	COMMIT	COMMIT
OFF	OFF	ROLLBACK	ROLLBACK

12.41.23 SET FEED[BACK] {6 | *n* | ON | OFF | ONLY} [SQL_ID]

スクリプトが*n*個以上のレコードを選択した場合に、スクリプトから戻されるレコード数を表示します。

ONまたはOFFによって、この表示をオンまたはオフにできます。フィードバックをONに設定すると、*n*が1に設定されます。フィードバックを0に設定することは、OFFに設定するのと同じです。データが表示されている間、フィードバック・メッセージは表示されません。

また、SET FEEDBACK OFFによって、SQLまたはPL/SQL文が正常に実行された後に表示される確認メッセージ(「表が作成されました。」、「PL/SQLプロシージャが正常に完了しました。」など)を非表示にできます。

ONLYは、データを表示せずに問合せで選択された行数を返します。

SQL_IDは、実行されたSQL文またはPL/SQL文のsql_idを返します。sql_idが事前定義済変数_SQL_IDに割り当てられます。この事前定義済変数を使用して、実行されたSQL文をデバッグできます。変数は、_USERおよび_DATEなど、他の事前定義済の変数のように使用できます。

```
SQL> SET FEEDBACK ON SQL_ID
SQL> SELECT * FROM DUAL;

D
-
X

1 row selected.

SQL_ID: a5ks9fhw2v9s1
SQL> COLUMN sql_text FORMAT a50
SQL> SELECT sql_text FROM v$sql WHERE sql_id = '&_sql_id';

SQL_TEXT
-----
SELECT * FROM DUAL

1 row selected.

SQL_ID: cf9bgxbfytv5b
```

SQL_IDオプションが指定され、フィードバックがONになっている場合、フィードバック・メッセージとともにSQL IDが表示されます。フィードバックがOFFの場合、SQL IDのみが表示されます。

例

SET FEEDBACK ONLYを有効にするには、次のように入力します。

```
SQL> SET FEEDBACK ONLY
SQL> SHOW FEEDBACK
feedback ONLY
SQL> SELECT * FROM EMP;

14 rows selected.
```

SET_FEEDBACKおよびSQL_IDを有効にするには、次のように入力します

```
SQL> SET FEEDBACK ON SQL_ID
SQL> SELECT * FROM DEPT;

DEPTNO DNAME LOC
-----
10 ACCOUNTING NEW YORK
20 RESEARCH DALLAS
30 SALES CHICAGO
40 OPERATIONS BOSTON

4 rows selected.

SQL_ID: 3154rqzb8xudy
SQL> CONNECT SYSTEM
```

```

ENTER PASSWORD:
Connected.
SQL> COLUMN sql_text FORMAT a50
SQL> SELECT sql_text FROM v$sql WHERE sql_id = '&_sql_id';

SQL_TEXT
-----
SELECT * FROM DEPT

1 row selected.
SQL_ID: 81a5n8q6g2vvr

```

12.41.24 SET FLAGGER {OFF | ENTRY | INTERMED[IATE] | FULL}

SQL文がANSI/ISO SQL92規格に準拠しているかどうかを調べます。

規格外の構造体が見つかった場合、Oracle Databaseサーバーは、その構造体にエラーのフラグを付け、違反のある構文を表示します。これは、SQL言語のALTER SESSION SET FLAGGERコマンドと同じです。

データベースに接続されていない場合でも、SET FLAGGERを実行できます。FIPSフラグ付けは、SET FLAGGER OFF(またはALTER SESSION SET FLAGGER = OFF)コマンドが成功するか、またはSQL*Plusを終了するまでは、SQL*Plusセッション全体にわたって有効な状態のままになります。

FIPSフラグ付けを使用可能にすると、CONNECT、DISCONNECTおよびALTER SESSION SET FLAGGERコマンドが成功した場合でも、SQL*Plusはこれらのコマンドに対してアラートを表示します。

12.41.25 SET FLU[SH] {ON | OFF}

出力をユーザーの表示デバイスに、いつ送るかを制御します。OFFを指定すると、オペレーティング・システムは出力をバッファに格納できます。ONを指定すると、バッファリングは使用禁止になります。FLUSHは表示出力にのみ影響し、スプール出力には影響を与えません。

OFFは、スクリプトを非対話形式で実行する場合(スクリプトの実行が完了するまで出力またはプロンプト(あるいはその両方)を表示させる必要がない場合)のみ使用します。FLUSH OFFを使用すると、プログラムI/Oの量が減り、パフォーマンスが向上する場合があります。

12.41.26 SET HEA[DING] {ON | OFF}

レポートへの列ヘッダーの出力を制御します。

ONを指定すると、列ヘッダーがレポートに出力されます。OFFを指定すると、列ヘッダーが出力されなくなります。

SET HEADING OFFコマンドは、表示される列幅に影響しません。そのコマンドは、列ヘッダーを出力しないようにするのみです。

例

レポート内で列ヘッダーを非表示にするには、次のように入力します。

```
SET HEADING OFF
```

次のようなSQL SELECTコマンドを実行したとします。

```
SELECT LAST_NAME, SALARY
```



```
FROM EMP_DETAILS_VIEW
WHERE JOB_ID = 'AC_MGR';
```

出力結果は次のようになります。

```
Higgins 12000
```

列ヘッダーの表示を戻すには、次のように入力します。

```
SET HEADING ON
```

12.41.27 SET HEADS[EP] { | c | ON | OFF }

列ヘッダーの行ブレイクとして使用する文字を定義します。

ヘッダーのセパレータ文字に、英数字または空白は使用できません。ヘッダーのセパレータ文字を、COLUMNコマンドの中で、および旧形式のBTITLEおよびTTITLEの中で使用して、列ヘッダーまたはタイトルを複数の行に分割できます。ONまたはOFFによって、このヘッダー分割をオンまたはオフにできます。ヘッダー分割をOFFにすると、SQL*Plusは、ヘッダーのセパレータ文字を他の文字と同様に出力します。ONを指定すると、cの値がデフォルトの「|」に戻ります。

12.41.28 SET HIST[ORY] { ON | OFF | n }

コマンドの履歴を有効または無効にします。有効にした場合、SQL*Plus文、SQL文およびPL/SQL文はコマンド履歴リストに格納されます。[HISTORY](#)コマンドを使用することで、SQL*Plus文、SQL文およびPL/SQL文をリコールできます。

例

コマンド履歴を有効にしてリストに200件のエンTRIESを格納するには、次のように入力します。

```
SET HIST[ORY] 200
```

nの値を指定しない場合、デフォルトは100となります。

ノート:



PL/SQL ブロックなどの複数行のENTRIESは、コマンド履歴リストでは1つのENTRIESと見なされます。コマンド履歴には最大 100000 個のコマンドを格納できます。すでに有効となっているコマンド履歴を有効にしようとした場合、古いコマンド履歴がクリアされます。

12.41.29 SET INSTANCE [*instance_path* | LOCAL]

セッションのデフォルトのインスタンスを、指定したインスタンス・パスに変更します。

SET INSTANCEコマンドを使用して、データベースへは接続しません。デフォルト・インスタンスは、インスタンスが何も指定されていないときにコマンドに対して使用されます。SET INSTANCEを最初に使用する前の一部のコマンドは、デフォルト・インスタンスと通信します。

ご使用のオペレーティング・システムでデフォルト値のインスタンスをリセットするには、*instance_path*を指定しないでSET INSTANCEを入力するか、またはSET INSTANCE LOCALを入力します。

インスタンスの変更は、いずれのインスタンスにも現在接続されていない場合にのみ可能です。つまり、まず現行のインスタンスから切断されていることを確認してからインスタンスを設定または変更し、新しい設定を有効にするにはインスタンスに再接続する

必要があります。

このコマンドは、Oracle Netが実行されているときにのみ実行されます。指定するインスタンス・パスとして、一部の有効なOracle Net接続識別子を使用できます。ご使用のオペレーティング・システムでOracle Net接続識別子を指定する方法の詳細は、オペレーティング・システム固有のOracle Databaseのマニュアルを参照してください。インスタンス・パスの最大長は、64文字です。

例

デフォルト・インスタンスに「PROD1」を設定するには、次のように入力します。

```
DISCONNECT
SET INSTANCE PROD1
```

インスタンスをローカルのデフォルトに戻すには、次のように入力します。

```
SET INSTANCE local
```

インスタンスを変更するには、接続しているインスタンスから切断する必要があります。

12.41.30 SET LIN[ESIZE] {80 | *n* | WINDOW}

SQL*Plusが新しい行を開始する前に、1行に表示する文字の合計数を設定します。

TTITLE、BTITLE、REPHEADERおよびREPFOOTERの場合の、中央揃えおよび右揃えのテキストの位置も、この値によって制御されます。行サイズの設定の変更は、DESCRIBEコマンドによる出力で、テキストの折返しに影響する場合があります。DESCRIBEコマンドで出力される列には、通常、行サイズの割合が割り当てられます。行サイズを増減することによって、表示上、テキストの折返しが意図しない位置で行われる場合があります。LINESIZEは、値1から最大値(システムによって異なります)の範囲内で定義できます。

WINDOWによって、画面の幅と高さに応じた書式設定された出力にあわせて行サイズとページサイズが調整されます。出力が画面の幅を超えている場合、出力は対応して折り返されます。

画面サイズがSET LINESIZE *n*コマンドを使用して手動で設定されている場合、後続の出力は新しい行サイズにあわせて表示されます。

ノート:



SET LINESIZE WINDOW コマンドは、スクリプトに無視され、出力は画面のサイズに応じては表示されません。

例

出力の行サイズを20に設定するには、次のように入力します

```
SQL> SET LINESIZE 20
SQL> SHOW LINESIZE
linesize 20
```

画面を手動でサイズ変更した後に出力表示を動的に変更するには、次のように入力します

```
SQL> SET LINESIZE Window
SQL> SHOW LINESIZE
linesize 160 WINDOW
```

12.41.31 SET LOBOF[FSET] {1 | n}

BLOB、BFILE、CLOBおよびNCLOBデータを取得し、表示する開始位置を設定します。

例

CLOB列のデータを取得するための開始位置を22桁目に設定するには、次のように入力します。

```
SET LOBOFFSET 22
```

CLOBデータは、画面上で折り返されます。SQL*Plusは、23番目の文字になるまで切捨てを行いません。

12.41.32 SET LOBPREFETCH {0 | n}

SQL*Plusがデータベースから一度にプリフェッチするLOBデータの量(バイト数)を設定します。

例

プリフェッチされるLOBデータの量を8000バイトに設定するには、次のように入力します。

```
SET LOBPREFETCH 8000
```

*n*の値を指定しない場合、デフォルトは0となります。つまり、LOBデータのプリフェッチはオフとなります。

ノート:



指定できる最大値は 2147483648 バイト(2 GB)です。oraaccess.xml ファイル内の設定により、SQL*Plus における SET LOBPREFETCH の設定をオーバーライドできます。oraaccess.xml の詳細は、*Oracle Call Interface プログラマーズ・ガイド*を参照してください。

SQL*Plusがデータベースから一度にプリフェッチするLOBデータの量について現在の設定を表示するには、次のように入力します。

```
SHOW LOBPREF[ETCH]
```

12.41.33 SET LOGSOURCE [*pathname*]

リカバリ中に取得されるアーカイブ・ログの位置を指定します。

デフォルト値は、Oracle Database初期化パラメータ・ファイルinit.oraのLOG_ARCHIVE_DEST初期化パラメータで設定されます。*pathname*を指定せずにSET LOGSOURCEコマンドを使用すると、デフォルト位置がリストアップされます。

例

リカバリするログ・ファイルのデフォルト位置を、/usr/oracle10/dbs/archディレクトリに設定するには、次のように入力します。

```
SET LOGSOURCE "/usr/oracle10/dbs/arch"  
RECOVER DATABASE
```

12.41.34 SET LONG {80 | n}

BLOB、BFILE、CLOB、LONG、NCLOBおよびXMLType値を表示するため、およびLONG値をコピーするための最大幅をバイト単位で設定します。

SET LONGCHUNKSIZEコマンドの値に関係なく、LONG列の問合せには、SET LONGで指定されたデータの量を格納するための十分なローカル・メモリーが必要です。この要件は、LOBの問合せには適用されません。

LONG列を含む表は、作成しないことをお勧めします。LONG列は、下位互換性のためにのみサポートされています。かわりにLOB列(BLOB、BFILE、CLOB、NCLOB)を使用してください。LOB列は、LONG列より制限が少なく、現在でも機能が拡張されています。

n の最大値は2,000,000,000バイトです。SET LONGコマンドの実行に必要なメモリーがマシンで使用可能かどうかを、次のようにチェックすることが重要です。

```
SET LONG 2000000000
```

この例の場合は、マシンで使用可能なRAM(ランダム・アクセス・メモリー)が2GBを超えていることが必要です。

例

LONG値を表示およびコピーするためにフェッチする最大バイト数を500に設定するには、次のように入力します。

```
SET LONG 500
```

LONGデータは、画面上で折り返されます。SQL*Plusは、501バイト目から切捨てを行います。LONGのデフォルト値は80バイトです。

12.41.35 SET LONGC[HUNKSIZE] {80 | n }

SQL*PlusがBLOB、BFILE、CLOB、LONG、NCLOBまたはXMLType値を取得するときに使用する増分値のサイズ(バイト単位)を設定します。

LONGCHUNKSIZEは、CLOB、NCLOBなどのオブジェクト・リレーショナル問合せには使用されません。

例

SQL*PlusがLONG値を取得するときに使用する増分値のサイズを100バイトに設定するには、次のように入力します。

```
SET LONGCHUNKSIZE 100
```

LONGデータは、値全体が取得されるまで、またはSET LONGの値に到達するまで(いずれか早い方)、100バイトの増分値で取得されます。

12.41.36 SET MARK[UP]

構文

```
SET MARK[UP] markup_option
```

*markup_option*の構成は、次のとおりです。

- *csv_option*
- *html_option*

*csv_option*の構文は、次のとおりです。

```
CSV {ON|OFF} [DELIMI[TER] character] [QUOTE {ON|OFF}]
```

*html_option*の構文は、次のとおりです。

```
HTML {ON|OFF} [HEAD text] [BODY text] [TABLE text] [ENTMAP {ON|OFF}] [SPOOL {ON|OFF}] [PRE[FORMAT] {ON|OFF}]
```

csv_option

レポートをCSV形式で出力します。

有効にするには、動的レポート出力の値を変更するSET MARKUPコマンドを、問合せ出力を生成する文の前に発行する必要があります。問合せ出力を生成する最初の文によって、DELIMITERおよびQUOTEの設定を反映したCSVデータの出力がトリガーされます。

CSVは、生成される出力タイプをCSVに指定するSET MARKUPの必須引数です。CSV引数ONおよびOFFは、CSV出力を生成するかどうかを指定します(オプション)。デフォルトはOFFです。

CSV出力のONおよびOFFは、セッション中、必要に応じて切り替えることができます。

DELIMI[TER] *character*

DELIMI[TER] *character*オプションにより、列セパレータ文字を指定できます。

QUOTE {ON|OFF}

QUOTE {ON|OFF}オプションにより、テキストに対する引用符の付加のオンとオフを切り替えることができます。デフォルトはOFFです。

QUOTE ONにすると、すべてのテキストに引用符が付加された状態でCSV出力が生成されます。テキストに埋め込まれた二重引用符(" ")はエスケープされます。

引用符が付加されたテキストのONおよびOFFは、セッション中、必要に応じて切り替えることができます。

SET MARKUP CSVが有効な場合にサポートされているコマンド

SET MARKUP CSVが有効な場合、次の[COLUMN](#)コマンドは有効なままとなります。

- COLUMN FORMAT
- COLUMN HEADING
- COLUMN NULL

SET MARKUP CSVが有効な場合にサポートされていないコマンド

SET MARKUP CSVが有効な場合、次のSQL*Plusコマンドは出力において有効となりません。

- [BREAK](#)
- [BTITLE](#)
- [COMPUTE](#)
- [REPFOOTER](#)
- [REPHEADER](#)

SET MARKUP CSVが有効な場合、次のSETコマンドは出力において有効となりません。

- [SET COLSEP](#)
- [SET HEADSEP](#)
- [SET LINESIZE](#)
- [SET NEWPAGE](#)
- [SET PAGESIZE](#)

- [SET PAUSE](#)
- [SET RECSEP](#)
- [SET SHIFTINOUT](#)
- [SET TAB](#)
- [SET TRIMOUT](#)
- [SET TRIMSPool](#)
- [SET UNDERLINE](#)
- [SET WRAP](#)

SET MARKUP CSVが有効な場合、次の[COLUMN](#)コマンドは出力において有効となりません。

- COLUMN ENTMAP
- COLUMN FOLD_AFTER
- COLUMN FOLD_BEFORE
- COLUMN JUSTIFY
- COLUMN NEWLINE
- COLUMN NEW_VALUE
- COLUMN NOPRINT
- COLUMN OLD_VALUE
- COLUMN WRAP

MARKUPオプションの状態を表示するには、SHOW MARKUPコマンドを使用します。

例

次の例は、SET MARKUP CSVが有効な場合の出力を示しています。

```
SQL> SET MARKUP CSV ON
SQL> SELECT * FROM EMP;

"EMPNO","ENAME","JOB","MGR","HIREDATE","SAL","COMM","DEPTNO"
7369,"SMITH","CLERK",7902,"17-DEC-80",800,,20
7499,"ALLEN","SALESMAN",7698,"20-FEB-81",1600,300,30
7521,"WARD","SALESMAN",7698,"22-FEB-81",1250,500,30
7566,"JONES","MANAGER",7839,"02-APR-81",2975,,20
7654,"MARTIN","SALESMAN",7698,"28-SEP-81",1250,1400,30
7698,"BLAKE","MANAGER",7839,"01-MAY-81",2850,,30
7782,"CLARK","MANAGER",7839,"09-JUN-81",2450,,10
7788,"SCOTT","ANALYST",7566,"19-APR-87",3000,,20
7839,"KING","PRESIDENT",,"17-NOV-81",5000,,10
7844,"TURNER","SALESMAN",7698,"08-SEP-81",1500,0,30
7876,"ADAMS","CLERK",7788,"23-MAY-87",1100,,20
7900,"JAMES","CLERK",7698,"03-DEC-81",950,,30
7902,"FORD","ANALYST",7566,"03-DEC-81",3000,,20
7934,"MILLER","CLERK",7782,"23-JAN-82",1300,,10

14 rows selected.
```

次の例は、テキスト文字列に引用符を付加せずに、データベースの従業員表からすべてのレコードを抽出する方法を示していま

す。

```
SQL> SET MARKUP CSV ON QUOTE OFF
SQL> SELECT * FROM EMP;
```

```
EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO
7369, SMITH, CLERK, 7902, 17-DEC-80, 800, , 20
7499, ALLEN, SALESMAN, 7698, 20-FEB-81, 1600, 300, 30
7521, WARD, SALESMAN, 7698, 22-FEB-81, 1250, 500, 30
7566, JONES, MANAGER, 7839, 02-APR-81, 2975, , 20
7654, MARTIN, SALESMAN, 7698, 28-SEP-81, 1250, 1400, 30
7698, BLAKE, MANAGER, 7839, 01-MAY-81, 2850, , 30
7782, CLARK, MANAGER, 7839, 09-JUN-81, 2450, , 10
7788, SCOTT, ANALYST, 7566, 19-APR-87, 3000, , 20
7839, KING, PRESIDENT, , 17-NOV-81, 5000, , 10
7844, TURNER, SALESMAN, 7698, 08-SEP-81, 1500, 0, 30
7876, ADAMS, CLERK, 7788, 23-MAY-87, 1100, , 20
7900, JAMES, CLERK, 7698, 03-DEC-81, 950, , 30
7902, FORD, ANALYST, 7566, 03-DEC-81, 3000, , 20
7934, MILLER, CLERK, 7782, 23-JAN-82, 1300, , 10
```

14 rows selected.

次の例は、パイプ(|)文字がデリミタとして指定された出力を示しています。

```
SQL> SET MARKUP CSV ON DELIMITER |
SQL> SELECT * FROM EMP;
```

```
EMPNO|ENAME|JOB|MGR|HIREDATE|SAL|COMM|DEPTNO
7369|SMITH|CLERK|7902|17-DEC-80|800||20
7499|ALLEN|SALESMAN|7698|20-FEB-81|1600|300|30
7521|WARD|SALESMAN|7698|22-FEB-81|1250|500|30
7566|JONES|MANAGER|7839|02-APR-81|2975||20
7654|MARTIN|SALESMAN|7698|28-SEP-81|1250|1400|30
7698|BLAKE|MANAGER|7839|01-MAY-81|2850||30
7782|CLARK|MANAGER|7839|09-JUN-81|2450||10
7788|SCOTT|ANALYST|7566|19-APR-87|3000||20
7839|KING|PRESIDENT||17-NOV-81|5000||10
7844|TURNER|SALESMAN|7698|08-SEP-81|1500|0|30
7876|ADAMS|CLERK|7788|23-MAY-87|1100||20
7900|JAMES|CLERK|7698|03-DEC-81|950||30
7902|FORD|ANALYST|7566|03-DEC-81|3000||20
7934|MILLER|CLERK|7782|23-JAN-82|1300||10
```

14 rows selected.

html_option

HTMLマークアップ・テキストを出力します。

有効にするには、動的レポート出力の値を変更するSET MARKUPコマンドを、問合せ出力を生成する文の前に指定する必要があります。問合せ出力を生成する最初の文によって、HEAD設定、TABLE設定などのSET MARKUPによって影響を受ける情報の出力がトリガーされます。後続のSET MARKUPコマンドは、すでにレポートに送信されている情報には影響を与えません。

SET MARKUPは、SQL*Plusの出力をHTML形式にエンコードすることを指定するのみです。スプール・ファイルの作成、ファイル名の指定およびファイルへのHTML出力の書き込みを開始するには、SET MARKUP HTML ON SPOOL ONおよびSQL*PlusのSPOOLコマンドを使用する必要があります。SET MARKUPのオプションと動作は、SQLPLUS -MARKUPと同じです。

詳細は、「[MARKUPオプション](#)」を参照してください。使用例については、[SQL*Plusからのレポートの生成](#)を参照してください。

MARKUPオプションの状態を表示するには、SHOW MARKUPコマンドを使用します。

例

次に、SET MARKUP HTMLコマンドを使用し、HTMLマークアップ・テキストを指定したファイルにスプールするスクリプトを示します。

ノート:



SET MARKUP の例では、コマンドは、行継続文字「-」および空白を使用して読みやすいようにレイアウトされています。通常、コマンド・オプションは連結して入力されます。

任意のテキスト・エディタを使用して、レポート用のHTMLオプションおよび問合せの設定に必要なコマンドを入力します。

```
SET MARKUP HTML ON SPOOL ON HEAD "<TITLE>SQL*Plus Report</title> -
<STYLE TYPE='TEXT/CSS'><!--BODY {background: ffffc6} --></STYLE>"
SET ECHO OFF
SPOOL employee.htm
SELECT FIRST_NAME, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000;
SPOOL OFF
SET MARKUP HTML OFF
SET ECHO ON
```

このスクリプトにはSQL*Plusコマンドが含まれるため、バッファから/(スラッシュ)を使用すると正常に実行されません。テキスト・エディタでスクリプトを保存し、STARTを使用して次のように実行します。

```
START employee.sql
```

HTMLスプール・ファイルemployee.htmの書込みと同様に、SET TERMOUTがONにデフォルト設定されているため、出力は画面にも表示されます。Webブラウザにスプール・ファイルemployee.htmを表示できます。次のように表示されるはずです。

FIRST_NAME	LAST_NAME	SALARY
Steven	King	24000
Neena	Kochhar	17000
Lex	De Haan	17000
Michael	Hartstein	13000
John	Russell	14000
Karen	Partners	13500

6 rows selected.

12.41.37 SET NEWP[AGE] {1 | n | NONE}

各ページの最上部から上部タイトルまでの間に入れる空白行の数を設定します。値0を指定すると、(最初のページも含めて)各ページの開始位置に改ページ文字が設定され、ほとんどの端末で画面がクリアされます。NEWPAGEをNONEに設定した場合、SQL*Plusはレポートのページ間で空白行または改ページ文字を出力しません。

12.41.38 SET NULL テキスト

SQL SELECTコマンドの結果にNULL値がある場合に表示されるテキストを設定します。

指定した列のNULL変数の設定を上書きするには、COLUMNコマンドのNULL句を使用します。NULLは、デフォルトでは空白("")で出力されます。

12.41.39 SET NUMF[ORMAT] 形式

数値を表示するためのデフォルトの書式を設定します。数値書式の詳細は、「[COLUMN](#)」コマンドのFORMAT句を参照してください。*format*に、数値書式を入力します。SET NUMWIDTHで指定されたデフォルトのフィールド幅と書式設定モデルを使用するには、次のように入力します。

```
SET NUMFORMAT ""
```

12.41.40 SET NUM[WIDTH] {10 | *n*}

数値を表示するためのデフォルトの幅を設定します。数値書式の詳細は、「[COLUMN](#)」コマンドのFORMAT句を参照してください。

COLUMN FORMAT設定は、SET NUMFORMAT設定より優先されます。SET NUMFORMAT設定は、SET NUMWIDTH設定より優先されます。

12.41.41 SET PAGES[IZE] {14 | *n*}

各出力ページの行数を設定します。PAGESIZEを0に設定すると、ヘッダー、ページ・ブレイク、タイトル、初期空白行およびその他の書式設定情報をすべて非表示にできます。

12.41.42 SET PAU[SE] {ON | OFF | テキスト}

レポートの実行中に端末のスクロール制御を実現します。SQL*Plusが一時停止するたびにテキストを表示する場合は、SET PAUSE *text*を指定した後で、SET PAUSE ONを指定します。

SET PAUSE ONは各PAGESIZEで指定した行数の各レポートの出力開始時に出力を一時停止します。以降の出力を表示するには、[Return]を押します。SET PAUSE *text*には、SQL*Plusが一時停止するたびに表示するテキストを指定します。テキストに複数の語を含める場合は、*text*を引用符で囲む必要があります。

端末固有のエスケープ・シーケンスをPAUSEコマンドの中に入れることができます。このようなシーケンスを使用すると、反転表示などの効果を持つメッセージを端末上に作成できます。

12.41.43 SET RECSEP {WR[APPED] | EA[CH] | OFF}

RECSEPには、SQL*Plusがどこでレコードを分離するかを指定します。

たとえば、RECSEPをWRAPPEDに設定すると、SQL*Plusは折り返された行の直後にのみレコード・セパレータを出力します。RECSEPをEACHに設定すると、SQL*Plusは各行の後にレコード・セパレータを出力します。RECSEPにOFFを設定すると、SQL*Plusはレコード・セパレータを出力しません。

12.41.44 SET RECSEPCHAR { | *c*}

表示または出力時にレコードを分離する文字を定義します。

レコード・セパレータは、LINESIZEに指定した回数だけ繰り返される1行分のRECSEPCHAR(レコード・セパレータ文字)です。デフォルトは単一の空白です。

12.41.45 SET ROWLIMIT {*n* | OFF}

問合せに対して表示する行数の制限を設定します。デフォルトでは、ROWLIMITはOFFになっています。

*n*には1 - 2,000,000,000の任意の数値を指定できます。入力された値がこの範囲内でない場合には、エラーが表示されます。

OFFを指定すると、出力のすべての行が表示されます。

SET ROWLIMITは、SET FEEDBACKコマンドで設定した値を無視します。

SET ROWLIMITコマンドは、多くの行数を返す問合せの場合に便利です。ユーザーは問合せを変更せずに表示する行数を制限できます。

SET ROWLIMITコマンドの目的は、問合せのパフォーマンスを向上させることではなく、表示する行を制限することです。データベースからフェッチされる行数は、SET ARRAYSIZE値によって決定されます。このため、サーバーはROWLIMITで設定された値を超える行を返す場合があります。制限を超えた行が返されると、問合せは終了し、サーバーからそれ以上データをフェッチしません。たとえば、表を問い合わせるSQL文に100行が含まれているが、ROWLIMITは10に設定されており、ARRAYSIZEは15に設定されているとします。問合せが実行されると、15行が最初にデータベースからフェッチされます。返された行がROWLIMIT設定に到達しているため、これ以上フェッチせずに問合せは終了します。ただし、ROWLIMITが20に設定されている場合、ROWLIMIT設定に到達させるために次の15行をフェッチする2回目のラウンドトリップが必要となります。

例

表示される行数を2に設定するには、次のように入力します。

```
SQL> SET ROWLIMIT 2
SQL> SHOW ROWLIMIT
rowlimit 2

SQL> SELECT * FROM DEPT;

DEPTNO  DNAME
-----  -
10      ACCOUNTING
20      RESEARCH

2 rows selected. (rowlimit reached)
```

例

デフォルトでは、FEEDBACKメッセージは10行が表示された後にのみ表示されますが、ROWLIMITが2に設定されている場合、FEEDBACKが10に設定されていても、2行のFEEDBACKメッセージが表示されます。

```
SQL> SET FEEDBACK 10
SQL> SET ROWLIMIT 2
SQL> SELECT * from EMP;

2 rows selected. (rowlimit reached)
```

12.41.46 SET ROWPREFETCH {1 | *n*}

SQL*Plusがデータベースから一度にプリフェッチする行数を設定します。

デフォルト値は1です。

例

プリフェッチされる行数を200に設定するには、次のように入力します。

```
SET ROWPREFETCH 200
```

*n*の値を指定しない場合、デフォルトは1行となります。つまり、行のプリフェッチはオフとなります。

ノート:



プリフェッチされた行に含まれるデータの量が、最大値 2147483648 バイト(2 GB)を超えないようにしてください。oraaccess.xml ファイル内の<prefetch>の設定により、SQL*Plus における SET ROWPREFETCH の設定をオーバーライドできます。oraaccess.xml の詳細は、*Oracle Call Interface* プログラマーズ・ガイドを参照してください。

SQL*Plusがデータベースから一度にプリフェッチする行の数について現在の設定を表示するには、次のように入力します。

```
SHOW ROWPREF[ETCH]
```

12.41.47 SET SECUREDCOL {OFF | ON} [UNAUTH[ORIZED] テキスト] [UNK[NOWN] テキスト]

SQLPLUS出力で、セキュアな列値を、列を表示する権限のないユーザーおよびセキュリティが不明の列に対して表示する方法を設定します。デフォルト・テキストを選択するか、表示するテキストを指定できます。デフォルトはOFFです。

列レベルのセキュリティが有効で、SET SECUREDCOLがONに設定されている場合、セキュアな列または不明なセキュリティ・レベルの列に対するSQLPLUSからの出力は、カスタマイズしたテキストまたはデフォルトのインジケータで置き換えられます。これは、スカラー・データ型にのみ適用されます。複合オブジェクト・データ出力には影響ありません。

ONを指定すると、認可のないユーザーの列値のかわりに*****、セキュリティ・レベルが不明な列値のかわりに?????が表示されます。定義されている列長または現在のCOLUMNコマンドで定義した列長に合せて*および?が埋め込まれます。

OFFを指定すると、認可のないユーザーの列値およびセキュリティ・レベルが不明な列値のかわりにNULL値が表示されます。

UNAUTH[ORIZED]テキストでは、認可のないユーザーに対してセキュアな列に表示するテキストを指定できます。このテキストは、デフォルトの*****のかわりに表示されます。

列長以内または30文字以内の任意の英数字テキストを指定できます。これより長いテキストは切り捨てられます。空白を含むテキストは、引用符で囲む必要があります。

UNK[NOWN]テキストでは、セキュリティ・レベルが不明の列に表示するテキストを指定できます。このテキストは、デフォルトの??????のかわりに表示されます。

列長以内または30文字以内の任意の英数字テキストを指定できます。これより長いテキストは切り捨てられます。空白を含むテキストは、引用符で囲む必要があります。

例

```
SET SECUREDCOL ON
SELECT empno, ename, sal FROM emp ORDER BY deptno;

EMPNO ENAME  DEPTNO SAL
```

```

7539 KING    10    *****
7369 SMITH   20      800
7566 JONES   20    2975
7788 SCOTT   20    3000
7521 WARD    30    *****
7499 ALLEN   30    *****

```

```

SET SECUREDCOL ON UNAUTH notallow
SELECT empno, ename, sal FROM emp ORDER BY deptno;

```

```

EMPNO ENAME  DEPTNO SAL
7539 KING    10    notallow
7369 SMITH   20      800
7566 JONES   20    2975
7788 SCOTT   20    3000
7521 WARD    30    notallow
7499 ALLEN   30    notallow

```

12.41.48 SET SERVEROUT[PUT] {ON | OFF} [SIZE {*n* | UNL[IMITED]}] [FOR[MAT] {WRA[PPED] | WOR[D_WAPPED] | TRU[NCATED]}]

ストアド・プロシージャの出力(DBMS_OUTPUT.PUT_LINE)またはPL/SQLブロックの出力を、SQL*Plusで表示するかどうかを制御します。DBMS_OUTPUTによる行の長さの制限は32767バイトです。

OFFを指定すると、DBMS_OUTPUT.PUT_LINEの出力が非表示になります。ONを指定すると、出力が表示されます。

ONを指定すると、以前のSET SERVEROUTPUT ON SIZE *n* FORMAT *f*のSIZEおよびFORMATが使用されるか、または現在の接続で以前にSET SERVEROUTPUTコマンドが発行されていない場合はデフォルト値が使用されます。

SIZEには、Oracle Databaseサーバーの内部バッファに格納できる出力のバイト数を設定します。デフォルトはUNLIMITEDです。*n*は2000以上または1,000,000以下にしてください。

SERVEROUTPUTが設定されている場合、リソースは事前に割り当てられません。パフォーマンスは低下しないため、物理メモリーを保存する場合以外は、UNLIMITEDを使用します。

各サーバー出力行は新しい出力行から開始されます。

WRAPPEDを使用可能にすると、SQL*PlusはSET LINESIZEに指定された行サイズに従ってサーバー出力を折り返し、必要に応じて新しい行を開始します。

WORD_WRAPPEDを使用可能にすると、サーバー出力の各行がSET LINESIZEで指定した行サイズに従って折り返されます。行はワード境界で改行されます。SQL*Plusは、各行を左揃えし、先行空白をすべてスキップします。

TRUNCATEDを使用可能にすると、サーバー出力の各行がSET LINESIZEで指定した行サイズになるように切り捨てられます。

UTL_FILEユーティリティおよび関連ユーティリティの使用の詳細は、『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』を参照してください。

DBMS_OUTPUT.PUT_LINEの詳細は、『[Oracle XAでのアプリケーションの開発](#)』を参照してください。

例

DBMS_OUTPUT.PUT_LINEを使用してPL/SQLブロック内のテキストを表示するには、次のように入力します。

```

SET SERVEROUTPUT ON

```

次に、SET SERVEROUTPUT ONを使用して匿名プロシーダを実行した結果を示します。

```
BEGIN
  DBMS_OUTPUT.PUT_LINE('Task is complete');
END;
/

Task is complete.
PL/SQL procedure successfully completed.
```

次に、SET SERVEROUTPUT ONを使用してトリガーを作成した結果を示します。

```
CREATE TABLE SERVER_TAB (Letter CHAR);
CREATE TRIGGER SERVER_TRIG BEFORE INSERT OR UPDATE -
OR DELETE
ON SERVER_TAB
BEGIN
  DBMS_OUTPUT.PUT_LINE('Task is complete. ');
END;
/

Trigger Created.

INSERT INTO SERVER_TAB VALUES ('M');
DROP TABLE SERVER_TAB;
/* Remove SERVER_TAB from database */

Task is complete.
1 row created.
```

出力をWORD_WRAPPEDに設定するには、次のように入力します。

```
SET SERVEROUTPUT ON FORMAT WORD_WRAPPED
SET LINESIZE 20
BEGIN
  DBMS_OUTPUT.PUT_LINE('If there is nothing left to do');
  DBMS_OUTPUT.PUT_LINE('shall we continue with plan B?');
END;
/

If there is nothing
left to do
shall we continue
with plan B?
```

出力をTRUNCATEDに設定するには、次のように入力します。

```
SET SERVEROUTPUT ON FORMAT TRUNCATED
SET LINESIZE 20
BEGIN
  DBMS_OUTPUT.PUT_LINE('If there is nothing left to do');
  DBMS_OUTPUT.PUT_LINE('shall we continue with plan B?');
END;
/

If there is nothing
shall we continue wi
```

12.41.49 SET SHIFT[INOUT] {VIS[IBLE] | INV[ISIBLE]}

シフト文字を表示する端末に適した位置合せを実現します。SET SHIFTINOUTコマンドは、データとともにシフト文字を表示する端末(たとえば、IBM 3270端末)の場合に使用すると有効です。このコマンドを使用できるのは、シフト文字を検出する文字セット(たとえば、JA16DBCS)のときのみです。

端末でシフト文字を可視文字(たとえば、空白またはコロン)として表示するには、VISIBLEを使用します。INVISIBLEは、これと反対で、シフト文字が表示されません。

例

これらをサポートする端末にシフト文字が表示されるようにするには、次のように入力します。

```
SET SHIFTINOUT VISIBLE
SELECT LAST_NAME, JOB_ID FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000;
```

LAST_NAME	JOB_ID
:JJ00:	:AABBCC:
:AA:abc	:DDEE:e

「:」は、マルチバイト文字を表すシフト大文字です。

小文字は、シングルバイト・キャラクタを示します。

12.41.50 SET SHOW[MODE] {ON | OFF}

SETを使用して、SQL*Plusシステム変数の設定を変更したときに、新旧の設定値をSQL*Plusで表示するかどうかを制御します。ONにすると設定値がリスト表示され、OFFにすると設定値はリスト表示されません。SHOWMODE ONの動作は、すでに廃止されたSHOWMODE BOTHと同じです。

12.41.51 SET SQLBL[ANKLINES] {ON | OFF}

SQL*PlusがSQLコマンドまたはスクリプト内に空白行を入れるかどうかを制御します。ONを指定すると、空白行および新しい行はSQLコマンドまたはスクリプトの一部とみなされます。OFF(デフォルト)を指定すると、SQLコマンドまたはスクリプト内に空白行または新しい行を許可しません。

SQLコマンドのエントリをSQLコマンドを実行しないで停止するには、BLOCKTERMINATORを入力します。SQLコマンドのエントリを停止して、SQL文を実行するには、SQLTERMINATORを入力します。

例

SQL文での空白行を許可するには、次のように入力します。

```
SET SQLBLANKLINES ON
REM Using the SQLTERMINATOR (default is ";")
REM Could have used the BLOCKTERMINATOR (default is ".")
SELECT *

FROM

DUAL

;
```

出力結果は、次のようになります。

```
D
-
X
```

12.41.52 SET SQLC[ASE] {MIX[ED] | LO[WER] | UP[PER]}

SQLコマンドおよびPL/SQLブロックの大/小文字を実行直前に変換します。

SQL*Plusは、引用符で囲まれたリテラルおよび識別子も含めて、コマンド内部のすべてのテキストを変換します。SQLCASEがUPPERの場合は、大文字に変換されます。SQLCASEがLOWERの場合は、小文字に変換されます。SQLCASEがMIXEDの場合は、変更されません。

SQLCASEでは、SQLバッファの内容自体は変更されません。

12.41.53 SET SQLCO[NTINUE] {> | テキスト}

1つのSQL*Plusコマンドをハイフン(-)を使用して次の行に続けた後で、プロンプトとしてSQL*Plusによって表示する文字列順序を設定します。

例

感嘆符とそれに続く1個の空白をSQL*Plusコマンド継続プロンプトとして設定するには、次のように入力します。

```
SET SQLCONTINUE ' ! '
```

SQL*Plusは、継続のプロンプトを次のように表示します。

```
TTITLE 'MONTHLY INCOME' -
! RIGHT SQL.PNO SKIP 2 -
! CENTER 'PC DIVISION'
```

継続プロンプトのデフォルトは「>」です。

12.41.54 SET SQLN[UMBER] {ON | OFF}

SQLコマンドまたはPL/SQLブロックの2行目以降の行のためのプロンプトを設定します。ONを設定すると、プロンプトは行番号に設定されます。OFFを設定すると、プロンプトはSQLPROMPTの値に設定されます。

12.41.55 SET SQLPLUSCOMPAT[IBILITY] {x.y[.z]}

動作をx.y[.z]で指定されたリリースまたはバージョンの動作に設定します。

xはバージョン番号を、yはリリース番号を、zはアップデート番号を示します。たとえば、8.1.7、9.0.1または10.2です。

SQLPLUSCOMPATIBILITYによって影響を受ける機能は、後述のSQL*Plus互換性マトリックスに示します。また、コマンドラインからSQL*Plusを起動する際に、SQLPLUSコマンドの-C[OMPATIBILITY]引数を使用して、SQLPLUSCOMPATIBILITYの値を設定できます。

SQLPLUSCOMPATIBILITYのデフォルト設定は、SQL*Plusクライアントの値です。

SET SQLPLUSCOMPATIBILITY 12.2をスクリプトに追加して、SQL*Plusの将来のバージョンとの互換性を高めることをお勧めします。

12.41.55.1 SQL*Plus互換性マトリックス

SQL*Plus互換性マトリックスに、各SQL*Plusの互換性の設定によって影響を受ける動作を示します。SQL*Plusの互換性モードは、3つの方法で設定できます。

- SET SQLPLUSCOMPATIBILITYコマンドを、サイトまたはユーザーのプロファイル内に含めることができます。インストール時には、glogin.sqlにSET SQLPLUSCOMPATIBILITY設定はありません。したがって、デフォルトの互換性は12.2です。
- 起動時に、SQLPLUS -C[OMPATIBILITY] {x.y[.z]}コマンド引数を使用して、そのセッションの互換性モードを設定できます。
- セッション中に、SET SQLPLUSCOMPATIBILITY {x.y[.z]}コマンドを使用して、セッションに必要なSQL*Plusの動作を設定できます。

次の表に、動作が変更されるSQL*Plusのリリース、およびそのように動作させるためのSQLPLUSCOMPATIBILITYの最小値を示します。たとえば、VARIABLEコマンドを以前のように動作させるには、9.0.1より前のバージョンのSQL*Plusを使用するか、SQLPLUSCOMPATIBILITYに9.0.1より小さい値を使用する必要があります。SQLPLUSCOMPATIBILITYに設定できる最も低い値は7.3.4です。

表12-6 互換性マトリックス

値	結果	使用可能なバージョン
>=10.1	SHOW ERRORS は、Oracle Database 10g でのみ使用可能な新しい列を使用し、PL/SQL エラー・メッセージをソートします。	10.1
>=10.1	SPOOL オプション CREATE、REPLACE および SAVE が追加されました。これらのオプションは、一部のプラットフォームでファイル名の解析に影響する場合があります。	10.1
>=10.1	SET SQLPROMPT	10.1
>=10.1	空白文字は、引用符で囲むと Windows ファイル名に使用できます。その他の特殊句読記号には、現在 Windows では使用できないものもあります。	10.1
>=10.1	glogin/login ファイルは、再接続するたびにコールされます。	10.1
<10.1	/*コメントをエコー表示する場合は廃止された DOC>プロンプトを使用します。	10.1
>= 9.2	FOLD_AFTER で定義される幅の広い列が、新しい行の先頭に表示されます。その他の場合は、前の行の終わりに想定された幅よりも小さい幅で不適切に挿入されます。	9.2.
>= 9.0	SQL 文のスラッシュ(/)の前の空白は無視され、スラッシュは SQL 文の実行とみなされます。その他の場合は、スラッシュは SQL 文の一部(除算記号など)として扱われます。	9.0.1.4.
>= 9.0	NCHAR および NVARCHAR2 型に指定する長さは文字です。その他の場合は、長さは	9.0.1

値	結果
	文字セットに応じてバイトまたは文字を表します。

12.41.56 SET SQLPRE[FIX] {# | c}

SQL*Plusの接頭文字を設定します。SQLコマンドまたはPL/SQLブロックを入力している途中で、別の行に、SQL*Plus接頭文字を付けてSQL*Plusコマンドを入力できます。SQL*Plusは、入力中のSQLコマンドまたはPL/SQLブロックに影響を与えずに、別の行に入力したコマンドを即時実行します。接頭文字は、英数字以外の文字にする必要があります。

12.41.57 SET SQLP[ROMPT] {SQL> | テキスト}

SQL*Plusコマンド・プロンプトを設定します。SET SQLPROMPTによって変数が動的に置換されます。これによって、実行時の変数(現在の接続識別子など)を挿入できます。SQLPROMPTで使用される置換変数の前に「&」を付ける必要はありません。他の置換変数と同様に使用およびアクセスできます。デフォルトのプロンプト「SQL>」に対しては、変数の置換は試行されません。

変数の置換は、SQLPROMPTにSETを指定するたびに行われます。SQLPROMPTがglogin.sqlに含まれている場合、SQLPROMPTの置換変数はログインまたは接続するたびに更新されます。

例

接続識別子を表示するSQL*Plusプロンプトを変更するには、次のように入力します。

```
SET SQLPROMPT "_CONNECT_IDENTIFIER > "
```

現行のユーザーを表示するSQL*Plusコマンド・プロンプトを設定するには、次のように入力します。

```
SET SQLPROMPT "_USER > "
```

SQL*Plusプロンプトを変更して現在の日付、現行のユーザーおよびユーザーの権限レベルを表示するには、次のように入力します。

```
SET SQLPROMPT "_DATE _USER _PRIVILEGE> "
```

SQL*Plusプロンプトを変更して定義済の変数を表示するには、次のように入力します。

```
DEFINE mycon = Prod1
SET SQLPROMPT "mycon> "
```

```
Prod1>
```

ネストした引用符内のテキストは、置換用には解析されません。SQL*Plusプロンプトに、ユーザー名、「@」および接続識別子をこの順序で表示するには、次のように入力します。

```
SET SQLPROMPT "_USER'@'_CONNECT_IDENTIFIER > "
```

12.41.58 SET SQLT[ERMINATOR] {; | c | ON | OFF}

PL/SQLブロックまたはSQL文のスクリプトまたはデータ・エントリの終了、スクリプトの実行およびバッファへのスクリプトのロードのために使用する文字を設定します。

英数字または空白は使用できません。OFFを設定すると、SQL*Plusは、コマンド終了記号を認識しません。SQLコマンドを終了するには、空白行かスラッシュ(/)を入力します。SQLBLANKLINESがONに設定されている場合は、BLOCKTERMINATORを使用してSQLコマンドを終了する必要があります。ONを設定すると、終了記号はデフォルトのセミコロン(;)にリセットされます。

12.41.59 SET STATEMENTC[ACHE] {0 | n}

文キャッシュ・サイズを設定します。キャッシュの値を指定すると、繰り返される文のキャッシュが可能になります。したがって、これらの文は再度解析する必要がないため、パフォーマンスが向上します。

例

文キャッシュ・サイズを15に設定するには、次のように入力します。

```
SET STATEMENTCACHE 15
```

*n*の値を指定しない場合、デフォルトは0となります。つまり、文キャッシュはオフとなります。

文キャッシュ・サイズには、0から32767の範囲の値を指定できます。

ノート:



Oracle Database 初期化ファイル init.ora のオープン・カーソル・パラメータに指定された値未満のキャッシュ・サイズを指定します。値 0 を指定すると、文キャッシュがオフになります。oraaccess.xml ファイル内の **<statement_cache>** の設定により、SQL*Plus における SET STATEMENTCACHE の設定をオーバーライドできます。oraaccess.xml の詳細は、*Oracle Call Interface プログラマーズ・ガイド* を参照してください。

文キャッシュ・サイズの現在の設定を表示するには、次のように入力します。

```
SHOW STATEMENTC[ACHE]
```

12.41.60 SET SUF[IX] {SQL | テキスト}

スクリプトを参照するコマンドの中でSQL*Plusが使用するデフォルトのファイル拡張子を設定します。SUFFIXでは、スプール・ファイルの拡張子は制御できません。

例

デフォルトのコマンド・ファイル拡張子を、SQLから、TXTに設定するには、次のように入力します。

```
SET SUFFIX TXT
```

次のように入力したとします。

```
GET EXAMPLE
```

SQL*Plusは、EXAMPLE.SQLのかわりに、EXAMPLE.TXTという名前のファイルを検索します。

12.41.61 SET TAB {ON | OFF}

SQL*Plusが空白をどのように端末に出力するかを指定します。OFFを指定すると、空白は空白として出力されます。ONを指

定すると、TAB文字が使用されます。TABの設定間隔は8文字です。TABのデフォルト値は、システムによって異なります。

12.41.62 SET TERM[OUT] {ON | OFF}

@、@@またはSTARTを使用して実行するスクリプトのコマンドによって生成される出力の表示を制御します。OFFを指定すると非表示になるため、出力を、画面に表示せずにファイルにスプールできます。ONを指定すると、画面に出力が表示されます。TERMOUT OFFは、対話方式で入力するコマンド、またはオペレーティング・システムからSQL*Plusにリダイレクトするコマンドによる出力には影響しません。

12.41.63 SET TI[ME] {ON | OFF}

現在の時刻表示を制御します。ONを指定すると、各コマンド・プロンプトの前に現在の時刻が表示されます。OFFを指定すると、時刻が非表示になります。

12.41.64 SET TIMI[NG] {ON | OFF}

タイミング統計の表示を制御します。

ONを指定すると、それぞれのSQLコマンドまたはPL/SQLブロックが実行されるたびに、そのタイミング統計が表示されます。OFFを指定すると、各コマンドのタイミング統計が表示されません。

複数のコマンドのタイミング情報については、「[TIMING](#)」を参照してください。

例

タイミング統計の書式は、オペレーティング・システムによって異なります。LinuxおよびWindowsでは、タイミング統計は24時間形式で、時間、分、秒および100分の1秒を表示します。

```
SET SUFFIX TXT
```

次のように入力したとします。

```
GET EXAMPLE
```

SQL*Plusは次のような出力を表示します。

```
GET EXAMPLE
```

12.41.65 SET TRIM[OUT] {ON | OFF}

SQL*Plusで、それぞれの表示行の終わりに後続の空白を入れるかどうかを指定します。ONを指定すると、各行の終わりの空白が削除されるため、パフォーマンスが向上します。特に、低速の通信デバイスからSQL*Plusにアクセスする場合に効果的です。OFFを指定すると、SQL*Plusで後続空白を表示できます。TRIMOUT ONは、スプール出力には影響を与えません。

12.41.66 SET TRIMS[POOL] {ON | OFF}

SQL*Plusで、それぞれのスプール行の終わりに後続の空白を入れるかどうかを指定します。ONを指定すると、各行の終わりの空白が削除されます。OFFを指定すると、SQL*Plusで後続空白を表示できます。TRIMSPool ONは、端末出力には影響を与えません。

12.41.67 SET UND[ERLINE] {- | c | ON | OFF}

レポートの列ヘッダーに下線を付けるために使用する文字を設定します。下線文字には、英数字または空白は使用できません。ONまたはOFFによって、下線付けをオンまたはオフにできます。ONを指定すると、cの値がデフォルトの「-」に戻ります。

12.41.68 SET VER[IFY] {ON | OFF}

置換変数を値と置き換える前後に、SQL文またはPL/SQLコマンドのテキストをリスト表示するかどうかを制御します。ONにするとテキストがリスト表示され、OFFにするとテキストはリスト表示されません。

12.41.69 SET WRA[P] {ON | OFF}

現在の行の幅に対して、選択した行の表示が長すぎる場合に、切り捨てるかどうかを制御します。OFFにすると、選択した行が切り捨てられます。ONにすると、選択した行は次の行に折り返されます。

特定の列のWRAPの設定を上書きするには、COLUMNコマンドのWRAPPED句およびTRUNCATED句を使用します。

12.41.70 SET XMLOPT[IMIZATIONCHECK] [ON|OFF]

完全に最適化されたXML問合せおよびDML操作のみを実行するかどうかを制御します。ONを指定すると、XML問合せまたはDML操作を完全に最適化できない場合にその実行を抑止し、その理由がトレース・ファイルに書き込まれます。OFFを指定すると、このような問合せおよび操作の実行を抑止しません。デフォルトはOFFです。

SET XMLOPT[IMIZATIONCHECK] ONは、XML問合せまたはDML操作の開発中やデバッグの作業中にのみ使用してください。

12.41.71 SET XQUERY BASEURI {テキスト}

ファンクション内の相対URIを解決するために使用されるベースURIを指定します。これによって、XQueryによりアクセスされるファイルの接頭辞を変更できます。

BASEURIの設定を解除するには、次のように空の文字列を設定します。

```
SET XQUERY BASEURI ''
```

値はXQUERYコマンドの発行時にのみ確認されるため、注意して有効な値を入力してください。

例

```
SET XQUERY BASEURI '/public/scott'  
XQUERY for $i in doc("foo.xml") return $i  
/
```

これは、次の操作に相当します。

```
XQuery declare base-uri "/public/hr";  
for $i in doc("foo.xml") return $i
```

12.41.72 SET XQUERY ORDERING {UNORDERED | ORDERED | DEFAULT}

XQueryの出力の順序付けを設定します。次の3つの値があります。

UNORDERED: データベースから取得された順序で結果が格納されることを指定します。

ORDERED: XQueryで定義された順序で結果が格納されることを指定します。

DEFAULT: データベースのデフォルトを指定します。Oracle Database 10gでは、デフォルトはUNORDEREDです。

SET XQUERY ORDERINGが設定されていない場合、デフォルトはDEFAULT(UNORDERED)です。

例

```
SET XQUERY ORDERING ORDERED
XQUERY for $i in doc("foo.xml") return $i
/
```

これは、次の操作に相当します。

```
XQuery declare ordering ordered;
for $i in doc("foo.xml") return $i
/
```

12.41.73 SET XQUERY NODE {BYVALUE | BYREFERENCE | DEFAULT}

ノードID保存モードを設定します。保存モードは、新規ノードを作成するすべての式(要素コンストラクタなど)またはノードを含む項目あるいは順序を戻すすべての式に適用されます。次の3つの値があります。

BYVALUE: ノードIDを保存する必要がないことを指定します。これは、作成などのノード操作または式の結果として戻されたノード操作がディープ・コピーされ、元のツリーでのコンテキストが失われることを意味します。このノード上でのノードID、親軸、兄弟軸または順序付けをテストする後続の操作は定義されません。

BYREFERENCE: ノードIDを保存する必要があることを指定します。このノード上での後続の操作で、ノードのコンテキストおよび定義が保存されます。

DEFAULT: データベースのデフォルトを指定します。Oracle Database 10gでは、デフォルトはBYVALUEです。

SET XQUERY NODEが設定されていない場合、デフォルトはDEFAULT(BYVALUE)です。

例

```
SET XQUERY NODE BYREFERENCE
XQUERY for $i in doc("foo.xml") return $i
/
```

これは、次の操作に相当します。

```
XQuery declare node byreference;
for $i in doc("foo.xml") return $i
/
```

12.41.74 SET XQUERY CONTEXT {テキスト}

XQueryコンテキスト項目の式を指定します。コンテキスト項目の式によって、コンテキスト項目に対して評価が行われます。コンテキスト項目は、ノード(式fn:doc("bib.xml")//book[fn:count(/author)>1]を指定)または原子価(式(1 to 100)[. mod 5 eq 0]を指定)のいずれかです。

XQUERY CONTEXTの設定を解除するには、次のように空の文字列を設定します。

```
SET XQUERY CONTEXT ''
```

値はXQUERYコマンドの発行時にのみ確認されるため、注意して有効な値を入力してください。

例

```
SET XQUERY CONTEXT 'doc("foo.xml")'  
XQUERY for $i in /a return $i  
/
```

これは、次の操作に相当します。

```
XQuery for $i in /a return $i  
passing XMLQuery("doc('foo.xml')")  
/
```

12.42 SHOW

構文

```
SHOW [W] option
```

*option*には、次の項または句を1つ指定します。

```
system_variable ALL BTI[TLE] CON_ID CON_NAME EDITION ERR[ORS] [ {ANALYTIC VIEW | ATTRIBUTE  
DIMENSION | HIERARCHY | FUNCTION | PROCEDURE | PACKAGE | PACKAGE BODY | TRIGGER | VIEW | TYPE | TYPE  
BODY | DIMENSION | JAVA CLASS } [schema.]name] HISTORY LNO LOBPREF[ETCH] PARAMETER[S]  
[parameter_name] PDBS PNO RECYC[LEBIN] [original_name] REL[EASE] REPF[OOTER] REPH[EADER]  
ROWPREF[ETCH] SGA SPOOL] SPPARAMETER[S] [parameter_name] SQLCODE STATEMENTC[ACHE] TTI[TLE] USER  
XQUERY
```

SQL*Plusシステム変数の値または現行のSQL*Plus環境を表示します。SHOW SGAの場合は、DBA権限でのログインが必要です。

項

```
system_variable
```

SETコマンドによって設定される任意のシステム変数を指定します。

```
ALL
```

ERRORSおよびSGAを除いて、SHOWのすべてのオプション設定がアルファベット順に表示されます。

```
CON_ID
```

統合データベースに接続している場合は接続先のコンテナのIDが表示されます。非統合データベースに接続している場合にこのコマンドを発行すると、0が戻されます。

```
CON_NAME
```

統合データベースに接続している場合は接続先のコンテナの名前が表示されます。非統合データベースの場合、Non Consolidatedが戻されます。

```
EDITION
```

既存のデータベース・サービスのエディション属性が表示されます。

```
BTI [TLE]
```

現行のBTITLE定義が表示されます。

```
ERR[ORS] [ {ANALYTIC VIEW | ATTRIBUTE DIMENSION | HIERARCHY | FUNCTION | PROCEDURE | PACKAGE | PACKAGE  
BODY | TRIGGER | VIEW | TYPE | TYPE BODY | DIMENSION | JAVA CLASS} [schema.]name]
```

ストアド・プロシージャ(ストアド・ファンクション、プロシージャおよびパッケージを含む)のコンパイル・エラーが表示されます。CREATEコマンドを使用してストアド・プロシージャを作成した後で、ストアド・プロシージャにコンパイル・エラーが検出されると、メッセージが表示されます。エラーを表示するには、SHOW ERRORSを使用します。

引数を指定しないでSHOW ERRORSのみを指定すると、SQL*Plusは最後に作成または変更されたストアド・プロシージャに対するコンパイル・エラーを表示します。PL/SQLストアド・プロシージャのタイプ(分析ビュー、属性ディメンション、階層、ファンクション、プロシージャ、パッケージ、パッケージ本体、トリガー、ビュー、型、型本体、ディメンションまたはJavaクラス)および名前を指定すると、SQL*Plusは、そのストアド・プロシージャのエラーを表示します。コンパイル・エラーの詳細は、『PL/SQL ユーザー

ズ・ガイドおよびリファレンス』を参照してください。

`schema`には、名前付きオブジェクトが含まれています。`schema`を省略すると、SHOW ERRORSはオブジェクトが現行のスキーマ内にあるとみなします。

SHOW ERRORSの出力には、エラーの行番号および列番号(LINE/COL)と同時にエラーそのもの(ERROR)も表示されます。LINE/COLおよびERRORのデフォルトの幅は、それぞれ8および65です。デフォルトの幅は、COLUMNコマンドを使用して変更できます。

HISTORY

[SET HISTORY](#)コマンドを使用して設定された現在のコマンド履歴ステータスを表示します。

LN0

カレント行の番号(表示出力またはスプール出力(あるいはその両方)の現行ページの中の位置)が表示されます。

LOBPREFETCH

SQL*Plusがデータベースから一度にプリフェッチするLOBデータの量について現在の設定を表示します。SQL*Plusがデータベースから一度にプリフェッチするLOBデータの量の設定方法は、[SET LOBPREFETCH](#)を参照してください。

PARAMETERS [*parameter_name*]

1つ以上の初期化パラメータに対して、現行の値を表示します。このコマンドの後の文字列を使用して、その文字列を含む名前のパラメータのサブセットを参照できます。たとえば、次のように入力したとします。

SHOW PARAMETERS COUNT

NAME	TYPE	VALUE
-----	-----	-----
db_file_multiblock_read_count	integer	12
spin_count	integer	0

コマンドに続く任意の文字列なしでSHOW PARAMETERSコマンドのみを指定すると、すべての初期化パラメータが表示されます。

出力結果は、接続しているOracle Databaseサーバーのバージョンおよび構成によって異なります。PARAMETERS句を使用するには、SELECT ON V_\$PARAMETERオブジェクト権限が必要です。権限がない場合、次のメッセージが表示されます。

```
ORA-00942: table or view does not exist
```

PDBS

接続している統合データベース内のプラガブル・データベースの名前、ID、モードおよび制限ステータスが表示されます。非統合データベースに接続している場合はコマンドを使用できません。

PDBSオプションは、SYSDBAとしてログインし、SYSDBA権限を持つ場合にのみ使用可能です。DBA以外のユーザーがPDBSオプションの使用を試みると、エラー「SP2-0382: SHOW PDBSコマンドは使用できません。」が返されます。

PN0

現行のページ番号が表示されます。

RECYC[LEBIN] [*original_name*]

FLASHBACK BEFORE DROPコマンドでリカバリ可能なごみ箱内のオブジェクトを表示します。列名を記録したり、解読が容

易でない問合せの出力結果を解釈する必要はありません。

```
SELECT * FROM USER_RECYCLEBIN
```

問合せによって、次の順序で表示される4つの列が戻されます。

列名	説明
ORIGINAL NAME	オブジェクト作成時に使用した元の名前を表示します。
RECYCLEBIN NAME	ごみ箱内のオブジェクトの識別に使用する名前を表示します。
OBJECT TYPE	オブジェクトの型を表示します。
DROP TIME	オブジェクトが削除された時刻を表示します。

出力列の書式はCOLUMNコマンドで設定できます。

DBAは、独自のuser_recyclebinビューを持っているため、このコマンドを実行して所有しているオブジェクトを表示できます。

```
REL[EASE]
```

SQL*PlusがアクセスしているOracle Databaseのリリース番号が表示されます。

```
REPF[OOTER]
```

現行のREPFOOTER定義が表示されます。

```
REPH[EADER]
```

現行のREPHEADER定義が表示されます。

```
ROWPREFETCH
```

SQL*Plusがデータベースから一度にプリフェッチする行の数について現在の設定を表示します。SQL*Plusがデータベースから一度にプリフェッチする行の数の設定方法は、[SET ROWPREFETCH](#)を参照してください。

```
SP00[L]
```

出力がスプールされているかどうかが表示されます。

```
SGA
```

現行のインスタンスのシステム・グローバル領域に関する情報が表示されます。ユーザーはSELECT ON V_\$SGAオブジェクト権限が必要です。この権限がない場合は、次のメッセージが表示されます。

```
ORA-00942: table or view does not exist
```

```
SPPARAMETERS [parameter_name]
```

SHOW PARAMETERSの場合(SHOW SPPARAMETERSがすべてのインスタンスの初期化パラメータの現行値を表示する場合を除く)。このコマンドの後の文字列を使用して、その文字列を含む名前のパラメータのサブセットを参照できます。

コマンドに続く任意の文字列なしでSHOW SPPARAMETERSコマンドのみを指定すると、すべてのインスタンスの初期化パラメータがすべて表示されます。

出力結果は、接続しているOracle Databaseサーバーのバージョンおよび構成によって異なります。SPPARAMETERS句を使用するには、SELECT ON V_\$PARAMETERオブジェクト権限が必要です。

SQLCODE

SQL.SQLCODE(最新の操作のSQLリターン・コード)の値が表示されます。

STATEMENTCACHE

文キャッシュ・サイズの現在の設定を表示します。文キャッシュ・サイズの設定方法は、[SET STATEMENTCACHE](#)を参照してください。

TTI [TLE]

現行のTTITLE定義が表示されます。

USER

SQL*Plusのアクセスで現在使用しているユーザー名が表示されます。/ AS SYSDBAで接続すると、SHOW USERコマンドで表示されます。

```
USER is "SYS"
```

XQUERY

XQUERY設定、BASEURI、CONTEXT、NODEおよびORDERINGの現行の値を示します。

```
xquery BASEURI "public/scott" CONTEXT "doc('test.xml')" NODE byreference ORDERING ordered
```

値が設定されない場合は、次の出力が表示されます。

```
xquery BASEURI "" CONTEXT "" NODE default ORDERING default
```

例

SGAの情報を表示するには、次のように入力します。

SHOW SGA

Total System Global Area	7629732 bytes
Fixed Size	60324 bytes
Variable Size	6627328 bytes
Database Buffers	409600 bytes
Redo Buffers	532480 bytes

次の例は、ストアード・プロシージャを作成し、そのコンパイル・エラーを表示する方法を示しています。

```
CONNECT SYSTEM/MANAGER
CREATE PROCEDURE HR.PROC1 AS
BEGIN
:P1 := 1;
END;
/
```

Warning: Procedure created with compilation errors.

```
SHOW ERRORS PROCEDURE PROC1
```

```
NO ERRORS.
```

```
SHOW ERRORS PROCEDURE HR.PROC1
```

```
Errors for PROCEDURE HR.PROC1:  
LINE/COL ERROR
```

```
-----  
3/3      PLS-00049: bad bind variable 'P1'
```

AUTORECOVERYが使用可能かどうかを表示するには、次のように入力します。

```
SHOW AUTORECOVERY
```

```
AUTORECOVERY ON
```

接続しているコンテナのIDを表示するには、次のように入力します。

```
SHOW CON_ID
```

```
CON_ID
```

```
-----  
1
```

接続しているコンテナの名前を表示するには、次のように入力します。

```
SHOW CON_NAME
```

```
CON_NAME
```

```
-----  
CDB$ROOT
```

[SET HIST\[ORY\] {ON | OFF | n}](#) コマンドを発行して設定された現在のコマンド履歴ステータスを表示するには、次のように入力します。

```
SHOW HISTORY
```

```
SQL> set history on  
SQL> show history  
History is ON and set to "100"  
SQL> set history off  
SQL> show history  
History is OFF  
SQL> set history 1000  
SQL> show history  
History is ON and set to "1000"
```

接続している統合データベース内のプラガブル・データベースの名前、IDおよびモードを表示するには、次のように入力します。

```
SHOW PDBS
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	CDB1_PDB1	READ WRITE	NO

デフォルト・インスタンスの接続識別子を表示するには、次のように入力します。

```
SHOW INSTANCE
```

```
INSTANCE "LOCAL"
```

アーカイブ・ログの位置を表示するには、次のように入力します。

```
SHOW LOGSOURCE
```

```
LOGSOURCE "/usr/oracle90/dbs/arch"
```

CJ1およびABCが削除された場所に、FLASHBACKコマンドでリカバリ可能なオブジェクトを表示するには、次のように入力します。

```
SHOW RECYCLEBIN
```

ORIGINAL NAME	RECYCLEBIN NAME	OBJECT TYPE	DROP TIME
-----	-----	-----	-----
CJ1	RB\$\$29458\$TABLE\$0	TABLE	2003-01-22:14:54:07
ABC	RB\$\$29453\$TABLE\$0	TABLE	2003-01-20:18:50:29

CJ1をリストアするには、次のように入力します。

```
FLASHBACK TABLE CJ1 TO BEFORE DROP;
```

12.43 SHUTDOWN

構文

```
SHUTDOWN [ABORT | IMMEDIATE | NORMAL | TRANSACTIONAL [LOCAL]]
```

現在実行中のOracle Databaseインスタンスを停止します。必要に応じて、データベースのクローズおよびデスマウントが実行されます。現在のデータベースがプラガブル・データベースの場合、そのプラガブル・データベースのみが閉じられます。統合インスタンスは引き続き実行されます。

現在のコールの完了またはSHUTDOWN NORMALおよびSHUTDOWN TRANSACTIONALを使用したユーザーによる切断を待機するSHUTDOWNコマンドには待機時間制限があります。停止を阻止するすべてのイベントが時間制限以内に発生しなかった場合、SHUTDOWNコマンドは次のメッセージを表示して取消しを実行します。

```
ORA-01013: user requested cancel of current operation
```

PDBを停止するための前提条件

現在のコンテナがプラガブル・データベース(PDB)のとき、SHUTDOWNコマンドを使用できるのは、次の場合のみです。

- 現在のユーザーにSYSDBA、SYSOPER、SYSBACKUPまたはSYSDGシステム権限がある。
- その権限が共通的に付与されているか、そのPDBでローカルに付与されている。
- 接続時に現在のユーザーがAS SYSDBA、AS SYSOPER、AS SYSBACKUPまたはAS SYSDGを使用して権限を実行する。
- PDBをクローズするには、そのPDBがオープンされている必要があります。

詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。

項

ABORT

コールの完了またはユーザーの切断を待たずに、高速でデータベースの停止処理を進めます。

コミットされていないトランザクションはロールバックされません。現在実行中のクライアントから要求されたSQL文は、終了されます。現在データベースに接続しているユーザーはすべて暗黙的に切断され、次のデータベース起動時にインスタンス・リカバリが必要になります。

バックグラウンド・プロセスが異常終了した場合、このオプションを使用する必要があります。

IMMEDIATE

現行のコールの完了またはユーザーによるデータベース接続の切断を待ちません。

新しい接続は禁止されます。データベースはクローズされ、デスマウントされます。次のデータベース起動時に、インスタンス・リカバリは必要ありません。

NORMAL

デフォルト・オプションです。ユーザーによってデータベース接続が切断されるまで待機します。

新しい接続は禁止されます。データベースはクローズされ、デスマウントされます。次のデータベース起動時に、インスタンス・リカバリは必要ありません。

TRANSACTIONAL [LOCAL]

実行中のトランザクションがすべて完了したとき、インスタンスの計画停止が実行されます。すべてのユーザーにログオフを要求しない場合でも、クライアントの処理が失われるのを防止できます。

この状態のインスタンスでは、クライアントからは新しいトランザクションを開始できません。新しいトランザクションを開始しようとすると、切断されます。すべてのトランザクションの完了後、インスタンスに接続しているクライアントは切断されます。

SHUTDOWN IMMEDIATE文が実行された場合と同様に、インスタンスが停止します。次にデータベースを起動するときに、インスタンス・リカバリ手順は必要ありません。

LOCALモードは、ローカル・インスタンスのみでのトランザクション停止を指定します。これは、すべてのトランザクションではなく、ローカル・トランザクションの完了を待機するのみです。たとえば、計画した停止のメンテナンスに便利です。

使用方法

引数なしのSHUTDOWNは、SHUTDOWN NORMALと同じ動作をします。

SYSDBA、SYSOPER、SYSBACKUPまたはSYSDGでデータベースに接続している必要があります。マルチスレッド・サーバーを経由して接続できません。データベースへの接続の詳細は、「[CONNECT](#)」を参照してください。

例

CDBにログインした場合、停止によりそのCDBインスタンスが閉じられます。

CDBまたは非CDBを停止するには、閉じるCDBまたは非CDBインスタンスに接続してから次を入力する必要があります。

```
SHUTDOWN
```

```
Database closed.  
Database dismounted.  
Oracle instance shut down.
```

PDBを停止するには、そのPDBにログインしてSHUTDOWNコマンドを発行する必要があります。

```
SHUTDOWN
```

```
Pluggable Database closed.
```

12.44 SPOOL

構文

```
SPOOL [ file_name [. ext] ] [ CRE [ATE] | REP [LACE] | APP [END] ] | OFF | OUT
```

問合せの結果をファイルに格納し、必要に応じてそのファイルをプリンタに送信します。

項

```
file_name [. ext]
```

スプール出力を格納するファイルの名前を指定します。SPOOLに続けて*file_name*を入力すると、指定ファイルへの表示出力のスプールが開始されます。拡張子を指定しない場合、SPOOLはデフォルトの拡張子(ほとんどのシステムではLSTまたはLIS)が使用されます。/dev/nullおよび/dev/stderrなどのシステム・ファイルには拡張子は追加されません。

```
CRE [ATE]
```

指定した名前で、ファイルを新規作成します。

```
REP [LACE]
```

既存のファイルの内容を置換します。ファイルが存在しない場合、ファイルが作成されます。これはデフォルトの動作です。

```
APP [END]
```

指定したファイルの終わりに、バッファの内容を追加します。

```
OFF
```

スプールを停止します。

```
OUT
```

スプールを停止して、ファイルをコンピュータの標準(デフォルト)プリンタに送ります。このオプションは、一部のオペレーティング・システムでは使用できません。

現行のスプール状態を表示するには、句を指定しないでSPOOLのみを入力します。

使用方法

スクリプト内のコマンドによって生成された出力を画面に表示しないで、その出力をスプールするには、SET TERMOUT OFFを使用します。SET TERMOUT OFFは、対話形式で実行するコマンドからの出力には影響を与えません。

空白を含むファイル名の前後に引用符を使用する必要があります。

SPOOL APPENDコマンドを使用して有効なHTMLファイルを作成するには、PROMPTまたは同様のコマンドを使用してHTMLページのヘッダーとフッターを作成する必要があります。SPOOL APPENDコマンドはHTMLタグを解析しません。

CREATE、APPENDおよびSAVEの各パラメータを無効にするには、SET SQLPLUSCOMPAT[IBILITY]を9.2以前に設定します。SET SQLPLUSCOMPAT[IBILITY]コマンドで制御する機能を判断するには、[「SQL*Plus互換性マトリックス」](#)を参照してください。

SPOOLコマンドの例

デフォルトのファイル拡張子を使用して、新しいファイルDIARYに出力を記録するには、次のように入力します。

```
SPOOL DIARY CREATE
```

既存のファイルDIARYに出力を追加するには、次のように入力します。

```
SP00L DIARY APPEND
```

ファイルDIARYに出力を記録し、既存の内容に上書きするには、次のように入力します。

```
SP00L DIARY REPLACE
```

スプールを停止し、ファイルをデフォルトのプリンタで印刷するには、次のように入力します。

```
SP00L OUT
```


12.45 START

構文

```
STA[RT] { url | file_name[. ext] } [arg...]
```

指定したスクリプトのSQL*Plus文を実行します。スクリプトは、ローカル・ファイル・システムまたはWebサーバーからコールされます。

項

url

指定したWebサーバーで実行するスクリプトのURLを指定します。SQL*Plusは、HTTPおよびFTPプロトコルをサポートしますが、HTTPSはサポートしません。今回のリリースでは、`http://username:password@machine_name.domain...`形式のHTTP認証はサポートされていません。

file_name[. *ext*]

実行するスクリプトを指定します。このファイルには、対話形式で実行できるコマンドも含まれています。

拡張子を指定しない場合、SQL*Plusは、デフォルトのコマンド・ファイル拡張子(通常はSQL)が指定されたものとみなします。デフォルトの拡張子を変更する方法については、「[SET SUF\[FIX\] {SQL | text}](#)」を参照してください。

START *file_name.ext*を入力すると、SQL*Plusは現行のデフォルト・ディレクトリの中で、指定したファイル名および拡張子を持つファイルを検索します。該当するファイルが見つからない場合、SQL*Plusは、そのファイルを見つけるためにシステム依存パスを検索します。オペレーティング・システムによっては、このパス検索がサポートされていない場合もあります。オペレーティング・システム固有の情報については、ご使用のオペレーティング・システムのプラットフォーム固有のOracleマニュアルを参照してください。

arg...

スクリプト内のパラメータに渡すデータ項目を指定します。1つ以上の引数を入力すると、SQL*Plusは、該当する値をスクリプトのパラメータ(&1、&2など)に代入します。最初の引数によって、出現したすべての&1が置換され、2番目の引数によって、出現したすべての&2が置換されます。

STARTコマンドは、引数の値でパラメータを定義します。このセッションで同じスクリプトをSTARTによって再起動する場合は、新しい引数を入力することも、引数の指定を省略して古い値を使用することもできます。

パラメータの使用方法的詳細は、「[置換変数の定義](#)」および「[置換変数の使用](#)」を参照してください。

使用方法

スクリプト起動時は、COLUMNコマンドなどで事前に設定されたすべての内容が有効です。スクリプトで設定が変更される場合、新しい値が有効になるのは、スクリプト終了後です。

@(アットマーク)および@@(二重アットマーク)コマンドは、STARTと同じ機能です。製品ユーザー・プロフィールでSTARTコマンドを使用禁止にすると、@コマンドおよび@@コマンドも使用禁止になります。これらのコマンドの詳細は、「[@\(アットマーク\)](#)」および「[@@\(二重アットマーク\)](#)」を参照してください。詳細は、[SQL*Plus、SQLおよびPL/SQLコマンドの使用禁止](#)を参照してください。

スクリプト内でEXITまたはQUITコマンドを使用すると、そこでSQL*Plusが終了します。

例

名前がPROMOTEで拡張子がSQLの、社員の昇進に使用するファイルには、次のコマンドが含まれていることがあります。

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID, SALARYFROM EMP_DETAILS_VIEWWHERE JOB_ID=' &1' AND SALARY>&2;
```

スクリプトを実行するには、次のように入力します。

```
START PROMOTE ST_MAN 7000
```

Webサーバーにある場合は、次の形式でコマンドを入力します。

```
START HTTP://machine_name.domain:port/PROMOTE.SQL ST_MAN 7000
```

*machine_name.domain*をhost.domain名に、*port*をスクリプトがあるWebサーバーが使用しているポート番号に置き換える必要があります。

次のコマンドが実行されます。

```
SELECT LAST_NAME, LAST_NAME  
FROM EMP_DETAILS_VIEW  
WHERE JOB_ID=' ST_MAN' AND SALARY>7000;
```

その結果が表示されます。

12.46 STARTUP

構文

```
STARTUP db_options | cdb_options | upgrade_options
```

*db_options*の構文は、次のとおりです。

```
[FORCE] [RESTRICT] [PFILE=filename] [QUIET] [ MOUNT [dbname] | [ OPEN [open_db_options] [dbname] ] |  
NOMOUNT ]
```

*open_db_options*の構文は、次のとおりです。

```
READ [ONLY | WRITE [RECOVER]] | RECOVER
```

*cdb_options*の構文は、次のとおりです。

```
root_connection_options | pdb_connection_options
```

*root_connection_options*の構文は、次のとおりです。

```
PLUGGABLE DATABASE pdbname [FORCE] | [UPGRADE] | [RESTRICT] [ OPEN {open_pdb_options}]
```

*pdb_connection_options*の構文は、次のとおりです。

```
[FORCE] | [UPGRADE] | [RESTRICT] [ OPEN {open_pdb_options}]
```

*open_pdb_options*の構文は、次のとおりです。

```
READ WRITE | READ ONLY
```

*upgrade_options*の構文は、次のとおりです。

```
[PFILE=filename] {UPGRADE | DOWNGRADE} [QUIET]
```

データベースのマウントおよびオープンを含む様々なオプションを付けて、Oracle Databaseインスタンスを起動します。

PDB STARTUPのための前提条件

現在のコンテナがプラグブル・データベース(PDB)のとき、STARTUPコマンドを使用できるのは、次の場合のみです。

- 現在のユーザーにSYSDBA、SYSOPER、SYSBACKUPまたはSYSDGシステム権限がある。
- その権限が共通的に付与されているか、そのPDBでローカルに付与されている。
- 接続時に現在のユーザーがAS SYSDBA、AS SYSOPER、SYSBACKUPまたはAS SYSDGを使用して権限を実行する。
- PDBがMOUNTEDモードである(FORCEオプションを使用する場合を除く)。
- MOUNTEDモードになるには、PDBはREAD ONLYまたはREAD WRITEモードである必要があります。

詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。

ノート:



db_options は、データベース(統合または非統合)を起動する場合にのみ使用します。

root_connection_options は、ルートに接続しているときにプラグブル・データベースを起動する場合にのみ使用します。

pdb_options は、接続しているプラグブル・データベースを起動する場合にのみ使用します。

upgrade_options は、データベース(統合または非統合)をアップグレードまたはダウングレード目的で起動する場合にのみ使用します。

READ WRITE と READ ONLY がどちらも指定されていないとき、PDB は、属する CDB が物理スタンバイ・データベースとして使用される場合は READ ONLY で開かれ、それ以外の場合は READ WRITE で開かれます。

統合データベースおよびプラグブル・データベースの使用方法の詳細は、「[Oracle Database の作成および構成](#)」を参照してください。

項

FORCE

再起動する前に、現行の Oracle Database インスタンス(実行されている場合)を ABORT モードの SHUTDOWN で停止しておく必要があります。現行のインスタンスが実行されている、および FORCE が指定されていない場合は、エラーになります。FORCE は、デバッグ中および異常な環境下で有効となります。通常は使用しないでください。

RESTRICT

データベースに接続できるのは、RESTRICTED SESSION のシステム権限がある Oracle Database ユーザーのみです。制限されたセッション機能を使用禁止にするには、ALTER SYSTEM コマンドを使用します。

PFILE=*filename*

インスタンスの起動中に使用されるクライアント・パラメータ・ファイルを指定します。PFILE の指定を省略すると、サーバーはデフォルトのサーバー・パラメータ・ファイル(spfile)にアクセスしようとします。デフォルトの spfile が見つからないと、サーバーはデフォルトの pfile にアクセスしようとします。デフォルトのファイルはプラットフォーム固有です。たとえば、デフォルトのファイルは、UNIX では \$ORACLE_HOME/dbs/init\$ORACLE_SID.ora で、Windows では ORACLE_HOME¥database¥initORCL.ora です。

QUIET

インスタンスを起動する際、システム・グローバル領域の情報を非表示にします。

MOUNT *dbname*

データベースをオープンしないで、マウントします。

dbname には、マウントまたはオープンするデータベースの名前を指定します。データベース名を指定しない場合、初期化パラメータ DB_NAME のデータベース名が使用されます。

OPEN

指定されたデータベースを、マウントおよびオープンします。

NOMOUNT

インスタンスの起動によって、データベースがマウントされなくなります。

MOUNTまたはOPENと同時に指定できません。

RECOVER

インスタンスを起動する前に、必要に応じてメディア・リカバリを実行するように指定します。STARTUP RECOVERは、RECOVER DATABASEコマンドの実行、およびインスタンスの起動と同じ結果になります。完全リカバリが可能なのは、RECOVERオプションのみです。

必要に応じて、AUTORECOVERYが使用可能かどうかにかかわらず、AUTORECOVERYにONが設定されているものとしてリカバリを続行します。REDOログ・ファイルが予想した位置に見つからない場合、AUTORECOVERYが使用禁止であるとみなして、リカバリが続行されます。このとき、適用が必要な次のログ・ファイルの位置と名前がプロンプトに提示されます。

UPGRADE

OPEN UPGRADEモードでデータベースを起動し、システム初期化パラメータに、データベースのアップグレード・スクリプトを実行可能にするために必要な特定の値を設定します。UPGRADEはOracle Databaseサーバーの新しいバージョンを使用して、初めてデータベースを起動するときのみ使用されます。

アップグレード・スクリプトを実行すると、インストール済のOracle Databaseが新しいバージョンに変換されます。たとえば、Oracle9データベースがOracle Database 10gにアップグレードされます。アップグレードが完了したら、データベースを停止し、通常どおり再起動する必要があります。

DOWNGRADE

OPEN DOWNGRADEモードでデータベースを起動し、システム初期化パラメータに、データベースのダウングレード・スクリプトを実行可能にするために必要な特定の値を設定します。

ダウングレード・スクリプトを実行すると、インストール済のOracle Databaseが以前のバージョンに変換されます。たとえば、Oracle Database 10gがOracle9データベースにダウングレードされます。ダウングレードが完了したら、データベースを停止し、通常どおり再起動する必要があります。

PLUGGABLE DATABASE

プラグブル・データベースのオプションを使用して、STARTUPコマンドで起動させるプラグブル・データベースを指定します。

使用方法

SYSDBA、SYSOPER、SYSBACKUPまたはSYSDGでデータベースに接続している必要があります。デイスパッチャを経由して共有サーバーには接続できません。

引数なしのSTARTUPは、STARTUP OPENと同じ動作をします。

STARTUP OPEN RECOVERは、リカバリが失敗したときもデータベースをマウントおよびオープンします。

例

標準パラメータ・ファイルを使用してCDBまたは非CDBインスタンスを起動し、デフォルトのデータベースをマウントして、オープンするには、次のように入力します。

STARTUP

または、次のように入力します。

STARTUP OPEN database

標準パラメータ・ファイルを使用してインスタンスを起動し、デフォルトのデータベースをマウントして、オープンするには、次のように入力します。

```
STARTUP FORCE RESTRICT MOUNT
```

データベースをマウントしないで、パラメータ・ファイルTESTPARMを使用してインスタンスを起動するには、次のように入力します。

```
STARTUP PFILE=testparm NOMOUNT
```

現在稼働中のインスタンスを強制終了し、パラメータ・ファイルmyinit.oraを使用して再起動後にデータベースをオープンします。データベースへの接続は、RESTRICTED SESSION権限を持つユーザーのみが可能です。次のように入力します。

```
STARTUP FORCE RESTRICT PFILE=myinit.ora OPEN database
```

インスタンスを起動して、データベースをオープンしないでマウントするには、次のように入力します。

```
CONNECT / as SYSDBA
```

```
Connected to an idle instance.
```

```
STARTUP MOUNT
```

```
ORACLE instance started.
```

Total System Global Area	7629732 bytes
Fixed Size	60324 bytes
Variable Size	6627328 bytes
Database Buffers	409600 bytes
Redo Buffers	532480 bytes

PDBコンテナからPDBを起動するには、次のシーケンスを入力します。

```
CONNECT SYS/<password>@CDB1_PDB1 AS SYSDBA
```

```
Connected.
```

```
SHOW CON_NAME
```

```
CON_NAME
```

```
-----  
CDB1_PDB1
```

```
SHOW PDBS
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
3	CDB1_PDB1	MOUNTED	

```
STARTUP
```

```
Pluggable Database opened.
```

```
SHOW PDBS
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
3	CDB1_PDB1	READ WRITE	NO

ルートからPDBを起動するには、次のシーケンスを入力します。

```
CONNECT / AS SYSDBA
```

```
Connected.
```

```
SHOW CON_NAME
```

```
CON_NAME
```

```
-----  
CDB$ROOT
```

```
SHOW PDBS
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	CDB1_PDB1	MOUNTED	

```
STARTUP PLUGGABLE DATABASE CDB1_PDB1
```

```
Pluggable Database opened.
```

```
SHOW PDBS
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	CDB1_PDB1	READ WRITE	NO

12.47 STORE

構文

```
STORE SET file_name[. ext] [CRE[ATE] | REP[LACE] | APP[END]]
```

現行のSQL*Plus環境の属性をスクリプトに保存します。

項

STOREコマンド構文の項および句については、「[SAVE](#)」を参照してください。

SET

システム変数の値が保存されます。

使用方法

このコマンドを実行すると、「[START](#)」コマンド、「[@\(アットマーク\)](#)」コマンドまたは「[@@\(二重アットマーク\)](#)」コマンドを使用して実行できるスクリプトが作成されます。

STOREコマンド句(CREATE、REPLACEまたはAPPEND)と同じ名前でファイルを保存するには、名前を一重引用符で囲むか、またはファイル拡張子を指定する必要があります。

例

現行のSQL*Plusシステム変数を、名前がDEFAULTENVでデフォルトのコマンド・ファイル拡張子が付いたファイルに格納するには、次のように入力します。

```
STORE SET DEFAULTENV
```

現行のSQL*Plusシステム変数を、名前がDEFAULTENVで拡張子がOLDの既存ファイルに追加するには、次のように入力します。

```
STORE SET DEFAULTENV.OLD APPEND
```


12.48 TIMING

構文

```
TIMI[NG] [START text | SHOW | STOP]
```

経過時間に関するタイミング・データの記録、現行のタイマーの名前とタイミング・データの表示、またはアクティブなタイマー数の表示を行います。

項

```
START text
```

タイマーが設定され、*text*がタイマーの名前になります。最初のタイマーを停止する前に追加のタイマーを起動することによって、複数のアクティブ・タイマーを同時に使用できます。SQL*Plusは、それぞれの新規タイマーをその前のタイマーにネストします。最後に起動したタイマーが現行のタイマーになります。

```
SHOW
```

現行のタイマー名およびタイミング・データが表示されます。

```
STOP
```

現行のタイマー名および計時データが表示されてから、そのタイマーが削除されます。他にアクティブになっているタイマーがあれば、削除されたタイマーの次に起動されたタイマーが現行のタイマーになります。

アクティブ・タイマーの数を表示するには、句を指定しないでTIMINGのみを入力します。TIMINGについては、SET AUTOTRACEも参照してください。

使用方法

このデータを使用して、特定の期間中に実行されるコマンドまたはブロックに関するパフォーマンスを分析できます。

各SQLコマンドまたはPL/SQLブロックを実行した後、自動的にTIMINGのデータを表示する場合の詳細は、SET TIMINGコマンドを参照してください。

すべてのタイマーを削除するには、CLEAR TIMINGコマンドを使用します。

例

SQL_TIMERという名前のタイマーを作成するには、次のように入力します。

```
TIMING START SQL_TIMER
```

現行のタイマーのタイトルおよび累積時間を表示するには、次のように入力します。

```
TIMING SHOW
```

現行のタイマーのタイトルおよび累積時間を表示して、そのタイマーを削除するには、次のように入力します。

```
TIMING STOP
```

12.49 TTITLE

構文

```
TTI [TLE] [printspec [text | variable] ...] [ON | OFF]
```

*printspec*には、テキストの配置および書式設定に使用される次の句のうちの1つ以上を指定します。

```
BOLD CE[NTER] COL n FORMAT text LE[FT] R[IGHT] S[KIP] [n] TAB n
```

指定したタイトルを各レポート・ページの上部に配置して書式設定します。TTITLEの現行の定義を表示するには、句を指定しないでTTITLEのみを入力します。TTITLEコマンドの後に引用符で囲まれた単一の語または文字列のみを指定した場合は、旧形式のTTITLEが使用されます。

旧形式のTTITLEの詳細は、「[TTI\[TLE\] text\(旧形式は廃止\)](#)」を参照してください。

項

これらの項および句は、BTITLEコマンドにも適用されます。

text

タイトル・テキストを指定します。1行に2つ以上の語を入れるには、テキストを一重引用符で囲んで入力します。

variable

置換変数、または次のシステム管理値のいずれかを指定します。SQL.LNO(カレント行の番号)、SQL.PNO(現行のページ番号)、SQL.RELEASE(現行のOracle Databaseのリリース番号)、SQL.SQLCODE(現行のエラー・コード)またはSQL.USER(現行のユーザー名)。

これらの値の1つを出力するには、適切な変数をタイトルの中で参照します。FORMAT句で*variable*の形式を設定できます。

SQL*Plusの置換変数(& variables)は、TTITLEの実行前に展開されます。生成された文字列はTTITLEテキストとして格納されます。後続の各結果ページの生成中に予期しない結果が発生して、展開された変数の値自身が置換変数として解釈される場合があります。

TTITLEコマンドでのこの二重置換を回避するには、各結果ページで置換される変数に&接頭辞を使用しないでください。置換変数を使用してTTITLEに未変更のテキストを挿入する場合は、1回のみ置換されるようにテキストを引用符で囲みます。

OFF

定義に影響を与えないで、タイトルをオフ(非表示)にします。

ON

タイトルをオン(表示)にします。上部タイトルを定義すると、SQL*Plusは自動的にTTITLEをONに設定します。

COL *n*

カレント行の列*n*までインデントします(列*n*を過ぎている場合は、後退します)。ここでいう「列」とは、印刷位置のことで、表の列ではありません。

S[KIP] [*n*]

新規行の先頭まで*n*回スキップします。*n*を省略すると、1回スキップします。*n*に0を入力すると、カレント行の先頭まで戻ります。

TAB *n*

n 列分前方に(n に負の値を入力した場合は後方に)スキップします。ここでいう「列」とは、印刷位置のことで、表の列ではありません。

LE[FT] | CE[NTER] | R[IGHT]

カレント行のデータを、それぞれ左揃え、中央揃えおよび右揃えにします。SQL*Plusは、*printspec*の終わりまで、あるいは次のLEFT、CENTER、RIGHTまたはCOLコマンドまでのデータ項目を、1グループとして整列させます。CENTERおよびRIGHTでは、SET LINESIZE値を使用し、後続のデータ項目の位置が計算されます。

BOLD

データを太字で印刷します。SQL*Plusは、端末上で同じデータを3回続けて出力することによって、太字印刷を行います。一部のオペレーティング・システムでは、SQL*Plusがプリンタにテキストを太字でなく3回連続で印刷するように指示することがあります。

FORMAT *text*

次のFORMAT句まで、またはこのコマンドの終わりまでの、後続のデータ項目の書式を決定する書式モデルを指定します。書式モデルは、A10や\$999などのテキスト定数である必要があります。書式設定および有効な書式モデルの詳細は、[「COLUMN」](#)コマンドを参照してください。

書式モデルのデータ型が、指定したデータ項目のデータ型と一致しない場合、FORMAT句はその項目には影響しません。

指定したデータ項目の前に適切な書式モデルがない場合、SQL*Plusは、SET NUMFORMATで指定された書式を使用して、NUMBER値を出力します。また、SET NUMFORMATが指定されていない場合はデフォルトの書式に従って、NUMBER値を出力します。SQL*Plusは、デフォルトの書式に従ってDATE値を出力します。

現行のTTITLE定義を表示するには、句を指定しないでTTITLEのみを入力します。

使用方法

最初に現れるテキストの前に*printspec*句を入力しない場合、TTITLEはテキストを左揃えにします。有効な*printspec*句(LEFT、SKIP、COLなど)がコマンド名のすぐ後に続いている場合、SQL*Plusは新形式でTTITLEを解釈します。

上部タイトルの列およびDATE値の出力については、[「COLUMN」](#)を参照してください。

*printspec*には、定数および変数をいくつでも指定できます。SQL*Plusは、それらを指定した順序で表示し、それぞれの定数または変数をその直前の*printspec*句で指定されている位置および書式に設定します。

TTITLEに指定するタイトルの長さは、2400文字を超えないようにしてください。

一重引用符で囲まれたタイトル・テキスト文字列の内部では、継続文字(ハイフン)は認識されません。これを認識させるには、次のように、継続文字を引用符の外側に置く必要があります。

```
TTITLE CENTER 'Summary Report for' -  
> ' the Month of May'
```

例

「Monthly Analysis」を上部タイトルとして定義して左揃え、日付を中央揃え、ページ番号を3桁書式で右揃え、次の行の中央に「Data in Thousands」を表示するには、次のように入力します。

```
TTITLE LEFT 'Monthly Analysis' CENTER '01 Jan 2003' -  
RIGHT 'Page:.' FORMAT 999 SQL.PNO SKIP CENTER -  
'Data in Thousands'
```

Monthly Analysis	01 Jan 2003	Page: 1
Data in Thousands		

上部タイトルの定義を変更しないで非表示にするには、次のように入力します。

TTITLE OFF

12.50 UNDEFINE

構文

```
UNDEF[INE] variable ...
```

*variable*には、定義を参照する置換変数の名前を指定します。

明示的に(DEFINEコマンドを使用して)定義したか、または暗黙的に(STARTコマンドでの引数を使用して)定義した1つ以上の置換変数を削除します。

例

POSという名前の置換変数の定義を削除するには、次のように入力します。

```
UNDEFINE POS
```

MYVAR1およびMYVAR2という名前の2つの置換変数の定義を削除するには、次のように入力します。

```
UNDEFINE MYVAR1 MYVAR2
```

12.51 VARIABLE

構文

```
VAR[iable] [variable [type [=value]]]
```

PL/SQLで参照可能なバインド変数を宣言するか、1つまたはすべての変数の現在の表示特性を表示します。

*type*には、次のいずれかを指定します。

```
NUMBER CHAR CHAR (n [CHAR | BYTE]) NCHAR NCHAR (n) VARCHAR2 (n [CHAR | BYTE]) NVARCHAR2 (n) CLOB NCLOB  
REFCURSOR BINARY_FLOAT BINARY_DOUBLE
```

引数なしでVARIABLEを指定すると、セッション内で宣言されているすべての変数が表示されます。VARIABLEの後に変数名のみを指定すると、その変数が表示されます。

CLOBおよびNCLOBバインド変数に使用されるリソースを解放するには、次のように入力して一時LOBを手動で解放する必要があります。

```
EXECUTE DBMS_LOB.FREETEMPORARY (:cv)
```

バインド変数の詳細は、[バインド変数の使用方法について](#)を参照してください。PL/SQLの詳細は、[『Oracle Database PL/SQL言語リファレンス』](#)を参照してください。

項

```
variable
```

作成するバインド変数の名前を指定します。

```
value
```

入力バインディングの変数に値を割り当てることができます。

```
NUMBER
```

NUMBER型で固定長の変数が作成されます。

```
CHAR
```

CHAR(文字)型で長さ1の変数が作成されます。

```
CHAR (n[CHAR | BYTE])
```

CHAR型で長さが*n*バイトまたは*n*文字の変数が作成されます。*n*の最大値は2000バイトで、最小値は1バイトまたは1文字です。キャラクタ・セマンティクスを持つCHAR変数の最大値*n*は、選択した文字セットで各文字を格納するために必要なバイト数で決定されますが、最大2000バイトです。長さセマンティクスは、長さ修飾子CHARまたはBYTEで決定されます。明示的に記述しない場合は、環境変数NLS_LENGTH_SEMANTICSの値がバインド変数に適用されます。変数定義の段階で長さセマンティクスを明示的に記述すると、NLS_LENGTH_SEMANTICS設定より常に優先されます。

```
NCHAR
```

NCHAR(各国語キャラクタ)型で長さ1の変数が作成されます。

```
NCHAR (n)
```

NCHAR型で長さが*n*文字の変数が作成されます。*n*の最大値は、選択した文字セットで各文字を格納するために必要なバイ

ト数で決定されますが、最大2000バイトです。SQL*PlusセッションがOracle8i以下のサーバーに接続している場合、またはSQLPLUSCOMPATIBILITYシステム変数が9.0.0未満に設定されている場合は例外となります。この場合、長さ n は、選択した各国語文字セットによってバイト単位または文字単位になりますが、いずれの場合も最大値は2000バイトです。

VARCHAR2 (n [CHAR | BYTE])

VARCHAR2型で長さ n バイトまたは n 文字までの変数が作成されます。 n の最大値は32KB (ノートを参照)で、最小値は1バイトまたは1文字です。文字セマンティクスを持つVARCHAR2変数 n の最大値は、選択した文字セットで各文字を格納するために必要なバイト数で決定されますが、最大32KBです。長さセマンティクスは、長さ修飾子CHARまたはBYTEで決定されます。明示的に記述しない場合は、環境変数NLS_LENGTH_SEMANTICSの値がバインド変数に適用されます。変数定義の段階で長さセマンティクスを明示的に記述すると、NLS_LENGTH_SEMANTICS設定より常に優先されます。

ノート:

デフォルトでは、VARCHAR2 の最大長は 4000 バイトです。4000 バイトを超える最大長の使用を試みると、ORA-01460 : 「リクエストされた変換はできません。」が発生します。

32K の最大長を有効にするには、init. ora ファイルに MAX_STRING_SIZE=extended パラメータを追加する必要があります。



NVARCHAR2 (n)

NVARCHAR2 型で長さ n 文字までの変数が作成されます。 n の最大値は、選択した文字セットで各文字を格納するために必要なバイト数で決定されますが、最大 32KB (ノートを参照)です。SQL*Plus セッションが Oracle8i 以下のサーバーに接続している場合、または SQLPLUSCOMPATIBILITY システム変数が 9.0.0 未満に設定されている場合は例外となります。この場合、長さ n は、選択した各国語文字セットによってバイト単位または文字単位になりますが、いずれの場合も最大値は 32KB です。

ノート:

デフォルトでは、NVARCHAR2 の最大長は 4000 バイトです。4000 バイトを超える最大長の使用を試みると、ORA-01460 : 「リクエストされた変換はできません。」が発生します。



32K の最大長を有効にするには、init. ora ファイルに MAX_STRING_SIZE=extended パラメータを追加する必要があります。

CLOB

CLOB 型の変数が作成されます。

NCLOB

NCLOB型の変数が作成されます。

REFCURSOR

REF CURSOR型の変数が作成されます。

BINARY_FLOAT型の変数が作成されます。BINARY_FLOATは、米国電気電子技術者協会(IEEE)の2進浮動小数点演算についてのIEEE規格(IEEE規格754-1985)に実質的に準拠する浮動小数点数です。

BINARY_DOUBLE

BINARY_DOUBLE型の変数が作成されます。BINARY_DOUBLEは、米国電気電子技術者協会(IEEE)の2進浮動小数点演算についてのIEEE規格(IEEE規格754-1985)に実質的に準拠する浮動小数点数です。

使用方法

バインド変数は、ストアド・プロシージャに対するパラメータとして使用されます。また、無名PL/SQLブロックの中で直接参照できます。

VARIABLEで作成したバインド変数の値を表示するには、PRINTコマンドを使用します。詳細は、「[PRINT](#)」を参照してください。

VARIABLEで作成したバインド変数の値を自動的に表示するには、SET AUTOPRINTコマンドを使用します。詳細は、[SET AUTOP\[RINT\] {ON | OFF}](#)を参照してください。

バインド変数は、PL/SQLブロックに入れる場合を除いて、COPYコマンドまたはSQL文の中で使用できません。かわりに、置換変数を使用します。

VARIABLE ... CLOBまたはNCLOBコマンドを実行すると、SQL*Plusは、LOBロケータをバインド変数に対応付けます。PL/SQLブロック内のSELECT clob_column INTO :cv文を実行すると、LOBロケータには自動的にデータが挿入されます。SQL*Plusを終了すると、SQL*PlusはLOBロケータをクローズします。

CLOBおよびNCLOBバインド変数に使用されるリソースを解放するには、次のように入力して一時LOBを手動で解放する必要があります。

```
EXECUTE DBMS_LOB.FREETEMPORARY (:cv)
```

すべての一時LOBは、SQL*Plusの終了時に解放されます。

CLOBまたはNCLOBバインド変数を出力するときに、SQL*PlusのSETコマンド(SET LONG、SET LONGCHUNKSIZE、SET LOBOFFSETなど)を使用して、バッファのサイズを制御できます。

SQL*PlusのREFCURSORのバインド変数を使用すると、PL/SQL 2.3以上のカーソル変数を参照できます。また、PL/SQL出力をSQL*Plusで書式設定できます。PL/SQLのカーソル変数の詳細は、「[カーソル変数](#)」を参照してください。

VARIABLE ... REFCURSORコマンドを実行する場合、SQL*Plusはカーソル・バインド変数を作成します。PL/SQLブロック内のバインド変数を参照するOPEN ... FOR SELECT文によって、カーソルが自動的にオープンされます。SQL*Plusは、バインド変数用のPRINT文の処理が完了した後、または終了時に、カーソルをクローズします。

SQL*Plusの書式設定コマンド(BREAK、COLUMN、COMPUTE、SETなど)を使用して、REFCURSORの印刷出力を書式設定できます。

REFCURSORバインド変数を2回以上印刷するには、PL/SQL OPEN ... FOR文を再実行する必要があります。

例

次の例では、バインド変数を作成し、その値を変更して、現在の値を表示する方法を示します。

バインド変数を作成するには、次のように入力します。

```
VARIABLE ret_val NUMBER
```


SQL*Plusでこのバインド変数を変更するには、PL/SQLブロックを使用する必要があります。

```
BEGIN
  :ret_val:=4;
END;
/
```

PL/SQL procedure successfully completed.

SQL*Plusのバインド変数の値を表示するには、次のように入力します。

```
PRINT ret_val
```

```
RET_VAL
-----
      4
```

次の例では、バインド変数を作成し、その変数を関数から戻される値に設定する方法を示します。

```
VARIABLE id NUMBER
BEGIN
  :id := EMP_MANAGEMENT.HIRE
    ('BLAKE', 'MANAGER', 'KING', 2990, 'SALES');
END;
/
```

ストアド・プロシージャから戻された値がバインド変数:idに設定されています。この変数は、PRINTコマンドを使用して表示したり、後続のPL/SQLサブプログラムで使用できます。

次の例では、バインド変数を自動的に表示する方法を示します。

```
SET AUTOPRINT ON
VARIABLE a REFCURSOR
BEGIN
  OPEN :a FOR SELECT LAST_NAME, CITY, DEPARTMENT_ID
    FROM EMP_DETAILS_VIEW
    WHERE SALARY > 12000
    ORDER BY DEPARTMENT_ID;
END;
/
```

PL/SQL procedure successfully completed.

LAST_NAME	CITY	DEPARTMENT_ID
Hartstein	Toronto	20
Russell	Oxford	80
Partners	Oxford	80
King	Seattle	90
Kochhar	Seattle	90
De Haan	Seattle	90

6 rows selected.

前述の例では、変数を表示するためにPRINTコマンドを実行する必要はありません。

次の例では、いくつかの変数を作成します。

```
VARIABLE id NUMBER
VARIABLE txt CHAR (20)
VARIABLE myvar REFCURSOR
```

定義済変数を表示するには、引数を指定せずにVARIABLEを入力します。

```
VARIABLE

variable id
datatype NUMBER

variable txt
datatype CHAR(20)

variable myvar
datatype REFCURSOR
```

次の例では、1つの変数を表示します。

```
VARIABLE txt

variable txt
datatype CHAR(20)
```

次の例では、入力バインディングの変数に対する値の割当てを示します。

```
VARIABLE tmp_var VARCHAR2(10)=Smith
```

次の例では、前の例と同じ結果を得るための代替方法を示します。

```
VARIABLE tmp_var VARCHAR2(10)
VARIABLE tmp_var=Smith
EXECUTE DBMS_OUTPUT.PUT_LINE(:tmp_var)
```

次の例では、個人別に給与を表示し、1か月の給与が\$12,000を超える従業員の部門別の給与を計算するレポートを作成する方法を示します。

```
VARIABLE rc REFCURSOR
BEGIN
  OPEN :rc FOR SELECT DEPARTMENT_NAME, LAST_NAME, SALARY
  FROM EMP_DETAILS_VIEW
  WHERE SALARY > 12000
  ORDER BY DEPARTMENT_NAME, LAST_NAME;
END;
/

PL/SQL procedure successfully completed.

SET PAGESIZE 100 FEEDBACK OFF
TTITLE LEFT '*** Departmental Salary Bill ***' SKIP 2
COLUMN SALARY FORMAT $999,990.99 HEADING 'Salary'
COLUMN DEPARTMENT_NAME HEADING 'Department'
COLUMN LAST_NAME HEADING 'Employee'
COMPUTE SUM LABEL 'Subtotal:' OF SALARY ON DEPARTMENT_NAME
COMPUTE SUM LABEL 'Total:' OF SALARY ON REPORT
BREAK ON DEPARTMENT_NAME SKIP 1 ON REPORT SKIP 1
PRINT rc
```

*** Departmental Salary Bill ***

DEPARTMENT_NAME	Employee	Salary
Executive	De Haan	\$17,000.00
	King	\$24,000.00

	Kochhar	\$17,000.00

Subtotal:		\$58,000.00
Marketing	Hartstein	\$13,000.00

Subtotal:		\$13,000.00
Sales	Partners	\$13,500.00
	Russell	\$14,000.00

Subtotal:		\$27,500.00

Total:		\$98,500.00

次の例では、入力バインドを作成してCLOBデータをCLOB列に挿入する方法を示します。

```
SQL> create table xyz (col1 clob);

Table created.

SQL> var abc varchar2(100)='This is a clob input'
SQL> insert into xyz values (:abc);

1 row created.
```

次の例では、CLOB列を含むレポートを作成し、SET LOBOFFSETコマンドを使用してそのレポートを表示する方法を示します。

CLOB型のclob_colという名前の列を含むclob_tabという名前の表を、すでに作成してあるとします。clob_colには、次のデータが含まれています。

```
Remember to run the Departmental Salary Bill report each month. This report
contains confidential information.
```

col_clob列にデータを表示するレポートを作成するには、次のように入力します。

```
VARIABLE T CLOB
BEGIN
  SELECT CLOB_COL INTO :T FROM CLOB_TAB;
END;
/

PL/SQL PROCEDURE SUCCESSFULLY COMPLETED
```

clob_col列から200文字を出力するには、次のように入力します。

```
SET LINESIZE 70
SET LONG 200
PRINT T

T
-----
Remember to run the Departmental Salary Bill report each month This r
eport contains confidential information.
```

出力位置を21番目の文字に設定するには、次のように入力します。

```
SET LOBOFFSET 21
```

PRINT T

T

Departmental Salary Bill report each month This report contains confi
dential information.

12.52 WHENEVER OSERROR

構文

```
WHENEVER OSERROR {EXIT [SUCCESS | FAILURE | n | variable | :BindVariable] [COMMIT | ROLLBACK] |  
CONTINUE [COMMIT | ROLLBACK | NONE]}
```

オペレーティング・システムのエラー(ファイル書込みエラーなど)が発生した場合に、指定した操作(デフォルトはSQL*Plusを終了)を実行します。

項

```
[SUCCESS | FAILURE | n | variable | :BindVariable]
```

オペレーティング・システムのエラーが検出されたら、すぐに指定した処理を実行するように、SQL*Plusに指示します。成功または失敗を示すコード、オペレーティング・システム障害コード、またはユーザー指定の数値や変数をSQL*Plusが戻すように指定することもできます。

```
EXIT [SUCCESS | FAILURE | n | variable | :BindVariable]
```

オペレーティング・システムのエラーが検出されたらすぐに終了するように、SQL*Plusに指示します。成功または失敗を示すコード、オペレーティング・システム障害コード、またはユーザー指定の数値や変数をSQL*Plusが戻すように指定することもできます。詳細は、「[EXIT](#)」を参照してください。

```
CONTINUE
```

EXITオプションをオフにします。

```
COMMIT
```

終了または継続する前にCOMMITを実行し、データベースに対する保留中の変更を保存するようにSQL*Plusに指示します。

```
ROLLBACK
```

終了または継続する前にROLLBACKを実行し、データベースに対する保留中の変更を破棄するようにSQL*Plusに指示します。

```
NONE
```

継続する前に何の操作もしないようにSQL*Plusに指示します。

使用方法

WHENEVER OSERRORコマンドを入力しない場合、SQL*Plusのデフォルト動作では、オペレーティング・システムのエラーが発生すると、何も処置せずに処理が続行されます。

WHENEVER SQLERRORコマンドを入力しない場合、SQL*Plusのデフォルト動作では、SQLエラーが発生すると、何も処置せずに処理が続行されます。

例

出力ファイルからの読取り時に障害が発生した場合、次のスクリプト内のコマンドは、SQL*Plusを終了し、保留中の変更があればすべてコミットします。

```
WHENEVER OSERROR EXIT  
START no_such_file
```

OS Message: No such file or directory
Disconnected from Oracle.....

12.53 WHENEVER SQLERROR

構文

```
WHENEVER SQLERROR {EXIT [SUCCESS | FAILURE | WARNING | n | variable | :BindVariable] [COMMIT | ROLLBACK] | CONTINUE [COMMIT | ROLLBACK | NONE]}
```

SQLコマンドまたはPL/SQLブロックでエラーが発生した場合に、指定した操作(デフォルトはSQL*Plusを終了)を実行します。

項

```
[SUCCESS | FAILURE | WARNING | n | variable | :BindVariable]
```

SQLコマンドまたはPL/SQLブロックでエラーが検出されたら、すぐに(ただし、エラー・メッセージを出力した後で)指定した操作を実行するようにSQL*Plusに指示します。SQL*Plusエラーの場合は、SQL*Plusは終了しません。

```
EXIT [SUCCESS | FAILURE | WARNING | n | variable | :BindVariable]
```

SQLコマンドまたはPL/SQLブロックでエラーが検出されたらすぐに(ただし、エラー・メッセージを出力した後で)終了するように、SQL*Plusに指示します。SQL*Plusエラーの場合は、SQL*Plusは終了しません。WHENEVER SQLERRORのEXIT句は、EXITコマンドと同じ構文規則に従います。詳細は、「[EXIT](#)」を参照してください。

```
CONTINUE
```

EXITオプションをオフにします。

```
COMMIT
```

終了または継続する前にCOMMITを実行し、データベースに対する保留中の変更を保存するようにSQL*Plusに指示します。

```
ROLLBACK
```

終了または継続する前にROLLBACKを実行し、データベースに対する保留中の変更を破棄するようにSQL*Plusに指示します。

```
NONE
```

継続する前に何の操作もしないようにSQL*Plusに指示します。

使用方法

WHENEVER SQLERRORコマンドは、SQLコマンドまたはPL/SQLブロックのエラーによってトリガーされます。SQL*Plusコマンド・エラーではトリガーされません。

例

次のスクリプト中のコマンドは、SQL UPDATEコマンドが失敗すると、SQL*Plusを終了し、SQLエラー・コードを戻します。

```
WHENEVER SQLERROR EXIT SQL.SQLCODE
UPDATE EMP_DETAILS_VIEW SET SALARY = SALARY*1.1;
```

次の例は、SQL*Plusコマンドでエラーが発生した後は、WHENEVER SQLERRORコマンドは実行されませんが、SQLコマンドまたはPL/SQLブロックでエラーが発生した場合は、実行されることを示します。

```
WHENEVER SQLERROR EXIT SQL.SQLCODE
column LAST_name heading "Employee Name"

Unknown COLUMN option "heading"
```

```
SHOW non_existed_option
```

次のPL/SQLブロック・エラーの場合、SQL*Plusが終了し、SQLエラー・コードが戻ります。

```
WHENEVER SQLERROR EXIT SQL.SQLCODE
begin
  SELECT COLUMN_DOES_NOT_EXIST FROM DUAL;
END;
/

SELECT COLUMN_DOES_NOT_EXIST FROM DUAL;
      *
ERROR at line 2:
ORA-06550: line 2, column 10:
PLS-00201: identifier 'COLUMN_DOES_NOT_EXIST' must be declared
ORA-06550: line 2, column 3:
PL/SQL: SQL Statement ignored

Disconnected from Oracle.....
```


12.54 XQUERY

構文

```
XQUERY xquery_statement
```

SQL*PlusのXQUERYコマンドを使用すると、指定したデータベースでXQuery 1.0問合せを実行できます。XQUERYは、Oracle Database 10gリリース2以上のバージョンでサポートされています。それより前のOracle DatabaseでXQUERYを使用しようとすると、エラーが発生します。

```
SP2-614 Server version too low
```

項

```
xquery_statement
```

実行するXQUERY文を指定します。この文は、標準のXQuery構文で入力します。XQUERY文は、スラッシュ「/」で終了します。

使用方法

SQL*PlusコマンドXQUERYをXQuery文の接頭辞として設定し、スラッシュ「/」でXQUERYコマンドを終了します。XQUERYは、SQL*Plusキーワードです。XQueriesを他のツールで実行する場合は、キーワードが必要ない場合があります。XQUERYコマンドのXML出力は、アクティブなSETコマンド・オプションに従ってネイティブXMLとして表示されます。通常、SET LONGを設定する必要があります。次の設定を行うと有効な場合があります。

- デフォルトの80文字より長い行の行サイズ(SET LINESIZE)。
- デフォルトの80文字より長い行のLOB、LONGおよびXMLTypeのサイズ(SET LONG)。
- 出力と一致する出力ページ・サイズ(SET PAGESIZE)。
- 「Result Sequence」列ヘッダーを抑制するヘッダーの表示(SET HEADING OFF)。

XQUERYコマンドには、アクティブなデータベース接続が必要です。このコマンドは、SQLPLUS /NOLOGでは動作しません。

XQUERYコマンドでは、バインド変数はサポートされていません。

XQUERYコマンドには固有の4つのSETコマンドがあります。SHOW XQUERYコマンドを実行すると、これらの設定の状態が表示されます。これらを次に示します。

- [SET XQUERY BASEURI {text}](#)
- [SET XQUERY ORDERING {UNORDERED | ORDERED | DEFAULT}](#)
- [SET XQUERY NODE {BYVALUE | BYREFERENCE | DEFAULT}](#)
- [SET XQUERY CONTEXT {text}](#)

例

次のスクリプトのXQuery文は、HRスキーマのEMP_DETAILS_VIEWビューを問い合わせます。

```
set long 160
set linesize 160
xquery for $i in fn:collection('oradb:/SCOTT/EMP_DETAILS_VIEW') return $i
/
```

Result Sequence

<ROW><EMPNO>7369</EMPNO><ENAME>SMITH</ENAME><JOB>CLERK</JOB><MGR>7902</MGR><HIREDATE>17-
DEC-80</HIREDATE><SAL>800</SAL><DEPTNO>20</DEPTNO></ROW>

14 item(s) selected.

13 SQL*Plusのエラー・メッセージ

ここでは、SQL*Plusで表示される、接頭辞SP2-およびCPY-のエラー・メッセージについて説明します。

- [SQL*Plusのエラー・メッセージ](#)
- [COPYコマンドのメッセージ](#)

Oracle Databaseで表示される接頭辞ORA-、TNS-およびPLS-のエラー・メッセージについては、『[Oracle Databaseエラー・メッセージ](#)』を参照してください。

13.1 SQL*Plusのエラー・メッセージ

SP2-00002 ACCEPT 文には、変数名の指定が必要です。

原因: ACCEPT コマンドの後に必要な変数名が欠落しています。

処置: 入力値を格納する可変引数を使用して ACCEPT コマンドを再入力してください。

SP2-00003 この ACCEPT コマンドは command_string で始まっており、形式が正しくありません。

原因: ACCEPT コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、ACCEPT コマンドの構文を確認してください。

SP2-00004 追加すべきものがありません。

原因: APPEND コマンドの後に入力されるテキストが指定されませんでした。

処置: 指定されたテキストを使用して APPEND コマンドを再入力してください。

SP2-00006 計算をフォーマットするには空間が不足しています。

原因: 計算をフォーマットするためのメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00015 ブレークが定義されていません。

原因: ブレークが定義されていません。

処置: ブレークを定義してください。正しいオプションについては、BREAK コマンドの構文を確認してください。

SP2-00016 ブレーク指定は、ON/BY または ACROSS キーワードで始めなければなりません。

原因: BREAK コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、BREAK コマンドの構文を確認してください。

SP2-00017 keyword_name キーワードの後に列名が欠落しています。

原因: 指定されたキーワードの後に列名がありませんでした。

処置: 指定されたキーワードの後に列名を入力してください。

SP2-00019 option_name オプションの数値引数が無効です。

原因: 指定されたオプションに無効な数値引数が使用されました。

処置: 引数を修正し、再試行してください。

SP2-00020 'column_name'に使用可能な記憶域がありません。

原因: エラーが発生しました。SQL*Plus は、BREAK コマンドにメモリーを割り当てられませんでした。

処置: いくつかのアプリケーションをクローズして、より多くのメモリーを割り当ててください。

SP2-00022 buffer_name バッファ変数の変更領域に領域を割り当てできません。

原因: 内部エラーが発生しました。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00023 文字列が見つかりません。

原因: 指定された検索文字列が見つかりませんでした。

処置: 検索文字列を確認して、それが有効であることを確認してください。

SP2-00024 変更するものがありません。

原因: CHANGE コマンドの使用時、SQL バッファ内が空でした。

処置: CHANGE コマンドを使用する前に、SQL バッファが空でないことを確認してください。

SP2-00025 変更文字列が無効です。

原因: CHANGE コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、CHANGE コマンドの構文を確認してください。

SP2-00026 削除する行はありません。

原因: DEL コマンドの使用時、SQL バッファ内が空でした。

処置: DEL コマンドを使用する前に、SQL バッファが空でないことを確認してください。

SP2-00027 入力が長すぎます(>max_characters 文字)。この行は無視されました。

原因: 指定された入力値が長すぎました。

処置: 文字数を減らして再入力してください。

SP2-00029 コマンド・バッファ領域を使い果たしました。

原因: サイズの大きい SQL または PL/SQL スクリプトが SQL*Plus から実行されています。

処置: 次のいずれかの方法で、SQL 文または PL/SQL ブロックのサイズを減らしてください。

- 余分な空白およびコメントを減らす。
- 再コード化し、コマンドを減らすか、または変数名を短くする(あるいはその両方)。
- ブロックのセクションをストアド・プロシージャまたは、パッケージ・プロシージャに置き、ブロックからこれらのプロシージャをコールする。

SP2-00030 他の行を入れる領域がありません。

原因: SQL 文または PL/SQL ブロックの最大行数を超えています。

処置: 行数を減らし、再試行してください。

SP2-00038 コマンドが長すぎます(max_characters 文字)。

原因: 指定されたコマンドを入力しましたが、長すぎました。

処置: 制限の詳細は、コマンド構文を確認してください。

SP2-00039 command_name への代入中にコマンドラインがオーバーフローしました。

原因: コマンドラインの最大長を超えています。

処置: コマンドで使用する置換変数のデータの長さを減らしてください。

SP2-00042 不明なコマンドです(command_name)。行の残りは無視されました。

原因: 入力したコマンドは無効です。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00044 既存コマンドのリストを見るには HELP を、元に戻るには EXIT を入力します。

原因: 不明なコマンドが入力されました。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00045 * column_name が定義されていません。

原因: 列が定義されていません。

処置: 処置は必要ありません。

SP2-00046 column_name が定義されていません。

原因: 指定された列名が定義されていません。

処置: 有効な列名を使用して再試行してください。

SP2-00047 option_name オプションの数値が無効です。

原因: このオプションに無効な数値が使用されました。

処置: 有効な数値で操作を再試行してください。

SP2-00052 類似 column_name 'column_name'が定義されていません。

原因: 書式の基になる列が定義されていません。

処置: COLUMN コマンドを使用して、書式の基になる列が最初に定義されていることを確認してください。

SP2-00054 definition_name 定義を割り当てる領域がありません。無視されました。

原因: COLUMN コマンドを処理するためのメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00055 新規 definition_name の部分の割当て中に領域がなくなりました。旧定義が(ある場合)保持されました。

原因: 新しい定義を格納するためのメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00080 現在、COMPUTE は定義されていません。

原因: COMPUTE が定義されていません。

処置: COMPUTE を定義してください。正しいオプションについては、COMPUTE コマンドの構文を確認してください。

SP2-00081 一度に最大 number 個の COMPUTE 関数まで可能です。

原因: COMPUTE 関数の最大数を超えています。

処置: COMPUTE 関数の数を減らしてください。

SP2-00082 COMPUTE 関数が要求されていません。

原因: COMPUTE 関数が要求されていません。

処置: 処置は必要ありません。

SP2-00083 警告: COMPUTE オプション function_name が number 回指定されました。

原因: ラベルまたは関数が 2 回以上指定されました。

処置: 不要なラベルまたは関数を削除してください。

SP2-00084 COMPUTE ON キーワードはすでに指定されています。

原因: ON キーワードが 2 回以上指定されました。

処置: コマンドに ON キーワードを 1 回のみ指定してください。

SP2-00085 COMPUTE OF キーワードはすでに指定されています。

原因: OF キーワードが 2 回以上指定されました。

処置: コマンドに OF キーワードを 1 回のみ指定してください。

SP2-00087 column_name に COMPUTE 制御ブロックを割り当てる領域がありません。

原因: COMPUTE コマンドを処理するためのメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00088 keyword_name キーワードが欠落しています。使用方法: STORE {SET} filename[.ext] [CRE[ATE]|REP[LACE]|APP[END]]

原因: 文にキーワードが欠落しています。

処置: 使用するコマンドの正しいオプションについては、コマンドの構文を確認し、適切な位置にキーワードを使用してください。

SP2-00092 keyword_name キーワードの列が欠落しています。

原因: キーワードに列名が指定されませんでした。

処置: 列名を指定し、再試行してください。

SP2-00096 INTO 変数 variable_name を割り当てる領域がありません。

原因: COMPUTE コマンドを処理するためのメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00097 ON 列 column_name を割り当てる記憶域がありません。

原因: COMPUTE コマンドを処理するためのメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00098 column_name 用に COMPUTE ブロックを割り当てる記憶域がありません。

原因: COMPUTE コマンドを処理するためのメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00103 SQL バッファ内に実行対象がありません。

原因: SQL バッファ内に実行対象がありませんでした。

処置: 有効な SQL コマンドを入力してください。SQL*Plus コマンドは、SQL バッファには格納されません。

SP2-00105 エンティティ名が正しくないか欠落しています。

原因: GET または SAVE コマンドにファイル名が指定されていません。

処置: ファイル名を指定し、再試行してください。

SP2-00107 保存対象がありません。

原因: ファイルに内容を保存しようとしたときに、SQL バッファ内に保存対象がありませんでした。

処置: SQL コマンドを入力し、保存してください。SQL*Plus コマンドは、SQL バッファには格納されません。

SP2-00108 CREATE、REPLACE、APPEND、FILE というファイル名および略称は使用できません。

原因: FILE という単語を使用したファイル名、コマンド・キーワード CREATE、REPLACE、APPEND のいずれか、またはこれらのコマンド・キーワードのいずれかの略称を入力しようとしました。このコマンドで指定されたファイル名は許可されませんでした。

処置: 次のコマンド構文を確認し、有効なファイル名を入力してください。

command filename[.ext] [CR[EATE]|REP[LACE]|AP[PEND]]

command には SAVE、SPOOL または STORE SET を指定してください。

ファイル名として、コマンド・キーワード CREATE、REPLACE、APPEND、またはこれらのいずれかの略称、あるいは FILE という単語を使用する場合は、ファイル名を一重引用符で囲んでください。

SP2-00109 ファイル"file_name"に追加できません。

原因: SQL バッファの内容をファイルに追加しようとしたが、書き込めませんでした。次のような原因が考えられます。

- 定義ファイル作成時にエラーが検出された。
- SAVE 文で指定されたディレクトリ名が見つからない。
- システム・エラーのため、ファイルをオープンできない。

処置: 次の処置を行ってください。

- 出力先が有効であり、出力先デバイスに十分な空き領域があることを確認してください。
- 文のディレクトリ名に入力ミスがないことを確認してください。ディレクトリ名を修正して、文を再発行してください。

SP2-00110 保存ファイル"file_name"を作成できません。

原因: SQL バッファの内容をファイルに保存しようとしたが、書き込めませんでした。次のような原因が考えられます。

- 定義ファイル作成時にエラーが検出された。
- SAVE 文で指定されたディレクトリ名が見つからない。
- システム・エラーのため、ファイルをオープンできない。

処置: 次の処置を行ってください。

- 出力先が有効であり、出力先デバイスに十分な空き領域があることを確認してください。
- 文のディレクトリ名に入力ミスがないことを確認してください。ディレクトリ名を修正して、文を再発行してください。

SP2-00111 保存ファイル"file_name"をクローズできません。

原因: ファイルが使用中です。

処置: ファイルを他のプロセスから解放してください。

SP2-00116 SAVE コマンドが不正です。

原因: SAVE コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、SAVE コマンドの構文を確認してください。

SP2-00134 現在定義されている記号はありません。

原因: DEFINE 記号が定義されていません。

処置: 処置は必要ありません。

SP2-00135 記号 `symbol_name` は定義されていません。

原因: 指定された記号は定義されていません。

処置: 代入句、有効な記号または変数名を使用して DEFINE コマンドを再入力してください。

SP2-00136 DEFINE には等価符号(=)が必要です。

原因: DEFINE コマンドの記号または変数名の後に等価符号が必要です。

処置: 記号または変数名の後に等価符号を指定してください。

SP2-00137 DEFINE には等価符号の後に値が必要です。

原因: 変数または記号に値がありませんでした。SQL*Plus では、等価符号の後に記号または変数名に代入する値が必要です。

処置: 記号または変数に値を指定してください。

SP2-00138 置換変数"variable"を追加する領域がありません

原因: SQL*Plus セッションで定義される変数の最大数を超えました。

処置: この変数に領域を割り当てるために使用されていない変数の定義を解除して、コマンドを再実行してください。

SP2-00146 必要な動的領域(number_of_bytes バイト)を割当てできません。終了します。

原因: 内部エラーが発生しました。

処置: メッセージおよび番号を記録して、システム管理者に連絡してください。

SP2-0152 「Oracle が正常に機能していない可能性があります。」

原因: Oracle インスタンスへのセッションを初期化できませんでした。

処置: メッセージおよび番号をノートにとって、データベース管理者に連絡してください。

SP2-00157 3 回試行しましたが Oracle に接続できませんでした。SQL*Plus を終了します。

原因: 3 回試行しましたが、Oracle に接続できませんでした。

処置: ログインの詳細を検証して、再試行してください。

SP2-00158 不明な command_name オプションです。"option_name"

原因: 指定されたコマンドに無効なオプションが指定されました。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00160 "file_name"をオープンできません。

原因: 次のような原因が考えられます。

- 指定された位置に指定された名前のファイルが見つからない。
- ファイルをオープンするために必要な権限が不足している。
- システム・エラーのため、ファイルをオープンできない。

処置: 次の処置を行ってください。

- 指定されたファイル名が適切なディレクトリに格納されていることを確認してください。
- ファイルに必要なアクセス権限があることを確認してください。権限がない場合、権限を変更してください。
- オペレーティング・システムのドキュメントを参照するか、またはシステム管理者に連絡してください。

SP2-00161 行 line_number が切り捨てられました。

原因: ファイル内の行が長すぎました。

処置: 処置は必要ありません。または、行の長さを短くしてください。

SP2-00162 "file_name"をクローズできません。

原因: 指定されたファイルが使用されているため、クローズできませんでした。

処置: ファイルを他のプロセスから解放してください。

SP2-00171 HELP にアクセスできません。

原因: SQL*Plus のコマンドライン・ヘルプがこの Oracle インスタンスにインストールされていません。

処置: SQL*Plus のコマンドライン・ヘルプがこの Oracle インスタンスにインストールされていません。
sqlplus/admin/help/hlpbld.sql スクリプトを使用して、このデータベースにヘルプをインストールしてください。

```
sqlplus system @hlpbld.sql helpus.sql
```

SP2-00172 このトピックと一致する HELP が見つかりませんでした。

原因: 指定されたコマンドに使用可能なヘルプ情報がありません。

処置: 項目を表示する場合は、HELP INDEX を入力します。

SP2-0176 オプション?が無効です。

原因: このコマンドでは、オプション?は無効です。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00187 変数の割当てエラーが発生しました。

原因: 指定された変数の割当てが不適切でした。

処置: 正しいオプションについては、ACCEPT コマンドの構文を確認してください。

SP2-00223 buffer_name バッファ内に行がありません。

原因: バッファ内に格納されている行がありません。

処置: バッファに SQL 文を入力してください。

SP2-00224 開始行番号が無効です。

原因: 指定された行番号に誤りがあります。

処置: 行番号が正しいことを確認して、再試行してください。

SP2-00225 終了行番号が無効です。

原因: 指定された行番号に誤りがあります。

処置: 行番号が正しいことを確認して、再試行してください。

SP2-00226 行番号 current_line_number が無効です。

原因: 無効な行番号が指定されました。

処置: 有効な行番号で再入力してください。

SP2-00232 入力が長すぎます。number_of_characters 文字以内である必要があります。

原因: 入力値が長すぎました。

処置: 値のサイズを小さくして、再入力してください。

SP2-00233 number_of_attempts 回試行しましたがユーザーID を取得できません。コマンドを再試行してください。

原因: 3 回試行しましたが、SQL*Plus はログインできませんでした。

処置: ユーザーID およびパスワードが正しいことを確認して、再試行してください。

SP2-00240 variable_name に値を入力してください。

原因: SQL*Plus は、置換変数の値を見つけることができませんでした。

処置: 置換変数の入力を求めるプロンプトで値を入力してください。

SP2-00241 記号 symbol_name 用の領域がありません(定義されていません)。

原因: 記号用にメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00244 PAGE ブレーク内では PRINT コマンドを発行できません。

原因: PRINT コマンドは、PAGE ブレーク内では許可されていません。

処置: 正しいオプションについては、PRINT コマンドの構文を確認してください。

SP2-00245 出力用の一時記憶域を割当てできません。

原因: 印刷用の一時記憶域を割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00246 FORMAT 文字列"column_ format_name"が不正です。

原因: 列に無効な書式が指定されました。

処置: 列に有効な書式を指定してください。

SP2-00249 variable_name は出力用の変数タイプとして無効です。

原因: 指定された変数は、印刷には無効です。

処置: 変数タイプを確認して、コマンドを再入力してください。

SP2-00253 データ項目 line_number("data_item_name")が行に適合しません。

原因: カレント行のサイズ設定が小さすぎるため、指定されたデータ項目が行に適合しません。

処置: 項目を表示できるように、行サイズを増やしてください。

SP2-00258 変数 variable_name を列 column_name 用に作成できませんでした。

原因: 内部エラーまたはメモリー不足のために、指定された変数は、列に作成できませんでした。

処置: メモリー使用量を確認してください。

SP2-00259 変数 variable_name を COMPUTE INTO 用に作成できませんでした。

原因: 指定された変数は作成されませんでした。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00260 列 column_name 用の計算が一意に修飾されていません。表 table_name または table_name 用のものである可能性があります。計算は無視されました。

原因: 指定された列は、文で一意に修飾されていませんでした。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00262 CCBDEF ポインタ配列を割り当てる領域がありません。

原因: 内部メモリー・エラーが発生しました。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00263 column_name ON page/report/column_name 用に COMPUTE ブロックを割り当てる領域がありません。

原因: COMPUTE ブロックに割り当てるためのメモリーが不十分でした。

処置: 他のアプリケーションをクローズして、より多くのメモリーを割り当ててください。

SP2-00265 option_name を ON または OFF に設定する必要があります。

原因: 無効な SET オプション名が指定されました。

処置: SET オプションの 1 つとして ON または OFF のいずれかを再入力してください。

SP2-00266 内部エラー: バッファ(buffer_size)が l(buffer_limit)より小さくなっています。

原因: 内部エラーが発生しました。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00267 option_name オプション parameter_number が範囲(lower_range から

upper_range)外です。

原因: パラメータの値が指定された範囲外です。

処置: パラメータの制限を確認して、範囲内の値を入力してください。

SP2-00268 option_name オプションが有効数値ではありません。

原因: パラメータに数値(整数)以外の値が入力されました。

処置: 有効な数値(整数)を入力してください。

SP2-00271 variable_name はバッファ変数ではありません。

原因: 指定された変数はバッファとして定義されていません。

処置: バッファ変数名が正しいことを確認して、再試行してください。

SP2-00272 character_name 文字に英数字または空白は使用できません。

原因: SET コマンドの指定する文字には、英数字または空白は使用できません。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00277 entered_value 値が無効です。

原因: 入力した値に誤りがあります。

処置: 有効な値で再入力してください。

**SP2-0281 option_name に設定オプションが欠落しています。使用方法: SET SHIFT[INOUT]
[VIS[IBLE|INV[ISIBLE]]または使用方法: SET MARKUP HTML [ON|OFF] [HEAD
text] [BODY text] [TABLE text] [ENTMAP [ON|OFF]] [SPOOL [ON|OFF]]
[PRE[FORMAT] [ON|OFF]] [-M[ARKUP] ¥"HTML [ON|OFF] [HEAD text] [BODY
text]**

原因: コマンドに SET オプションが欠落しています。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-0306 オプションが無効です。使用方法: CONN[ECT] [login] [AS

{SYSDBA|SYSOPER}]ただし<login> ::=

<username>[/<password>][@<connect_string>] | /または使用方法: CONN[ECT]

username/password[@connect_identifier] [AS {SYSOPER|SYSDBA}]または:

CONN[ECT] /[@connect_identifier] AS {SYSOPER|SYSDBA}

原因: コマンドに無効なオプションが指定されました。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00308 スプール・ファイルをクローズできません。

原因: ファイルは、現在使用されています。

処置: ファイルを他のプロセスから解放してください。

SP2-00309 SQL*Plus コマンド・プロシージャが number_of_nested_procedures の深度までしかネストされていない可能性があります。

原因: ネストしたプロシージャまたはスクリプトの最大数に達しました。

処置: ネストしたプロシージャまたはスクリプトの数を減らしてください。

SP2-00310 ファイル file_name をオープンできません。

原因: 指定されたファイルをオープンできませんでした。

処置: ファイル名が有効であることを確認してください。

SP2-00311 必要な文字列が見つかりませんでした。

原因: SQL*Plus のコマンドの終わりに必要な文字列が見つかりませんでした。

処置: 有効な文字列を使用してコマンドを再試行してください。正しいオプションについては、コマンドの構文を確認してください。

SP2-00312 終了引用符(quote_type)が欠落しています。

原因: DESCRIBE コマンド・スキーマまたはオブジェクトに終了引用符がありませんでした。

処置: 対応する終了引用符で開始引用符をクローズしてください。

SP2-00317 必要な記号名が欠落しています。

原因: SQL*Plus で必要な記号が指定されていません。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00318 variable_name..."で始まる記号名が長すぎます(最大 max_name_length)。変数名 "variable_name"が不正です。

原因: 指定された変数名が名前の最大長を超えました。

処置: 記号名のサイズを小さくして、再入力してください。

SP2-00323 タイミング要素を追加する領域がありません。要求は拒否されました。

原因: TIMING コマンド実行中にメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00324 オペレーティング・システム・タイミング・エラーerror_option_number - 要求は拒否されました。

原因: オペレーティング・システム・エラーのために、TIMING コマンドの初期化に失敗しました。

処置: オペレーティング・システム・エラーを解決し、再試行してください。

SP2-00325 option_name に対するタイミング要素はありません。

原因: SHOW または STOP に記録されるタイマーがありません。

処置: タイマーが TIMING コマンドで作成されたことを確認してください。

SP2-00328 タイトル・バッファを割り当てる領域がありません。

原因: TTITLE または BTITLE コマンド実行中にメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-0331 SPOOL OUT が使用禁止になりました。

原因: SPOOL OUT を使用しようとしたが、サポートされていません。

処置: 対応策はありません。SPOOL OUT は、オペレーティング・システム・レベルで印刷がサポートされていないために使用禁止になった可能性があります。

SP2-00332 スプール・ファイルを作成できません。

原因: 次のような原因が考えられます。

- ファイルを作成する権限が不十分である。
- システム・エラーのため、ファイルを作成できない。

処置: 次の処置を行ってください。

- ファイルを作成できるように権限を変更してください。
- オペレーティング・システムのドキュメントを参照するか、またはシステム管理者に連絡してください。

SP2-00333 不正なスプール・ファイル名: "spool_name"(不正文字: 'character_name')

原因: SPOOL コマンドに無効なファイル名が入力されました。

処置: ファイル名を修正して、再入力してください。

SP2-00341 変数の置換中に行がオーバーフローしました(>number_of_characters 文字、行 line_number)。

原因: 置換変数を拡張した後、SQL バッファで文字の最大数を超えました。

処置: 置換変数の長さを短くし、再試行してください。

SP2-00357 一時記憶域が足りません。

原因: コマンド実行中にメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00359 メモリーを使い果たしました。

原因: コマンド実行中にメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00381 command_name は使用できません。

原因: 指定されたコマンドは実装されていません。

処置: 適切な SQL*Plus コマンドを使用してください。コマンドのリストおよびそれらの正しい構文の詳細は、ドキュメントを参照してください。

SP2-00382 command_name コマンドは使用できません。

原因: コマンドが認識されていないか、または使用禁止です。これは、コマンドが SQL*Plus で意味を持たない(SQL バッファの編集コマンドなど)場合、またはセキュリティ上の理由から許可されていない場合に発生します。

処置: スクリプトからコマンドを削除してください。コマンドのリストおよびそれらの正しい構文の詳細は、ドキュメントを参照してください。

SP2-00392 現行の編集バッファを UNDEFINE できません。

原因: 現行の編集バッファは UNDEFINE できません。

処置: 処置は必要ありません。

SP2-00394 バッファ名: "buffer_name"が不正です。

原因: バッファ名に不正な文字(たとえば、ハイフン(-))が含まれています。

処置: バッファ名から不正な文字を修正および削除してください。

SP2-00423 GET コマンドが不正です。

原因: GET コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00425 "値"は有効なデータ型ではありません

原因: ACCEPT コマンドに入力された値が、指定されたデータ型に対して無効でした。

処置: 有効な値を入力してください(例: NUMBER 変数の場合、123)。

SP2-00426 入力は number_of_characters 文字に切り捨てられました。

原因: SQL 文の最後の行に改行がありませんでした。

処置: 改行を挿入してください。

SP2-00495 FROM 句と TO 句の両方が欠落しています。少なくとも 1 つを指定してください。

原因: FROM および TO 句が COPY 文から欠落しています。

処置: 1 つ以上の句を指定してください。正しいオプションについては、コマンドの構文を確認してください。

SP2-00496 FROM 句の位置が正しくありません。

原因: FROM キーワードが COPY コマンドの不適切な位置にありました。

処置: 正しいオプションについては、COPY コマンドの構文を確認してください。

SP2-00497 TO 句の位置が正しくありません。

原因: TO キーワードが COPY コマンドの不適切な位置にありました。

処置: 正しいオプションについては、COPY コマンドの構文を確認してください。

SP2-00498 カッコ付きの列のリストか USING キーワードが欠落しています。

原因: カッコ付きのリストが列リスト内で欠落しているか、または USING キーワードが COPY コマンド内で欠落しています。

処置: 正しいオプションについては、COPY コマンドの構文を確認してください。

SP2-00499 APPEND キーワードの位置が正しくありません。

原因: APPEND キーワードが COPY コマンドの不適切な位置にありました。

処置: 正しいオプションについては、COPY コマンドの構文を確認してください。

SP2-00501 SELECT 文のエラー: Oracle_database_error_message

原因: COPY コマンドで無効な SELECT 文が見つかりました。

処置: 正しいオプションについては、COPY コマンドの構文を確認してください。

SP2-00513 CREATE キーワードの位置が正しくありません。

原因: CREATE キーワードが COPY コマンドの不適切な位置にありました。

処置: 正しいオプションについては、COPY コマンドの構文を確認してください。

SP2-00514 REPLACE キーワードの位置が正しくありません。

原因: REPLACE キーワードが COPY コマンドの不適切な位置にありました。

処置: 正しいオプションについては、COPY コマンドの構文を確認してください。

SP2-00515 最大列数(max_num_columns)を超えました。

原因: COPY コマンドの最大列数を超えました。

処置: 列数を減らし、再試行してください。

SP2-00516 無効な command_name 名"NULL"が検出されました。

原因: COLUMN コマンドまたは ATTRIBUTE コマンドで、無効な列名または NULL が指定されました。

処置: 有効な列名を使用して操作を再試行してください。

SP2-00517 カンマまたは右カッコが欠落しています。

原因: COPY コマンドで右カッコの欠落が検出されました。

処置: カンマまたは右カッコを追加して操作を再試行してください。

SP2-00518 USING 句が欠落しています。

原因: COPY コマンドの USING 句で USING キーワードが欠落しています。

処置: COPY コマンドの USING 句の前に USING キーワードを指定してください。

SP2-00519 FROM 文字列の Oracle Net @database 指定が欠落しています。

原因: COPY コマンドでコピーされるデータを含むデータベースの接続文字列が欠落しています。

処置: FROM 句を挿入して、デフォルト以外のソース・データベースを指定してください。

SP2-00520 TO 文字列の Oracle Net @database 指定が欠落しています。

原因: COPY コマンドで宛先の表を含むデータベースの接続文字列が欠落しています。

処置: TO 句を挿入して、デフォルト以外のソース・データベースを指定してください。

SP2-00526 INSERT キーワードの位置が正しくありません。

原因: COPY コマンドで INSERT キーワードの位置が正しくありませんでした。

処置: 正しいオプションについては、COPY コマンドの構文を確認してください。

SP2-00540 ファイル"file_name"はすでに存在しています。"SAVE filename[.ext] REPLACE"を使用してください。

原因: 指定されたファイルは、すでに存在しています。

処置: REPLACE オプションを使用して、既存のファイルを上書きしてください。または、別のファイル名を指定してください。

SP2-00544 プロダクト・ユーザー・プロファイルではコマンド"command_name"を使用できません

原因: このデータベースのスキーマに対して明示的に使用禁止にされているコマンドを使用しようとしました。

処置: プロダクト・ユーザー・プロファイル(PUP)表で、スキーマに対してこのコマンドが使用禁止に設定されている理由について、システム管理者に問い合わせてください。

SP2-00545 SET コマンドには引数が必要です。

原因: SET コマンドに引数が欠落しています。

処置: 正しいオプションについては、SET コマンドの構文を確認してください。

SP2-00546 ユーザー要求の割込みまたは EOF が検出されました。

原因: EOF に達したか、または[Ctrl]キーを押しながら[C]を押して処理を取り消しました。

処置: 処置は必要ありません。

SP2-00547 option_name オプション value が範囲(lower_value から upper_value)外です。

原因: 指定した SET オプションは範囲外です。

処置: SET オプションの範囲内で値を入力し、SET コマンドを再試行してください。

SP2-0548 使用方法: VAR[TABLE] [<variable> [NUMBER | CHAR | CHAR (n [CHAR|BYTE]) | VARCHAR2 (n [CHAR|BYTE]) | NCHAR | NCHAR (n) | NVARCHAR2 (n) | CLOB | NCLOB | REFCURSOR | BINARY_FLOAT | BINARY_DOUBLE][=value]]

原因: 不適切な構文で VARIABLE コマンドが入力されました。

処置: 正しい使用方法については、VARIABLE コマンドの構文を確認してください。

SP2-00549 使用方法: PRINT [:<variable> ...]

原因: 不適切な構文で PRINT コマンドが入力されました。

処置: 正しい使用方法については、PRINT コマンドの構文を確認してください。

SP2-0550 使用方法: SHOW ERRORS [{ANALYTIC VIEW | ATTRIBUTE DIMENSION | HIERARCHY | FUNCTION | PROCEDURE | PACKAGE | PACKAGE BODY | TRIGGER | VIEW | TYPE | TYPE BODY | DIMENSION| JAVA SOURCE | JAVA CLASS}] [schema.]name]

原因: 不適切な構文で SHOW ERRORS コマンドが入力されました。

処置: 正しいオプションについては、SHOW ERRORS コマンドの構文を確認してください。

SP2-00552 バインド変数"variable_name"が宣言されていません。

原因: 指定されたバインド変数は、宣言されていません。

処置: VARIABLE コマンドを実行し、使用したバインド変数が、SQL 文に存在するかどうかを確認

してください。バインド変数を使用した SQL 文を実行する前に、VARIABLE コマンドを使用して、各変数を宣言する必要があります。

**SP2-0556 ファイル名が無効です。使用方法: STORE {SET} filename[.ext]
[CRE[ATE]]|REP[LACE]]|APP[END]]または EDIT コマンドを完了できません。**

原因: ファイル名が欠落しているか、無効なファイル名が指定されています。

処置: ファイル名が指定されていることを確認してください。

SP2-00559 使用方法: EXEC[UTE] statement

原因: 不適切な構文で EXECUTE コマンドが入力されました。

処置: 正しい使用方法については、EXECUTE コマンドの構文を確認してください。

**SP2-00560 使用方法: DESCRIBE [schema.]object[.subobject]@db_link]
[column]**

原因: 不適切な構文で DESCRIBE コマンドが入力されました。

処置: 正しい使用方法については、DESCRIBE コマンドの構文を確認してください。

SP2-00561 オブジェクトが存在しません。

原因: DESCRIBE の対象として指定したオブジェクトはデータベースに存在しません。

処置: 有効なオブジェクト名を使用してコマンドを再試行してください。

SP2-00562 オブジェクトがパッケージに存在しません。

原因: DESCRIBE の対象として指定したオブジェクトはパッケージに存在しません。

処置: オブジェクト名が正しいことを確認してください。

SP2-00564 オブジェクト"object_name"は無効です。指定されていない可能性があります。

原因: DESCRIBE の対象として指定したオブジェクトが無効です。

処置: オブジェクトを再検証してください。

SP2-00565 識別子が不正です。

原因: DESCRIBE コマンドに無効な文字が使用されました。

処置: 文字を修正し、再試行してください。

SP2-00566 サブオブジェクトの指定が不正です。

原因: DESCRIBE コマンドで無効なサブオブジェクトが指定されました。

処置: サブオブジェクトの指定を修正し、再試行してください。

SP2-00567 PL/SQL オブジェクトの列指定が不正です。

原因: DESCRIBE コマンドのオブジェクト内で列を指定しました。

処置: DESCRIBE コマンドの列の指定を削除して、再試行してください。

SP2-00568 バインド変数が宣言されていません。

原因: バインド変数が宣言されていません。

処置: 処置は必要ありません。

SP2-00570 使用方法: SET SERVEROUTPUT {ON | OFF} [SIZE {n | UNLIMITED}] [FOR[MAT] {WRAPPED | WORD_WRAPPED} | TRUNCATED] }

原因: SET SERVEROUTPUT コマンドに無効なオプションが使用されました。

処置: 有効なオプションを指定してください。

SP2-00575 Oracle の SQL 機能の使用が SQL92 の Entry |Intermediate|Full レベルではありません。

原因: FIPS 準拠ではない SQL 文が試行されました。このエラーは、FIPS フラグ付けを使用する際に、Oracle 固有の SQL を使用する SQL*Plus 機能(たとえば、SET AUTOTRACE など)がオンにされている場合に発生する可能性があります。

処置: SET FLAGGER を使用して FIPS 準拠のチェックをオフにするか、または文を修正してください。

い。

SP2-0577 使用方法: SET FLAGGER {OFF | ENTRY | INTERMEDIATE | FULL}

原因: SET FLAGGER コマンドに無効なオプションが指定されました。

処置: 有効なオプションを指定してください。

SP2-00581 オブジェクト object_name はパッケージです。DESCRIBE

<package>.<procedure>を使用。

原因: パッケージの定義をスタンドアロンで表示しようとした。プロシージャなどのサブオブジェクトがありません。

処置: DESCRIBE コマンドを使用して、パッケージにサブオブジェクトの定義を表示してください。

**SP2-00582 使用方法: {EXIT | QUIT} [SUCCESS | FAILURE | WARNING | n
|<variable> | :<bindvariable>] [COMMIT | ROLLBACK]**

原因: EXIT のオプションは SQL*Plus では無効です。

処置: 有効なオプションを指定してください。

SP2-00584 EXIT 変数"variable_name"が非数値でした。

原因: 指定した EXIT 変数は数値以外です。

処置: 正しい使用方法については、EXIT コマンドの構文を確認してください。

SP2-00590 COMPUTE 関数は各 LABEL キーワードの前に来なければなりません。

原因: COMPUTE 関数は各 LABEL キーワードの前にある必要があります。

処置: 正しい使用方法については、COMPUTE コマンドの構文を確認してください。

SP2-00591 必要な動的領域(number_of_bytes バイト)を割当てできません。ARRAYSIZE または選択列数を減らしてみてください。

原因: コマンドを処理するためのメモリーを割り当てられませんでした。

処置: 不要なアプリケーションをクローズするか、コマンドや文のサイズを減らすか、または問合せを再

コード化しレコードの選択を減らして、余分なメモリーを解放してください。

SP2-00593 ラベル・テキストは LABEL キーワードの後に来なければなりません。

原因: COMPUTE コマンドで LABEL キーワードのラベル・テキストが欠落しています。

処置: 正しいオプションについては、COMPUTE コマンドの構文を確認してください。

SP2-00594 使用方法: SET COLSEP {" " | text}

原因: SET COLSEP コマンドに無効なオプションが使用されました。

処置: 有効なオプションを指定してください。

SP2-00596 使用方法: SET AUTO[COMMIT] {OFF | ON | IMM[EDIATE] | n}

原因: SET AUTO[COMMIT]コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、SET AUTOCOMMIT コマンドの構文を確認してください。

SP2-00597 "datatype _name"は無効な datatype _name 形式です。

原因: ACCEPT コマンドに入力された値が指定されたデータ型ではありませんでした。

処置: データ型を修正し、再入力してください。

SP2-00598 "value_name"は入力形式"format_name"と一致しません。

原因: ACCEPT コマンドに入力された値が指定された形式ではありませんでした。

処置: 形式を修正し、再試行してください。

SP2-00599 使用方法: SET EDITF[ILE] ファイル名[.ext]

原因: SET EDITFILE コマンドの後に必要なファイル名が欠落していました。

処置: 正しいオプションについては、SET EDITFILE コマンドの構文を確認してください。

**SP2-00603 STORE コマンドが不正です。使用方法: STORE {SET} filename[.ext]
[CRE[ATE]]|REP[LACE]]|APP[END]]**

原因: STORE コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、STORE コマンドの構文を確認してください。

SP2-00605 ファイル"file_name"はすでに存在しています。別のファイル名か"STORE {SET} filename[.ext] REPLACE"を使用してください。

原因: STORE コマンドで指定されたファイルは、すでに存在しています。

処置: REPLACE オプションを使用して、既存のファイルを上書きしてください。または、別のファイル名を指定してください。

SP2-00606 ファイル"file_name"を作成できません。

原因: STORE コマンドは、指定されたファイルを作成できません。オープンしているファイルが多すぎるためディスク領域が不十分であるか、または出力ディレクトリが読み取り専用になっています。

処置: ディスク領域が十分であること、およびディレクトリにファイルを作成できることを確認してください。

SP2-00607 ファイル"file_name"をクローズできません。

原因: STORE コマンドは、指定されたファイルをクローズできません。別のリソースがそのファイルをロックしている可能性があります。

処置: クローズする前にファイルがロックされていないことを確認してください。

SP2-00608 オブジェクト "object_name"はリモート・オブジェクトです。詳しい説明はできません。

原因: リモート・オブジェクトは DESCRIBE できません。

処置: 処置は必要ありません。

**SP2-0609 使用方法: SET AUTOT[RACE] {OFF | ON | TRACE[ONLY] }
[EXP[LAIN]] [STAT[ISTICS]]**

原因: SET AUTOTRACE コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、SET AUTOTRACE コマンドの構文を確認してください。

SP2-00610 feature_name の初期化中にエラーが発生しました。

原因: この機能を使用するためのメモリーが不十分です。

処置: 不要なアプリケーションをクローズしてメモリーを解放するか、または、コマンド、文または問合せの出力のサイズを減らしてください。

SP2-00612 report_name レポートの生成中にエラーが発生しました。

原因: AUTOTRACE を使用してレポートを生成できませんでした。

処置: メッセージおよび番号をノートにとって、データベース管理者に連絡してください。

SP2-00613 PLAN_TABLE の形式または存在を検証できません。autotrace_report レポートを使用可能にするときにエラーが発生しました。

原因: 権限のないユーザーまたは PLAN_TABLE を持たないユーザーが、AUTOTRACE コマンドを実行しました。

処置: ユーザーに PLUSTRACE ロールが付与され、そのユーザー用の PLAN_TABLE が作成済であることを確認してください。

SP2-00614 この機能に対してサーバーのバージョンが古すぎます。

原因: この機能に対して Oracle Server の現行のバージョンが古すぎます。

処置: Oracle Server の新しいバージョンを使用してください。

SP2-00617 一意の STATEMENT_ID を組み立てられません。

原因: AUTOTRACE で一意の文 ID を構成できませんでした。

処置: AUTOTRACE が構成され、PLUSTRACE ロールが使用可能であることを確認してください。

SP2-00618 セッション識別子が見つかりません。PLUSTRACE ロールが使用可能であることを確認してください。autotrace_report レポートを使用可能にするときにエラーが発生しました。

原因: セッション識別子を検出できませんでした。

処置: PLUSTRACE ロールが付与されていることを確認してください。

SP2-00619 接続中にエラーが発生しました。

原因: AUTOTRACE がデータベース・インスタンスへの 2 番目の接続を確立しようとして、エラーが発生しました。

処置: データベースのアクティブなセッション数の制限を超えていないことを確認してください。

SP2-00620 接続切断中にエラーが発生しました。

原因: AUTOTRACE がデータベース・インスタンスから切断しようとして、エラーが発生しました。

処置: データベースが使用可能であることを確認してください。

SP2-00621 統計表示収集中にエラー(ORA-error_number)が発生しました。

原因: 統計表示収集中で、AUTOTRACE を使用して PLAN_TABLE にデータを検出できませんでした。

処置: ORA エラー・メッセージについては、『Oracle Database エラー・メッセージ』を参照してください。

SP2-00622 開始行番号は終了行番号より小さくありません。

原因: 指定された開始行番号は、終了行番号より大きいです。

処置: 開始行番号を小さくして再入力してください。

SP2-00623 PRODUCT_USER_PROFILE へのアクセス中にエラーが発生しました。警告: 製品のユーザー・プロファイル情報がロードされていません。PUPBLD.SQL を SYSTEM として実行する必要があります。

原因: PRODUCT_USER_PROFILE 表が SYSTEM アカウントで作成されていません。

処置: ファイル拡張子の正確な形式およびファイルの位置は、システムによって異なります。ご使用のオペレーティング・システムでの詳細は、SQL*Plus のインストレーション・ガイドを参照してください。スクリプトは、ユーザー-SYSTEM で実行する必要があります。

SP2-00625 変数"variable_name"の出力中にエラーが発生しました。

原因: 指定された変数の出力中にエラーが発生しました。

処置: 指定した変数が正しいことを確認して、再試行してください。

SP2-00626 パッケージ DBMS_APPLICATION_INFO へのアクセス・エラーです。Oracle Procedural オプションをインストールする必要がある場合があります。SET APPINFO は Oracle

Server リリース 7.2 以上を必要とします。

原因：このメッセージの後に、Oracle Server への正常なログインが行われます。DBMS_APPLICATION パッケージは、Oracle にログインした特定のアプリケーションについてのオンラインの情報を保持するために使用されます。SET APPINFO は、初期化されません。

処置：このパッケージは、CATPROC.SQL の実行中に作成され、Oracle 7.2 以上のすべてのデータベースで使用可能である必要があります。データベースが正しくインストールされていることを確認してください。

SP2-0631 string_name で始まる文字列が長すぎます。最大サイズは 1 文字または string_length 文字です。

原因：指定された文字列が長すぎました。

処置：指定された文字列のサイズを小さくして、操作を再試行してください。

SP2-00640 接続されていません。

原因：Oracle インスタンスへ接続していないときに、PASSWORD コマンドが発行されました。

処置：PASSWORD コマンドを再実行する前に、Oracle Database に接続してください。

SP2-00641 "command_name"ではサーバーへの接続が必要です。

原因：データベースへ接続していないため、SQL*Plus はコマンドを実行できませんでした。

処置：データベースに接続して、操作を再試行してください。

SP2-0642 状態 error_state、コンテキスト error_number で SQL*Plus の内部エラーが発生しました。続行するのは危険です。続行できません。

原因：内部エラーが発生しました。

処置：メッセージをノートにとって、Oracle サポート・サービスに連絡してください。

SP2-00645 オペレーティング・システムのエラーが発生しました。EDIT コマンドを完了できません。

原因：EDIT コマンド使用時に、オペレーティング・システム・エラーが発生しました。

処置：ファイルが正常に作成されたことを確認し、書込み中のデバイスがまだ使用可能であることを

確認してください。

SP2-00650 新規パスワードが一致しません。

原因: 入力された新しいパスワードが一致しませんでした。

処置: PASSWORD コマンドを再発行して、新しいパスワードが正しく入力されていることを確認してください。

SP2-00659 パスワードが変更されませんでした。

原因: 次の理由のため、PASSWORD コマンドがパスワードの変更に失敗しました。

- パスワードが指定されていない。
- 新しいパスワードが一致していない。

処置: PASSWORD コマンドを再発行して、新しいパスワードが正しく入力されていることを確認してください。

SP2-00666 警告: SHIFTINOUT は、シフト文字を検出する文字セットのみに影響します。

原因: このセッションで使用する NLS 文字セットには、シフト対応の文字が含まれていません。SET SHIFTINOUT コマンドは不要です。

処置: 処置は必要ありません。

SP2-00667 メッセージ・ファイル facility<lang>.msb が見つかりません。

原因: SP1、SP2 または CPY メッセージ・ファイルが見つかりませんでした。SQL*Plus は実行できません。

処置: プラットフォーム固有の Oracle マニュアルを確認して、SQL*Plus が正しくインストールされていることを確認してください。環境変数 ORACLE_HOME または同等のレジストリが Oracle ソフトウェアの位置に設定されていないために、このエラーが発生します。この値が正しく設定されていることを確認してください。SQL*Plus バイナリ・メッセージ・ファイルが SQL*Plus メッセージ・ディレクトリ(たとえば、\$ORACLE_HOME/sqlplus/mesg)に存在することを確認してください。環境変数 NLS_LANG または同等のレジストリの値が正しいことを確認してください。

SP2-00668 変数名が無効です。

原因: 変数名の一部に無効な文字が指定されました。

処置: 有効な文字を使用して変数を指定してください。

SP2-00669 有効文字は英数字および '_' です。

原因: 変数名の一部に無効な文字が指定されました。

処置: 英数字および '_' を使用して変数を指定してください。

SP2-00670 内部数値変換が失敗しました。

原因: 文字列に英数字が含まれるために、変換要求が実行されませんでした。

処置: 文字列に数字のみが含まれていることを確認してください。

SP2-00675 COPY コマンドは使用できません。

原因: このバージョンの SQL*Plus では、COPY コマンドは使用できません。

処置: メッセージおよび番号をノートにとって、Oracle サポート・サービスに連絡してください。

SP2-00676 バインド変数の長さは variable_length units_of_variable 以下である必要があります。

原因: バインド変数のデータ型の長さを超えています。

処置: バインド変数のデータ型の長さを減らしてください。

SP2-00678 列または属性型は SQL*Plus では表示できません。

原因: 指定された型はサポートされていません。

処置: クエリー・リライトを実行して、SQL*Plus がサポートする型でデータを選択してください。

SP2-00685 日付 "entered_variable" が無効か、または書式 "string" が不整合です。

原因: 無効な日付が入力されたか、または書式と一致していません。

処置: 有効な日付または要求された書式で日付を入力してください。

SP2-00686 使用方法: DESCRIBE [スキーマ.]オブジェクト[@db_link]

原因: DESCRIBE コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、DESCRIBE コマンドの構文を確認してください。

SP2-0692 使用方法: CONN[ECT] [logon] [AS {SYSDBA|SYSOPER}]ただし

<logon> ::= <username>[/<password>][@<connect_string>] | /

原因: SQL*Plus CONNECT コマンドに、無効なオプションが入力されました。

処置: 正しい使用方法については、CONNECT コマンドの構文を確認してください。

SP2-00714 STARTUP オプションの組合せが無効です。

原因: STARTUP コマンドの指定されたオプションは、同時に使用できません。

処置: 正しい使用方法については、STARTUP コマンドの構文を確認してください。

SP2-00715 SHUTDOWN オプションの組合せが無効です。

原因: SHUTDOWN コマンドの指定されたオプションは、同時に使用できません。

処置: 正しい使用方法については、SHUTDOWN コマンドの構文を確認してください。

SP2-00716 ARCHIVE LOG オプションの組合せが無効です。

原因: ARCHIVE LOG コマンドの指定されたオプションは、同時に使用できません。

処置: 正しい使用方法については、ARCHIVE LOG コマンドの構文を確認してください。

SP2-00717 SHUTDOWN オプションが正しくありません。

原因: SHUTDOWN コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、SHUTDOWN コマンドの構文を確認してください。

SP2-00718 ARCHIVE LOG オプションが正しくありません。

原因: ARCHIVE LOG コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、ARCHIVE LOG コマンドの構文を確認してください。

SP2-00728 ログの指定: {<RET>=suggested | filename | AUTO | CANCEL}

原因: 適用する REDO ログ・ファイルの入力を求める RECOVER DATABASE コマンド・プロンプ

トです。

処置: REDO ログ・ファイル・オプションのうち 1 つを入力します。

SP2-00729 データベースに接続している間は、SET INSTANCE を行えません。

原因: SET INSTANCE コマンドの実行中に、インスタンスへの接続で問題が発生しました。

処置: インスタンスから切断して、コマンドを再実行してください。

SP2-00733 接続文字列が無効です。

原因: 無効な接続文字列が指定されました。

処置: 接続文字列が正しいことを確認してください。

SP2-00734 "command_name..."で開始するコマンドが不明です。残りの行は無視されました。

原因: 入力したコマンドが無効です。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00735 不明な command_name オプション("option_name..."で開始)です。

原因: コマンドに無効なオプションが指定されました。

処置: 正しいオプションについては、コマンドの構文を確認してください。

SP2-00736 "string_name..."で開始する行の置換で、コマンドラインがオーバーフローしました。

原因: コマンドラインの最大長を超えました。

処置: コマンドで使用する置換変数のデータの長さを減らしてください。

SP2-00737 使用方法: SET DESCRIBE [DEPTH {1|n|ALL}] [LINENUM {ON|OFF}] [INDENT {ON|OFF}]

原因: SET DESCRIBE コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、SET DESCRIBE コマンドの構文を確認してください。

SP2-00738 制限されたコマンド"command_name"は使用できません。

原因: セキュリティ上の理由から、コマンドは、-RESTRICT コマンドライン・オプションによって制限されています。

処置: SQL*Plus が-RESTRICT オプションで実行している理由をデータベース管理者に問い合わせてください。

SP2-00745 使用方法: SET SQLPLUSCOMPAT[IBILITY]バージョン リリース[update]

原因: SET SQLPLUSCOMPAT[IBILITY]コマンドに無効なオプションが使用されました。

処置: 正しいオプションについては、SET SQLPLUSCOMPATIBILITY コマンドの構文を確認してください。

SP2-00746 command_option オプションが範囲(lower から upper)外です。

原因: 指定された値は範囲外です。

処置: 範囲内で値を指定してください。

SP2-00747 LINESIZEline_size でこの問合せを実行するには、少なくとも max_page_size の PAGESIZE が必要です

原因: PAGESIZE の設定が小さすぎるため、指定された LINESIZE を表示できませんでした。

処置: PAGESIZE を増やして、少なくとも指定された LINESIZE と一致するようにしてください。

SP2-00749 シノニム"synonym_name"の循環パスを解決できません。

原因: ベース・オブジェクトとシノニムの名前が同じ場合に、すでに存在しないオブジェクトを指すシノニムを使用しようとしたか、または自身を指す循環パスを持つシノニムを使用しようとしていました。

処置: シノニムのパスの最新のシノニムが、存在するオブジェクトを指すこと、また、シノニム自身を指していないことを確認してください。

SP2-00750 Oracle ソフトウェア・ディレクトリに ORACLE_HOME を設定する必要がある場合があります。

原因: SQL*Plus は、プログラムの初期化中にメッセージ・ファイルを検索できませんでした。そのため、エラー・メッセージや通常の処理に必要なテキストを表示できませんでした。最も可能性の高い原因として、ORACLE_HOME が設定されていないことが考えられます。他には、メッセージ・ファイルが破損しているか読取り不能である可能性があります。また、Windows で SQLPLUS レジストリ・エント

リが無効な場合があります。

このメッセージは、SQL*Plus のソース・コードに(英語で)ハード・コードされているため、メッセージ・ファイル・エラーとして表示される場合があります。このエラーは、メッセージ・ファイルがオープンできない場合にのみ発生するため、このメッセージ・ファイルからは読み取れません。メッセージ・ファイルのこのエントリは、説明用に記載しています。

処置: すべての環境変数、または SQL*Plus の実行に必要なレジストリ・エントリが設定されていることを確認してください。変数はプラットフォーム固有ですが、ORACLE_HOME、ORACLE_SID、NLS_LANG および LD_LIBRARY_PATH を含む場合があります。

Windows で環境変数 SQLPLUS が設定されている場合は、変数に SQL*Plus メッセージ・ファイルのディレクトリ名(ORACLE_HOME¥sqlplus¥mesg など)を含める必要があります。

また、ファイル sp1XX.msb が、\$ORACLE_HOME/sqlplus/mesg ディレクトリまたは ORACLE_HOME¥sqlplus¥mesg ディレクトリにあることを確認してください。「XX」は、環境変数 NLS_LANG に関連付けられた地域接頭辞を示します。SQL*Plus は、sp1XX.msb ファイルから 1 つのみ読み取ります。たとえば、NLS_LANG が JAPANESE_JAPAN.JA16EUC の場合、sp1ja.msb が読み取られます。NLS_LANG が設定されていない場合、デフォルト(英語)の sp1us.msb が使用されます。適切なファイルのサイズが 0(ゼロ)以外であり、ファイルの読取り権限があることを確認してください。「.msb」ファイルはバイナリです。ファイルの内容を表示したり印刷しても意味がありません。どの言語のファイルが使用されているか不明な場合は、NLS_LANG を設定しないで SQL*Plus を実行し、sp1us.msb ファイルを読み取れるか検証してください。

SP2-00751 Oracle に接続できませんでした。SQL*Plus を終了します。

原因: Oracle Server と接続していません。

処置: 通常は、データベースが実行していないか、またはユーザー名およびパスワードが無効であることを示す他のエラーの後に発生します。

SP2-00752 使用方法: -C[OMPATIBILITY] version.release.[update]

原因: -C[OMPATIBILITY]コマンド・オプションに無効なオプションが使用されました。

処置: 正しいオプションについては、SQL*Plus の実行可能ファイルの構文を確認してください。

SP2-00753 STARTUP MIGRATE は、Oracle 9.2 以上で有効です。

原因: リリース 2(9.2)より前の Oracle Server を起動しようとして、STARTUP MIGRATE を使用しました。

処置: プラットフォーム固有の環境を確認し、リリース 2(9.2)以上の Oracle Server に接続していることを確認してください。

SP2-00754 FROM 句には AS SYSDBA または AS SYSOPER を指定できません

原因: COPY コマンドでは、AS SYSDBA または AS SYSOPER 接続はサポートされていません。

処置: FROM 句から AS SYSDBA または AS SYSOPER を削除してください。

SP2-00755 TO 句には AS SYSDBA または AS SYSOPER を指定できません

原因: COPY コマンドでは、AS SYSDBA または AS SYSOPER 接続はサポートされていません。

処置: TO 句から AS SYSDBA または AS SYSOPER を削除してください。

SP2-00756 FROM 句の長さ clause_len バイトは最大長 max_len を超えています

原因: FROM 句が長すぎます。

処置: FROM 句で指定した文字列を短くしてください。

SP2-00757 TO 句の長さ clause_len バイトは最大長 max_len を超えています

原因: TO 句が長すぎます。

処置: TO 句で指定した文字列を短くしてください。

SP2-00758 FROM 句にユーザー名が欠落しています

原因: COPY コマンドの FROM 句にユーザー名および接続識別子が含まれていません。

処置: FROM 句にユーザー名と接続識別子を指定してください。

SP2-00759 TO 句にユーザー名が欠落しています

原因: COPY コマンドの TO 句にユーザー名および接続識別子が含まれていません。

処置: TO 句にユーザー名と接続識別子を指定してください。

SP2-0762 SHOW ERRORS [object]の引用符が一致していません

原因: SHOW ERRORS に対する引数として送信されるオブジェクト名に、無効な構文が見つかり

ました。

処置：引用符を使用している場合、正しく一致していることを確認してください。引数全体を引用符で囲むか、スキーマとオブジェクト・コンポーネントを別々に引用符で囲んでください。

SP2-00768 無効な SPOOL コマンドです

原因：SPOOL コマンドに無効なオプションが使用されました。

処置：正しいオプションについては、SPOOL コマンドの構文を確認してください。

SP2-00769 使用方法: SPOOL { <file> | OFF | OUT }

ただし<file>は file_name[.ext] [CRE[ATE]|REP[LACE]|APP[END]]

原因：不適切な構文で SPOOL コマンドが入力されました。

処置：正しい使用方法については、SPOOL コマンドの構文を確認してください。

SP2-00771 ファイル"filename"はすでに存在します。別のファイル名か"SPOOL filename[.ext] REPLACE"を使用してください。

原因：SPOOL コマンドで指定されたファイルは、すでに存在しています。

処置：REPLACE オプションを使用して、既存のファイルを上書きしてください。または、別のファイル名を指定してください。

SP2-00772 ASM インスタンスが起動されました

原因：ドキュメント：フィードバック・メッセージです。

処置：

SP2-00773 ASM ディスク・グループがマウントされました

原因：ドキュメント：フィードバック・メッセージです。

処置：

SP2-00774 ASM インスタンスがシャットダウンされました

原因：ドキュメント：フィードバック・メッセージです。

処置:

SP2-00775 ASM ディスク・グループがディスマウントされました

原因: ドキュメント: フィードバック・メッセージです。

処置:

SP2-0776 SHOW ERRORS [object]のスキーマおよびオブジェクト・セパレータが無効です

原因: SHOW ERRORS に対する引数として送信されるオブジェクト名に、無効な構文が見つかりました。

処置: スキーマを指定した場合は、スキーマとオブジェクト名がピリオドで区切られていることを確認してください。

SP2-0777 SHOW ERRORS [object]の一重引用符が無効です

原因: SHOW ERRORS に対する引数として送信されるオブジェクト名に、無効な構文が見つかりました。

処置: SHOW ERROR の引数が引用符で囲まれている場合、二重引用符のみが使用されていることを確認してください。引数全体を引用符で囲むか、スキーマとオブジェクト・コンポーネントを別々に引用符で囲んでください。

SP2-00778 スクリプトのファイル名と引数が長すぎます

原因: スクリプトのファイル名と引数を結合した長さが SQL*Plus では長すぎます。

処置: スクリプト名とパスの長さを短くしてください。スクリプトの引数の数またはサイズ(あるいはその両方)を減らしてください。

SP2-00780 入力した値は有効な datatype ではありません

原因: ACCEPT コマンドに入力された値が、指定されたデータ型に対して無効でした。

処置: データ型の有効範囲内の有効な数値を入力してください。

SP2-00781 コマンド・オプションが範囲(最小値から最大値)外です

原因: コマンド・オプションに対する有効範囲外の値を入力しようとしていました。

処置: コマンド・オプションの制限を確認して、有効範囲内の値を入力してください。

SP2-00782 予備接続が確立されました

原因: ドキュメント: フィードバック・メッセージです。

処置:

SP2-00783 データベースに接続している間は、SET variable を行えません

原因: データベース・インスタンスに接続している間は設定できないシステム変数を設定しようとした。

処置: データベース・インスタンスから切断してから、システム変数を設定してください。

SP2-00784 byte で始まる無効または不完全な文字が戻されました

原因: データベースから無効または不完全な文字を含む文字列を戻そうとしました。

処置: データベース内の無効または不完全な文字列を、有効または完全な文字列に置き換えてください。

SP2-00804 プロシージャが作成されましたが、コンパイル警告があります。

原因: PL/SQL プロシージャが作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL プロシージャを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00805 プロシージャが変更されましたが、コンパイル警告があります。

原因: PL/SQL プロシージャが変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL プロシージャを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00806 ファンクションが作成されましたが、コンパイル警告があります。

原因: PL/SQL ファンクションが作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL ファンクションを改善する際に役立ちます。

ンス・メッセージがあります。これらは、ご使用の PL/SQL ファンクションを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00807 ファンクションが変更されましたが、コンパイル警告があります。

原因: PL/SQL ファンクションが変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL ファンクションを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00808 パッケージが作成されましたが、コンパイル警告があります。

原因: PL/SQL パッケージが作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL パッケージを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00809 パッケージが変更されましたが、コンパイル警告があります。

原因: PL/SQL パッケージが変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL パッケージを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00810 パッケージ本体が作成されましたが、コンパイル警告があります。

原因: PL/SQL パッケージ本体が作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL パッケージ本体を改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00811 パッケージ本体が変更されましたが、コンパイル警告があります。

原因: PL/SQL パッケージ本体が変更されましたが、1 つ以上の警告、情報メッセージまたはパ

パフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL パッケージ本体を改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00812 ビューが作成されましたが、コンパイル警告があります。

原因: PL/SQL ビューが作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL ビューを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00813 ビューが変更されましたが、コンパイル警告があります。

原因: PL/SQL ビューが変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL ビューを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00814 トリガーが作成されましたが、コンパイル警告があります。

原因: PL/SQL トリガーが作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL トリガーを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00815 トリガーが変更されましたが、コンパイル警告があります。

原因: PL/SQL トリガーが変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL トリガーを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00816 型が作成されましたが、コンパイル警告があります。

原因: PL/SQL 型が作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッ

セージがあります。これらは、ご使用の PL/SQL 型を改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00817 型が変更されましたが、コンパイル警告があります。

原因: PL/SQL 型が変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL 型を改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00818 型本体が作成されましたが、コンパイル警告があります。

原因: PL/SQL 型本体が作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL 型本体を改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00819 型本体が変更されましたが、コンパイル警告があります。

原因: PL/SQL 型本体が変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL 型本体を改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00820 ライブラリが作成されましたが、コンパイル警告があります。

原因: PL/SQL ライブラリが作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL ライブラリを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00821 ライブラリが変更されましたが、コンパイル警告があります。

原因: PL/SQL ライブラリが変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL ライブラリを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00822 Java が作成されましたが、コンパイル警告があります。

原因: PL/SQL Java が作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL Java を改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00823 Java が変更されましたが、コンパイル警告があります。

原因: PL/SQL Java が変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL Java を改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00824 PL/SQL コンパイル警告

原因: PL/SQL ブロックが作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL ブロックを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00825 ディメンションが作成されましたが、コンパイル警告があります。

原因: PL/SQL ディメンションが作成されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL ディメンションを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00826 ディメンションが変更されましたが、コンパイル警告があります。

原因: PL/SQL ディメンションが変更されましたが、1 つ以上の警告、情報メッセージまたはパフォーマンス・メッセージがあります。これらは、ご使用の PL/SQL ディメンションを改善する際に役立ちます。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

い。

SP2-00827 プロシージャが作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL プロシージャが作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00828 プロシージャが変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL プロシージャが変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00829 ファンクションが作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL ファンクションが作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00830 ファンクションが変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL ファンクションが変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00831 パッケージが作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL パッケージが作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00832 パッケージが変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL パッケージが変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

い。

SP2-00833 パッケージ本体が作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL パッケージ本体が作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00834 パッケージ本体が変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL パッケージ本体が変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00835 ビューが作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL ビューが作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00836 ビューが変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL ビューが変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00837 トリガーが作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL トリガーが作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00838 トリガーが変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL トリガーが変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

い。

SP2-00839 型が作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL 型が作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00840 型が変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL 型が変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00841 型本体が作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL 型本体が作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00842 型本体が変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL 型本体が変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00843 ライブラリが作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL ライブラリが作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00844 ライブラリが変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL ライブラリが変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

い。

SP2-00845 Java が作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL Java が作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00846 Java が変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL Java が変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00847 PL/SQL コンパイル・エラー

原因: PL/SQL ブロックが作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00848 ディメンションが作成されましたが、コンパイル・エラーがあります。

原因: PL/SQL ディメンションが作成されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-00849 ディメンションが変更されましたが、コンパイル・エラーがあります。

原因: PL/SQL ディメンションが変更されましたが、1 つ以上のエラー・メッセージがあります。

処置: SQL*Plus SHOW ERR[ORS]コマンドを使用して、警告およびメッセージを参照してください。

SP2-01500 STARTUP UPGRADE は、Oracle 9.2 以上で有効です

原因: リリース 2(9.2)より前の Oracle Server を起動しようとして、STARTUP UPGRADE を使用しました。

処置：プラットフォーム固有の環境を確認し、リリース 2(9.2)以上の Oracle Server に接続していることを確認してください。

SP2-01501 STARTUP DOWNGRADE は Oracle 9.2 以上で有効です

原因：リリース 2(9.2)より前の Oracle Server を起動しようとして、STARTUP DOWNGRADE を使用しました。

処置：プラットフォーム固有の環境を確認し、リリース 2(9.2)以上の Oracle Server に接続していることを確認してください。

SP2-01502 http_proxy で指定されている HTTP プロキシ・サーバーにアクセスできません

原因：SQL*Plus で使用する HTTP プロキシ・サーバーにアクセスできません。SQL*Plus では、Web サーバーにあるスクリプトを実行できません。

処置：プロキシに正しい値が設定されているかどうかを確認するか、プロキシが不要な場合は設定を解除します。SQL*Plus は、環境変数 http_proxy からプロキシ名を取得するか、または別の方法でシステムにその値を設定します。指定したプロキシ・サーバーが動作可能であることを確認します。ほとんどの Web ブラウザは、プロキシを使用するために構成できます。必要なプロキシを使用するためにブラウザを構成し、Web ページがロード可能であることを確認します。

SP2-01503 Oracle Call Interface を初期化できません

原因：SQL*Plus でデータベースとの通信に使用するライブラリが正しく初期化されていません。

処置：Oracle 環境またはレジストリ・エントリが一貫していて、正しいことを確認してください。SQL*Plus Instant Client を使用している場合は、SQL*Plus および Oracle ライブラリのリリースが同じであることを確認してください。ライブラリへの読取りアクセス権を所有していることを確認してください。

SP2-01504 初期化されていない LOB 変数は印刷できません

原因：指定した LOB 変数は、印刷する前に初期化する必要があります。

処置：指定した変数が初期化されていることを確認して、再試行してください。

SP2-01505 入力された COL または TAB の位置が無効です

原因：入力した BTITLE、TTITLE、REPHEADER または REPFOOTER コマンドで、240 を超える COL 値または TAB 値の使用が試行されています。あるいは、LINESIZE が 240 を超えている

る場合に、LINESIZE を超える COL 値または TAB 値の使用が試行されています。

処置：入力した BTITLE、TTITLE、REPHEADER または REPFOOTER コマンドでは、240 以下の COL 値または TAB 値を使用してください。また、LINESIZE が 240 を超えている場合は、COL 値または TAB 値を LINESIZE 以下にしてください。

SP2-01507 エラー・ロギング表、ロールまたは権限が欠落しているか、あるいはそれらにアクセスできません。

原因：エラー・ロギング表/ロール/権限が欠落しているか、あるいはそれらにアクセスできません。

処置：デフォルト以外のエラー・ログ表の作成方法については、sqlplus doc ディレクトリの elgsetup.txt を参照してください。権限の付与の方法については、『Oracle Database 管理者ガイド』を参照してください。

SP2-01508 SET ERRORLOGGING OFF のオプションが無効です。

原因：追加オプションを指定して SET ERRORLOGGING OFF コマンドを発行しようとした。SET ERRORLOGGING OFF は、その他のオプションを受け入れません。

処置：その他のオプションを指定せずに SET ERRORLOGGING OFF コマンドを再入力します。

SP2-01509 SET ERRORLOGGING ON のオプションが無効です。

原因：SET ERRORLOGGING ON に無効なオプションを入力しようとした。

処置：有効なオプションを使用して SET ERRORLOGGING ON コマンドを再入力してください。有効なオプションは、TABLE、IDENTIFIER および TRUNCATE です。

SP2-01510 SET ERRORLOGGING ON のオプション *option_name* が無効です。

原因：SET ERRORLOGGING ON に無効なオプションを入力しようとした。

処置：有効なオプションを使用して SET ERRORLOGGING ON コマンドを再入力してください。有効なオプションは、TABLE、IDENTIFIER および TRUNCATE です。

SP2-01511 SET ERRORLOGGING ON TABLE の表名が欠落しています。

原因：SET ERRORLOGGING ON TABLE コマンドで、表名が入力されていないか、あるいは無効な表名またはスキーマ名が入力されました。*schema.table* という構文を使用して別のスキーマが指定されていないかぎり、表名は現行のスキーマ内にあるとみなされます。

処置: 有効な表名を使用して SET ERRORLOGGING ON TABLE コマンドを再入力してください。

SP2-01512 SET ERRORLOGGING ON TABLE の表名として *table_name* を設定できませんでした。

原因: SET ERRORLOGGING ON TABLE コマンドで、表名が入力されていないか、あるいは無効な表名またはスキーマ名が入力されました。 *schema.table* という構文を使用して別のスキーマが指定されていないかぎり、表名は現行のスキーマ内にあるとみなされます。

処置: 有効な表名を使用して SET ERRORLOGGING ON TABLE コマンドを再入力してください。

SP2-01513 SET ERRORLOGGING ON FILE のファイル名 *file_name* が無効です。

原因: SET ERRORLOGGING ON FILE コマンドで、ファイル名が入力されていないか、あるいは無効なファイル名またはパスが入力されました。

処置: 有効なファイル名とパスを使用して SET ERRORLOGGING ON FILE コマンドを再入力してください。

SP2-01514 SET ERRORLOGGING ON IDENTIFIER の識別子 *identifier* が無効です。

原因: SET ERRORLOGGING ON IDENTIFIER コマンドで、識別子の文字列が入力されていないか、あるいは無効な識別子の文字列が入力されました。識別子の文字列には、特殊文字は使用できません。

処置: 有効な識別子の文字列を使用して SET ERRORLOGGING ON IDENTIFIER コマンドを再入力してください。

SP2-01515 SET ERRORLOGGING ON IDENTIFIER の識別子 *identifier* が無効です。

原因: SET ERRORLOGGING ON IDENTIFIER コマンドで、識別子の文字列が入力されていないか、あるいは無効な識別子の文字列が入力されました。識別子の文字列には、特殊文字は使用できません。

処置: 有効な識別子の文字列を使用して SET ERRORLOGGING ON IDENTIFIER コマンドを再入力してください。

SP2-01516 SET ERRORLOGGING ON IDENTIFIER の識別子 *identifier* で引用符が一致

していません。

原因: SET ERRORLOGGING ON IDENTIFIER コマンドで無効な文字列が見つかりました。
識別子の文字列は、一重または二重引用符で囲む必要があります。

処置: 識別子内の文字列の引用符を確認してから、SET ERRORLOGGING ON IDENTIFIER コマンドを再入力してください。

SP2-01517 エラー・ログ表 *schema.table* をフラッシュできませんでした。

原因: エラー・ログ表のフラッシュに必要な権限がユーザーにない可能性があります。

処置: エラー・ログ表にアクセスするために付与された権限を確認してください。権限の付与の方法については、『Oracle Database 管理者ガイド』を参照してください。

SP2-01518 エラー・ロギング表 *table_name* がスキーマ *schema_name* に存在しません。

原因: 存在しないエラー・ログ表に書き込もうとしました。

処置: デフォルト以外のエラー・ログ表の作成方法については、sqlplus doc ディレクトリの elgsetup.txt を参照してください。

SP2-01519 エラー・ログ表 *schema.table* に書き込めません。

原因: エラー・ログ表への書き込みに必要な権限がユーザーにない可能性があります。

処置: エラー・ログ表にアクセスするために付与された権限を確認してください。権限の付与の方法については、『Oracle Database 管理者ガイド』を参照してください。

SP2-01538 エディションは Oracle Database に接続する場合にのみ有効です。

原因: データベースが使用可能な状態でないときにエディション・セッションに接続しようとしてしました。

処置: エディション・セッションに接続する前に、データベースが使用可能な状態であることを確認してください。

SP2-01539 エディションには Oracle Database 11g 以上が必要です。

原因: Oracle Database 11g より前のバージョンのエディション・セッションに接続しようとしてしました。

処置: エディションは Oracle Database 11g 以上でのみ有効です。

Mahanteshに割当済の1530。2006年6月23日。

Bug-4732049。メッセージ・ファイルに新規メッセージ「コマンドが失敗しました」があります。

Mahanteshに割当済の1531。2006年7月7日。

SPOOLまたはSTART/@コマンドでファイル名の一部に使用する@、?および\$に関連する問題を解決するために、新規SETコマンドが導入されました。構文は次のとおりです。

SET ESCCHAR [OFF|@|?|\$] - デフォルトはOFFです。

ユーザーがSET ESCCHARコマンドで前述とは異なる文字を使用した場合、エラー・メッセージ「オプションが無効です」が新しい使用方法コマンドとともに表示されます。

"使用方法: SET ESCCHAR [OFF|@|?|\$]"

13.2 COPYコマンドのメッセージ

CPY-00002 APPEND、CREATE、INSERT または REPLACE のオプションが不正または欠落しています。

原因: 内部 COPY 関数が、有効範囲外の作成オプション(フラグ)値を使用して COPY を起動しました。

処置: Oracle サポート・サービスに連絡してください。

CPY-00003 内部エラー: 論理ホスト番号が範囲外です。

原因: 有効範囲外の論理ホスト番号値を使用して内部 COPY 関数を起動しました。

処置: Oracle サポート・サービスに連絡してください。

CPY-00004 ソース表と宛先表にある列の名前が一致していません。

原因: APPEND 操作または INSERT(表が存在する場合)で、宛先表の中の 1 つ以上の列名が、オプションの列名リストまたは SELECT コマンドの対応する列名と一致していません。

処置: COPY コマンドを指定しなおします。宛先表の中のそれぞれの列名およびその順序が、オプションの列名リストまたは SELECT コマンドの列名およびその順序と一致していることを確認します。

CPY-00005 ソースと宛先の列属性が一致していません。

原因: APPEND 操作または INSERT(表が存在する場合)で、宛先表の 1 つ以上の列のデータ型が、SELECT コマンドの対応する列のデータ型と一致していません。

処置: 選択する項目のデータ型が宛先と一致することを確認したうえで、再度 COPY コマンドを指

定してください。変換するには、TO_DATE、TO_CHAR および TO_NUMBER を使用します。

CPY-0006 選択リストに宛先表より多い列があります。

原因: APPEND 操作または INSERT(表が存在する場合)で、SELECT コマンドの列の数が、宛先表の列の数を上回っています。

処置: 選択する列数がコピー先の表の列数と一致することを確認したうえで、再度 COPY コマンドを指定してください。

CPY-0007 選択リストに宛先表より少ない列があります。

原因: APPEND 操作または INSERT(表が存在する場合)で、SELECT コマンドの列の数が、宛先表の列の数を下回っています。

処置: 選択する列数がコピー先の表の列数と一致することを確認したうえで、再度 COPY コマンドを指定してください。

CPY-00008 列のリスト名が宛先表にある列より多くあります。

原因: APPEND 操作または INSERT(表が存在する場合)で、列名リストの列の数が、宛先表の列の数を上回っています。

処置: 列名リストの列数がコピー先の表の列数と一致することを確認したうえで、再度 COPY コマンドを指定してください。

CPY-00009 列のリスト名が宛先表にある列より少数です。

原因: APPEND 操作または INSERT(表が存在する場合)で、列名リストの列の数が、宛先表の列の数を下回っています。

処置: 列名リストの列数がコピー先の表の列数と一致することを確認したうえで、再度 COPY コマンドを指定してください。

CPY-00012 データ型はコピーできません

原因: COPY コマンドでサポートされていないデータ型をコピーしようとした。COPY コマンドでサポートされるデータ型は、CHAR、DATE、LONG、NUMBER および VARCHAR2 です。

サポートされていないデータ型列が削除されていることを確認して、COPY コマンドを再指定します。詳細は、「[SQL*Plus の COPY コマンド](#)」を参照してください。

第IV部 SQL*Plusの付録

第IV部の内容は次のとおりです。

- [SQL*Plusの制限](#)
- [SQL*PlusのCOPYコマンド](#)
- [廃止されたSQL*Plusコマンド](#)
- [SQL*Plus Instant Client](#)

A SQL*Plusの制限

この付録に記載するSQL*Plusの一般的な制限値は、ほとんどのオペレーティング・システムで有効です。

表A-1 SQL*Plusの制限

項目	制限
ファイル名の長さ	システム依存
ユーザー名の長さ	128 バイト
置換変数名の長さ	128 バイト
置換変数値の長さ	240 文字
コマンドラインの長さ	5000 文字
LONG	2,000,000,000 バイト
LINESIZE	システム依存
LONGCHUNKSIZE 値	システム依存
出力行のサイズ	システム依存
変数置換後の SQL または PL/SQL コマンドライン・サイズ	3,000 文字(内部のみ)
COMPUTE コマンド・ラベルの文字数	500 文字
1 つの SQL コマンドの行数	500(1 行 80 文字の場合)
最大 PAGESIZE	50,000 行
合計行幅	32,767 文字
最大 ARRAYSIZE	5000 行
ネストされたスクリプトの最大数	20
最大ページ数	99,999

項目	制限
PL/SQL エラー・メッセージの最大サイズ	2K
ACCEPT 文字列の最大長	240 バイト
置換変数の最大数	2048

B SQL*PlusのCOPYコマンド

ここでは、次の項目について説明します。

- [COPYコマンドの構文](#)
- [データベース間でのデータのコピー](#)
- [1つのデータベース上にある表の間でのデータのコピーについて](#)

この章を読むときは、コンピュータ上で、示されている例を実際に試してみてください。始める前に、「[SQL*Plusのクイック・スタート](#)」で説明したサンプル表へのアクセス権限があることを確認してください。

COPYコマンドは、SQL*Plusの今後のリリースで非推奨となる予定です。Oracle 9iより後では、新しいデータ型はCOPYでサポートされていません。

B.1 COPYコマンドの構文

```
COPY {FROM database | TO database | FROM database TO database}  
[APPEND|CREATE|INSERT|REPLACE] destination_table [(column, column, column, ...)] USING query
```

*database*では、次の構文に従います。

```
username[/password]@connect_identifier
```

問合せから、同じまたは別のデータベース内の表にデータをコピーします。COPYは、次のデータ型をサポートします。

- CHAR
- DATE
- LONG
- NUMBER
- VARCHAR2

警告:



パスワードをプレーン・テキストで指定すると、セキュリティ上の危険があります。パスワードを省略し、パスワードの入力を求めるプロンプトが表示された場合にのみ入力することで、この危険を回避できます。

B.1.1 条件

項または句の詳細は、次の説明を参照してください。

```
FROM database
```

コピーするデータを含むデータベースを指定します。FROM句を省略した場合のデフォルトのソースは、SQL*Plusが接続されているデータベース(他のコマンドがアドレス指定するデータベース)です。デフォルト以外のソース・データベースを指定するには、FROM句を使用する必要があります。COPYコマンドのFROM句では、SYSDBAまたはSYSOPER権限での接続はサポートされていません。

```
TO database
```

宛先表を含むデータベースを指定します。TO句を省略した場合のデフォルトの宛先は、SQL*Plusが接続されているデータベース(他のコマンドがアドレス指定するデータベース)です。デフォルト以外の宛先データベースを指定するには、TO句を使用する必要があります。COPYコマンドのTO句では、SYSDBAまたはSYSOPER権限での接続はサポートされていません。

database

FROM句またはTO句で指定するコピー元またはコピー先のOracle Databaseを*username[/password]* @*connect_identifier*で指定します。COPYコマンドでは、SYSDBAまたはSYSOPER権限での接続はサポートされていません。ユーザー名を指定する必要があります。SQL*Plusによって、COPY FROMまたはCOPY TO句で指定したユーザー名に関連付けられたパスワードの入力を求められます。SQL*Plusでは、このパスワードへの応答は表示されません。

コピー元またはコピー先データベースの指定に、*connect_identifier*句を含める必要があります。正確な構文はOracle Netの構成によって異なります。詳細は、Oracle Netのマニュアルを参照するか、またはデータベース管理者に問い合わせてください。

APPEND

*destination_table*が存在する場合は、*query*からその表へ行が挿入されます。*destination_table*が存在しない場合、COPYによってその表が作成されます。

CREATE

*destination_table*を作成した後、*query*からその表に行が挿入されます。*destination_table*がすでに存在している場合は、COPYによってエラーが戻されます。

INSERT

*query*から*destination_table*に行が挿入されます。*destination_table*が存在しない場合は、COPYによってエラーが戻されます。INSERTを使用するときは、USING *query*によって*destination_table*の中の各列について1列ずつ選択する必要があります。

REPLACE

*destination_table*およびその内容を*query*からの行に置換します。*destination_table*が存在しない場合、COPYによってその表が作成されます。存在する場合は、COPYによってその既存の表が削除され、コピーしたデータを含む表で置換されます。

destination_table

作成する表またはデータを追加する表を指定します。

(*column, column, column, ...*)

*destination_table*の中の列の名前を指定します。名前に小文字または空白が含まれている場合は、その名前を二重引用符で囲む必要があります。

列を指定する場合は、問合せで選択された列と同じ数の列を指定する必要があります。COPYによって*destination_table*を作成する場合に、列を指定しないと、宛先表内でのコピーされた列の名前は、ソースでの名前と同じものになります。

USING *query*

COPYによってコピーする行および列を決定するSQL問合せ(SELECTコマンド)を指定します。

B.1.2 使用方法

Oracle DatabaseとOracle以外のデータベースの間でデータをコピーできるようにするため、NUMBER列は宛先の表の中で

DECIMAL列に変更されます。したがって、Oracle Database間でコピーする場合、精度がないNUMBER列はDECIMAL(38)列に変更されます。Oracle Database間でコピーする場合は、SQLコマンド(CREATE TABLE ASおよびINSERT)を使用するか、列に精度が指定されている必要があります。

SQL*PlusのSET LONG変数は、コピーするLONG列の長さを制限するために使用します。LONG列にLONGの値より長いデータが含まれている場合、COPYによってそのデータが切り捨てられます。

SQL*Plusは、COPYが正常に完了するたびにコミットを実行します。SQL*PlusのSET COPYCOMMIT変数を正の値nに設定すると、SQL*Plusはレコード・バッチをn回コピーするたびにコミットを1回行います。パッチのサイズは、SQL*PlusのSET ARRAYSIZE変数によって決まります。

操作環境によっては、サービス名を二重引用符で囲む必要があります。

B.1.3 例

次のコマンドは、EMPLOYEES表の全体をWESTEMPLOYEESという名前の表にコピーします。これらの表は2つの異なるデータベース内にあります。WESTEMPLOYEESがすでに存在している場合は、SQL*Plusによって、その表およびその内容が置換されます。WESTEMPLOYEESの中の列は、ソース表EMPLOYEESの中の列と同じ名前になります。

```
COPY FROM HR@HQ TO JOHN@WEST -  
REPLACE WESTEMPLOYEES -  
USING SELECT * FROM EMPLOYEES
```

次のコマンドは、EMPLOYEESから選択したレコードを、SQL*Plusが接続されているデータベースにコピーします。SQL*Plusは、コピーによってSALESMENを作成します。SQL*Plusは、列EMPLOYEE_IDおよびLAST_NAMEのみをコピーし、コピー先でそれらをEMPLOYEE_IDおよびSA_MANと命名します。

```
COPY FROM HR@ORACLE01 -  
CREATE SALESMEN (EMPLOYEE_ID, SA_MAN) -  
USING SELECT EMPLOYEE_ID, LAST_NAME FROM EMPLOYEES -  
WHERE JOB_ID=' SA_MAN' ;
```

B.2 データベース間でのデータのコピー

データベース間および同じデータベースの表同士の間でCHAR、DATE、LONG、NUMBERまたはVARCHAR2のデータをコピーするには、SQL*PlusのCOPYコマンドを使用します。COPYコマンドを使用すると、次のようなデータベース間でのデータのコピーが可能です。

- リモート・データベースからローカル・データベースへのデータのコピー
- ローカル(デフォルト)・データベースから(ほとんどのシステム上の)リモート・データベースへのデータのコピー
- あるリモート・データベースから(ほとんどのシステム上の)別のリモート・データベースへのデータのコピー

ノート:



一般に、COPY コマンドは、Oracle Database と Oracle 以外のデータベースの間でデータをコピーするために使用するように設計されています。Oracle Database 間でデータをコピーするには、SQL コマンド (CREATE TABLE AS および INSERT)を使用してください。

B.2.1 COPYコマンド構文について

COPYコマンドは、次の形式で入力します。

```
COPY FROM database TO database action -  
destination_table (column_name, column_name, -  
column_name ...) USING query
```

次にCOPYコマンドの例を示します。

```
COPY FROM HR@BOSTONDB -  
TO TODD@CHICAGODB -  
CREATE NEWDEPT (DEPARTMENT_ID, DEPARTMENT_NAME, CITY) -  
USING SELECT * FROM EMP_DETAILS_VIEW
```

FROM句またはTO句の中でデータベースを指定するには、ローカル・データベースおよびリモート・データベースに対して有効なユーザー名およびパスワードが必要で、適切なOracle Netサービス名を指定する必要があります。COPYは、Oracle Databaseセキュリティに従うため、表にアクセスするには、指定するユーザー名に表へのアクセス権限が必要です。どのデータベースが使用可能であるかについては、DBAに問い合わせてください。

リモート・データベースからローカル・データベースにコピーする場合は、TO句を省略できます。ローカル・データベースからリモート・データベースにコピーする場合は、FROM句を省略できます。リモート・データベース間でコピーする場合は、必ず両方の句を指定します。両方の句を挿入すると、スクリプトが読みやすくなります。

COPYコマンドの動作は、宛先の表がすでに存在するかどうかおよびユーザーが入力したアクション句(例ではCREATE)によって異なります。詳細は、[宛先の表の扱いを制御する方法について](#)を参照してください。

デフォルトでは、宛先の表にコピーされた列は元の表での名前と同じ名前になります。宛先の表の中で列に新しい名前を指定する場合は、宛先の表名の後に新しい名前をカッコで囲んで入力します。1つでも列名を入力する場合は、コピーするすべての列の名前を入力する必要があります。

ノート:



Oracle Database と Oracle 以外のデータベースの間でデータをコピーできるようにするため、NUMBER 列は宛先の表の中で DECIMAL 列に変更されます。したがって、Oracle Database 間でコピーする場合は、精度がない NUMBER 列は DECIMAL(38)列に変更されます。Oracle Database 間でコピーする場合は、SQL コマンド (CREATE TABLE AS および INSERT)を使用するか、列に精度が指定されている必要があります。

USING句を使用すると、元の表に対して問合せを行うことによって、COPYで宛先の表にコピーするデータを指定できます。COPYコマンドでコピーするデータを選択するには、任意の形式でSQLのSELECTコマンドを使用できます。

次に示すCOPYコマンドの例では、ソース表から2つの列のみをコピーし、DEPARTMENT_IDの値が30である行のみをコピーします。

```
COPY FROM HR@BOSTONDB -  
REPLACE EMPCOPY2 -  
USING SELECT LAST_NAME, SALARY -  
FROM EMP_DETAILS_VIEW -  
WHERE DEPARTMENT_ID = 30
```

長いCOPYコマンドは、コマンド・プロンプトから直接入力するよりも、スクリプトに入力して編集する方が簡単です。

B.2.2 宛先の表の扱いを制御する方法について

宛先の表の扱いを制御するには、REPLACE、CREATE、INSERTまたはAPPENDの4つの制御句のいずれかを入力します。

REPLACE句は、宛先のデータベース内に作成する表の名前を指定し、次のアクションを指定します。

- 宛先の表がすでに存在する場合、COPYは既存の表を削除し、コピーされたデータが入った表に置き換えます。
- 宛先の表が存在しない場合、COPYはコピーされたデータを使用してその表を作成します。

既存の表を誤って上書きしないように、CREATE句を使用することができます。CREATEは、次のアクションを指定します。

- 宛先の表がすでに存在する場合、COPYはエラーを報告して停止します。
- 宛先の表が存在しない場合、COPYはコピーされたデータを使用してその表を作成します。

既存の表にデータを挿入するには、INSERTを使用します。INSERTは、次のアクションを指定します。

- 宛先の表がすでに存在する場合、COPYはコピーされたデータを宛先の表に挿入します。
- 宛先の表が存在しない場合、COPYはエラーを報告して停止します。

既存の表にデータを挿入する場合、または宛先の表が存在しないときに新しい表を作成する場合は、APPENDを使用します。APPENDは、次のアクションを指定します。

- 宛先の表がすでに存在する場合、COPYはコピーされたデータを宛先の表に挿入します。
- 表が存在しない場合、COPYは表を作成し、コピーされたデータをその表に挿入します。

ノート:



EMPLOYEE_COPY のコピーが入っているリモート・コンピュータ用の適切なユーザー名、パスワードおよびサービス名については、DBA に問い合わせてください。

```
COPY FROM HR@BOSTONDB -
```

```
CREATE EMPCOPY - USING SELECT * FROM HR
```

```
Array fetch/bind size is 15. (arraysize is 15)
Will commit when done. (copycommit is 0)
Maximum long size is 80. (long is 80)
```

その後、SQL*Plusは表EMPLOYEE_COPYを作成し、行をコピーします。

```
Table SALESMAN created.
```

```
5 rows selected from HR@BOSTONDB.
5 rows inserted into SALESMAN.
5 rows committed into SALESMAN at DEFAULT HOST connection.
```

このCOPYコマンドにFROM句を指定した場合は、BOSTONDBで指定されたデータベースに、ユーザー名HRで接続されます。

コマンドの終わりにセミコロンを付ける必要はありません。COPYは問合せを含んでも、SQLコマンドではなくSQL*Plusコマンドです。ほとんどのCOPYは長さが1行を超えるので、最終行を除く各行の終わりにハイフン(-)を使用する必要があります。ハイフンの前に空白を付けてもかまいません。

例B-1 CREATEを使用したリモート・データベースからローカル・データベースへのコピー

リモート・データベースからユーザー自身のデータベース上にあるEMPLOYEE_COPYという表にHRをコピーするには、次のコマン

ドを入力します。

B.2.3 COPYで表示されるメッセージの解釈について

COPYによって表示される最初の3つのメッセージは、そのCOPY操作に影響するSETコマンド変数の値を示しています。最も重要なのはLONGで、これはLONG列の値の長さを制限します。(LONGは、CHARと同じように1つのデータ型です。)元の表にLONG列が含まれている場合、COPYでは、その列の値はシステム変数LONGで指定された長さまで切り捨てられます。

ARRAYSIZE変数は、SQL*Plusがデータベースから一度にフェッチする行数を制限します。これらの行は、バッチを形成します。COPYCOMMIT変数には、COPYでデータベースに変更がコミットされる前にコピーされるバッチの数を設定します。(COPYCOMMITを0に設定した場合、COPYはすべてのバッチがコピーされた後にのみ、変更をコミットします。)設定の変更方法など、SET変数の詳細は、「[SET](#)」コマンドを参照してください。

3つのシステム変数およびそれらの値が表示された後、コピーの間に表の削除、作成または変更があったかどうかが表示されます。その後、選択、挿入およびコミットが実行された行数が表示されます。

B.2.4 別ユーザーの表の指定

ローカル・データベースの場合やデータベース・リンクを使用した問合せの場合と同様に、COPYコマンドの中で表名をユーザー名で修飾することによって、別のユーザー表を参照できます。

たとえば、ユーザー名ADAMSが所有し、Oracle Net接続識別子BOSTONDBに対応付けられているデータベース上にあるDEPARTMENTという名前の表のローカル・コピーを作成するには、次のように入力します。

```
COPY FROM HR@BOSTONDB -  
CREATE EMPLOYEE_COPY2 -  
USING SELECT * FROM ADAMS.DEPARTMENT
```

COPYにリモート・データベースへログインするよう指示することによって、ADAMSと同じ結果を得ることができます。ただし、ユーザー名ADAMSに対応付けられているパスワードを知らないと実行できません。

B.3 1つのデータベース上にある表の間でのデータのコピーについて

単一のデータベース(ローカルまたはリモート)内で、1つの表から別の表にデータをコピーできます。ローカル・データベース内の表と表の間でコピーするには、次のようにFROM句またはTO句のいずれかで(その他の句は省略し)、ユーザー自身のユーザー名およびローカル・データベースのサービス名を指定します。

```
COPY FROM HR@MYDATABASE -  
INSERT EMPLOYEE_COPY2 -  
USING SELECT * FROM EMPLOYEE_COPY
```

リモート・データベース上にある表と表の間でコピーするには、次のようにFROM句およびTO句の両方に同じユーザー名およびサービス名を挿入します。

```
COPY FROM HR@BOSTONDB -  
TO HR@BOSTONDB -  
INSERT EMPLOYEE_COPY2 -  
USING SELECT * FROM EMPLOYEE_COPY
```

C 廃止されたSQL*Plusコマンド

ここでは、一部のSQL*Plusコマンドの旧バージョンについて説明します。これらの旧コマンドは、これまで同様にSQL*Plusで動作しますが、現在はサポートされていません。かわりに、次の表に示すSQL*Plusコマンドを使用することをお勧めします。

C.1 SQL*Plusで廃止されたコマンドとその代替コマンド

廃止されたコマンドは、現行リリースのSQL*Plusで使用可能です。今後のリリースでは、SQLPLUSCOMPATIBILITY変数を設定した場合のみ、使用可能になる予定です。代替コマンドを使用するには、廃止されたコマンドを使用して、スクリプトを変更する必要があります。

廃止されたコマンド	代替コマンド	代替コマンドの説明
BTITLE (old form)	BTITLE	各レポート・ページの下部にタイトルを配置し書式設定するか、または現行の BTITLE 定義を表示します。
COLUMN DEFAULT	COLUMN CLEAR	列の表示属性をデフォルトにリセットします。
DOCUMENT	REMARK	SQL*Plus がコマンドとして解釈しないようにコメントを付けます。
NEWPAGE	SET NEWP[AGE] {1 n NONE}	各ページの最上部から上部タイトルまでの間に入れる空白行の数を設定します。
SET BUFFER	EDIT	SQL*Plus コマンド・バッファ、または保存されたファイルの内容を編集できます。外部ファイルの作成および使用には、SQL*Plus の SAVE、GET、@および START コマンドを使用します。
SET COMPATIBILITY	none	廃止
SET CLOSECURSOR	none	廃止
SET DOCUMENT	none	廃止
SET MAXDATA	none	廃止
SET SCAN	SET DEF[INE] {& c ON OFF}	接頭辞の置換変数に使用される文字を設定します。
SET SPACE	SET COLSEP { text}	SELECT によって選択された列の間に出力するテキ

廃止されたコマンド	代替コマンド	代替コマンドの説明
		ストを設定します。
SET TRUNCATE	SET WRA[P] {ON OFF}	SELECT によって選択された行が長すぎてカレント行の幅に収まらない場合に、その行を SQL*Plus で切り捨てるかどうかを制御します。
SHOW LABEL	none	廃止
TTITLE (old form)	TTITLE	各レポート・ページの上部にタイトルを配置し書式設定するか、または現行の TTITLE 定義を表示します。

C.2 BTI[TLE] テキスト(旧形式は廃止)

それぞれのレポート・ページの下部にタイトルを表示します。

旧形式のBTITLEコマンドは、書式設定機能に関しては、新形式と比べると制限されていますが、UFI(SQL*Plusの前身)との互換性を提供します。旧形式で定義される下部タイトルでは、空の行が1行入り、その後に中央揃えされたテキスト行が続きます。詳細は、「[TTI\[TLE\] text\(旧形式は廃止\)](#)」を参照してください。

C.3 COL[UMN] {列|*expr*} DEF[AULT] (廃止)

特定の列の表示属性をデフォルト値にリセットします。

COLUMN CLEARと同じ働きをします。

C.4 DOC[UMENT] (廃止)

スクリプト内のドキュメントのブロックを開始します。

スクリプトにコメントを挿入するための現行の方法については、[スクリプトへのコメントの挿入について](#)および[REMARK](#)コマンドを参照してください。

DOCUMENTを入力し[Return]を押すと、SQL*Plusでは、ドキュメントを終了するまで、「SQL>」のかわりに「DOC>」が表示されます。ドキュメントは、シャープ記号(#)のみが単独で入っている行で終わります。

DOCUMENTをOFFに設定した場合、SQL*Plusは、DOCUMENTコマンドで作成したドキュメントのブロックを表示しません。詳細は、「[SET DOC\[UMENT\] {ON|OFF}\(廃止\)](#)」を参照してください。

C.5 NEWPAGE [1|n] (廃止)

n行文のスプール出力を、次のページの先頭より先へ進めます。

スプール出力でのページ間隔を決める現行の方法については、「[SET NEWP\[AGE\] {1 | n | NONE}](#)」を参照してください。

C.6 SET BUF[FER] {バッファ|SQL} (廃止)

指定したbufferを現行バッファにします。

最初は、SQLバッファが現行バッファです。SQL*Plusでは、複数のバッファを使用する必要はありません。SQLバッファのみで要件が満たされます。

入力したバッファが存在しない場合、SET BUFFERは、そのバッファを定義(作成および命名)します。SQL*Plusを終了すると、SQL*Plusは、バッファおよびその内容を削除します。

問合せを実行すると、SQLバッファは、自動的に現行バッファになります。バッファから別のバッファへテキストをコピーするには、GETおよびSAVEコマンドを使用します。現行バッファからテキストを消去するには、CLEAR BUFFERを使用します。別のバッファを使用しているときにSQLバッファからテキストを消去するには、CLEAR SQLを使用します。

C.7 SET COM[patibility]{V7 | V8 | NATIVE} (廃止)

使用する構文を解析するSQL言語のバージョンを指定します。

COMPATIBILITYは、Oracle7の場合はV7に、Oracle8以上の場合はV8に設定します。COMPATIBILITYのデフォルトは、常にNATIVEです。そうでない場合は、COMPATIBILITYを、接続先のデータベースで使用するOracle DatabaseのSQL構文のバージョンに合わせて設定します。

デフォルトの互換性設定であるNATIVEは、最新のOracle Databaseに最も適した設定です。

SQL*Plusバージョンの互換性設定の詳細は、[SET SQLPLUSCOMPAT\[IBILITY\] {x.y\[.z\]}](#)を参照してください。

例

Oracle7のSQL構文で作成されたスクリプトSALARY.SQLを実行するには、次のように入力します。

```
SET COMPATIBILITY V7
START SALARY
```

このファイルを実行した後で、Oracle Database 10g用に作成されたスクリプトを実行するために互換性をNATIVEにリセットするには、次のように入力します。

```
SET COMPATIBILITY NATIVE
```

これ以外の方法として、スクリプトの最初にコマンドSET COMPATIBILITY V7を追加し、ファイルの終わりでCOMPATIBILITYをNATIVEにリセットすることもできます。

C.8 SET CLOSECUR[SOR] {ON|OFF} (廃止)

カーソル使用動作を設定します。

カーソルが、それぞれのSQL文の後でクローズおよび再オープンするかどうかを、ONまたはOFFで設定します。環境によっては、この機能が、データベース・サーバーのリソースを解放する際に有効な場合もあります。

C.9 SET DOC[UMENT] {ON|OFF} (廃止)

DOCUMENTコマンドで作成したドキュメントのブロックを表示または非表示にします。

SET DOCUMENT ONを指定すると、ドキュメントのブロックが画面にエコー表示されます。SET DOCUMENT OFFを指定すると、ドキュメントのブロックを非表示にします。

DOCUMENTコマンドについては、「[DOC\[UMENT\]\(廃止\)](#)」を参照してください。

C.10 SET MAXD[ATA] *n* (廃止)

SQL*Plusが処理できる最大行幅を設定します。

現在、SQL*Plusでは、最大行幅は制限されていません。SQL*Plusは、SET MAXDATAを使用して設定した値を無視します。

C.11 SET SCAN {ON|OFF} (廃止)

置換変数およびパラメータの存在確認のためのスキャンを制御します。OFFを指定すると、置換変数およびパラメータの処理を行わず、ONを指定すると、通常の処理ができます。

ONを指定すると、SET DEFINE ONと同じ機能が得られます。

C.12 SET SPACE {1|*n*} (廃止)

出力内の列間の空白数を設定します。*n*の最大値は10です。

SET SPACE 0コマンドの機能は、SET COLSEPコマンドと同じです。このコマンドは、SET COLSEPに置き換えられましたが、下位互換性のためにまだ使用できます。SHOWコマンドを使用すると、COLSEPは認識しますがSPACEは認識しないので、COLSEPを使用する方が便利です。

C.13 SET TRU[NCATE] {ON|OFF} (廃止)

SQL*Plusが、カレント行の幅に収まらないデータ項目を切り捨てるか折り返すかを制御します。

ONは、SET WRAP OFFと同じ働きをし、OFFは逆の働きをします。SHOWコマンドを使用すると、WRAPは認識しますがTRUNCATEは認識しないので、WRAPを使用する方が便利です。

C.14 TTI[TLE] テキスト (旧形式は廃止)

それぞれのレポート・ページの上部にタイトルを表示します。

旧形式のTTITLEコマンドは、書式設定機能に関しては、新形式と比べると制限されていますが、UFI(SQL*Plusの前身)との互換性を提供します。旧形式で定義される上部タイトルは、最初の行に日付が左揃えで、ページ番号が右揃えで入り、次に、中央揃えのテキストを含む行が続き、最後に空白行が1行入ります。

TTITLEは、ユーザーが入力するtextをタイトルとして表示します。

SQL*Plusは、SET LINESIZEが決める行サイズに基づいてテキストを中央揃えにします。セパレータ文字(|)があると新しい行が始まります。行の中に2個のセパレータ文字(||)があると、空白行が1行挿入されます。行セパレータ文字は、SET HEADSEPで変更できます。

旧形式のTTITLEおよびBTITLEのページ番号の書式設定を制御するには、_pageという名前の変数を定義します。_pageのデフォルト値は、書式設定文字列page &P4です。書式を変更するには、次のように、新しい書式設定文字列を指定したDEFINE _pageを使用します。

```
SET ESCAPE / SQL> DEFINE _page = 'Page /&P2'
```

この書式設定文字列の場合、pageの頭文字が大文字で表示され、ページ番号の書式が2文字の幅に設定されます。pageは、すべてのテキストで置き換えることができ、幅は、すべての数値で置き換えることができます。SQL*Plusがアンパサンド(&)を置換変数と解釈しないように、エスケープを設定する必要があります。エスケープ文字を設定する方法の詳細は、[SET ERRORL\[OGGING\] {ON | OFF} \[TABLE \[schema.\]tablename\] \[TRUNCATE\] \[IDENTIFIER identifier\]](#)

を参照してください。

コマンド名のすぐ後に有効な新形式の句が続いていない場合、SQL*Plusは、TTITLEを旧形式のコマンドと解釈します。

TTITLEとともにCENTERを使用して、1行に複数の語を入れるには、新形式のTTITLEを使用します。詳細は、「[TTITLE](#)」コマンドを参照してください。

例

旧形式のTTITLEを使用して、最初の行に左寄せの日付と右寄せのページ番号が示され、次の行にSALES DEPARTMENTが表示され、3番目の行にPERSONNEL REPORTが表示されるように上部タイトルを設定するには、次のように入力します。

```
TTITLE ' SALES DEPARTMENT | PERSONNEL REPORT '
```


D SQL*Plus Instant Client

SQL*Plus Instant Clientは、SQL*Plusのコマンドラインのすべての機能を備えたスタンドアロン製品です。既存の Remote Oracle Databaseに接続しますが、独自のデータベースはありません。簡単にインストールでき、SQL*Plusコマンドラインの使用に必要なOracle Database Clientの完全なインストールと比較すると、非常に少ないディスク領域を使用します。

SQL*Plus Instant Clientは、OCI Instant Clientがサポートされているプラットフォームで使用できます。OCI Instant Clientの詳細は、「[About OCI Instant Clientについて](#)」を参照してください。

SQL*Plus Instant Clientをインストールするには、次の2つのパッケージが必要です。

- SQL*Plus Instant Clientパッケージ
- 基本OCI Instant Clientパッケージまたは軽量OCI Instant Clientパッケージ

D.1 インストールするSQL*Plus Instant Clientの選択について

SQL*Plus Instant Clientは、次の2つの方法でインストールできます。

- Oracle Technology Network(OTN)からパッケージをダウンロードする方法
- Oracle Database 19c Client Administratorインストールのパッケージ内にある同じファイルをコピーする方法

SQL*PlusおよびOCIのパッケージの両方が、Oracle Databaseの同じバージョンからである必要があります。

D.1.1 基本Instant Client

基本OCIパッケージを使用するSQL*Plus Instant Clientは、Oracle DatabaseでサポートされているNLS_LANG設定で動作します。Oracle Databaseで使用可能なすべての文字セットおよび言語設定をサポートしています。

D.1.2 軽量Instant Client

量OCIパッケージを使用するSQL*Plus Instant Clientは、英語でのみエラー・メッセージを表示し、特定の文字セットのみをサポートしています。基本OCIパッケージを使用するSQL*Plus Instant Clientより大幅に小さくなります。

軽量Instant ClientでのNLS_LANGのパラメータに有効な値は、次のとおりです。

- *language*は、Oracle Databaseでサポートされている任意の有効な言語にできます。ただし、エラー・メッセージは英語でのみ報告されます。
- *territory*は、Oracle Databaseでサポートされている任意の有効な地域にできます。
- *charset*は、次の文字セットのいずれかにします。
 - US7ASCII
 - WE8DEC
 - WE8MSWIN1252
 - WE8ISO8859P1
 - UTF8
 - AL16UTF16

- AL32UTF8

次に例を示します。

```
NLS_LANG=AMERICAN_AMERICA.UTF8
```

NLS設定の詳細は、「[グローバル化のサポート環境の設定](#)」および「[環境変数NLS_LANG](#)」を参照してください。

D.1.2.1 サポートされていない文字セットによる軽量SQL*Plus Instant Clientエラー

サポートされていない文字セットでSQL*Plus Instant Clientを起動しようとすると、失敗し、次のエラーが戻されます。

```
Error 5 initializing SQL*Plus
NLS initialization error
```

D.2 OTNからのダウンロードによるSQL*Plus Instant Clientのインストールについて

Linux用のOTNダウンロードは、RPMパッケージです。UNIXおよびWindows用のOTNダウンロードは、zipファイルです。

SQL*Plus Instant Clientパッケージは、ORACLE_HOMEにはインストールしないでください。

D.2.1 Linux RPMパッケージからのSQL*Plus Instant Clientのインストール

1. SQL*Plusインスタント・クライアントパッケージを含むRPMパッケージおよびOCIパッケージを[OTNインスタント・クライアント](#)からダウンロードします。両方のパッケージは、同じバージョンのものである必要があります。
2. RPMパッケージを初期インストールするには`rpm -i`、新しいバージョンのパッケージにアップグレードするには`rpm -u`を使用します。
3. SQL*Plus Instant Clientを構成します。[SQL*Plus Instant Clientの構成について](#)を参照してください。

D.2.2 UNIXまたはWindows zipファイルからのSQL*Plus Instant Clientのインストール

1. SQL*Plusインスタント・クライアントパッケージを含むZipファイルおよびOCIパッケージを[OTNインスタント・クライアント](#)からダウンロードします。両方のパッケージは、同じバージョンのものである必要があります。
2. 新規ディレクトリ(UNIXでは`//home/instantclient12_2`、Windowsでは`c:\instantclient12_2`など)を作成します。
3. 2つのパッケージを新規ディレクトリに解凍します。
4. SQL*Plus Instant Clientを構成します。[SQL*Plus Instant Clientの構成について](#)を参照してください。

D.2.3 SQL*Plus Instant Clientに必要なファイルのリスト

表E-1、E-2およびE-3は、SQL*PlusおよびOCIのパッケージそれぞれに含まれる必要なファイルのリストです。OCIパッケージのうちのいずれか1つのパッケージのみのファイルが必要です。インストールされるその他のファイルでこのリストに含まれていないファイルは、無視するか、ディスク領域を節約するために削除できます。

表D-1 SQL*Plusパッケージ内のInstant Clientファイル

LinuxおよびUNIX	Windows	説明
sqlplus	sqlplus.exe	SQL*Plus 実行可能ファイル

LinuxおよびUNIX	Windows	説明
libsqlplus.so	該当なし	SQL*Plus ライブラリ
libsqlplusic.so	orasqlplusic19.dll	SQL*Plus データ共有ライブラリ

表D-2 基本OCIパッケージ内のInstant Clientファイル

LinuxおよびUNIX	Windows	説明
libclntsh.so.19.1	oci.dll	クライアント・コード・ライブラリ
libclntshcore.so	該当なし	OCI Instant Client データ共有ライブラリ
libociei.so	oraociei19.dll	OCI Instant Client データ共有ライブラリ
libnnz19.so	orannzsbb19.dll	セキュリティ・ライブラリ
libons.so	oraons.dll	ONS ライブラリ

表D-3 軽量OCIパッケージ内のInstant Clientファイル

LinuxおよびUNIX	Windows	説明
libclntsh.so.19.1	oci.dll	クライアント・コード・ライブラリ
libociicus.so	oraociicus19.dll	OCI Instant Client データ共有ライブラリ(英語のみ)
libnnz19.so	orannzsbb19.dll	セキュリティ・ライブラリ

D.3 19c Clientリリース・メディアからのSQL*Plus Instant Clientのインストール

1. Oracle Database 19c Clientリリース・メディアでインストーラを実行し、管理者オプションを選択します。
2. 新規ディレクトリ(UNIXおよびLinuxでは`/home/instantclient_19_1`、Windowsでは`c:\instantclient_19_1`など)を作成します。
3. SQL*Plus Instant ClientおよびOCI Instant Clientファイルを新規ディレクトリにコピーします。すべてのファイルを同じORACLE_HOMEからコピーする必要があります。

コピーするファイルのリストは、「[UNIXまたはLinuxでのSQL*Plus Instant Clientのインストール](#)」または「[WindowsでのSQL*Plus Instant Clientのインストール](#)」を参照してください。
4. SQL*Plus Instant Clientを構成します。[SQL*Plus Instant Clientの構成について](#)を参照してください。

D.3.1 UNIXまたはLinuxでのSQL*Plus Instant Clientのインストール

UNIXおよびLinuxで基本OCIパッケージを使用してSQL*Plus Instant Clientをインストールするには、次のファイルをコピーします。

```
$ORACLE_HOME/instantclient/liboci12.so
$ORACLE_HOME/lib/libc12sh.so.19.1
$ORACLE_HOME/lib/libnnz12.so
$ORACLE_HOME/lib/libsqlplus.so
$ORACLE_HOME/lib/libsqlplusic.so
$ORACLE_HOME/bin/sqlplus
```

UNIXおよびLinuxで軽量OCIパッケージを使用してSQL*Plus Instant Clientをインストールするには、次のファイルをコピーします。

```
$ORACLE_HOME/instantclient/light/liboci12us.so
$ORACLE_HOME/lib/libc12sh.so.19.1
$ORACLE_HOME/lib/libnnz12.so
$ORACLE_HOME/lib/libsqlplus.so
$ORACLE_HOME/lib/libsqlplusic.so
$ORACLE_HOME/bin/sqlplus
```

D.3.2 WindowsでのSQL*Plus Instant Clientのインストール

Windowsで基本OCIパッケージを使用してSQL*Plus Instant Clientをインストールするには、次のファイルをコピーします。

```
ORACLE_HOME¥instantclient¥oraoci12.dll
ORACLE_HOME¥bin¥oci.dll
ORACLE_HOME¥bin¥orannzsb12.dll
ORACLE_HOME¥bin¥orasqlplusic12.dll
ORACLE_HOME¥bin¥sqlplus.exe
```

Windowsで軽量OCIパッケージを使用してSQL*Plus Instant Clientをインストールするには、次のファイルをコピーします。

```
ORACLE_HOME¥instantclient¥light¥oraoci12us.dll
ORACLE_HOME¥bin¥oci.dll
ORACLE_HOME¥bin¥orannzsb12.dll
ORACLE_HOME¥bin¥orasqlplusic12.dll
ORACLE_HOME¥bin¥sqlplus.exe
```

D.4 SQL*Plus Instant Clientの構成について

SQL*Plus Instant Client実行可能ファイルは、バージョンが一致するOCI Instant Clientとともにのみ使用する必要があります。

環境変数ORACLE_HOMEまたはORACLE_SIDを設定する必要はありません。

D.4.1 Linuxでの(RPMからの)SQL*Plus Instant Clientの構成

OTNからダウンロードしたRPMは、/usrファイル・システム内のOracle固有のサブディレクトリにインストールされます。サブディレクトリ構造によって、複数のバージョンのInstant Clientを使用できます。

1. Instant Clientライブラリを含むディレクトリの名前をLD_LIBRARY_PATHに追加します。その他のOracleディレクトリは削除します。

たとえば、BourneシェルまたはKornシェルでSolarisにLD_LIBRARY_PATHを設定するには、次のように指定しま

す。

```
LD_LIBRARY_PATH=/usr/lib/oracle/19.1/client/lib:${LD_LIBRARY_PATH}
export LD_LIBRARY_PATH
```

2. RPMからインストールした`sqlplus`実行可能ファイルがPATHの先頭にあることを確認します。テストするには、`which sqlplus`と入力します。`/usr/bin/sqlplus`が戻されるはずですが、そうでない場合は、他のOracleディレクトリをPATHから削除するか、PATH内の他のSQL*Plus実行可能ファイルの前に`/usr/bin`を入力するか、あるいは絶対パスまたは相対パスを使用してSQL*Plusを起動します。

たとえば、bashシェルでPATHを設定するには、次のように指定します。

```
PATH=/usr/bin:${PATH}
export PATH
```

複数のバージョンのSQL*Plusをインストールする場合は、シンボリック・リンク`/usr/bin/sqlplus`を、`LD_LIBRARY_PATH`内のライブラリと一致するSQL*Plusのバージョンに変更する必要がある場合があります。19.1の場合、`/usr/bin/sqlplus`は、`/usr/lib/oracle/19.1/client/bin/sqlplus`にあるSQL*Plusバイナリのシンボリック・リンクです。

3. ロケールに必要なOracleグローバル化変数を設定します。変数を設定しない場合は、デフォルトのロケールが指定されたものとみなされます。詳細は、「[ロケール・データ](#)」を参照してください。

次に例を示します。

```
NLS_LANG=AMERICAN_AMERICA.UTF8
export NLS_LANG
```

D.4.2 Linux (Clientメディアまたはzipファイルから)およびUNIXでのSQL*Plus Instant Clientの構成

1. Instant Clientファイルを含むディレクトリの名前を、適切な共有ライブラリ・パス`LD_LIBRARY_PATH`、`LIBPATH`または`SHLIB_PATH`に追加します。その他のOracleディレクトリは削除します。

たとえば、BourneシェルまたはKornシェルでSolarisに設定するには、次のように指定します。

```
LD_LIBRARY_PATH=/home/instantclient_19_1:${LD_LIBRARY_PATH}
export LD_LIBRARY_PATH
```

2. Instant Clientファイルを含むディレクトリを環境変数PATHに追加します。設定されていない場合は、絶対パスまたは相対パスを使用してSQL*Plusを起動する必要があります。その他のOracleディレクトリはPATHから削除します。次に例を示します。

```
PATH=/home/instantclient_19_1:${PATH}
export PATH
```

3. ロケールに必要なOracleグローバル化変数を設定します。変数を設定しない場合は、デフォルトのロケールが指定されたものとみなされます。詳細は、「[ロケール・データ](#)」を参照してください。

次に例を示します。

```
NLS_LANG=AMERICAN_AMERICA.UTF8
export NLS_LANG
```

D.4.3 WindowsでのSQL*Plus Instant Clientの構成

環境は、Windowsコマンド・プロンプトでSETコマンドを使用して構成するか、またはシステム・プロパティの環境変数を設定して永続的にできます。

たとえば、Windows 2000でシステム・プロパティを使用して環境変数を設定するには、「コントロール パネル」から「システム」を開き、「詳細設定」タブをクリックしてから、「環境変数」をクリックします。

1. Instant Clientファイルを含むディレクトリをシステム環境変数PATHに追加します。その他のOracleディレクトリはPATHから削除します。

たとえば、PATHの先頭にc:\instantclient12_2を追加します。

2. ロケールに必要なOracleグローバル化変数を設定します。変数を設定しない場合は、デフォルトのロケールが指定されたものとみなされます。詳細は、「[ロケール・データ](#)」を参照してください。たとえば、日本語環境用のNLS_LANGを設定するには、JAPANESE_JAPAN.JA16EUCに設定されたユーザー環境変数NLS_LANGを作成します。

軽量Instant Clientをインストールした場合、サポートされているNLS_LANG設定については、「[軽量Instant Client](#)」を参照してください。

D.5 SQL*Plus Instant Clientによるデータベースへの接続について

SQL*Plus Instant Clientは、すべてのデータベース・サーバーから常にリモートの状態になります。データベースに接続するには、Oracle Net接続識別子を使用して、データベースを指定する必要があります。

簡単な接続識別子を使用して、*mymachine*で実行されているMYDBデータベース内のHRスキーマに接続する例は、次のとおりです。

```
sqlplus hr/your_password@¥"/mymachine.mydomain:port/MYDB¥"
```

また、ネット・サービス名も使用できます。

```
sqlplus hr/your_password@MYDB
```

ネット・サービス名は、LDAPなどの様々な場所に格納できます。Oracle Database 12cの新機能を利用するには、LDAPを使用することをお勧めします。詳細は、「[tnsnames.oraファイルのローカル・ネーミング・パラメータ](#)」を参照してください。

ローカルのOracle Net *tnsnames.ora*ファイル内に構成されたネット・サービス名を使用する場合は、*tnsnames.ora*ファイルを含むディレクトリに環境変数TNS_ADMINを設定します。たとえば、UNIXで、*/home/user1*にある*tnsnames.ora*ファイルによってネット・サービス名MYDB2が定義されている場合は、次のように設定します。

```
TNS_ADMIN=/home/user1
export TNS_ADMIN
sqlplus hr@MYDB2
```

TNS_ADMINが設定されていない場合は、オペレーティング・システム依存のディレクトリ内で*tnsnames.ora*が検索されます。この検索パスには*network/admin/tnsnames.ora*も含まれ、ORACLE_HOME環境変数で指定されているディレクトリも検索されます。環境変数ORACLE_HOMEは、このためにのみ、SQL*Plus Instant Clientに設定します。Instant Clientアプリケーションの実行時にORACLE_HOMEを設定する場合は、既存のディレクトリに設定する必要があります。

この例は、環境変数ORACLE_HOMEが設定されていて、*\$ORACLE_HOME/network/admin/tnsnames.ora*または*ORACLE_HOME¥network¥admin¥tnsnames.ora*ファイルによってネット・サービス名MYDB3が定義されていることを前提としています。

```
sqlplus hr@MYDB3
```

環境変数TWO_TASK(UNIXの場合)またはLOCAL(Windowsの場合)には接続識別子を設定できます。これによって、SQL*PlusまたはSQL*Plus Instant Clientで接続が行われるたびに接続識別子を明示的に入力する必要がなくなります。このUNIXの例では、MYDB4という名前のデータベースに接続します。

```
TNS_ADMIN=/home/user1
export TNS_ADMIN
TWO_TASK=MYDB4
export TWO_TASK
sqlplus hr
```

Windowsでは、TNS_ADMINおよびLOCALは、システム・プロパティで設定できます。「[WindowsでのSQL*Plus Instant Clientの構成](#)」を参照してください。

D.6 SQL*Plus Instant ClientによるAS SYSDBAまたはAS SYSOPER接続

AS SYSDBAまたはAS SYSOPERで接続してDBA作業を実行するには、データベースの*orapwd*ユーティリティを使用して、データベース・サーバーにOracleパスワード・ファイルを設定する必要があります。これを設定すると、SQL*Plus Instant Client接続文字列は次のようになります。

```
sqlplus sys@MYDB AS SYSDBA
```

Oracleパスワード・ファイルの詳細は、「[パスワード・ファイル認証の使用](#)」を参照してください。

D.7 Instant Clientのアンインストールについて

SQL*Plus Instant Clientパッケージは、OCI Instant Clientとは別に削除できます。SQL*Plus Instant Clientパッケージをアンインストールした後も、残されたOCI Instant Clientライブラリによって、カスタム作成されたOCIプログラムまたはサード・パーティのデータベース・ユーティリティでデータベースに接続できます。

D.7.1 SQL*Plus Instant Clientのアンインストール

1. LinuxでのRPMパッケージからのインストールの場合は、SQL*Plus Instant Clientパッケージに対してのみ`rpm -e`を使用します。

または

UNIXおよびWindowsでのインストールの場合、およびLinuxでのClientリリース・メディアからのインストールの場合は、次のSQL*Plus固有のファイルを手動で削除します。

表D-4 SQL*Plusパッケージ内のInstant Clientファイル

UNIX	Windows	説明
sqlplus	sqlplus.exe	SQL*Plus 実行可能ファイル
libsqlplus.so	該当なし	SQL*Plus ライブラリ
libsqlplusic.so	orasqlplusic12.dll	SQL*Plus データ共有ライブラリ

2. 環境変数を再設定し、必要に応じて`tnsnames.ora`を削除します。

D.7.2 Instant Clientの完全なアンインストール

1. LinuxでのRPMパッケージからのインストールの場合は、`rpm -qa`を使用してSQL*Plus Instant Clientおよび基本OCIパッケージの名前を検索し、`rpm -e`を実行してそれらを削除します。

または

UNIXおよびWindowsでのインストールの場合、およびLinuxでのClientリリース・メディアからのインストールの場合は、SQL*Plus実行可能ファイルおよびOracleライブラリを含むディレクトリを手動で削除します。

ディレクトリにコピーしたファイルのリストは、「[12c Clientリリース・メディアからのSQL*Plus Instant Clientのインストール](#)」を参照してください。

2. PATH、SQLPATH、LD_LIBRARY_PATH、TNS_ADMINなどの環境変数を再設定します。
3. 必要に応じて、`tnsnames.ora`を削除します。

索引

記号 数値 [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#)

記号

- `_CONNECT_IDENTIFIER`事前定義変数 [2.2.2.1](#), [12.17.1](#)
- `_DATE`事前定義変数 [12.17.1](#)
- `_EDITOR`、`EDIT`コマンド [5.1.1](#), [12.17.1](#), [12.21](#)
- `_EDITOR`事前定義変数 [5.1.1](#), [12.17.1](#), [12.21](#)
- `_EDITOR`置換変数 [12.17.1](#)
- `_O_RELEASE`事前定義変数 [12.17.1](#)
- `_O_VERSION`事前定義変数 [12.17.1](#)
- `_PRIVILEGE`事前定義変数 [12.17.1](#)
- `_RC`事前定義変数 [12.27](#)
- `_SQLPLUS_RELEASE`事前定義変数 [12.17.1](#)
- `_USER`事前定義変数 [12.17.1](#)
- `;` (セミicolon) [4.4](#), [5.2](#), [12.29](#)
- `/*...*/` (コメント・デリミタ) [5.3.2](#)
- `~`無限大記号 [12.13](#)
- `-~`負の無限大記号 [12.13](#)
- `((amp))` (`(amp)`) [5.9.2](#)
- `((amp))` (アンパサンド)
 - 置換変数の無効化 [5.9.4](#)
 - 置換変数 [5.9](#)
- `((colon))` (colon)
 - バインド変数 [5.12](#)
- `((colon))` `BindVariable`句
 - `EXIT`コマンド [12.23](#)
- `((nbsp))` (スラッシュ)コマンド
 - デフォルト・ログオン [3.5.2](#), [12.15](#)
 - バッファ行番号プロンプトでの入力 [4.4.1.2](#), [12.4](#)
 - コマンド・プロンプトでの入力 [12.4](#)
 - 現行のPL/SQLブロックの実行 [4.5](#)
 - `RUN`との類似 [12.4](#), [12.38](#)
 - 使用方法 [12.4](#)
- `((nbsp)).` (ピリオド) [4.5](#)
- `((nbsp))--` (コメント・デリミタ) [5.3.3](#)
- `((nbsp))-` (ハイフン)
 - 句 [3.5.1.1](#)
 - 長いSQL*Plusコマンドの継続 [4.6.1.1](#), [12](#)
- `@` (アットマーク)
 - コマンド [2.2.3.1](#), [5.4](#), [12.2](#)
 - コマンド引数 [12.2](#), [12.3](#)

- CONNECTコマンド [12.15](#)
- COPYコマンド [B.1](#), [B.2.1](#)
- SQLPLUSコマンド [3.5](#)
- スクリプトへのパラメータの受渡し [12.2](#), [12.3](#)
- スクリプト [5.4](#), [12.2](#)
- STARTとの類似 [5.4](#), [12.45](#)
- @@ (二重アットマーク)コマンド [2.2.3.1](#), [12.3](#)
 - スクリプト [12.3](#)
 - STARTとの類似 [12.3](#), [12.45](#)
- @ (アットマーク)
 - STARTとの類似 [12.2](#)
- \$数値書式 [6.1.2.2](#)
- * (アスタリスク)
 - DELコマンド [5.2](#), [12.18](#)
 - LISTコマンド [5.2](#), [12.29](#)
- #シャープ記号
 - オーバーフローの通知 [12.13](#)
 - SET SQLPREFIX文字 [12.41.56](#)
- -MARKUP [3.5.1.6](#), [7.1](#)
 - SQLPLUSコマンド句 [3.5.1.6](#), [12.41.36](#)
- -SILENTオプション [3.5.1.10](#), [7.1.3](#)

数値

- 0, 数値書式 [6.1.2.2](#)
 - 9, 数値書式 [6.1.2.2](#)
-

A

- ABORTモード [12.43](#)
- 問合せの中断 [4.8](#)
- ACCEPTコマンド [5.11.1](#), [12.5](#)
 - DEFINEコマンド [12.17](#)
 - BINARY_DOUBLE句 [12.5](#), [12.51](#)
 - BINARY_FLOAT句 [12.5](#), [12.51](#)
 - 値の入力を求めるプロンプトのカスタマイズ [5.11.2](#)
 - DATE句 [12.5](#)
 - DEFAULT句 [12.5](#)
 - FORMAT句 [12.5](#)
 - HIDE句 [12.5](#)
 - NOPROMPT句 [12.5](#)
 - NUMBER句 [5.11.2](#)
 - PROMPT句 [5.11.1](#), [12.5](#)
- 別名 [3.2.2](#)

- ALIAS句 [12.13](#)
 - ATTRIBUTEコマンド [12.8](#)
- ALL句 [12.42](#)
- アンパサンド(((amp)))
 - パラメータ [5.10](#), [12.2](#), [12.3](#), [12.45](#)
 - 置換変数 [5.9](#)
- APPEND句
 - COPYコマンド [B.1.1](#), [B.2.2](#)
 - SAVEコマンド [12.39](#), [12.44](#)
- APPENDコマンド [5.2](#), [5.2.3](#), [12.6](#)
- APPINFO句 [8.5.2](#), [12.41.1](#)
- ARCHIVE LOG
 - コマンド [10.3.1](#), [12.7](#)
 - モード [10.3.1](#)
- 引数
 - STARTコマンド [5.10](#)
- ARRAYSIZE変数 [8.5.3](#), [12.41](#), [12.41.2](#)
 - COPYコマンドとの関連 [B.1.2](#), [B.2.3](#)
- ATTRIBUTEコマンド [12.8](#)
 - ALIAS句 [12.8](#)
 - CLEAR COLUMNコマンド [12.8](#)
 - CLEAR句 [12.8](#)
 - 列のクリア [12.12](#), [12.13](#)
 - 表示特性の制御 [12.8](#)
 - 表示特性 [12.8](#)
 - 複数入力 [12.8](#)
 - FORMAT句 [12.8](#)
 - LIKE句 [12.8](#)
 - 属性の表示特性の表示 [12.8](#)
 - OFF句 [12.8](#)
 - ON句 [12.8](#)
 - 列の表示属性の表示 [12.8](#)
 - 列の表示属性の非表示 [12.8](#)
- AUTOCOMMIT変数 [4.11](#), [12.41](#), [12.41.3](#)
- AUTOMATIC句 [12.34](#)
- AUTOPRINT変数 [12.41](#), [12.41.4](#)
- AUTORECOVERY変数 [12.41](#), [12.41.5](#)
- 自動トレース・レポート [8.1.1](#)
- AUTOTRACE変数 [8.1.1](#), [12.41.6](#)

B

- バックグラウンド・プロセス
 - 異常終了後の起動 [12.43](#)

- BASEURI変数 [12.41](#)
- BASEURI XQUERYオプション [12.41.71](#)
- 基本OCIパッケージ [D.1.1](#)
- バッチ・ジョブ、ユーザーの認証 [3.1.1](#)
- バッチ・モード [12.23](#)
- BEGINコマンド [4.5](#)
- BINARY_DOUBLE句
 - ACCEPTコマンド [12.5](#), [12.51](#)
 - VARIABLEコマンド [12.51](#)
- BINARY_FLOAT句
 - ACCEPTコマンド [12.5](#), [12.51](#)
 - VARIABLEコマンド [12.51](#)
- バインド変数 [5.12](#)
 - 作成 [12.51](#)
 - 表示 [12.32](#)
 - 自動表示 [12.41.4](#), [12.51](#)
 - PL/SQLブロック [12.51](#)
 - SQL文 [12.51](#)
 - COPYコマンド [12.51](#)
- 空白行
 - PL/SQLブロック [4.5](#)
 - SQLコマンド [4.4.1.2](#)
 - SQLコマンドでの保存 [12.41](#), [12.41.51](#)
- BLOB
 - 列幅 [6.1.3.1](#)
 - レポート内での書式設定 [6.1.3](#)
- BLOB列
 - デフォルト書式 [12.13](#)
 - 最大幅の設定 [12.41](#), [12.41.34](#)
 - 取得位置の設定 [12.41](#), [12.41.31](#)
 - 取得サイズの設定 [8.5.7](#), [12.41](#), [12.41.35](#)
- ブロック, PL/SQL
 - 継続 [4.5](#)
 - バッファ内の編集 [5.2](#)
 - システム・エディタでの編集 [5.1](#), [12.21](#)
 - 入力および実行 [4.5](#)
 - バッファ内の現在の状態をリスト [5.2.1](#)
 - 現行値の保存 [12.39](#)
 - 終了のために使用する文字設定 [12.41](#), [12.41.7](#)
 - SQLバッファに格納 [4.5](#)
 - タイミング統計 [12.41.64](#)
 - SQLコマンド [4.5.1](#)
- BLOCKTERMINATOR [12.41](#), [12.41.7](#), [12.41.51](#), [12.41.58](#)
- BODY句 [3.5.1.6](#)
- BODYオプション [3.5.1.6](#)

- BOLD句 [12.37](#), [12.49](#)
- ブレーク列 [6.2](#), [12.9](#)
 - 値が変更されたときの空白の挿入 [6.2.2](#)
 - 複数の指定 [6.2.4](#)
 - 値の重複の抑止 [6.2.1](#)
- BREAKコマンド [6.2](#), [12.9](#)
 - SQLのORDER BY句 [6.2](#), [6.2.1](#), [6.2.4](#), [12.9](#)
 - BREAKSのクリア [6.2.5](#)
 - タイトルへの列値の表示 [6.3.4](#)
 - DUPLICATES句 [12.9](#)
 - 各行の後への空白の挿入 [6.2.3](#)
 - ブレーク列が変更されたときの空白の挿入 [6.2.2](#)
 - 現行のブレーク定義の表示 [6.2.5](#), [12.9](#)
 - ON column句 [6.2.1](#), [12.9](#)
 - ON expr句 [12.9](#)
 - ON REPORT句 [6.2.7](#), [12.9](#)
 - ON ROW句 [6.2.3](#), [12.9](#)
 - 総サマリーおよびサブサマリーの出力 [6.2.7](#)
 - レポートの終わりでサマリー行の出力 [6.2.7](#)
 - 定義の取消し [12.12](#)
 - SKIP句 [6.2.3](#), [12.9](#)
 - SKIP PAGE句 [6.2.2](#), [6.2.3](#), [12.9](#)
 - 複数のブレーク列の指定 [6.2.4](#), [12.9](#)
 - 値の重複の抑止 [6.2.1](#)
 - COMPUTEとの併用 [6.2.6](#)
 - SET COLSEPとの併用 [12.41.10](#)
 - REFCURSOR変数の書式設定に使用 [12.51](#)
 - COMPUTEでの使用 [12.9](#), [12.14](#)
- ブレーク定義
 - 現行の表示 [6.2.5](#), [12.9](#)
 - 現行の取消し [6.2.5](#), [12.12](#)
- BREAKS句 [6.2.5](#), [12.12](#)
- ブラウザ, Web [7.1.1](#)
- BTITLE句 [12.42](#)
- BTITLEコマンド [6.3.1](#), [12.10](#)
 - タイトル要素の整列 [12.49](#)
 - BOLD句 [12.49](#)
 - CENTER句 [12.49](#)
 - COL句 [12.49](#)
 - FORMAT句 [12.49](#)
 - タイトルのインデント [12.49](#)
 - LEFT句 [12.49](#)
 - OFF句 [12.49](#)
 - 旧形式 [C.2](#)
 - 下部タイトルの前に空白行を出力 [6.3.1.1](#)

- 列値変数の参照 [12.13](#)
- RIGHT句 [12.49](#)
- SKIP句 [12.49](#)
- 現行の定義の非表示 [12.49](#)
- TAB句 [12.49](#)
- TTITLEコマンド [12.10](#)
- バッファ [4.1.1](#)
 - 行へのテキストの追加 [5.2.3](#), [12.6](#)
 - 1行の削除 [5.2](#)
 - カレント行の削除 [5.2](#)
 - 最終行の削除 [5.2](#)
 - すべての行の削除 [5.2](#), [12.12](#), [12.18](#)
 - ある範囲の行の削除 [5.2](#), [12.18](#)
 - 1行の削除 [12.18](#)
 - 行の削除 [5.2.5](#), [12.18](#)
 - カレント行の削除 [12.18](#)
 - 最終行の削除 [12.18](#)
 - 内容の実行 [12.4](#), [12.38](#)
 - 新規行の挿入 [5.2.4](#), [12.28](#)
 - すべての行の表示 [5.2](#), [12.29](#)
 - ある範囲の行の表示 [5.2](#), [12.29](#)
 - 1行の表示 [5.2](#), [12.29](#)
 - 内容の表示 [5.2.1](#), [12.29](#)
 - カレント行の表示 [5.2](#), [12.29](#)
 - 最終行の表示 [5.2](#), [12.29](#)
 - システム・エディタへのロード [12.21](#)
 - 内容の保存 [12.39](#)
- BUFFER句 [5.1.1](#), [5.2](#), [12.12](#)
- BUFFER変数 [C.6](#)

C

- CANCEL句 [12.34](#)
- 取消しキー [4.8](#)
- 問合せの取消し [4.8](#)
- CENTER句 [6.3.1.1](#), [12.37](#), [12.49](#)
- CHANGEコマンド [5.2](#), [5.2.2](#), [12.11](#)
- CHAR句
 - VARIABLEコマンド [12.51](#)
- CHAR列
 - 書式の変更 [12.13](#)
 - デフォルト書式 [6.1.3.1](#)
 - DESCRIBEによる定義 [12.19](#)
- charset

- SQL*Plus Instant Client [D.1.2](#)
- CLEAR句 [6.1.5](#), [12.13](#), [12.26](#)
 - ATTRIBUTEコマンド [12.8](#)
- CLEARコマンド [12.12](#)
 - BREAKS句 [6.2.5](#), [12.12](#)
 - BUFFER句 [5.1.1](#), [5.2](#), [12.12](#)
 - COLUMNS句 [12.12](#)
 - COMPUTES句 [12.12](#)
 - SCREEN句 [5.11.4](#), [12.12](#)
 - SQL句 [12.12](#)
 - TIMING句 [12.12](#)
- CLOB句
 - VARIABLEコマンド [12.51](#)
- CLOB列
 - 書式の変更 [12.13](#)
 - デフォルト書式 [12.13](#)
 - 最大幅の設定 [12.41](#), [12.41.34](#)
 - 取得位置の設定 [12.41](#), [12.41.31](#)
 - 取得サイズの設定 [8.5.7](#), [12.41](#), [12.41.35](#)
- CLOSECURSOR変数 [C.1](#), [C.8](#)
- CMDSEP変数 [12.41](#), [12.41.8](#)
- COL句 [6.3.1.2](#), [12.37](#), [12.49](#)
- COLINVISIBLE変数 [12.41.9](#)
- コロン(((colon)))
 - バインド変数 [5.12](#)
- COLSEP変数 [12.41](#), [12.41.10](#)
- COLUMNコマンド [6.1](#), [12.13](#)
 - ALIAS句 [12.13](#)
 - BREAKコマンド [12.9](#)
 - DEFINEコマンド [12.17](#)
 - CLEAR句 [6.1.5](#), [12.13](#)
 - DEFAULT句 [C.3](#)
 - 下部タイトルへの列値の表示 [6.3.4](#), [12.13](#)
 - 上部タイトルへの列値の表示 [6.3.4](#), [12.13](#)
 - 複数入力 [12.13](#)
 - ENTMAP句 [12.13](#)
 - FOLD_AFTER句 [12.13](#)
 - FOLD_BEFORE句 [12.13](#)
 - FORMAT句 [6.1.2.2](#), [6.1.3.2](#), [12.13](#)
 - REFCURSOR変数の書式設定 [12.51](#)
 - NUMBER列の書式設定 [6.1.2.1](#), [12.13](#)
 - HEADING句 [6.1.1.2](#), [12.13](#)
 - HEADSEP文字 [12.13](#)
 - JUSTIFY句 [12.13](#)
 - LIKE句 [6.1.4](#), [12.13](#)

- 列の表示属性の表示 [6.1.5](#), [12.13](#)
- NEW_VALUE句 [6.3.4](#), [12.13](#)
- NEWLINE句 [12.13](#)
- NOPRINT句 [6.3.4](#), [8.5.1](#), [12.13](#)
- NULL句 [12.13](#)
- OFF句 [6.1.6](#), [12.13](#)
- OLD_VALUE句 [6.3.4](#), [12.13](#)
- ON句 [6.1.6](#), [12.13](#)
- PRINT句 [12.13](#)
- 列の表示をデフォルトにリセット [C.1](#)
- 表示をデフォルトにリセット [6.1.5](#), [12.13](#), [C.1](#)
- 列の表示属性の表示 [6.1.6](#), [12.13](#)
- タイトルに対する変数内の現在の日付の格納 [12.13](#)
- 列の表示属性の非表示 [6.1.6](#), [12.13](#)
- TRUNCATED句 [6.1.3.2](#), [12.13](#)
- WORD_WRAPPED句 [6.1.3.2](#), [6.1.7](#), [12.13](#)
- WRAPPED句 [6.1.3.2](#), [12.13](#)
- 列ヘッダー
 - 整列 [12.13](#)
 - 変更 [6.1.1.1](#), [12.13](#)
 - 下線に使用する文字の変更 [12.41](#), [12.41.67](#)
 - 複数の単語に変更 [6.1.1.2](#), [12.13](#)
 - 複数行への表示 [6.1.1.2](#), [12.13](#)
 - レポートに出力しない [12.41](#), [12.41.26](#)
 - 切り捨てられる場合 [12.13](#)
 - CHARおよびLONG列で切り捨てられる場合 [6.1.3.2](#)
 - DATE列で切り捨てられる場合 [6.1.3.2](#)
 - NUMBER列で切り捨てられる場合 [6.1.2.1](#)
- 列
 - 別名の割当て [12.13](#)
 - サマリー行の計算 [6.2.6](#), [12.14](#)
 - 表示属性のコピー [6.1.4](#), [12.8](#), [12.13](#)
 - 表間での値のコピー [B.1](#), [B.2](#), [B.3](#)
 - 下部タイトルへの値の表示 [6.3.4](#), [12.13](#)
 - 上部タイトルへの値の表示 [6.3.4](#), [12.13](#)
 - CHAR、VARCHAR、LONGおよびDATEの書式設定 [12.13](#)
 - レポート内での書式設定 [6.1](#), [12.13](#)
 - MLSLABEL、RAW MLSLABEL、ROWLABELの書式設定 [12.13](#)
 - NUMBERの書式設定 [6.1.2.1](#), [12.13](#)
 - すべての列の表示属性の表示 [6.1.5](#), [12.13](#)
 - 1つの列の表示属性の表示 [6.1.5](#), [12.13](#)
 - コピー時の宛先の表での名前 [B.1.1](#), [B.2.1](#)
 - オーバーフローした値の後の行の出力 [6.1.7](#), [12.41](#), [12.41.44](#)
 - 列の表示をデフォルトにリセット [6.1.5](#), [12.13](#), [C.1](#)
 - すべての列の表示をデフォルトにリセット [12.12](#)

- 表示属性の表示 [6.1.6](#), [12.8](#), [12.13](#)
- 出力をOFFまたはONに設定 [6.3.4](#), [8.5.1](#), [12.13](#)
- 新しい行の開始 [12.13](#)
- 変数への値の格納 [6.3.4](#), [12.13](#)
- 表示属性の非表示 [6.1.6](#), [12.8](#), [12.13](#)
- 値がオーバーフローした場合の全列値の切捨て表示 [6.1.3.2](#), [12.41.69](#)
- 値がオーバーフローした場合の列値の切捨て表示 [6.1.3.2](#), [12.13](#)
- 値がオーバーフローした場合の全列値の折返し表示 [6.1.3.2](#), [12.41.69](#)
- 値がオーバーフローした場合の列値の折返し表示 [6.1.3.2](#), [12.13](#)
- すべての単語を1列に折返し [6.1.7](#)
- COLUMNS句 [12.12](#)
- 列セパレータ [12.41](#), [12.41.10](#), [C.1](#)
- カンマ, 数値書式 [6.1.2.2](#)
- コマンド・ファイル
 - リターン・コードを伴う強制終了 [5.6](#), [12.52](#), [12.53](#)
 - システム・エディタでの作成 [5.1.1](#)
 - SAVEでの作成 [12.39](#), [12.41.17](#)
 - システム・エディタでの編集 [12.21](#)
 - @ (アットマーク)コマンド [5.4](#), [12.2](#)
 - コメントの挿入 [5.3](#), [12.35](#)
 - 複数のPL/SQLブロックの挿入 [5.1.1](#)
 - 複数のSQLコマンドの挿入 [5.1.1](#)
 - EDITコマンド [12.21](#)
 - GETコマンド [12.24](#)
 - SAVEコマンド [5.1.1](#), [12.39](#)
 - SQLPLUSコマンド [3.5.3](#), [5.4.1](#)
 - STARTコマンド [5.4](#), [12.45](#)
 - ネスト [5.5](#)
 - パラメータの受渡し [5.10](#), [12.2](#), [12.3](#), [12.45](#)
 - 登録 [12.41](#), [12.41.1](#)
 - 取得 [12.24](#)
 - 実行 [5.4](#), [12.2](#), [12.45](#)
 - 一連の順次実行 [5.5](#)
 - SQL*Plus起動時の実行 [3.5.3](#), [5.4.1](#)
 - バッチ・モードでの実行 [5.6](#), [12.23](#)
 - Uniform Resource Locator [12.2](#), [12.3](#), [12.45](#)
- コマンド履歴
 - SQL*Plus [12.26](#)
- コマンドライン
 - グローバリゼーション・サポートの構成 [11.1](#)
 - ヘルプのインストール [2.2.4](#)
- コマンドライン・インタフェース
 - フォントおよびサイズの変更 [1.2](#)
- コマンド・プロンプト
 - SET SQLPROMPT [8.5.10](#), [12.41](#), [12.41.57](#)

- SQL*Plus [3.3.1](#)
- コマンド
 - タイミング統計の収集 [8.2](#), [12.48](#)
 - スキーマでの使用禁止 [13.1](#)
 - 使用禁止 [9.1](#)
 - 画面にエコー表示 [12.41.16](#)
 - ホスト、SQL*Plusからの実行 [4.9](#), [12.27](#)
 - バッファ内の現在の状態をリスト [12.29](#)
 - 再度使用可能 [9.1](#)
 - 空白 [4.1](#)
 - SQL
 - 次の行への継続 [4.4.1.1](#)
 - バッファ内の編集 [5.2](#)
 - システム・エディタでの編集 [12.21](#)
 - 終了 [4.4.1.2](#)
 - 入力および実行 [4.4](#)
 - 実行せずに入力 [4.4.1.2](#)
 - 現行の実行 [12.4](#), [12.38](#)
 - 構文に従う [4.4.1](#)
 - バッファ内の現在の状態をリスト [5.2.1](#)
 - 現行値の保存 [12.39](#)
 - 終了および実行のために使用する文字設定 [12.41](#)
 - SQL*Plus
 - コマンド概要 [12.1](#)
 - 次の行への継続 [4.6.1.1](#), [12](#)
 - 終了 [4.6.1](#), [12](#)
 - 入力および実行 [4.6](#)
 - SQLコマンドのエントリ中に入力 [12.41.56](#)
 - 廃止されたコマンドとその代替コマンド [C.1](#)
 - 実行中の停止 [4.8](#)
 - タブ [4.1](#)
 - タイプ [4.1](#)
 - 実行に影響する変数 [4.7](#)
- コメント
 - コマンド・ファイルに挿入 [C.1](#)
 - スクリプトに挿入 [5.3](#), [12.35](#), [C.1](#)
 - /*...*/を使用した作成 [5.3.2](#)
 - REMARKの使用 [C.1](#)
 - REMARKを使用した作成 [5.3.1](#), [12.35](#), [C.1](#)
 - --を使用した作成 [5.3.3](#)
- COMMIT句 [12.23](#)
 - WHENEVER OSERROR [12.52](#)
 - WHENEVER SQLERROR [12.53](#)
- COMMITコマンド [4.11](#)
- 層間の通信

- COMPATIBILITY変数 [C.1](#), [C.7](#)
- コンパイル・エラー [4.5.1](#), [12.42](#), [13.1](#)
- COMPUTEコマンド [6.2](#), [12.14](#)
 - 異なる列についてのサマリーの計算 [6.2.8](#)
 - LABEL句 [6.2.6](#), [6.2.7](#), [12.14](#)
 - すべての定義の表示 [6.2.9](#), [12.14](#)
 - 最大LABEL長 [12.14](#)
 - OF句 [6.2.6](#)
 - ON [12.14](#)
 - ON column句 [6.2.6](#), [12.14](#)
 - ON expr句 [12.14](#)
 - ON REPORT句 [6.2.7](#), [12.14](#)
 - 総サマリーおよびサブサマリーの出力 [6.2.7](#)
 - 同じ列についての複数のサマリーの出力 [6.2.8](#)
 - レポートの終わりでのサマリー行の出力 [6.2.7](#)
 - ブレークでのサマリー行の出力 [6.2.6](#)
 - OFでのSELECT式の参照 [12.14](#)
 - ONでのSELECT式の参照 [12.14](#)
 - 定義の取消し [6.2.9](#), [12.12](#)
 - REFCURSOR変数の書式設定に使用 [12.51](#)
- COMPUTES句 [12.12](#)
- CON_ID句 [12.42](#)
- CON_NAME句 [12.42](#)
- CONCAT変数 [5.9.4](#), [12.41](#), [12.41.11](#)
- 構成
 - グローバリゼーション・サポート [11.1](#)
- 構成
 - Oracle Net [2.2.5](#)
 - SQL*Plus [2](#)
- CONNECT /機能 [3.1.1](#)
- CONNECTコマンド [3.1](#), [12.15](#)
 - @ (アットマーク) [12.15](#)
 - パスワードの変更 [12.15](#), [12.30](#)
 - SYSASM句 [12.15](#)
 - SYSBACKUP句 [12.15](#)
 - SYSDBA句 [12.15](#)
 - SYSDG句 [12.15](#)
 - SYSKM句 [12.15](#)
 - SYSOPER句 [12.15](#)
 - SYSRAC句 [12.15](#)
 - ユーザー名/パスワード [12.15](#)
- 接続識別子 [12.15](#)
 - CONNECTコマンド [12.15](#)
 - COPYコマンド [B.1](#)
 - DESCRIBEコマンド [12.19](#)

- SQLPLUSコマンド [3.5.2](#)
- 接続識別子 [3.2.1](#)
 - 省略した簡単な [3.2.3](#)
 - フル [3.2.2](#)
 - ネット・サービス名 [3.2.2](#)
- CONTEXT変数 [12.41](#)
- CONTEXT XQUERYオプション [12.41.74](#)
- CONTINUE句
 - WHENEVER OSERROR [12.52](#)
 - WHENEVER SQLERROR [12.53](#)
- 長いSQL*Plusコマンドの継続 [4.6.1.1](#), [12](#)
- COPYコマンド [12.16](#), [B.1](#), [B.2](#)
 - @ (アットマーク) [B.1](#), [B.2.1](#)
 - ARRAYSIZE変数 [B.1.2](#), [B.2.3](#)
 - COPYCOMMIT変数 [B.1.2](#), [B.2.3](#)
 - LONG変数 [B.1.2](#), [B.2.3](#)
 - APPEND句 [B.1.1](#), [B.2.2](#)
 - データベース間のデータのコピー [B.2](#)
 - 1つのデータベース上にある表の間でのデータのコピー [B.3](#)
 - CREATE句 [B.1.1](#), [B.2.2](#)
 - 表の作成 [B.1.1](#), [B.2.2](#)
 - 宛先の表 [B.1.1](#), [B.2.1](#)
 - アクションの決定 [B.2.1](#)
 - ソースの行および列の決定 [B.1.1](#), [B.2.1](#)
 - エラー・メッセージ [13.2](#)
 - FROM句 [B.2.1](#)
 - INSERT句 [B.1.1](#), [B.2.2](#)
 - 表へのデータの挿入 [B.1.1](#), [B.2.2](#)
 - メッセージの解釈 [B.2.3](#)
 - 必須の接続識別子 [B.1.1](#)
 - SELECTによるソース表の命名 [B.1.1](#), [B.2.1](#)
 - 問合せ [B.1.1](#), [B.2.1](#)
 - 別のユーザーの表への参照 [B.2.4](#)
 - REPLACE句 [B.1.1](#), [B.2.2](#)
 - 表内のデータの置換 [B.1.1](#), [B.2.2](#)
 - サンプル・コマンド [B.2.1](#)
 - サービス名 [B.2.1](#), [B.2.2](#), [B.3](#)
 - 宛先の列の指定 [B.1.1](#), [B.2.1](#)
 - コピーするデータの指定 [B.1.1](#), [B.2.1](#)
 - TO句 [B.2.1](#)
 - ユーザー名/パスワード [B.1.1](#), [B.2.1](#), [B.2.2](#), [B.3](#)
 - USING句 [B.1.1](#), [B.2.1](#)
- COPYCOMMIT変数 [12.41](#), [12.41.12](#)
 - COPYコマンドとの関連 [B.1.2](#), [B.2.3](#)
- COPYTYPECHECK変数 [12.41](#), [12.41.13](#)

- CREATE句
 - COPYコマンド [B.1.1](#), [B.2.2](#)
 - CREATEコマンド
 - PL/SQLの入力 [4.5.1](#)
 - PLAN_TABLEの作成 [8.1.1](#)
 - フラット・ファイルの作成 [6.4.1](#)
 - PLUSTRACEロールの作成 [8.1.1](#)
 - サンプル表の作成
 - CSV [7.1.2](#)
 - 句 [3.5.1.6](#), [12.41.36](#)
 - オプション [3.5.1.6](#)
 - csv, レポートの出力 [7.1](#)
 - カーソル変数 [12.51](#)
-

D

- データベース
 - 管理者 [10.1](#)
 - 接続識別子 [12.15](#)
 - マウント [12.46](#)
 - オープン [12.46](#)
- データベースの変更、自動保存 [12.41](#), [12.41.3](#)
- DATABASE句 [12.34](#)
- データベース・ファイル
 - リカバリ [12.34](#)
- 起動時のデータベース名 [12.46](#)
- データベース
 - デフォルトへの接続 [12.15](#)
 - リモートへの接続 [12.15](#)
 - データのコピー [B.1](#), [B.2](#)
 - 1つのデータベース上にある表の間でのデータのコピー [B.3](#)
 - SQL*Plusを終了せずに切断 [3.1](#), [12.20](#)
 - マウント [10.2.1](#)
 - オープン [10.2.1](#)
 - リカバリ [10.4](#), [12.34](#)
 - シャットダウン [10.2](#) [10.2.3](#)
 - 起動 [10.2](#)
- データベース・スキーマ [8.1.1](#)
 - DESCRIBEパラメータ [12.19](#)
 - SHOW [12.42](#)
- DATAFILE句 [12.34](#)
- DATE
 - DESCRIBEによる列定義 [12.19](#)
- 日付、タイトルの変数に現在の日付を格納 [6.3.5](#), [12.13](#)

- DATE句 [12.5](#)
- DATE列
 - 書式の変更 [12.13](#)
 - デフォルト書式 [6.1.3.1](#)
- DB2 [12.41.13](#)
- DBA [10.1](#)
 - モード [12.46](#)
 - 権限 [12.46](#)
- DBMS_APPLICATION_INFOパッケージ [8.5.2](#), [12.41](#), [12.41.1](#)
- DBMS出力 [8.5.9](#), [12.41.48](#)
- DECLAREコマンド
 - PL/SQL [4.5](#)
- DEFAULT句 [12.5](#)
- DEFINEコマンド [5.7](#), [12.17](#)
 - システム・エディタ [5.1.1](#), [12.17.1](#)
 - UNDEFINEコマンド [5.7](#), [12.50](#)
 - CHAR値 [12.17](#)
 - SET DEFINE ON|OFF [12.41](#), [12.41.14](#)
 - 置換変数 [12.17](#)
- DEFINE変数
 - 「置換変数」を参照
- DEL[ETE]句 [12.26](#)
- DELコマンド [5.2](#), [5.2.5](#), [12.18](#)
 - アスタリスクの使用 [5.2](#), [12.18](#)
- DELIMITER句 [12.41.36](#)
- DESCRIBEコマンド(SQL*Plus) [4.2](#), [12.19](#)
 - connect_identifier [12.19](#)
 - 表示できるPL/SQLプロパティ [12.19](#)
 - 表示できる表プロパティ [12.19](#)
- 無効化
 - PL/SQLコマンド [9.1](#)
 - SQL*Plusコマンド [9.1](#)
 - SQLコマンド [9.1](#)
- DISCONNECTコマンド [3.1](#), [12.20](#)
- DOCUMENTコマンド [C.1](#), [C.4](#)
 - 新しいバージョンのREMARK [C.4](#)
- DOCUMENT変数 [C.1](#), [C.9](#)
- DUPLICATES句 [12.9](#)

E

- ECHO
 - SETコマンド [12.41.16](#)
- ECHO変数 [5.4](#), [12.41](#), [12.41.16](#)

- EDIT句 [12.26](#)
- EDITコマンド [5.1.1](#), [12.17.1](#), [12.21](#)
 - スクリプトの作成 [5.1.1](#)
 - _EDITORの定義 [12.21](#)
 - スクリプトの変更 [12.21](#)
 - デフォルトのファイル名の設定 [12.41](#), [12.41.17](#)
- EDITFILE変数 [12.41](#), [12.41.17](#)
- エディション [3.5.2](#), [12.15](#)
 - CONNECTコマンド [12.15](#)
 - SQLPLUSコマンド [3.5.2](#)
- EDITORオペレーティング・システム変数 [12.17.1](#)
- Ed (UNIX) [12.17.1](#)
- EMBEDDED変数 [12.41](#), [12.41.18](#)
- エンティティ, HTML [7.1.1.1](#)
- ENTMAP [3.5.1.6](#)
- ENTMAP句 [3.5.1.6](#), [7.1.1.1](#), [12.13](#)
- 環境変数
 - LD_LIBRARY_PATH [2.1](#)
 - LOCAL [2.1](#)
 - NLS_LANG [2.1](#)
 - ORA_EDITION [2.1](#)
 - ORA_NLS10 [2.1](#)
 - ORACLE_HOME [2.1](#)
 - ORACLE_PATH [2.1](#)
 - ORACLE_SID [2.1](#)
 - PATH [2.1](#)
 - SQL*Plus [2.1](#)
 - SQLPATH [2.1](#)
 - TNS_ADMIN [2.1](#)
 - TWO_TASK [2.1](#)
- エラー
 - SQL*Plus Instant Clientのサポートされていないcharset [D.1.2.1](#)
- ERRORLOGGING変数 [12.41.19](#)
- エラー・メッセージ
 - COPYコマンド [13.2](#)
 - 解釈 [4.12](#)
 - sqlplus [13.1](#)
- エラー
 - コンパイル・エラー [4.5.1](#), [12.42](#), [13.1](#)
 - 含まれている行をカレント行にする [5.2.2](#)
- エスケープ文字、定義 [12.41](#), [12.41.20](#)
- ESCAPE変数 [5.9.4](#), [12.41](#), [12.41.20](#)
- ESCCHAR変数 [12.41.21](#)
- 例
 - CSVレポート [7.1.2](#)

- 対話型HTMLレポート [7.1.1](#)
 - EXECUTEコマンド [12.22](#)
 - 実行
 - CREATEコマンド [4.5.1](#)
 - 実行計画 [8.1.2](#)
 - 実行統計
 - レポートへの挿入 [12.41.6](#)
 - 終了, 条件付き [12.52](#)
 - EXIT句
 - WHENEVER OSERROR [12.52](#)
 - WHENEVER SQLERROR [12.53](#)
 - EXITコマンド [3.4](#), [12.23](#)
 - ((colon))BindVariable句 [12.23](#)
 - COMMIT句 [12.23](#)
 - FAILURE句 [12.23](#)
 - スクリプトで使用 [12.45](#)
 - ROLLBACK句 [12.23](#)
 - SET MARKUPでの使用 [7.1.1](#)
 - WARNING句 [12.23](#)
 - EXITCOMMIT変数 [12.41.22](#)
 - 拡張子 [12.39](#), [12.41.60](#), [12.47](#)
-

F

- FAILURE句 [12.23](#)
- FEEDBACK変数 [12.41](#), [12.41.23](#)
- ファイル拡張子 [2.2.3](#), [12.39](#), [12.41.60](#), [12.47](#)
- ファイル名
 - @ (アットマーク)コマンド [12.2](#)
 - @@ (二重アットマーク)コマンド [12.3](#)
 - EDITコマンド [12.21](#)
 - GETコマンド [12.24](#)
 - SAVEコマンド [12.39](#)
 - SPOOLコマンド [6.4.2](#), [12.44](#)
 - SQLPLUSコマンド [3.5.3](#)
- ファイル
 - フラット [6.4.1](#)
 - SQL*Plus Instant Clientに必須 [D.2.3](#)
- FLAGGER変数 [12.41](#), [12.41.24](#)
- フラット・ファイル [6.4.1](#)
- FLUSH変数 [8.5.5](#), [12.41](#), [12.41.25](#)
- FOLD_AFTER句 [12.13](#)
- FOLD_BEFORE句 [12.13](#)
- フォント

- コマンドラインのフォントおよびフォント・サイズの変更 [1.2](#)
 - フッター
 - 要素の整列 [12.37](#)
 - ページの下部に表示 [12.36](#)
 - システム管理値の表示 [12.37](#)
 - 要素の書式設定 [12.37](#)
 - インデント [12.37](#)
 - 現行の定義の表示 [12.36](#)
 - レポートの終わりに設定 [6.3](#)
 - 定義の非表示 [12.37](#)
 - FORCE句 [12.46](#)
 - FORMAT句 [12.5](#), [12.13](#)
 - ATTRIBUTEコマンド [12.8](#)
 - COLUMNコマンド [6.1.2.2](#), [6.1.3.2](#)
 - REPHEADERおよびREPFOOTERコマンド [12.37](#)
 - TTITLEおよびBTITLEコマンド [6.3.2](#), [12.49](#)
 - 書式モデル, 数値 [6.1.2.2](#), [12.13](#)
 - 改ページ、新しいページの開始 [6.3.6](#), [12.41.37](#)
 - FROM句 [12.34](#), [B.2.1](#)
-

G

- GETコマンド [12.24](#)
 - LIST句 [12.24](#)
 - NOLIST句 [12.24](#)
 - スクリプトの取得 [12.24](#)
 - グローバリゼーション・サポート
 - Oracle10g [11.3](#)
 - glogin [2.2.1](#)
 - 「login.sql」も参照
 - プロファイル [12.41.55.1](#)
 - サイト・プロファイル [2.2](#), [2.2.1](#), [2.2.2.1](#), [3.5.1.9](#), [12.41.55.1](#)
-

H

- HEAD句 [3.5.1.6](#)
- ヘッダー
 - 要素の整列 [6.3.1.1](#)
 - ページの上部に表示 [12.37](#)
 - システム管理値の表示 [12.37](#)
 - レポートの始めに設定 [6.3](#)
 - 非表示 [6.3.1](#)
- HEADING句 [6.1.1.2](#), [12.13](#)
- ヘッダー

- 要素の整列 [12.37](#)
- 列ヘッダー [12.41.26](#)
- 要素の書式設定 [12.37](#)
- インデント [12.37](#)
- 現行の定義の表示 [12.37](#)
- 定義の非表示 [12.37](#)
- HEADING変数 [12.41.26](#)
- HEADオプション [3.5.1.6](#)
- HEADSEP変数 [12.41](#), [12.41.27](#)
 - COLUMNコマンドでの使用 [6.1.1.2](#)
- ヘルプ
 - コマンドラインのインストール [2.2.4](#)
 - オンライン [3.3.2](#), [12.25](#)
- HELPコマンド, ?コマンド [12.25](#)
- HIDE句 [12.5](#)
- HISTORY句 [12.42](#)
- HISTORYコマンド [12.26](#)
 - CLEAR句 [12.26](#)
 - DEL[ETE]句 [12.26](#)
 - EDIT句 [12.26](#)
 - LIST句 [12.26](#)
 - RUN句 [12.26](#)
- HISTORY変数 [12.41](#)
- HOSTコマンド [4.9](#), [12.27](#)
- HTML [7.1.1](#)
 - 句 [3.5.1.6](#)
 - エンティティ [7.1.1.1](#)
 - オプション [3.5.1.6](#)
 - ファイルへのスプール [3.5.1.6](#)
 - タグ [7.1](#)
- ハイフン
 - 長いSQL*Plusコマンドの継続 [4.6.1.1](#), [12](#)

I

- IMMEDIATEモード [12.43](#)
- 無限大記号(~) [12.13](#)
- INIT.ORAファイル
 - パラメータ・ファイル [12.46](#)
- 初期化パラメータ
 - 表示 [12.42](#)
- 入力
 - [Return]のアクセプト [5.11.3](#)
 - ユーザーからの値のアクセプト [5.11.1](#), [12.5](#)

- INPUTコマンド [5.2](#), [5.2.4](#), [12.28](#)
 - 複数行の入力 [12.28](#)
- INSERT句 [B.1.1](#), [B.2.2](#)
- インストール
 - SQL*Plus Instant Client [D](#)
 - コピーによるSQL*Plus Instant Client [D.1](#)
 - 10g ClientからのコピーによるSQL*Plus Instant Client [D.3](#)
 - OTNからのダウンロードによるSQL*Plus Instant Client [D.1](#), [D.2](#)
 - LinuxでのSQL*Plus Instant Client [D.2.1](#)
 - UNIXまたはWindowsでのSQL*Plus Instant Client [D.2.2](#)
 - コピーするSQL*Plus Instant Client UNIXおよびLinuxファイル [D.3.1](#)
 - コピーするSQL*Plus Instant Client Windowsファイル [D.3.2](#)
- インストール, コピーするUNIXおよびLinuxファイル [D.3.1](#)
- インストール, コピーするWindowsファイル [D.3.2](#)
- コピーによるインストール [D.1](#)
- 10gクライアントからのコピーによるインストール [D.3](#)
- OTNからのダウンロードによるインストール [D.1](#), [D.2](#)
- Linuxでのインストール [D.2.1](#)
- UNIXまたはWindowsでのインストール [D.2.2](#)
- インスタンス
 - 停止 [12.43](#)
 - 起動 [12.46](#)
- INSTANCE変数 [12.41](#), [12.41.29](#)
- Instant Client
 - SQL*Plus [D](#)
- Instant Clientパッケージ [D.1](#)

J

- 日本語 [11.1](#)
- JUSTIFY句 [12.13](#)

L

- ラベル
 - COMPUTEコマンド [6.2.6](#), [12.14](#)
- LABEL変数
 - SHOWコマンド [C.1](#)
- 言語
 - SQL*Plus Instant Client [D.1.2](#)
- LD_LIBRARY_PATH
 - 環境変数 [2.1](#)
- LEFT句 [6.3.1.1](#), [12.37](#), [12.49](#)
- 軽量OCIパッケージ [D.1.2](#)

- LIKE句 [6.1.4](#), [12.8](#), [12.13](#)
- 制限, SQL*Plus [A](#)
- 行
 - バッファの最初に追加 [12.28](#)
 - バッファの最後に追加 [12.28](#)
 - カレント行の後への新規追加 [5.2.4](#), [12.28](#)
 - テキストの追加 [5.2.3](#), [12.6](#)
 - 幅の変更 [6.3.6](#), [8.5.6](#), [12.41](#), [12.41.30](#)
 - バッファ内のすべての削除 [12.18](#)
 - バッファからの削除 [5.2.5](#), [12.18](#)
 - カレント行の判断 [5.2.2](#)
 - カレント行の編集 [5.2.2](#)
 - バッファ内のすべての表示 [5.2](#), [12.29](#)
 - 終わりの空白の削除 [12.41.65](#)
- LINESIZE変数 [6.3.1.1](#), [6.3.6](#), [12.41](#), [12.41.30](#)
- Linux
 - SQL*Plus Instant Clientのインストール [D.2.1](#)
 - コピーするSQL*Plus Instant Clientファイル [D.3.1](#)
- LIST句 [12.7](#), [12.24](#), [12.26](#)
- LISTコマンド [5.2](#), [12.29](#)
 - カレント行の判断 [5.2.2](#), [12.29](#)
 - 最終行をカレント行にする [5.2.2](#), [12.29](#)
 - アスタリスクの使用 [5.2](#), [12.29](#)
- LNO句 [12.42](#)
- LOBデータ
 - プリフェッチ・サイズの設定 [12.41](#)
- LOBOFFSET変数 [12.41](#), [12.41.31](#)
- LOBPREFETCH句 [12.42](#)
- LOBPREFETCH変数 [12.41](#)
- LOCAL
 - 環境変数 [2.1](#)
- LOG_ARCHIVE_DESTパラメータ [12.7](#)
- LOGFILE句 [12.34](#)
- ログオフ
 - 条件付き [12.52](#), [12.53](#)
 - Oracle Database [3.1](#), [12.20](#)
 - SQL*Plus [3.4](#), [12.23](#)
- ログオン
 - Oracle Database [12.15](#)
 - SQL*Plus [3.3.1](#)
- ログイン
 - ユーザー・プロファイル [2.2.2.1](#)
- login.sql [2.2.2](#)
- LONG
 - DESCRIBEによる列定義 [12.19](#)

- LONGCHUNKSIZE変数 [6.1.3.1](#), [12.13](#), [12.41](#), [12.41.35](#), [12.41.36](#)
 - LONG列
 - 書式の変更 [12.13](#)
 - デフォルト書式 [12.13](#)
 - 最大幅の設定 [12.41](#), [12.41.34](#)
 - 取得サイズの設定 [8.5.7](#), [12.41](#), [12.41.35](#)
 - LONGRAW
 - DESCRIBEによる列定義 [12.19](#)
 - LONG変数 [12.41](#), [12.41.34](#)
 - COPYコマンドへの影響 [B.1.2](#), [B.2.3](#)
-

M

- MARKUP [3.5.1.6](#), [7.1](#), [12.41.36](#)
 - BODY句 [3.5.1.6](#)
 - DELIMITER句 [12.41.36](#)
 - ENTMAP句 [3.5.1.6](#)
 - HEAD句 [3.5.1.6](#)
 - PREFORMAT句 [3.5.1.6](#)
 - QUOTE句 [12.41.36](#)
 - TABLE句 [3.5.1.6](#)
 - MAXDATA変数 [C.1](#), [C.10](#)
 - メディア・リカバリ [12.46](#)
 - メッセージ、画面への送信 [5.11.1](#), [12.33](#)
 - MOUNT句 [12.46](#)
 - データベースのマウント [12.46](#)
-

N

- 各国語サポート [11.1](#)
 - 「グローバル化・サポート」も参照。
- NCHAR句
 - VARIABLEコマンド [12.51](#)
- NCHAR列
 - 書式の変更 [12.13](#)
 - デフォルトの書式 [6.1.3.1](#), [12.13](#)
- NCLOB句
 - VARIABLEコマンド [12.51](#)
- NCLOB列
 - 書式の変更 [12.13](#)
 - デフォルト書式 [12.13](#)
 - 最大幅の設定 [12.41](#), [12.41.34](#)
 - 取得位置の設定 [12.41](#), [12.41.31](#)
 - 取得サイズの設定 [8.5.7](#), [12.41](#), [12.41.35](#)

- 負の無限大記号(-∞) [12.13](#)
- ネット・サービス名 [3.2.1](#), [3.2.2](#)
- NEW_VALUE句 [6.3.4](#), [12.13](#)
 - タイトルに対する変数内の現在の日付の格納 [12.13](#)
- NEWLINE句 [12.13](#)
- NEWPAGEコマンド [C.1](#), [C.5](#)
- NEWPAGE変数 [6.3.6](#), [12.41](#), [12.41.37](#)
- NLS [11.1](#)
- NLS_DATE_FORMAT [12.5](#), [12.13](#)
- NLS_LANG
 - Instant Clientのcharsetパラメータ [D.1.2](#)
 - 環境変数 [2.1](#)
 - Instant Clientの言語パラメータ [D.1.2](#)
 - SQL*Plus Instant Client [D.1.2](#)
 - Instant Clientの地域パラメータ [D.1.2](#)
- NODE変数 [12.41](#)
- NODE XQUERYオプション [12.41.73](#)
- NOLIST句 [12.24](#)
- NOLOG [3.2.4](#), [3.5.2](#)
- nolongontime [3.5.1.8](#)
- NOMOUNT句 [12.46](#)
- NONE句
 - WHENEVER OSERROR [12.52](#)
 - WHENEVER SQLERROR [12.53](#)
- NOPARALLEL句 [12.34](#)
- NOPRINT句 [6.2.6](#), [6.3.4](#), [8.5.1](#), [12.13](#)
- NOPROMPT句 [12.5](#)
- NORMALモード [12.43](#)
- メモ帳(Windows) [12.17.1](#)
- NULL句 [12.13](#)
- NULL値
 - 表示するテキストの設定 [12.13](#), [12.41](#), [12.41.38](#)
- NULL変数 [12.41](#), [12.41.38](#)
- NUMBER
 - DESCRIBEによる列定義 [12.19](#)
- NUMBER句 [5.11.2](#)
 - VARIABLEコマンド [12.51](#)
- NUMBER列
 - 書式の変更 [6.1.2.1](#), [12.13](#)
 - デフォルトの書式 [6.1.2.1](#), [12.13](#)
- 数値書式
 - \$ [6.1.2.2](#)
 - 0 [6.1.2.2](#)
 - 9 [6.1.2.2](#)
 - カンマ [6.1.2.2](#)

- デフォルト設定 [5.9.4](#), [12.41](#), [12.41.39](#)
 - NUMFORMAT句
 - LOGIN.SQL [2.2.2.1](#)
 - NUMFORMAT変数 [12.41](#), [12.41.39](#)
 - NUMWIDTH変数 [12.41](#), [12.41.40](#)
 - NUMBER列の書式への影響 [6.1.2.1](#), [12.13](#)
 - NVARCHAR2列
 - 書式の変更 [12.13](#)
 - デフォルトの書式 [6.1.3.1](#), [12.13](#)
-

O

- オブジェクト, 定義の表示 [12.41.15](#)
- 廃止されたコマンド
 - BTITLE [C.2](#)
 - COLUMNコマンドDEFAULT句 [C.3](#)
 - DOCUMENT [C.1](#), [C.4](#)
 - NEWPAGE [C.1](#), [C.5](#)
 - SETコマンド BUFFER変数 [C.6](#)
 - SETコマンドCLOSECURSOR変数 [C.1](#), [C.8](#)
 - SETコマンドCOMPATIBILITY変数 [C.1](#), [C.7](#)
 - SETコマンドDOCUMENT変数 [C.1](#), [C.9](#)
 - SETコマンドMAXDATA変数 [C.1](#), [C.10](#)
 - SETコマンドSCAN変数 [C.1](#), [C.11](#)
 - SETコマンドSPACE変数 [C.1](#), [C.12](#)
 - SETコマンドTRUNCATE変数 [C.1](#), [C.13](#)
 - SHOWコマンドLABEL変数 [C.1](#)
 - TTITLEコマンドの旧形式 [C.14](#)
- OCI Instant Client [D](#)
- OCIパッケージ
 - 基本 [D.1.1](#)
 - 軽量 [D.1.2](#)
- OF句 [6.2.6](#)
- OFF句 [12.13](#)
 - ATTRIBUTEコマンド [12.8](#)
 - COLUMNコマンド [6.1.6](#), [12.13](#)
 - REPFOOTERコマンド [12.37](#)
 - REPHEADERコマンド [12.37](#)
 - SPOOLコマンド [6.4](#), [12.44](#)
 - TTITLEおよびBTITLEコマンド [6.3.3](#), [12.49](#)
- OLD_VALUE句 [6.3.4](#), [12.13](#)
- ON句
 - ATTRIBUTEコマンド [12.8](#)
 - COLUMNコマンド [6.1.6](#), [12.13](#)

- TTITLEおよびBTITLEコマンド [6.3.3](#)
- ON column句
 - BREAKコマンド [12.9](#)
 - COMPUTEコマンド [6.2.6](#), [12.14](#)
- ON expr句
 - BREAKコマンド [12.9](#)
 - COMPUTEコマンド [12.14](#)
- オンライン・ヘルプ [3.3.2](#), [12.25](#)
- ON REPORT句
 - BREAKコマンド [6.2.7](#), [12.9](#)
 - COMPUTEコマンド [6.2.7](#), [12.14](#)
- ON ROW句
 - BREAKコマンド [6.2.3](#), [12.9](#)
 - COMPUTEコマンド [12.14](#)
- OPEN句 [12.46](#)
- データベースのオープン [12.46](#)
- オペレーティング・システム
 - エディタ [5.1.1](#), [12.17.1](#), [12.21](#)
 - ファイル、バッファへのロード [12.24](#)
 - SQL*Plusからのコマンド実行 [4.9](#), [12.27](#)
 - テキスト・エディタ [5.1.1](#)
- ORA_EDITION
 - 環境変数 [2.1](#)
- ORA_NLS10
 - 環境変数 [2.1](#)
- ORACLE_HOME
 - 環境変数 [2.1](#)
- ORACLE_PATH
 - 環境変数 [2.1](#)
- ORACLE_SID
 - 環境変数 [2.1](#)
- Oracle10g
 - グローバリゼーション・サポート [11.3](#)
- Oracleアプリケーション・エディション
 - エディション [3.5.2](#)
- Oracle Database Client [D](#)
- Oracle Net
 - 構成 [2.2.5](#)
 - 接続識別子 [12.15](#)
- Oracleセッション・エディション
 - エディション [12.15](#)
- ORDER BY句
 - タイトルへの列値の表示 [6.3.4](#)
 - 出力で値を同時に表示 [6.2](#)
- ORDERING変数 [12.41](#)

- ORDERING XQUERYオプション [12.41.72](#)
- OUT句 [6.4.3](#), [12.44](#)
- 出力
 - 空白の書式設定 [8.5.11](#), [12.41.61](#)
 - 表示中の一時停止 [4.10](#), [12.41.42](#)

P

- パッケージ
 - SQL*PlusおよびOCIのInstant Client [D.1](#)
- PAGE句 [12.37](#)
- ページ番号、タイトルへの挿入 [6.2.4](#), [6.3.2](#)
- ページ
 - 長さの変更 [6.3.6](#), [8.5.8](#), [12.41](#), [12.41.41](#)
 - デフォルトのサイズ [6.3.6](#)
 - サイズを画面または用紙のサイズにあわせる [6.3.6](#)
 - サイズの設定 [6.3.6](#)
- PAGESIZE句
 - LOGIN.SQL [2.2.2.1](#)
- PAGESIZE変数 [4.4](#), [6.3.6](#), [8.5.8](#), [12.41](#), [12.41.41](#)
- パラメータ [5.10](#), [12.2](#), [12.3](#), [12.45](#)
 - SQLPATH [2.1.1](#)
- パラメータ・ファイル(INIT.ORAファイル)
 - 代替の指定 [12.46](#)
- PARAMETERS句 [12.42](#)
- パスワード
 - PASSWORDコマンドで変更 [12.30](#)
 - CONNECTコマンド [3.1](#), [12.15](#)
 - COPYコマンド [B.2.1](#), [B.2.2](#), [B.3](#)
 - SQLPLUSコマンド [3.3.1](#), [3.5.2](#)
 - 参照可能な警告 [3.5.2](#)
- PASSWORDコマンド [12.15](#), [12.30](#)
- PATH
 - 環境変数 [2.1](#)
- PAUSEコマンド [5.11.3](#), [12.31](#)
- PAUSE変数 [4.10](#), [12.41](#), [12.41.42](#)
- PDBS句 [12.42](#)
- パフォーマンス
 - SQL文 [8.1](#)
 - ダイアルアップ回線を超える [12.41.65](#)
- ピリオド(.)
 - PL/SQLブロックの終了 [4.5](#), [12.41](#), [12.41.7](#)
- PL/SQL [4.5](#)
 - ブロック, PL/SQL [4.5](#)

- 実行 [12.22](#)
 - SQL*Plusでの出力の書式設定 [12.51](#)
 - 定義の表示 [4.3](#)
 - SQL*Plusでのモード [4.5.1](#)
 - SQLコマンド [4.5.1](#)
- PLAN_TABLE
 - 作成 [8.1.1](#)
 - 表 [8.1.1](#)
- PLUGGABLE DATABASE句 [12.46](#)
- PLUSTRACE
 - ロールの作成 [8.1.1](#)
 - ロール [8.1.1](#)
- PNO句 [12.42](#)
- シャープ記号(#) [12.13](#)
- 事前定義変数
 - _CONNECT_IDENTIFIER [2.2.2.1](#), [12.17.1](#)
 - _DATE [12.17.1](#)
 - _EDITOR [5.1.1](#), [12.17.1](#), [12.21](#)
 - _O_RELEASE [12.17.1](#)
 - _O_VERSION [12.17.1](#)
 - _PRIVILEGE [12.17.1](#)
 - _RC [12.27](#)
 - _SQLPLUS_RELEASE [12.17.1](#)
 - _USER [12.17.1](#)
- PREFORMAT [3.5.1.6](#)
- PREFORMAT句 [3.5.1.6](#)
- PRINT句 [12.13](#)
- PRINTコマンド [12.32](#)
- 出力
 - バインド変数の自動出力 [12.41.4](#)
 - REFCURSOR変数 [12.51](#)
 - SPOOLコマンド [12.44](#)
- プロンプト
 - SET SQLPROMPT [8.5.10](#), [12.41](#), [12.41.57](#)
- PROMPT句 [5.11.1](#), [12.5](#)
- PROMPTコマンド [5.11.1](#), [12.33](#)
 - 値の入力を求めるプロンプトのカスタマイズ [5.11.2](#)
- 値の入力を求めるプロンプト
 - パラメータを使用した回避 [5.10](#)
 - カスタマイズ [5.11.2](#)
 - ACCEPTの使用 [5.11.1](#)
 - 置換変数の使用 [5.9.1](#)

- 問合せ
 - COPYコマンド [B.1.1](#), [B.2.1](#)
 - 取得したレコード数の表示 [4.4](#), [12.41](#), [12.41.23](#)
 - トレース [8.3](#), [8.4](#)
 - 問合せ実行パス
 - レポートへの挿入 [12.41.6](#)
 - 問合せ結果
 - 画面に表示 [4.4](#)
 - プリンタへの出力 [6.4.3](#), [12.44](#)
 - ファイルへの格納 [6.4.2](#), [12.44](#)
 - QUITコマンド [12.23](#), [12.45](#)
 - 「EXIT」も参照
 - QUOTE句 [12.41.36](#)
-

R

- RAW
 - DESCRIBEによる列定義 [12.19](#)
- レコード・セパレータ, 出力 [6.1.7](#), [12.41](#), [12.41.44](#)
- RECOVER句 [12.46](#)
- RECOVERコマンド [12.34](#)
 - データベースのリカバリ [10.4](#)
 - AUTOMATIC句 [12.34](#)
 - CANCEL句 [12.34](#)
 - CONTINUE句 [12.34](#)
 - DATABASE句 [12.34](#)
 - FROM句 [12.34](#)
 - LOGFILE句 [12.34](#)
 - NOPARALLEL句 [12.34](#)
 - SNAPSHOT TIME date句 [12.34](#)
 - STANDBY DATABASE句 [12.34](#)
 - STANDBY DATAFILE句 [12.34](#)
 - STANDBY TABLESPACE句 [12.34](#)
 - UNTIL CANCEL句 [12.34](#)
 - UNTIL CONTROLFILE句 [12.34](#)
 - UNTIL TIME句 [12.34](#)
 - USING BACKUP CONTROL FILE句 [12.34](#)
- リカバリ
 - RECOVERコマンド [12.34](#)
- RECSEPCHAR変数 [6.1.7](#), [12.41](#), [12.41.44](#)
- RECSEP変数 [6.1.7](#), [12.41](#), [12.41.44](#)
- REFCURSORバインド変数
 - ストアド・ファンクション [5.13](#)
- REFCURSOR句

- VARIABLEコマンド [12.51](#)
- レジストリ
 - エディタ [2.1.1](#)
- レジストリ・エントリ
 - SQLPATH [2.1](#), [2.1.1](#)
- RELEASE句 [12.42](#)
- REMARKコマンド [5.3.1](#), [12.35](#)
- サンプル表の削除
- REPFOOTER句 [12.42](#)
- REPFOOTERコマンド [6.3.1](#), [12.36](#)
 - フッター要素の整列 [12.37](#)
 - BOLD句 [12.37](#)
 - CENTER句 [12.37](#)
 - COL句 [12.37](#)
 - FORMAT句 [12.37](#)
 - レポート・フッターのインデント [12.37](#)
 - LEFT句 [12.37](#)
 - OFF句 [12.37](#)
 - RIGHT句 [12.37](#)
 - SKIP句 [12.37](#)
 - 現行の定義の非表示 [12.37](#)
 - TAB句 [12.37](#)
- REPHEADER句 [12.42](#)
- REPHEADERコマンド [6.3.1](#), [12.37](#)
 - ヘッダー要素の整列 [6.3.1.1](#)
 - ヘッダー要素の整列 [12.37](#)
 - BOLD句 [12.37](#)
 - CENTER句 [12.37](#)
 - COL句 [12.37](#)
 - FORMAT句 [12.37](#)
 - ヘッダーのインデント [12.37](#)
 - LEFT句 [12.37](#)
 - OFF句 [12.37](#)
 - PAGE句 [12.37](#)
 - RIGHT句 [12.37](#)
 - SKIP句 [12.37](#)
 - 現行の定義の非表示 [12.37](#)
 - TAB句 [12.37](#)
- REPLACE句
 - COPYコマンド [B.1.1](#), [B.2.2](#)
 - SAVEコマンド [12.39](#), [12.44](#)
- レポート
 - 自動トレース [8.1.1](#)
 - ブレーク [12.9](#)
 - 間隔の設定およびサマリー行による明確化 [6.2](#)

- 列 [12.13](#)
- 下部タイトルの作成 [6.3](#), [12.10](#), [C.1](#)
- フッターの作成 [12.36](#)
- ヘッダーの作成 [12.37](#)
- ヘッダーおよびフッターの作成 [6.3](#)
- マスター/ディテールの作成 [6.3.4](#), [12.13](#)
- 上部タイトルの作成 [6.3](#), [12.49](#), [C.1](#)
- csv [7.1](#)
- CSV例 [7.1.2](#)
- 表示 [12.41](#), [12.41.6](#)
- 列ヘッダーの書式設定 [6.1.1.2](#), [12.13](#)
- 列の書式設定 [6.1.2.1](#), [6.1.3.1](#), [12.13](#)
- 対話型HTMLの例 [7.1.1](#)
- Web上 [7.1](#)
- SILENTモード [7.1.3](#)
- 新しいページでの開始 [12.41.18](#)
- タイトル [12.49](#), [C.1](#)
- RESTRICT [3.5.1.9](#), [9.3](#), [12.46](#)
- リターン・コード, 指定 [5.6](#), [12.23](#), [12.53](#)
- RIGHT句 [6.3.1.1](#), [12.37](#), [12.49](#)
- ロール [9.2](#)
 - 使用禁止 [9.2.2](#)
 - 再度使用可能 [9.2.2](#)
- ROLLBACK句 [12.23](#)
 - WHENEVER OSERROR [12.52](#)
 - WHENEVER SQLERROR [12.53](#)
- 行データ
 - プリフェッチ・サイズの設定 [12.41](#)
- ROWID
 - DESCRIBEによる列定義 [12.19](#)
- ROWPREFETCH句 [12.42](#)
- ROWPREFETCH変数 [12.41](#)
- 行
 - 計算の実行 [6.2.6](#), [12.14](#)
 - 一度に取得する数の設定 [8.5.3](#), [12.41](#), [12.41.2](#)
 - COPYのコミットまでの数を設定 [12.41.12](#)
- RUN句 [12.26](#)
- RUNコマンド [12.38](#)
 - 現行のPL/SQLブロックの実行 [4.5](#)
 - 最終行をカレント行にする [5.2.2](#)
 - / (スラッシュ)コマンドとの類似 [12.38](#)

- サンプル・スキーマ , ,
 - 『Oracle Databaseサンプル・スキーマ』を参照
 - COLUMN例でHRを使用 [12.13](#)
 - 例でのHRの使用 [5](#), [6](#)
- サンプル表
 - アクセス
 - 作成
 - 削除
 - ロック解除
- SAVEコマンド [12.39](#)
 - APPEND句 [12.39](#)
 - CREATE句 [12.39](#)
 - REPLACE句 [12.39](#)
 - スクリプトへのコマンドの格納 [12.39](#)
 - INPUTと併用したスクリプトの作成 [5.1.1](#)
- 環境属性の保存 [12.47](#)
- SCAN変数 [C.1](#), [C.11](#)
- スキーマ
 - コマンド [13.1](#)
 - データベース [8.1.1](#)
 - DESCRIBEパラメータ [12.19](#)
 - 使用禁止コマンド [13.1](#)
 - HRサンプル
 - HRの独自のコピーをインストール
 - サンプル
 - SHOWパラメータ [12.42](#)
 - HRのロック解除
 - COLUMN例でHRを使用 [12.13](#)
 - 例でのHRの使用 [5](#), [6](#)
- SCREEN句 [5.11.4](#), [12.12](#)
- 画面
 - クリア [5.11.4](#), [12.12](#)
- スクリプト
 - 拡張子 [12.39](#), [12.41.60](#), [12.47](#)
 - 登録 [8.5.2](#)
- スクリプト、ユーザーの認証 [3.1.1](#)
- SECUREDCOL変数 [12.41.47](#)
- セキュリティ
 - パスワードの変更 [12.30](#)
 - nolongontime [3.5.1.8](#)
 - 参照可能なパスワード [3.5.2](#)
 - RESTRICT [3.5.1.9](#), [9.3](#)
- SELECTコマンド
 - BREAKコマンド [6.2](#), [12.9](#)
 - COLUMNコマンド [12.13](#)

- COMPUTEコマンド [6.2](#)
- COPYコマンド [B.1.1](#), [B.2.1](#)
- DEFINEコマンド [12.17](#)
- ORDER BY句 [6.2](#)
- 結果の書式設定 [5.13](#), [5.14](#)
- セミicolon (;)
 - PL/SQLブロック [4.5](#)
 - SQL*Plusコマンド [4.6.1](#), [12](#)
 - SQLコマンド [4.4](#), [4.4.1.2](#)
 - バッファに格納されない [5.2.1](#)
- SERVEROUTPUT変数 [12.41.48](#)
- サービス名
 - COPYコマンド [B.2.1](#), [B.2.2](#), [B.3](#)
- セッション・エディション [12.15](#)
- SET AUTOTRACE [8.1.1](#)
- SET句 [12.47](#)
- SETコマンド [2.2.3](#), [4.7](#), [12.40](#), [12.41](#)
 - APPINFO変数 [8.5.2](#), [12.41.1](#)
 - ARRAYSZ変数 [8.5.3](#), [12.41](#), [12.41.2](#), [B.2.3](#)
 - AUTOCOMMIT変数 [12.41](#), [12.41.3](#)
 - AUTOPRINT変数 [12.41](#), [12.41.4](#), [12.51](#)
 - AUTORECOVERY変数 [12.41](#), [12.41.5](#)
 - AUTOTRACE変数 [12.41.6](#)
 - BLOCKTERMINATOR変数 [12.41](#), [12.41.7](#)
 - BUFFER変数 [C.6](#)
 - CLOSECURSOR変数 [C.1](#), [C.8](#)
 - CMDSEP変数 [12.41](#), [12.41.8](#)
 - COLINVISIBLE変数 [12.41.9](#)
 - COLSEP変数 [6.4.1](#), [12.41](#), [12.41.10](#)
 - COMPATIBILITY変数 [C.1](#), [C.7](#)
 - CONCAT変数 [5.9.4](#), [12.41](#), [12.41.11](#)
 - COPYCOMMIT変数 [12.41](#), [12.41.12](#), [B.2.3](#)
 - COPYTYPECHECK変数 [12.41](#), [12.41.13](#)
 - DEFINE句 [5.9.4](#)
 - DEFINE変数 [12.41](#)
 - DESCRIBE変数 [12.41](#), [12.41.15](#)
 - DOCUMENT変数 [C.1](#), [C.9](#)
 - ECHO変数 [12.41](#), [12.41.16](#)
 - EDITFILE変数 [12.41](#), [12.41.17](#)
 - EMBEDDED変数 [12.41](#), [12.41.18](#)
 - ERRORLOGGING変数 [12.41.19](#)
 - ESCAPE変数 [5.9.4](#), [12.41](#), [12.41.20](#)
 - ESCCHAR変数 [12.41.21](#)
 - EXITCOMMIT変数 [12.41.22](#)
 - FEEDBACK変数 [12.41](#), [12.41.23](#)

- FLAGGER変数 [12.41](#), [12.41.24](#)
- FLUSH変数 [8.5.5](#), [12.41](#), [12.41.25](#)
- HEADING変数 [12.41.26](#)
- HEADSEP変数 [6.1.1.2](#), [12.41](#), [12.41.27](#)
- HISTORY変数 [12.41](#)
- INSTANCE変数 [12.41](#), [12.41.29](#)
- LINESIZE変数 [6.3.1.1](#), [6.3.6](#), [12.41](#), [12.41.30](#)
- LOBOFFSET変数 [12.41](#), [12.41.31](#)
- LOBPREFETCH変数 [12.41](#)
- LOGSOURCE変数 [12.41](#), [12.41.33](#)
- LONGCHUNKSIZE変数 [12.41](#), [12.41.35](#)
- LONG変数 [12.41](#), [12.41.34](#), [B.2.3](#)
- MARKUP句 [12.41.36](#)
- MAXDATA変数 [C.1](#), [C.10](#)
- NEWPAGE変数 [6.3.6](#), [12.41](#), [12.41.37](#)
- NULL変数 [12.41](#), [12.41.38](#)
- NUMFORMAT句 [2.2.2.1](#)
- NUMFORMAT変数 [12.41](#), [12.41.39](#)
- NUMWIDTH変数 [6.1.2.1](#), [12.13](#), [12.41](#), [12.41.40](#)
- PAGESIZE句 [2.2.2.1](#)
- PAGESIZE変数 [4.4](#), [6.3.6](#), [8.5.8](#), [12.41](#), [12.41.41](#)
- PAUSE変数 [12.41](#), [12.41.42](#)
- RECSEPCHAR変数 [6.1.7](#), [12.41](#), [12.41.44](#)
- RECSEP変数 [6.1.7](#), [12.41](#), [12.41.44](#)
- ROWPREFETCH変数 [12.41](#)
- SCAN変数 [C.1](#), [C.11](#)
- SECUREDCOL変数 [12.41.47](#)
- SERVEROUTPUT変数 [12.41.48](#)
- SHIFTINOUT変数 [12.41](#), [12.41.49](#)
- SPACE変数 [C.1](#), [C.12](#)
- SQLBLANKLINES変数 [12.41.51](#)
- SQLCASE変数 [12.41](#), [12.41.52](#)
- SQLCONTINUE変数 [12.41](#), [12.41.53](#)
- SQLNUMBER変数 [12.41](#), [12.41.54](#)
- SQLPLUSCOMPATIBILITY変数 [12.41](#), [12.41.55](#)
- SQLPREFIX変数 [12.41](#), [12.41.56](#)
- SQLPROMPT変数 [8.5.10](#), [12.41](#), [12.41.57](#)
- SQLTERMINATOR変数 [12.41](#), [12.41.58](#)
- STATEMENTCACHE変数 [12.41](#)
- 置換変数 [12.41.14](#)
- SUFFIX変数 [12.41](#), [12.41.60](#)
- TAB変数 [8.5.11](#), [12.41](#), [12.41.61](#)
- TERMOUT変数 [8.5.12](#), [12.41](#), [12.41.62](#)
- TIME変数 [12.41](#), [12.41.63](#)
- TIMING変数 [12.41](#), [12.41.64](#)

- TRIMOUT変数 [12.41](#), [12.41.65](#)
- TRIMSPool変数 [12.41](#), [12.41.66](#)
- TRUNCATE変数 [C.1](#), [C.13](#)
- UNDERLINE変数 [12.41](#), [12.41.67](#)
- REFCURSOR変数の書式設定に使用 [12.51](#)
- VERIFY句 [5.9.1](#)
- VERIFY変数 [5.9.4](#), [12.41](#), [12.41.68](#)
- WRAP変数 [6.1.3.2](#), [12.41](#), [12.41.69](#)
- XMLOPTIMIZATIONCHECK変数 [12.41](#), [12.41.70](#)
- XQUERY BASEURI変数 [12.41](#)
- XQUERY CONTEXT変数 [12.41](#)
- XQUERY NODE変数 [12.41](#)
- XQUERY ORDERING変数 [12.41](#)
- SET HISTORYコマンド [12.41.28](#)
- SET LOBPREFETCHコマンド [12.41.32](#)
- SET MARKUP
 - BODY句 [3.5.1.6](#)
 - CSV [3.5.1.6](#), [12.41.36](#)
 - CSV例 [7.1.2](#)
 - DELIMITER句 [12.41.36](#)
 - ENTMAP句 [3.5.1.6](#), [7.1.1.1](#)
 - HEAD句 [3.5.1.6](#)
 - HTML [3.5.1.6](#)
 - 対話型HTMLの例 [7.1.1](#)
 - PREFORMAT句 [3.5.1.6](#)
 - QUOTE句 [12.41.36](#)
 - TABLE句 [3.5.1.6](#)
- SET ROWPREFETCHコマンド [12.41.46](#)
- SET STATEMENTCACHEコマンド [12.41.59](#)
- SETシステム変数の要約 [12.41](#)
- SET変数 [4.7](#)
 - 「システム変数」を参照
- SET XQUERY BASEURI [12.41.71](#)
- SET XQUERY CONTEXT [12.41.74](#)
- SET XQUERY NODE [12.41.73](#)
- SET XQUERY ORDERING [12.41.72](#)
- SGA句 [12.42](#)
- SHIFTINOUT変数 [12.41](#), [12.41.49](#)
- SHOW
 - スキーマ・パラメータ [12.42](#)
- SHOW句 [12.48](#)
- SHOWコマンド [4.7](#), [12.42](#)
 - ALL句 [12.42](#)
 - BTITLE句 [12.42](#)
 - CON_ID句 [12.42](#)

- CON_NAME句 [12.42](#)
- ERRORS句 [12.42](#)
- HISTORY句 [12.42](#)
- LABEL変数 [C.1](#)
- 現行のページ・サイズの表示 [6.3.6](#)
- LNO句 [12.42](#)
- LOBPREFETCH句 [12.42](#)
- PDBS句 [12.42](#)
- PNO句 [12.42](#)
- RELEASE句 [12.42](#)
- REPFOOTER句 [12.42](#)
- REPHEADER句 [12.42](#)
- ROWPREFETCH句 [12.42](#)
- SPOOL句 [12.42](#)
- SQLCODE句 [12.42](#)
- STATEMENTCACHE句 [12.42](#)
- TTITLE句 [12.42](#)
- USER句 [12.42](#)
- XQUERY句 [12.42](#)
- SHOWMODE変数 [12.41](#), [12.41.50](#)
- SHUTDOWNコマンド [12.43](#)
 - ABORT [12.43](#)
 - IMMEDIATE [12.43](#)
 - NORMAL [12.43](#)
 - TRANSACTIONAL LOCAL [12.43](#)
- サイト・プロファイル
 - glogin [2.2](#), [2.2.1](#), [2.2.2.1](#), [3.5.1.9](#), [12.41.55.1](#)
- SKIP句
 - BREAKコマンド [6.2.2](#), [6.2.3](#), [12.9](#)
 - REPHEADERおよびREPFOOTERコマンド [12.37](#)
 - TTITLEおよびBTITLEコマンド [6.3.1.1](#), [12.49](#)
 - 下部タイトルの前に空白行を配置するために使用 [6.3.1.1](#)
- SKIP PAGE句 [6.2.2](#), [6.2.3](#), [12.9](#)
- slash (/)コマンド [12.4](#)
 - GETコマンドでロードされたファイル [12.24](#)
- SPACE変数 [C.1](#), [C.12](#)
- SPOOL句 [3.5.1.6](#), [12.42](#)
- SPOOLコマンド [6.3.6](#), [12.44](#)
 - APPEND句 [12.44](#)
 - CREATE句 [12.44](#)
 - ファイル名 [6.4.2](#), [12.44](#)
 - OFF句 [6.4](#), [12.44](#)
 - OUT句 [6.4.3](#), [12.44](#)
 - REPLACE句 [12.44](#)
 - HTMLファイル [3.5.1.6](#)

- スプールをOFFにする [6.4](#), [12.44](#)
 - SET MARKUPでの使用 [7.1.1](#)
- SQL.PNO、レポート・タイトル内での参照 [6.3.2](#)
- SQL.SQLCODE
 - EXITコマンドでの使用 [12.23](#)
- SQL*Plus
 - コマンド履歴 [12.26](#)
 - コマンド・プロンプト [3.3.1](#)
 - コマンド概要 [12.1](#)
 - グローバリゼーション・サポートの構成 [11.1](#)
 - Oracle Netの構成 [2.2.5](#)
 - データベース管理 [10.1](#)
 - 環境変数 [2.1](#)
 - エラー・メッセージ [13.1](#)
 - 実行計画 [8.1.2](#)
 - 終了 [3.4](#), [12.23](#)
 - 制限 [A](#)
 - 廃止されたコマンドとその代替コマンド [C.1](#)
 - 環境の設定 [2.2](#)
 - 起動 [3.3](#), [3.3.1](#), [3.5](#)
 - 統計 [8.1.3](#)
 - システム変数によるパフォーマンスへの影響 [8.5](#)
 - チューニング [8.1](#)
 - 使用権限を持つユーザー
- SQL*PlusおよびOCIパッケージ [D.1](#)
- SQL*PlusコマンドラインとSQL*Plus Instant Client [D](#)
- SQL*Plus Instant Client [D](#), [D.1](#), [D.2](#), [D.2.1](#), [D.2.2](#), [D.3](#), [D.3.1](#), [D.3.2](#)
 - 基本 [D.1.1](#)
 - インストール [D](#)
 - 軽量 [D.1.2](#)
 - NLS_LANG [D.1.2](#)
 - NLS_LANGのcharsetパラメータ [D.1.2](#)
 - NLS_LANGの言語パラメータ [D.1.2](#)
 - NLS_LANGの地域パラメータ [D.1.2](#)
 - パッケージ内の必須ファイル [D.2.3](#)
 - サポートされていないcharsetでのエラー [D.1.2.1](#)
- SQLBLANKLINES変数 [12.41](#), [12.41.51](#)
- SQLCASE変数 [12.41](#), [12.41.52](#)
- SQL句 [12.12](#)
- SQLCODE句 [12.42](#)
 - SHOWコマンド [12.42](#)
- SQLCONTINUE変数 [12.41](#), [12.41.53](#)
- SQL DML文
 - レポート [12.41](#), [12.41.6](#)
- SQLNUMBER変数 [12.41](#), [12.41.54](#), [12.41.55](#)

- SQLオブティマイザ [8.1.2](#)
- SQLPATH
 - 環境変数 [2.1](#)
 - レジストリ・エントリ [2.1](#), [2.1.1](#)
- SQLPLUSコマンド [3.3.1](#)
 - -?句 [3.5.1.2](#)
 - /NOLOG句 [3.5.2](#)
 - @ (アットマーク) [3.5](#)
 - EXIT FAILURE [3.4](#)
 - アプリケーション・エディション [3.5.2](#)
 - BODYオプション [3.5.1.6](#)
 - -句 [3.5.1.1](#)
 - コマンド
 - SQLPLUS [3.5](#)
 - 接続識別子 [3.5.2](#)
 - CSVオプション [3.5.1.6](#)
 - 構文の表示 [3.5.1.1](#)
 - エディション [3.5.2](#)
 - ENTMAPオプション [3.5.1.6](#)
 - HEADオプション [3.5.1.6](#)
 - HTMLオプション [3.5.1.6](#)
 - -MARKUP句 [3.5.1.6](#), [12.41.36](#)
 - -MARKUPオプション [3.5.1.6](#)
 - nolongontime [3.5.1.8](#)
 - PREFORMATオプション [3.5.1.6](#)
 - RESTRICT [3.5.1.9](#), [9.3](#)
 - サービス名 [3.5.2](#)
 - -SILENT句 [3.5.1.10](#)
 - -SILENTオプション [3.5.1.10](#), [7.1.3](#)
 - SPOOL句 [3.5.1.6](#)
 - 構文 [3.5](#)
 - SYSASM句 [3.5.2](#)
 - SYSBACKUP句 [3.5.2](#)
 - SYSDBA句 [3.5.2](#)
 - SYSDG句 [3.5.2](#)
 - SYSKM句 [3.5.2](#)
 - SYSOPER句 [3.5.2](#)
 - SYSRAC句 [3.5.2](#)
 - TABLEオプション [3.5.1.6](#)
 - 失敗した接続 [3.4](#)
 - ユーザー名/パスワード [3.3.1](#), [3.5.2](#)
- SQLPREFIX変数 [12.41](#), [12.41.56](#)
- SQLPROMPT変数 [8.5.10](#), [12.41](#), [12.41.57](#)
- SQLTERMINATOR変数 [12.27](#), [12.41](#), [12.41.51](#), [12.41.58](#)
- STANDBY DATAFILE句 [12.34](#)

- STANDBY TABLESPACE句 [12.34](#)
- START句 [12.48](#)
- STARTコマンド [5.4](#), [12.45](#)
 - 引数 [5.10](#)
 - スクリプトへのパラメータの受渡し [5.10](#)
 - スクリプト [5.4](#), [12.45](#)
 - @ (アットマーク)コマンドとの類似 [5.4](#), [12.2](#), [12.45](#)
 - @@ (二重アットマーク)コマンドとの類似 [12.45](#)
- 起動
 - SQL*Plus [1.1](#), [3.3](#), [3.3.1](#)
- STARTUPコマンド [12.46](#)
 - DOWNGRADE句 [12.46](#)
 - FORCE句 [12.46](#)
 - MOUNT句 [12.46](#)
 - NOMOUNT句 [12.46](#)
 - OPEN句 [12.46](#)
 - PFILE句 [12.46](#)
 - PLUGGABLE DATABASE句 [12.46](#)
 - RECOVER句 [12.46](#)
 - RESTRICT句 [12.46](#)
 - データベースの指定 [12.46](#)
 - UPGRADE句 [12.46](#)
- 文キャッシュ
 - サイズの設定 [12.41](#)
- STATEMENTCACHE句 [12.42](#)
- STATEMENTCACHE変数 [12.41](#)
- 統計 [8.1.3](#)
 - TIMING統計の収集 [8.2](#)
- STOP句 [12.48](#)
- 問合せの停止 [4.8](#)
- STOREコマンド [2.2.3](#), [12.47](#)
 - SET句 [12.47](#)
- ストアド・ファンクション [5.13](#)
- ストアド・プロシージャ
 - 作成 [4.5.1](#)
- サブキー, レジストリ [2.1.1](#)
- 置換変数 [5.7](#), [5.9](#), [5.9.4](#), [12.41](#), [12.41.14](#)
 - _EDITOR [12.17.1](#)
 - 直後に文字を追加 [5.9.1](#)
 - 値の入力を求める不要なプロンプトの回避 [5.9.2](#)
 - 連結文字 [12.41](#), [12.41.11](#)
 - DEFINEコマンド [12.17](#)
 - 定義 [5.7](#), [5.9.2](#), [12.17](#)
 - 削除 [5.7](#), [12.50](#)
 - ヘッダーおよびフッターの中に表示 [12.37](#)

- タイトル内に表示 [12.49](#)
- ACCEPTコマンド [5.11.1](#), [12.5](#)
- 定義の表示 [5.7](#), [12.17](#)
- 解析 [8.5.4](#)
- 接頭辞の追加 [12.41.14](#), [C.1](#)
- 関連するシステム変数 [5.9.4](#)
- 制限 [5.9.3](#)
- アンパサンドおよび二重アンパサンド [5.9.2](#)
- システム変数との併用 [5.9.4](#)
- 未定義 [5.9.1](#)
- 使用する位置および方法 [5.9.1](#)
- SUFFIX変数 [12.41](#), [12.41.60](#)
 - EDITコマンドとの併用 [12.21](#)
 - GETコマンドとの併用 [12.24](#)
 - SAVEコマンドとの併用 [12.39](#)
 - STARTコマンドとの併用 [12.45](#)
- SUM関数 [6.2.6](#)
- サマリー行
 - 計算および出力 [6.2.6](#), [12.14](#)
 - レポートの終わりでの計算および出力 [6.2.7](#)
 - 異なる列についての同じタイプの計算 [6.2.8](#)
 - 総サマリーおよびサブサマリー(合計)の出力 [6.2.7](#)
 - 同じブレイク列についての複数の出力 [6.2.8](#)
- 構文
 - COPYコマンド [B.2.1](#)
- 構文規則
 - SQL*Plusコマンド [4.6.1](#)
 - SQLコマンド [4.4.1](#)
- SYSASM句 [3.5.2](#), [12.15](#)
- SYSBACKUP句 [3.5.2](#), [12.15](#)
- SYSDBA句 [3.5.2](#), [12.15](#)
- SYSDG句 [3.5.2](#), [12.15](#)
- SYSKM句 [3.5.2](#), [12.15](#)
- SYSOPER句 [3.5.2](#), [12.15](#)
- SYSRAC句 [3.5.2](#), [12.15](#)
- システム管理値
 - ヘッダーおよびフッターの中に表示 [12.37](#)
 - タイトル内に表示 [6.3.2](#), [12.49](#)
 - タイトル内での書式設定 [6.3.2](#)
- システム変数 [4.7](#), [12.40](#)
 - SQL*Plusのパフォーマンスへの影響 [8.5](#)
 - 置換変数への影響 [5.9.4](#)
 - 現行の設定の表示 [4.7](#), [12.42](#)
 - 新旧の値の表示 [12.41](#), [12.41.50](#)
 - 格納およびリストア [2.2.3](#)

- SETコマンドの要約 [12.41](#)
-

T

- TAB句 [12.37](#), [12.49](#)
- TABLE句 [3.5.1.6](#)
- TABLEオプション [3.5.1.6](#)
- 表
 - サンプルへのアクセス
 - コピー時の宛先の制御 [B.1.1](#), [B.2.2](#)
 - 値のコピー [B.2](#), [B.3](#)
 - 列定義の表示 [4.2](#), [12.19](#)
 - コピー時における別のユーザーの表への参照 [B.2.4](#)
- TABLESPACE句 [12.34](#)
- 表領域, リカバリ [12.34](#)
- TAB変数 [8.5.11](#), [12.41](#), [12.41.61](#)
- タグ, HTML [7.1](#)
- TERMOUT変数 [8.5.12](#), [12.41](#), [12.41.62](#)
 - SPOOLコマンドでの使用 [12.44](#)
- 地域
 - SQL*Plus Instant Client [D.1.2](#)
- テキスト [3.5.1.6](#)
 - APPENDによるカレント行への追加 [5.2.3](#), [12.6](#)
 - CHANGEによる旧から新への変更 [5.2.2](#), [12.11](#)
 - バッファからのクリア [5.2](#), [12.12](#)
- テキスト・エディタ
 - オペレーティング・システム [5.1.1](#), [12.21](#)
- TIME変数 [12.41](#), [12.41.63](#)
- TIMING句 [12.12](#)
- TIMINGコマンド [8.2](#), [12.48](#)
 - 作成されたすべての領域の削除 [12.12](#)
 - 現行の領域の削除 [12.48](#)
 - SHOW句 [12.48](#)
 - START句 [12.48](#)
 - STOP句 [12.48](#)
- TIMING変数 [12.41](#), [12.41.64](#)
- タイトル
 - 要素の整列 [6.3.1.1](#), [12.49](#)
 - ページ下部への表示 [6.3](#), [12.10](#), [C.1](#)
 - ページ上部への表示 [6.3](#), [12.49](#), [C.1](#)
 - 列値の表示 [6.3.4](#), [12.13](#)
 - 現在の日付の表示 [6.3.5](#), [12.13](#)
 - ページ番号の表示 [6.3.2](#), [12.49](#)
 - システム管理値の表示 [6.3.2](#), [12.49](#)

- 要素の書式設定 [12.49](#)
- システム管理値の書式設定 [6.3.2](#)
- インデント [6.3.1.2](#), [12.49](#)
- 現行の定義の表示 [6.3.3](#), [12.10](#), [12.49](#)
- 定義の表示と非表示の切替え [6.3.3](#)
- レポートの始めまたは終わりに設定 [6.3](#)
- ページ最上部から上部タイトルまでの行数の設定 [6.3.6](#), [12.41](#), [12.41.37](#), [C.1](#)
- 上部タイトルからページの終わりまでの行の設定 [8.5.8](#), [12.41](#), [12.41.41](#)
- 上部および下部の設定 [6.3](#), [12.10](#), [12.49](#), [C.1](#)
- 最終行から下部タイトルまでの間隔 [6.3.1.1](#)
- 定義の非表示 [6.3.3](#), [12.49](#)
- TNS_ADMIN
 - 環境変数 [2.1](#)
- TO句 [B.2.1](#)
- 問合せのトレース [8.3](#), [8.4](#)
- 文のトレース
 - パフォーマンス統計用 [8.1.3](#)
 - 問合せ実行パス用 [8.1.3](#)
 - パラレル問合せ(PQ)オプション [8.3](#)
- TRIMOUT変数 [12.41](#), [12.41.65](#)
- TRIMSPPOOL変数 [12.41](#), [12.41.66](#)
- TRUNCATED句 [6.1.3.2](#), [12.13](#)
- TRUNCATE変数 [C.1](#), [C.13](#)
- TTITLE句 [12.42](#)
- TTITLEコマンド [6.3.1](#), [12.49](#)
 - タイトル要素の整列 [6.3.1.1](#), [12.49](#)
 - BOLD句 [12.49](#)
 - CENTER句 [6.3.1.1](#), [12.49](#)
 - COL句 [6.3.1.2](#), [12.49](#)
 - FORMAT句 [6.3.2](#), [12.49](#)
 - タイトルのインデント [6.3.1.2](#), [12.49](#)
 - LEFT句 [6.3.1.1](#), [12.49](#)
 - 現行の定義の表示 [6.3.3](#), [12.49](#)
 - OFF句 [6.3.3](#), [12.49](#)
 - 旧形式 [C.14](#)
 - ON句 [6.3.3](#)
 - 列値変数の参照 [6.3.4](#), [12.13](#)
 - 現行の定義の表示と非表示の切替え [6.3.3](#)
 - RIGHT句 [6.3.1.1](#), [12.49](#)
 - SKIP句 [6.3.1.1](#), [12.49](#)
 - 現行の定義の非表示 [6.3.3](#), [12.49](#)
 - TAB句 [12.49](#)
- チューニング
 - SET APPINFO OFF [8.5.2](#)
 - SET ARRAYSIZE [8.5.3](#)

- SET DEFINE OFF [8.5.4](#)
 - SET FLUSH OFF [8.5.5](#)
 - SET TRIMOUT ON [8.5.13](#)
 - SET TRIMSPool ON [8.5.13](#)
 - SQL*Plus [8.1](#)
 - システム変数 [8.5](#)
 - TWO_TASK
 - 環境変数 [2.1](#)
-

U

- UNDEFINEコマンド [5.7](#), [12.50](#)
 - DEFINEコマンド [12.17](#)
 - UNDERLINE変数 [12.41](#), [12.41.67](#)
 - unicode [11.1](#)
 - UNIXの場合
 - ed [12.17.1](#)
 - SQL*Plus Instant Clientのインストール [D.2.2](#)
 - コピーするSQL*Plus Instant Clientファイル [D.3.1](#)
 - サンプル表のロック解除
 - UNTIL CANCEL句 [12.34](#)
 - UNTIL CHANGE句 [12.34](#)
 - UNTIL CONTROLFILE句 [12.34](#)
 - UNTIL TIME句 [12.34](#)
 - USER句 [12.42](#)
 - ユーザー名 [3.1](#)
 - 異なるユーザー名での接続 [3.1](#), [12.15](#)
 - CONNECTコマンド [3.1](#), [12.15](#)
 - COPYコマンド [B.2.1](#), [B.2.2](#), [B.3](#)
 - SQLPLUSコマンド [3.3.1](#), [3.5.2](#)
 - ユーザー・プロファイル [2.2.2](#)
 - 「サイト・プロファイル」を参照
 - glogin.sql [2.2.2](#)
 - login.sql [2.2.2](#), [2.2.2.1](#)
 - ユーザー変数
 - 「置換変数」を参照
 - USING BACKUP CONTROL FILE句 [12.34](#)
 - USING句 [B.1.1](#), [B.2.1](#)
 - UTF-8 [11.1](#)
-

V

- V\$SESSION仮想表 [12.41.1](#)
- V\$SQLAREA仮想表 [12.41.1](#)

- VARCHAR2
 - DESCRIBEによる列定義 [12.19](#)
 - VARCHAR2句
 - VARIABLEコマンド [12.51](#)
 - VARCHAR2列
 - 書式の変更 [12.13](#)
 - デフォルト書式 [6.1.3.1](#)
 - VARCHAR列
 - デフォルト書式 [6.1.3.1](#)
 - VARIABLEコマンド [12.51](#)
 - BINARY_DOUBLE句 [12.51](#)
 - BINARY_FLOAT句 [12.51](#)
 - CHAR句 [12.51](#)
 - CLOB句 [12.51](#)
 - NCHAR句 [12.51](#)
 - NCLOB句 [12.51](#)
 - NUMBER句 [12.51](#)
 - REFCURSOR句 [12.51](#)
 - value句 [12.51](#)
 - VARCHAR2句 [12.51](#)
 - variable句 [12.51](#)
 - 変数
 - バインド変数 [5.12](#)
 - 置換変数 [12.17](#)
 - システム変数 [4.7](#)
 - VERIFY句 [5.9.1](#)
 - VERIFY変数 [5.9.4](#), [12.41](#), [12.41.68](#)
-

W

- WARNING句 [12.23](#)
- Web, レポートの出力 [7.1](#)
- Webブラウザ [7.1.1](#)
- WHENEVER OSERRORコマンド [12.52](#)
 - COMMIT句 [12.52](#)
 - CONTINUE句 [12.52](#)
 - EXIT句 [12.52](#)
 - NONE句 [12.52](#)
 - ROLLBACK句 [12.52](#)
- WHENEVER SQLERRORコマンド [12.53](#)
 - COMMIT句 [12.53](#)
 - CONTINUE句 [12.53](#)
 - EXIT句 [12.53](#)
 - NONE句 [12.53](#)

- ROLLBACK句 [12.53](#)
 - Windows
 - SQL*Plus Instant Clientのインストール [D.2.2](#)
 - ノート帳 [12.17.1](#)
 - コピーするSQL*Plus Instant Clientファイル [D.3.2](#)
 - WORD_WRAPPED句 [6.1.3.2](#), [6.1.7](#), [12.13](#)
 - WRAPPED句 [6.1.3.2](#), [12.13](#)
 - WRAP変数 [6.1.3.2](#), [12.41](#), [12.41.69](#)
-

X

- XMLOPTIMIZATIONCHECK変数 [12.41](#), [12.41.70](#)
- XML型
 - DESCRIBEによる列定義 [12.19](#)
 - 列の書式設定 [6.1.3.2](#)
 - 列幅 [6.1.3.1](#)
 - 作成 [6.1.3.2](#)
 - レポート内での書式設定 [6.1.3](#)
 - 値の挿入 [6.1.3.2](#)
 - データの選択 [6.1.3.2](#)
 - 列の取得サイズの設定 [8.5.7](#), [12.41.35](#)
 - 列の最大幅の設定 [12.41.34](#)
- XQUERY句 [12.42](#)
- XQUERYコマンド [12.54](#)
- XQUERYオプション
 - BASEURI [12.41.71](#)
 - CONTEXT [12.41.74](#)
 - NODE [12.41.73](#)
 - ORDERING [12.41.72](#)