

# SQL\*Plus

ユーザーズ・ガイドおよびリファレンス

リリース 9.2

2002 年 7 月

部品番号 : J06266-01

ORACLE®

---

SQL\*Plus ユーザーズ・ガイドおよびリファレンス, リリース 9.2

部品番号 : J06266-01

原本名 : SQL\*Plus User's Guide and Reference, Release 9.2

原本部品番号 : A90842-01

原本著者 : Simon Watt

原本協力者 : Andrew Code, Alison Goggin, Alison Holloway, Christopher Jones, Luan Nim, Marcus Pecher, Andrei Souleimanian, Christopher Tan, Ian Wu

Copyright © 1996, 2002 Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム (ソフトウェアおよびドキュメントを含む) の使用、複製または開示は、オラクル社との契約に記載された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されています。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

\* オラクル社とは、Oracle Corporation (米国オラクル) または日本オラクル株式会社 (日本オラクル) を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上記のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation (米国オラクル) およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

---

---

# 目次

<b>はじめに</b> .....	xiii
対象読者 .....	xiv
このマニュアルの構成 .....	xiv
関連文書 .....	xvi
表記規則 .....	xviii
<b>SQL*Plus の新機能</b> .....	xxiii
SQL*Plus リリース 9.2 の新機能 .....	xxiv
iSQL*Plus リリース 9.2 の新機能 .....	xxv
iSQL*Plus サーバー リリース 9.0.1 からリリース 9.2 へのアップグレード .....	xxviii
SQL*Plus リリース 9.0.1 の新機能 .....	xxx
<b>SQL*Plus クイック・スタート</b> .....	xxxiii
iSQL*Plus クイック・スタート .....	xxxiv
Windows 用の SQL*Plus コマンドライン・クイック・スタート .....	xxxiv
UNIX 用の SQL*Plus コマンドライン・クイック・スタート .....	xxxv
<b>第 I 部 SQL*Plus スタート・ガイド</b>	
<b>1 SQL*Plus 概要</b>	
SQL*Plus について .....	1-2
SQL*Plus コマンドラインのアーキテクチャ .....	1-4
iSQL*Plus アーキテクチャ .....	1-4
SQL*Plus のインストール .....	1-5

SQL*Plus の対象ユーザー .....	1-5
SQL*Plus の学習方法 .....	1-6
このマニュアルの使用法 .....	1-6
Oracle9i のサンプル表および SQL*Plus .....	1-6
サンプル表のロック解除 .....	1-7

## 2 SQL\*Plus ユーザー・インタフェース

SQL*Plus コマンドライン・ユーザー・インタフェース .....	2-2
コマンドラインの画面 .....	2-2
iSQL*Plus ユーザー・インタフェース .....	2-3
iSQL*Plus ナビゲーション .....	2-3
iSQL*Plus の「Login」画面 .....	2-4
iSQL*Plus の「Work Screen」画面 .....	2-5
iSQL*Plus の「History」画面 .....	2-6
iSQL*Plus の「Preferences」画面 .....	2-8
iSQL*Plus ユーザー・インタフェース・オプション .....	2-9
iSQL*Plus DBA の「Login」画面 .....	2-10
iSQL*Plus DBA の「Work Screen」画面 .....	2-12
iSQL*Plus サーバーの統計 .....	2-12
iSQL*Plus でのシステム変数の設定 .....	2-14
iSQL*Plus Extension for Windows .....	2-15

## 3 SQL\*Plus の構成

SQL*Plus の構成 .....	3-2
サイト・プロファイル .....	3-2
ユーザー・プロファイル .....	3-2
SQL*Plus システム変数の格納およびリストア .....	3-3
iSQL*Plus の構成 .....	3-5
クライアント層の構成 .....	3-5
中間層の構成 : iSQL*Plus サーバー .....	3-7
中間層の構成 : Oracle HTTP Server .....	3-12
中間層の構成 : Oracle Net .....	3-16
データベース層の構成 .....	3-16
iSQL*Plus Extension for Windows の構成 .....	3-16
「Configuration」ダイアログ・ボックス .....	3-16

## 4 SQL\*Plus の起動

SQL*Plus コマンドラインの起動 .....	4-2
SQLPLUS コマンド .....	4-3
Option .....	4-4
Logon .....	4-10
Start .....	4-11
コマンドライン・ヘルプの使用方法 .....	4-12
コマンドライン・ヘルプのインストール .....	4-12
SQL*Plus コマンドラインの終了 .....	4-13
iSQL*Plus の起動 .....	4-13
iSQL*Plus でのヘルプの使用方法 .....	4-14
iSQL*Plus の終了 .....	4-14
URL からの iSQL*Plus の起動 .....	4-15
ユーザー名およびパスワードのログイン .....	4-17
パスワードの変更 .....	4-18
iSQL*Plus でのパスワードの変更 .....	4-18
マルチユーザー・システム .....	4-19
シングルユーザー・システム .....	4-19
データベース接続識別子 .....	4-20
SQL*Plus の起動時にリモート・データベースに接続 .....	4-21
CONNECT コマンドを使用してリモート・データベースに接続 .....	4-21

## 第 II 部 SQL\*Plus の使用方法

### 5 SQL\*Plus の基本

コマンドの入力と実行 .....	5-2
SQL バッファ .....	5-2
コマンドの実行 .....	5-3
SQL コマンドの実行 .....	5-3
PL/SQL ブロックの実行 .....	5-7
SQL*Plus コマンドの実行 .....	5-8
SQL*Plus コマンドの構文について .....	5-9
コマンドの実行に影響するシステム変数 .....	5-10
データベースへの変更の自動保存 .....	5-11
実行中のコマンドの停止 .....	5-12

ホスト・オペレーティング・システムのコマンドの実行 .....	5-12
ヘルプの使用法 .....	5-13
表定義の表示 .....	5-13
PL/SQL 定義の表示 .....	5-14
表示の制御 .....	5-14
エラー・メッセージの解釈 .....	5-15

## 6 SQL\*Plus でのスクリプトの使用

スクリプトの編集 .....	6-2
システム・エディタでのスクリプトの作成 .....	6-2
スクリプトの変更 .....	6-3
SQL*Plus でのスクリプトの編集 .....	6-4
スクリプトへのコメントの挿入 .....	6-9
REMARK コマンドの使用法 .....	6-9
/*...*/ の使用法 .....	6-10
「--」 の使用法 .....	6-10
コメント挿入時の注意 .....	6-11
スクリプトの実行 .....	6-13
スクリプトのネスト .....	6-14
リターン・コードの受信 .....	6-15
リターン・コードを伴うスクリプトの終了 .....	6-16
対話型コマンドの作成 .....	6-16
ユーザー変数の定義 .....	6-16
置換変数の使用 .....	6-17
iSQL*Plus の「Substitution Variables」画面 .....	6-23
START コマンドを使用したパラメータの受渡し方法 .....	6-25
ユーザーとの対話 .....	6-26
バインド変数の使用法 .....	6-29
バインド変数の作成 .....	6-30
バインド変数の参照 .....	6-30
バインド変数の表示 .....	6-30
REFCURSOR バインド変数の使用法 .....	6-31

## 7 SQL\*Plus レポートの書式設定

列の書式設定 .....	7-2
列ヘッダーの変更 .....	7-2

NUMBER 列の書式設定 .....	7-4
データ型の書式設定 .....	7-5
列の表示属性のコピー .....	7-8
列の表示属性の表示およびリセット .....	7-9
列の表示属性の非表示および表示 .....	7-9
折り返した列の値の後に文字の行を出力する方法 .....	7-10
<b>間隔の設定およびサマリー行によるレポートの明確化 .....</b>	<b>7-11</b>
ブレイク列内の値の重複の抑止 .....	7-12
ブレイク列の値が変更されたときの空白の挿入 .....	7-13
各行の後に空白を挿入する方法 .....	7-13
複数の間隔設定技法の使用法 .....	7-14
ブレイク定義の表示および取消し .....	7-15
ブレイ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 .....	7-15
レポートの終わりでのサマリー行の計算 .....	7-19
複数のサマリー値およびサマリー行の計算 .....	7-20
COMPUTE 定義の表示および取消し .....	7-21
<b>ページおよびレポートのタイトルとサイズの定義 .....</b>	<b>7-22</b>
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 .....	7-22
タイトル内にシステム管理値を表示する方法 .....	7-26
ページ・タイトル定義の表示と非表示の切替え .....	7-27
タイトル内に列の値を表示する方法 .....	7-28
タイトル内に現在の日付を表示する方法 .....	7-29
ページ・サイズの設定 .....	7-30
<b>問合せ結果の格納および印刷 .....</b>	<b>7-32</b>
ファイルへの結果の格納 .....	7-33
プリンタへの結果の出力 .....	7-33

## 8 SQL\*Plus からの HTML レポートの生成

<b>SQL*Plus コマンドラインを使用したレポートの作成 .....</b>	<b>8-2</b>
レポートの作成 .....	8-2
CGI スクリプトを使用したレポートの作成 .....	8-7
レポートでの SQL*Plus コマンドの非表示 .....	8-11
HTML エンティティ .....	8-12
<b>iSQL*Plus を使用したレポートの作成 .....</b>	<b>8-12</b>

## 9 SQL\*Plus のチューニング

文のトレース .....	9-2
レポートの制御 .....	9-2
実行計画 .....	9-3
統計 .....	9-4
タイミング統計の収集 .....	9-7
パラレル問合せおよび分散問合せのトレース .....	9-7
システム変数による SQL*Plus のパフォーマンスへの影響 .....	9-10
iSQL*Plus サーバーの統計 .....	9-11
アクティブ統計の解釈 .....	9-13
iSQL*Plus のチューニングに関する注意事項 .....	9-14

## 10 SQL\*Plus セキュリティ

PRODUCT_USER_PROFILE 表 .....	10-2
PUP 表の作成 .....	10-2
PUP 表の構造 .....	10-3
PUP 列の説明および使用方法 .....	10-3
PUP 表の管理 .....	10-4
SQL*Plus、SQL および PL/SQL コマンドの使用禁止 .....	10-4
ロールの作成および制御 .....	10-6
SET ROLE の使用禁止 .....	10-7
ユーザー・ロールの使用禁止 .....	10-7
SQLPLUS -RESTRICT でのコマンドの使用禁止 .....	10-8
iSQL*Plus のセキュリティ .....	10-9
管理権限 .....	10-9
ユーザーのサーバー認証の有効化 .....	10-10
データベース・アクセス制限の有効化または無効化 .....	10-10
DBA アクセスの有効化 .....	10-11
Oracle HTTP Server 認証ファイルへのエントリの追加 .....	10-12
セキュリティの使用方法 .....	10-13

## 11 SQL\*Plus を使用したデータベース管理

概要 .....	11-2
データベースの起動および停止の概要 .....	11-2
データベースの起動 .....	11-2

データベースの停止 .....	11-3
REDO ログ・ファイル .....	11-4
ARCHIVELOG モード .....	11-4
データベースのリカバリ .....	11-5

## 12 SQL\*Plus グローバリゼーション・サポート

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 .....	12-2
iSQL*Plus でのグローバリゼーション・サポートの構成 .....	12-3

## 第 III 部 SQL\*Plus リファレンス

### 13 SQL\*Plus コマンド・リファレンス

SQL*Plus コマンド一覧 .....	13-2
@ (アットマーク) .....	13-5
@@ (二重アットマーク) .....	13-7
/ (スラッシュ) .....	13-9
ACCEPT .....	13-10
APPEND .....	13-12
ARCHIVE LOG .....	13-13
ATTRIBUTE .....	13-16
BREAK .....	13-18
BTITLE .....	13-23
CHANGE .....	13-24
CLEAR .....	13-27
COLUMN .....	13-29
COMPUTE .....	13-39
CONNECT .....	13-44
COPY .....	13-47
DEFINE .....	13-48
DEL .....	13-50
DESCRIBE .....	13-52
DISCONNECT .....	13-58
EDIT .....	13-59
EXECUTE .....	13-61
EXIT .....	13-62
GET .....	13-64

HELP .....	13-66
HOST .....	13-67
INPUT .....	13-69
LIST .....	13-71
PASSWORD .....	13-73
PAUSE .....	13-74
PRINT .....	13-75
PROMPT .....	13-76
RECOVER .....	13-77
REMARK .....	13-83
REPFOOTER .....	13-84
REPHEADER .....	13-86
RUN .....	13-89
SAVE .....	13-90
SET .....	13-92
SET APPI[INFO]{ON OFF text} .....	13-96
SET ARRAY[SIZE] {15 n} .....	13-97
SET AUTO[COMMIT]{ON OFF IMM[EDIATE] n} .....	13-98
SET AUTOP[RINT] {ON OFF} .....	13-98
SET AUTORECOVERY {ON OFF} .....	13-98
SET AUTOT[RACE] {ON OFF TRACE[ONLY]} [EXP[LAIN]] [STAT[ISTICS]] .....	13-99
SET BLO[CKTERMINATOR] {c ON OFF} .....	13-99
SET CMDS[EP] {; c ON OFF} .....	13-100
SET COLSEP {_ text} .....	13-101
SET COM[PATIBILITY]{V7 V8 NATIVE} .....	13-101
SET CON[CAT] {c ON OFF} .....	13-102
SET COPYC[OMMIT] {Q n} .....	13-102
SET COPYTYPECHECK {ON OFF} .....	13-102
SET DEF[INE] {& c ON OFF} .....	13-102
SET DESCRIBE [DEPTH {1 n ALL}][LINENUM {ON OFF}][INDENT {ON OFF}] .....	13-102
SET ECHO {ON OFF} .....	13-104
SET EDITF[ILE] file_name.ext .....	13-104
SET EMB[EDDED] {ON OFF} .....	13-104
SET ESC[APE] {x c ON OFF} .....	13-104
SET FEED[BACK] {6 n ON OFF} .....	13-105
SET FLAGGER {OFF ENTRY INTERMED[IATE] FULL} .....	13-105

SET FLU[SH] { <u>ON</u>   OFF} .....	13-105
SET HEA[DING] { <u>ON</u>   OFF} .....	13-106
SET HEADS[EP] { <u>1</u>   <i>c</i>   <u>ON</u>   OFF} .....	13-106
SET INSTANCE [ <i>instance_path</i>   <u>LOCAL</u> ] .....	13-106
SET LIN[ESIZE] { <u>80</u>   <i>n</i> } .....	13-107
SET LOBOF[FSET] { <i>n</i>   <u>1</u> } .....	13-107
SET LOGSOURCE [ <i>pathname</i> ] .....	13-108
SET LONG { <u>80</u>   <i>n</i> } .....	13-108
SET LONGC[HUNKSIZE] { <u>80</u>   <i>n</i> } .....	13-108
SET MARK[UP] HTML { <u>ON</u>   <u>OFF</u> } [HEAD <i>text</i> ] [BODY <i>text</i> ] [TABLE <i>text</i> ] [ENTMAP { <u>ON</u>   <u>OFF</u> }] [SPOOL { <u>ON</u>   <u>OFF</u> }] [PRE[FORMAT] { <u>ON</u>   <u>OFF</u> }] .....	13-109
SET NEWP[AGE] { <u>1</u>   <i>n</i>   NONE} .....	13-110
SET NULL <i>text</i> .....	13-110
SET NUMF[ORMAT] <i>format</i> .....	13-111
SET NUM[WIDTH] { <u>10</u>   <i>n</i> } .....	13-111
SET PAGES[IZE] { <u>24</u>   <i>n</i> } .....	13-111
SET PAU[SE] { <u>ON</u>   <u>OFF</u>   <i>text</i> } .....	13-111
SET RECSEP { <u>WR[APPED]</u>   EA[CH]   OFF} .....	13-111
SET RECSEPCHAR { <u>_</u>   <i>c</i> } .....	13-111
SET SERVEROUT[PUT] { <u>ON</u>   <u>OFF</u> } [SIZE <i>n</i> ] [FOR[MAT] { <u>WRA[PPED]</u>   <u>WOR[D WRAPPED]</u>   TRU[NCATED]}} .....	13-112
SET SHIFT[INOUT] { <u>VIS[IBLE]</u>   <u>INV[ISIBLE]</u> } .....	13-113
SET SHOW[MODE] { <u>ON</u>   <u>OFF</u> } .....	13-114
SET SQLBL[ANKLINES] { <u>ON</u>   <u>OFF</u> } .....	13-114
SET SQLC[ASE] { <u>MIX[ED]</u>   LO[WER]   UP[PER]} .....	13-115
SET SQLCO[NTINUE] { <u>&gt;=</u>   <i>text</i> } .....	13-115
SET SQLN[UMBER] { <u>ON</u>   OFF} .....	13-116
SET SQLPLUSCOMPAT[IBILITY] { <i>x.y.z</i> } .....	13-116
SET SQLPRE[FIX] { <u>#</u>   <i>c</i> } .....	13-116
SET SQLP[ROMPT] { <u>SQL&gt;</u>   <i>text</i> } .....	13-116
SET SQT[ERMINATOR] { <u>;</u>   <i>c</i>   <u>ON</u>   OFF} .....	13-117
SET SUF[FIX] { <u>SQL</u>   <i>text</i> } .....	13-117
SET TAB { <u>ON</u>   OFF} .....	13-118
SET TERM[OUT] { <u>ON</u>   OFF} .....	13-118
SET TI[ME] { <u>ON</u>   <u>OFF</u> } .....	13-118

SET TIMI[NG] {ON OFF} .....	13-118
SET TRIM[OUT] {ON OFF} .....	13-118
SET TRIMS[POOL] {ON OFF} .....	13-119
SET UND[ERLINE] {- c ON OFF} .....	13-119
SET VER[IFY] {ON OFF} .....	13-119
SET WRA[P] {ON OFF} .....	13-119
SHOW .....	13-120
SHUTDOWN .....	13-125
SPOOL .....	13-127
START .....	13-129
STARTUP .....	13-131
STORE .....	13-135
TIMING .....	13-136
TTITLE .....	13-138
UNDEFINE .....	13-141
VARIABLE .....	13-142
WHENEVER OSERROR .....	13-148
WHENEVER SQLERROR .....	13-150

## 14 SQL\*Plus エラー・メッセージ

SQL*Plus のエラー・メッセージ .....	14-2
iSQL*Plus のエラー・メッセージ .....	14-33
COPY コマンドのメッセージ .....	14-43

## 第 IV 部 SQL\*Plus の付録

### A SQL\*Plus の制限

### B SQL\*Plus の COPY コマンド

COPY コマンドの構文 .....	B-2
データベース間でのデータのコピー .....	B-5
COPY コマンド構文について .....	B-5
宛先の表の扱いを制御する方法 .....	B-6
COPY で表示されるメッセージの解釈 .....	B-8
別ユーザーの表の指定 .....	B-8

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-9
---------------------------------	-----

## C 廃止された SQL\*Plus コマンド

SQL*Plus で廃止されたコマンドとその代替コマンド .....	C-2
BTITLE (旧形式) .....	C-3
COLUMN DEFAULT .....	C-3
DOCUMENT .....	C-3
NEWPAGE .....	C-4
SET BUFFER .....	C-4
SET CLOSECURSOR .....	C-5
SET DOCUMENT .....	C-5
SET MAXDATA .....	C-6
SET SCAN .....	C-6
SET SPACE .....	C-6
SET TRUNCATE .....	C-7
SHOW LABEL .....	C-7
TTITLE (旧形式) .....	C-7

## D iSQL\*Plus では未サポートのコマンド

用語集

索引



---

# はじめに

このマニュアルでは、SQL\*Plus プログラムおよびその使用方法について説明します。それぞれの SQL\*Plus コマンドについても詳細に説明します。

ここでは、次の項目について説明します。

- 対象読者
- このマニュアルの構成
- 関連文書
- 表記規則

# 対象読者

このマニュアルは、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。

- SQL コマンドおよび PL/SQL ブロックの入力、編集、格納、取得および実行
- 問合せ結果の書式設定、計算、格納、印刷および Web への表示
- 表の列定義の表示
- エンド・ユーザーへのメッセージの送信およびエンド・ユーザーからの応答の受信
- 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、『Oracle9i SQL リファレンス』を参照してください。PL/SQL データベース言語を SQL\*Plus と組み合わせて使用する場合は、PL/SQL の使用方法について『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。

## このマニュアルの構成

このマニュアルは、次のように構成されています。

### SQL\*Plus クイック・スタート

インストール後の起動および実行について簡単に説明します。

## 第 I 部 SQL\*Plus スタート・ガイド

SQL\*Plus の概要、コマンドライン・ユーザー・インタフェースおよび iSQL\*Plus 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 1 章 「SQL\*Plus 概要」

SQL\*Plus の概要を説明し、このマニュアルの使用方法および SQL\*Plus の実行に必要な情報を提供します。

iSQL\*Plus アーキテクチャおよび SQL\*Plus に対するブラウザ・ベースの iSQL\*Plus インタフェースについての概要を説明します。

### 第 2 章 「SQL\*Plus ユーザー・インタフェース」

コマンドライン・ユーザー・インタフェース、iSQL\*Plus ユーザー・インタフェースおよび iSQL\*Plus Extension for Windows について説明します。

### 第 3 章 「SQL\*Plus の構成」

SQL\*Plus のコマンドライン環境および iSQL\*Plus 環境の構成方法について説明します。

## 第 4 章 「SQL\*Plus の起動」

Oracle データベースの起動、Oracle データベースへの接続、コマンドラインとオンライン・ヘルプへのアクセスおよび SQL\*Plus の終了方法について説明します。

## 第 II 部 SQL\*Plus の使用方法

SQL\*Plus のユーザーズ・ガイドとチュートリアル・コンテンツ、スクリプトの作成方法および SQL\*Plus のチューニング、セキュリティ、データベース管理とグローバリゼーション情報について説明します。

## 第 5 章 「SQL\*Plus の基本」

コマンドを入力して実行する方法について説明します。ここでは、サンプル表を使用した例を手順ごとに実践しながら学習します。

## 第 6 章 「SQL\*Plus でのスクリプトの使用」

SQL\*Plus、SQL および PL/SQL の文およびコマンドの作成および編集方法について、例を使用して説明します。

## 第 7 章 「SQL\*Plus レポートの書式設定」

問合せ結果の書式を設定して最終的なレポートを生成する方法について説明します。HTML 出力については説明しません。

## 第 8 章 「SQL\*Plus からの HTML レポートの生成」

問合せ結果を含む HTML レポートの生成方法について説明します。

## 第 9 章 「SQL\*Plus のチューニング」

統計およびその他のメカニズムを取得および使用して、SQL\*Plus から最適なパフォーマンスを得る方法について説明します。

## 第 10 章 「SQL\*Plus セキュリティ」

データベースへのアクセス、特定の SQL\*Plus および SQL コマンドへのアクセスを制限する方法について説明します。

## 第 11 章 「SQL\*Plus を使用したデータベース管理」

データベース管理者 (DBA) 用の SQL\*Plus の基本的なデータベース管理機能について説明します。

## 第 12 章 「SQL\*Plus グローバリゼーション・サポート」

SQL\*Plus コマンドライン・インタフェースおよび iSQL\*Plus ユーザー・インタフェースでのグローバリゼーション・サポートの構成方法について説明します。

## 第 III 部 SQL\*Plus のリファレンス

SQL\*Plus のコマンド・リファレンスおよびエラー・メッセージが含まれます。

### 第 13 章 「SQL\*Plus コマンド・リファレンス」

SQL\*Plus コマンドの概要および個々の SQL\*Plus コマンドの詳細をアルファベット順で説明します。

### 第 14 章 「SQL\*Plus エラー・メッセージ」

SQL\*Plus、iSQL\*Plus および COPY コマンドによって生成されるエラー・メッセージを示します。エラーの原因およびリカバリのための適切な処置について説明します。

## 第 IV 部 SQL\*Plus の付録

SQL\*Plus の付録が含まれます。

### 付録 A 「SQL\*Plus の制限」

SQL\*Plus の要素の最大値を示します。

### 付録 B 「SQL\*Plus の COPY コマンド」

COPY コマンドの構文および使用方法について説明します。

### 付録 C 「廃止された SQL\*Plus コマンド」

廃止された SQL\*Plus コマンドについて説明します。

### 付録 D 「iSQL\*Plus では未サポートのコマンド」

iSQL\*Plus でサポートされていない SQL\*Plus コマンドを示します。

### 用語集

Oracle および SQL\*Plus に関連する専門用語について説明します。

## 関連文書

詳細は、次の Oracle マニュアルを参照してください。

- 『SQL\*Plus クイック・リファレンス』
- 『PL/SQL ユーザーズ・ガイドおよびリファレンス』
- 『Oracle9i SQL リファレンス』
- 『Oracle9i データベース概要』
- 『Oracle9i データベース管理者ガイド』

- 『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』
- 『Oracle9i アプリケーション開発者ガイド - 基礎編』
- 『Oracle9i XML データベース開発者ガイド -Oracle XML DB』
- 『Oracle9i Database グローバリゼーション・サポート・ガイド』
- 『Oracle9i Heterogeneous Connectivity Administrator's Guide』
- 『Oracle9i アドバンスト・レプリケーション』
- 『Oracle9i データベース・ユーティリティ』
- 『Oracle9i データベース・エラー・メッセージ』
- 『Oracle9i データベース移行ガイド』
- 『Oracle9i データベース・リファレンス』
- 『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』
- 『Oracle9i Real Application Clusters 概要』
- 『Oracle9i Net Services 管理者ガイド』
- 『Oracle Call Interface プログラマーズ・ガイド』
- 『Pro\*COBOL Precompiler プログラマーズ・ガイド』
- 『Pro\*C/C++ Precompiler 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 インストール・ガイドおよび管理者リファレンス

このマニュアルの多くの例で、Oracle インストール時にデフォルトとしてインストールされるシード・データベースのサンプル・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の詳細は、『Oracle9i サンプル・スキーマ』を参照してください。

リリース・ノート、インストール・マニュアル、ホワイト・ペーパーまたはその他の関連文書は、OTN-J (Oracle Technology Network Japan) に接続すれば、無償でダウンロードできます。OTN-J を使用するには、オンラインでの登録が必要です。次の URL で登録できます。

<http://otn.oracle.co.jp/membership/>

すでに OTN-J のユーザー名およびパスワードを取得済であれば、次の OTN-J Web サイトの文書セクションに直接接続できます。

<http://otn.oracle.co.jp/document/>

# 表記規則

この項では、このマニュアルの本文およびコード例で使用される表記規則について説明します。この項の内容は次のとおりです。

- 本文中の表記規則
- コード例中の表記規則
- Windows オペレーティング・システム用の表記規則

## 本文中の表記規則

本文では、特別な用語をより迅速に識別できるように、様々な表記規則を使用します。次の表に、それらの表記規則を説明し、その使用例を示します。

規則	意味	例
太字	太字は、本文中で定義されている用語または用語集で記載されている用語（あるいはその両方）を示します。	この句を指定すると、 <b>索引構成表</b> が作成されます。
固定幅フォントの大文字	固定幅フォントの大文字は、システムが提供する要素を示します。このような要素には、パラメータ、権限、データ型、Recovery Manager のキーワード、SQL キーワード、SQL*Plus またはユーティリティ・コマンド、パッケージおよびメソッドが含まれます。また、システムが提供する列名、データベース・オブジェクト、データベース構造、ユーザー名およびロールも含まれます。	NUMBER 列に対してのみに、この句を指定できません。 BACKUP コマンドを使用して、データベースのバックアップを取ることができます。 USER_TABLES データ・ディクショナリ・ビュー内の TABLE_NAME 列を問い合わせます。 DBMS_STATS.GENERATE_STATS プロシージャを使用します。
固定幅フォントの小文字	固定幅フォントの小文字は、実行可能ファイル、ファイル名、ディレクトリ名およびユーザーが提供する要素のサンプルを示します。このような要素には、コンピュータ名、データベース名、ネット・サービス名および接続識別子が含まれます。また、ユーザーが提供するデータベース・オブジェクトとデータベース構造、列名、パッケージとクラス、ユーザー名とロール、プログラム・ユニットおよびパラメータ値も含まれます。 <b>注意:</b> 大文字と小文字を組み合わせて使用するプログラム要素もあります。これらの要素は、記載されているとおり入力してください。	sqlplus と入力して、SQL*Plus をオープンします。 パスワードは、orapwd ファイルで指定します。 /disk1/oracle/dbs ディレクトリ内のデータ・ファイルおよび制御ファイルのバックアップを取ります。 hr.departments 表には、department_id、department_name および location_id 列があります。 QUERY_REWRITE_ENABLED 初期化パラメータを true に設定します。 oe ユーザーとして接続します。 JRepUtil クラスが次のメソッドを実装します。

規則	意味	例
固定幅フォントの 小文字の イタリック	固定幅フォントの小文字のイタリックは、 プレースホルダまたは変数を示します。	<i>parallel_clause</i> を指定できます。  <i>Uold_release</i> .SQL を実行します。ここで、 <i>old_release</i> とはアップグレード前にインス トールしたリリースを示します。

## コード例中の表記規則

コード例では、SQL、PL/SQL、SQL\*Plus またはその他のコマンドライン文を示します。  
コード例は、固定幅フォントで示され、通常のテキストと区別されます。

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

ユーザーがシステムに入力するコマンドライン文は、次の例の左側にあるキーボードのアイコンで識別されます。コード例は、固定幅フォントで示され、通常のテキストと区別されません。



```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

同様に、出力例は、左側にあるコンピュータ画面のアイコンで識別されます。



```
PAGESIZE 24
```

両方のアイコンが表示される場合は、対話形式で入出力を行います。



```
1
  1* SELECT LAST_NAME, SALARY
APPEND , COMMISSION_PCT;
```

次の表に、コード例で使用される表記規則を説明し、その使用例を示します。

規則	意味	例
[ ]	大カッコは、任意に選択する1つ以上の項目を囲みます。大カッコは、入力しないでください。	DECIMAL ( <i>digits</i> [ , <i>precision</i> ])
{ }	中カッコは、2つ以上の項目を囲み、そのうち1つの項目は必須です。中カッコは、入力しないでください。	{ENABLE   DISABLE}
	縦線は、大カッコまたは中カッコ内の2つ以上のオプションの選択項目を表します。いずれかのオプションを入力します。縦線は、入力しないでください。	{ENABLE   DISABLE} [COMPRESS   NOCOMPRESS]
...	水平の省略記号は、次のいずれかを示します。 <ul style="list-style-type: none"> <li>■ 例に直接関連していないコードの一部が省略されている。</li> <li>■ コードの一部を繰り返すことができる。</li> </ul>	CREATE TABLE ... AS <i>subquery</i> ;  SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM <i>employees</i> ;
:	垂直の省略記号は、列に直接関連しない複数の行が省略されていることを示します。	
その他の句読点	大カッコ、中カッコ、縦線および省略記号以外の句読点は、表示されているとおりに入力する必要があります。	acctbal NUMBER(11,2); acct        CONSTANT NUMBER(4) := 3;
イタリック体	イタリック体は、特定の値を指定する必要があるプレースホルダや変数を示します。	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
大文字	大文字は、システムが提供する要素を示します。これらの用語は、ユーザー定義の用語と区別するために大文字で示されます。用語が大カッコ内にかぎりで表示されているとおりの順序および綴りで入力します。ただし、これらの用語は大 / 小文字が区別されないため、小文字でも入力できます。	SELECT last_name, employee_id FROM <i>employees</i> ;  SELECT * FROM USER_TABLES;  DROP TABLE hr.employees;
小文字	小文字は、ユーザー定義のプログラム要素を示します。たとえば、表名、列名、ファイル名などです。  <b>注意:</b> 大文字と小文字を組み合わせて使用するプログラム要素もあります。これらの要素は、記載されているとおりに入力してください。	SELECT last_name, employee_id FROM <i>employees</i> ;  <i>sqlplus hr/your_password</i>  CREATE USER mjones IDENTIFIED BY <i>ty3MU9</i> ;

## Windows オペレーティング・システム用の表記規則

次の表に、Windows オペレーティング・システム用の表記規則を説明し、その使用例を示します。

規則	意味	例
「スタート」>	プログラムの起動方法	Oracle Database Configuration Assistant を起動するには、「スタート」>「プログラム」>「Oracle - HOME_NAME」>「Configuration and Migration Tools」>「Database Configuration Assistant」を選択します。
ファイル名およびディレクトリ名	ファイル名およびディレクトリ名は、大 / 小文字が識別されません。特殊文字のうち、左山カッコ (<)、右山カッコ (>)、コロン (:)、二重引用符 (")、スラッシュ (/)、縦線 ( )、ダッシュ (-) は使用できません。特殊文字のうち円記号 (¥) は、引用符で囲まれている場合でも要素セパレータとして処理されます。Windows では、ファイル名が ¥¥ で始まっている場合、汎用ネーミング規則が使用されていることとなります。	c:¥winnt"¥"system32 は、 C:¥WINNT¥SYSTEM32 と同じです。
C:¥>	現行ハード・ディスク・ドライブの Windows コマンド・プロンプトを表します。コマンド・プロンプトのエスケープ文字はカレット (^) です。プロンプトは作業中のサブディレクトリを示します。このマニュアルでは、コマンド・プロンプトと呼びます。	C:¥oracle¥oradata>
	Windows コマンド・プロンプトの特殊文字のうち二重引用符 (") には、エスケープ文字として、特殊文字の円記号 (¥) が必要な場合があります。カッコおよび一重引用符 (') には、エスケープ文字は必要ありません。エスケープ文字および特殊文字の詳細は、ご使用の Windows オペレーティング・システムのマニュアルを参照してください。	C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥"  C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Oracle ホーム名を表します。ホーム名には、英数字で 16 文字まで使用できます。ホーム名に使用可能な特殊文字はアンダースコアのみです。	C:¥> net start OracleHOME_NAME_TNSListener

規則	意味	例
ORACLE_HOME および ORACLE_BASE	<p>Oracle8 リリース 8.0 以下のリリースでは、Oracle コンポーネントをインストールすると、すべてのサブディレクトリが最上位の ORACLE_HOME ディレクトリに配置されました。この ORACLE_HOME ディレクトリには、デフォルトで次のいずれかの名前が使用されました。</p> <ul style="list-style-type: none"> <li>■ Windows NT の場合 : C:\%orant</li> <li>■ Windows 98 の場合 : C:\%orawin98</li> </ul> <p>今回のリリースは、Optimal Flexible Architecture (OFA) のガイドラインに準拠しています。最上位の ORACLE_HOME ディレクトリに配置されないサブディレクトリもあります。ORACLE_BASE と呼ばれる最上位ディレクトリがあります。このディレクトリは、デフォルトで C:\%oracle となります。他の Oracle ソフトウェアがインストールされていないコンピュータに Oracle9i リリース 1 (9.0.1) をインストールする場合、最初の Oracle ホーム・ディレクトリのデフォルト設定は、C:\%oracle\ora90 となります。Oracle ホーム・ディレクトリは、ORACLE_BASE の直下に配置されます。</p> <p>このマニュアルに示すすべてのディレクトリ・パスの例は、OFA 表記規則に従っています。</p>	<p>%ORACLE_HOME%\rdbms\admin ディレクトリに移動します。</p>

---

# SQL\*Plus の新機能

ここでは、SQL\*Plus リリース 9.2 の新機能について説明し、追加情報の参照先を示します。現行リリースへ移行するユーザーのために、前リリースの新機能についても説明します。

ここでは、次の項目について説明します。

- [SQL\\*Plus リリース 9.2 の新機能](#)
- [iSQL\\*Plus リリース 9.2 の新機能](#)
- [iSQL\\*Plus サーバー リリース 9.0.1 からリリース 9.2 へのアップグレード](#)
- [SQL\\*Plus リリース 9.0.1 の新機能](#)

# SQL\*Plus リリース 9.2 の新機能

SQL\*Plus リリース 9.2 は、SQL\*Plus リリース 9.0.1 のスーパーセットです。ここでは、今回のリリースで導入された新機能について説明します。

## XMLType サポート

SQL\*Plus では、XMLType のサポートが追加されました。DESCRIBE コマンドを使用すると、XMLType 列の詳細、ファンクションおよびプロシージャにおける XMLType の使用方法の詳細を表示できます。取得された XMLType 列の最大サイズは、SET LONG によって決定され、SQL\*Plus で XMLType データの取得に使用される増分サイズまたはチャンク・サイズは、SET LONGCHUNKSIZE によって決定されます。

**参照：** 次の項を参照してください。

- 13-52 ページの「DESCRIBE」
- 13-108 ページの「SET LONG {80|n}」
- 13-108 ページの「SET LONGC[HUNKSIZE] {80|n}」
- 7-5 ページの「データ型の書式設定」

## SQLPLUS -L 引数

SQLPLUS コマンドに、コマンドライン引数 SQLPLUS-L が新しく追加されました。この引数を使用すると、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の入力を求めるプロンプトを再度表示しないように設定できます。この機能は、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、サーバーが実行されていないときに、接続の詳細の入力を求めるプロンプトが再表示されないようにする場合にも有効です。この機能は、Graphical User Interface (GUI) 版の SQL\*Plus for Windows では使用できません。

**参照：** 13-2 ページの「SQL\*Plus コマンド一覧」を参照してください。

## \_CONNECT\_IDENTIFIER DEFINE

DEFINE 変数 \_CONNECT\_IDENTIFIER が新しく追加されました。この変数には、使用可能な場所への接続のためにユーザーによって指定された SID が含まれています。この機能によって、他の DEFINE 変数と同様に、接続情報にアクセスできます。たとえば、TTITLE または「&」置換変数で使用したり、SET SQLPROMPT コマンドを使用して SQL\*Plus コマンドライン・プロンプトとしても使用できます。たとえば、ユーザー名および SID を表示するように SQL\*Plus プロンプトを変更するには、次のとおり入力します。

```
SET SQLPROMPT '&_CONNECT_IDENTIFIER > '
```

DEFINE コマンドを使用して、他の DEFINE 定義を表示する場合と同じ方法で `_CONNECT_IDENTIFIER` 定義を表示できます。引数が指定されていない DEFINE コマンドまたは表示する特定の引数が指定されている DEFINE コマンドを使用します。たとえば、次のように指定します。

```
DEFINE
```

または、次のように指定します。

```
DEFINE _CONNECT_IDENTIFIER
```

UNDEFINE コマンドを使用すると、`_CONNECT_IDENTIFIER` 定義を削除し、使用不可にします。

**参照：** 6-16 ページの「[ユーザー変数の定義](#)」を参照してください。

### @URL サポート

@URL 構文はすべてのプラットフォームでサポートされます。「@URL」を使用すると、Windows のみでなく、すべてのプラットフォームでスクリプトを実行できます。

**参照：** 第 13 章「[SQL\\*Plus コマンド・リファレンス](#)」の「[@ \(アットマーク\)](#)」、「[@@ \(二重アットマーク\)](#)」および「[START](#)」を参照してください。

## iSQL\*Plus リリース 9.2 の新機能

iSQL\*Plus ユーザー・インタフェースの機能およびロック・アンド・フィールドが変更されています。

### UNIX サポート

UNIX プラットフォームでは、iSQL\*Plus サーバーがサポートされます。

### システム変数の設定

iSQL\*Plus には、「System Variables」画面が新しく追加されました。この画面には、iSQL\*Plus へ適用可能なシステム変数に対するグラフィカル・インタフェースが提供されています。この機能によって、現行の設定を表示したり、1 画面からの 1 つ以上の設定を変更することができます。新しい iSQL\*Plus セッションを起動すると、必ずデフォルトのシステム変数が使用されます。必要な場合は、「Work Screen」画面で SET コマンドを使用することもできます。

**参照：** 5-11 ページの「データベースへの変更の自動保存」を参照してください。

### 「History」画面

iSQL\*Plus には、「History」画面が新しく追加されました。この画面では、同一セッションですでに実行された 1 つ以上のスクリプトを再ロードできます。

スクリプトを実行すると、履歴エントリが作成されます。表示されるエントリの数は、「Preferences」画面で定義できます。iSQL\*Plus の各セッションの履歴リストは、セッション終了時に消去されます。

**参照：** 2-6 ページの「iSQL\*Plus の「History」画面」を参照してください。

### 新しいセッション

iSQL\*Plus ユーザー・インターフェースには、「New Session」というアイコンが追加されました。「New Session」アイコンを使用すると、Web ブラウザ・ウィンドウで新しい iSQL\*Plus セッションを起動できます。新しいセッションへは、同一ユーザーまたは別の有効なユーザーとしてログインできます。

**参照：** 2-3 ページの「iSQL\*Plus ナビゲーション」を参照してください。

### スクリプトの取消し

iSQL\*Plus の「Work Screen」画面には、「Cancel」ボタンが新しく追加されました。「Cancel」ボタンを使用すると、iSQL\*Plus セッションで現在実行しているスクリプトを中断できます。

**参照：** 2-5 ページの「iSQL\*Plus の「Work Screen」画面」を参照してください。

### iSQL\*Plus DBA のログイン

iSQL\*Plus DBA の「Login」画面には、「Privilege」ドロップダウン・リストが含まれ、そのリストから SYSDBA または SYSOPER 権限でのログインを選択して、システム管理作業を実行します。このフィールドは、標準の iSQL\*Plus の「Login」画面では使用できません。iSQL\*Plus DBA の URL を使用して DBA の「Login」画面にアクセスします。

DBA 権限でログインすると、iSQL\*Plus の画面に「iSQL\*Plus DBA」ロゴが表示され、DBA 権限による管理セッションと標準の iSQL\*Plus セッションを明確に区別できます。

**参照：** 2-10 ページの「iSQL\*Plus DBA の「Login」画面」を参照してください。

## **iSQL\*Plus Extension for Windows**

iSQL\*Plus Extension for Windows では、iSQL\*Plus でローカル SQL スクリプトをロードまたは実行するためのコンテキスト・メニューが提供されています。iSQL\*Plus Extension for Windows は、.SQL ファイルで機能します。アクティブにするには、Windows エクスプローラで .SQL ファイルを右クリックします。

iSQL\*Plus サーバーを構成すると、コンテキスト・メニューを使用して、次のことができます。

- ワークステーションからアクセス可能なローカル・スクリプトの実行またはロード
- 使用するサーバーを、構成済 iSQL\*Plus サーバーのリストから選択
- ダブルクリックで、.SQL ファイルに関連付けられるファイルを選択
- 使用する言語を、構成済言語のリストから選択

## **iSQL\*Plus Extension for Windows での言語サポート**

Windows XP、Windows 2000 および Windows NT の場合、iSQL\*Plus Extension for Windows では次の言語がサポートされます。

- ポルトガル語（ブラジル）
- 英語
- フランス語 / フランス語（カナダ）
- ドイツ語
- イタリア語
- 日本語
- 韓国語
- 簡体字中国語
- スペイン語 / スペイン語（南米）
- 繁体字中国語

Windows 98 の場合、iSQL\*Plus Extension for Windows では英語のみ使用可能です。

**参照：** 2-15 ページの「[iSQL\\*Plus Extension for Windows](#)」を参照してください。

# iSQL\*Plus サーバー リリース 9.0.1 からリリース 9.2 へのアップグレード

iSQL\*Plus は、リリース 9.0.1 とリリース 9.2 の間で一部変更されています。

## iSQL\*Plus の URL

デフォルトの Oracle HTTP Server のインストールでは、ポート番号を指定します。インストール時にポート番号が記録されなかった場合は、`httpd.conf` の `Port` および `Listen` ディレクティブを確認します。Windows の場合、`httpd.conf` は、`%ORACLE_HOME%\Apache\Apache\conf` にあります。UNIX の場合、`httpd.conf` は、UNIX 上の `$ORACLE_HOME/Apache/Apache/bin/conf` にあります。

iSQL\*Plus リリース 9.2 の URL は次のとおりです。

`http://machine_name.domain:port/isqlplus`

iSQL\*Plus へのすべての HTML リンクが、新しい URL へ更新されているかどうかを確認する必要があります。

## iSQL\*Plus の構成ファイル

iSQL\*Plus の構成ファイル `isqlplus.conf` の形式は変更されています。ユーザーが行うすべてのカスタマイズは、手動で、デフォルトの iSQL\*Plus リリース 9.2 の構成ファイルにマージできます。Windows の場合、`isqlplus.conf` は、`%ORACLE_HOME%\sqlplus\admin` にあります。UNIX の場合、`isqlplus.conf` は、`$ORACLE_HOME/sqlplus/admin` にあります。

## iSQL\*Plus リリース 9.0.1 のパラメータ

iSQL\*Plus リリース 9.0.1 では、構成できるパラメータは次の 2 つでした。

- `iSQLPlusTimeOut`: このパラメータは削除されました。現在、期限切れは常に有効です。
- `iSQLPlusTimeOutInterval`: 新しいデフォルト値は 30 分です。現在、このディレクティブは、`FastCgiServer` ディレクティブに対するパラメータです。

## iSQL\*Plus リリース 9.2 のパラメータ

iSQL\*Plus リリース 9.2 のサイズ指定およびチューニング用に、新しくパラメータが追加されました。isqlplus.conf 構成ファイルを直接編集して、次の iSQL\*Plus サーバーのパラメータ設定を変更できます。

- iSQLPlusNumberOfThreads
- iSQLPlusLogLevel
- iSQLPlusTimeOutInterval
- iSQLPlusHashTableSize
- iSQLPlusConnectIdList
- -idle-timeout

これらのパラメータの詳細は、3-7 ページの「[中間層の構成 : iSQL\\*Plus サーバー](#)」を参照してください。

## セキュリティ

htpasswd ユーティリティを使用して、iSQL\*Plus リリース 9.2 のパスワード・ファイル ipplusdba.pw に、DBA 接続用の HTTP Server 認証ユーザー名を追加する必要があります。

リリース 9.0.1 でユーザー接続用の HTTP 認証を有効にした場合、新しい isqlplus.conf 形式が後に続いているかどうかを確認し、指定したパスワード・ファイルにユーザーを追加します。

iSQLPlusConnectIdList を有効にして、データベース・サーバーの指定したリストのみへのアクセスを可能にします。

詳細は、[第 10 章「SQL\\*Plus セキュリティ](#)」を参照してください。

## スタイルシート

スタイルシート ipplus.css は変更されています。iSQL\*Plus リリース 9.2 のカスタマイズ基準として、デフォルトのリリース 9.2 のスタイルシートを使用することをお勧めします。

# SQL\*Plus リリース 9.0.1 の新機能

## SQLPLUS コマンドライン・スイッチ

バージョンおよび使用方法のコマンドライン・スイッチは、次のように標準化されています。

```
-V [VERSION]
-H [HELP]
```

無効なオプションを使用すると、-HELP に相当する使用方法に関するメッセージが表示されます。「-」および「-?」オプションは廃止されました。

## iSQL\*Plus

iSQL\*Plus は、SQL\*Plus に対するブラウザ・ベースのインタフェースです。iSQL\*Plus は、このリリースでは、Microsoft Windows オペレーティング・システムのみで使用可能です。

## START、@、@@

URL から SQL スクリプトを読み込み、実行する新しいコマンドが追加されています。URL からスクリプトを読み込み、実行する機能は、@、@@ および START コマンドで使用可能です。次に構文を示します。

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

「url」には、指定した Web サーバーで実行するスクリプトを指定します。SQL\*Plus は、HTTP、FTP および Gopher プロトコルをサポートします。通常の方法でスクリプトに変数値を渡します。次に例を示します。

```
http://machine_name.domain:port/script.sql value1 value2
```

**注意：**この機能は、Microsoft Windows のみで使用可能です。

## SET APPINFO

SET APPINFO のデフォルトは OFF です。これを使用すると、チューニングが不完全なデータベースにおいて、実行中の処理を DBA が監視できます。SET APPINFO ON を glogin.sql に追加すると、以前のリリースの動作に戻すことができます。

## **SET SQLPLUSCOMPATIBILITY**

新しい SET コマンドとして、SET SQLPLUSCOMPAT[IBILITY] {x.y[z]} が追加されました。VARIABLE コマンドの動作を SQLPLUSCOMPATIBILITY で指定したリリースの動作に設定します。

SQL\*Plus の将来のリリースでは、SQLPLUSCOMPATIBILITY は VARIABLE 以外の機能もサポートする予定です。SQLPLUSCOMPATIBILITY コマンドは、他のリリースの動作との互換のために使用されます。

## **HR サンプル・スキーマ**

SQL\*Plus ドキュメントでは、すべての例に対してサンプル・スキーマの新しいセットを使用しています。サンプル・スキーマのアクセス、実装および使用の詳細は、『Oracle9i サンプル・スキーマ』を参照してください。



---

# SQL\*Plus クイック・スタート

ここでは、インストール後、SQL\*Plus を起動および実行する手順について説明します。

- [iSQL\\*Plus クイック・スタート](#)
- [Windows 用の SQL\\*Plus コマンドライン・クイック・スタート](#)
- [UNIX 用の SQL\\*Plus コマンドライン・クイック・スタート](#)

## iSQL\*Plus クイック・スタート

iSQL\*Plus を起動する手順は、次のとおりです。

1. インターネットまたはイントラネットに接続し、Web ブラウザを起動します。
2. Web ブラウザの「場所」または「アドレス」フィールドに、Oracle HTTP Server の URL に続けて `isqlplus` を入力します。iSQL\*Plus の URL は、次のようになります。

```
http://machine_name.domain:port/isqlplus
```

Oracle HTTP Server の URL が不明な場合は、システム管理者に問い合わせてください。

3. [Enter] キーを押して、その URL に移動します。iSQL\*Plus の「Login」画面が Web ブラウザに表示されます。
4. Oracle9i のユーザー名およびパスワードを、「Username」および「Password」フィールドに入力します。Oracle9i 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。
5. デフォルトのデータベースに接続するには、「Connection Identifier」フィールドを空のままにします。デフォルト以外のデータベースに接続するには、Oracle Net データベース別名を入力します。データベース・アクセス制限が構成されている場合、「Connection Identifier」フィールドには、使用可能なデータベースを選択できるドロップダウン・リストが表示されます。
6. 「Login」をクリックして、データベースに接続します。iSQL\*Plus の「Work Screen」画面が Web ブラウザに表示されます。

これで、「Work Screen」画面の入力領域で、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。

## Windows 用の SQL\*Plus コマンドライン・クイック・スタート

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は、次のとおりです。

1. Windows のコマンド・プロンプトを開きます。
2. コマンドライン・プロンプトで、次の形式で SQL\*Plus コマンドを入力します。

```
c:¥> sqlplus
```
3. プロンプトが表示されたら、Oracle9i のユーザー名およびパスワードを入力します。Oracle9i 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。
4. SQL\*Plus が起動し、デフォルト・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。

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は、次のとおりです。

1. SQL\*Plus を起動し、デフォルト・データベースに接続します。
2. CONNECT コマンドを使用して、接続するデータベースの Oracle Net データベース別名 (`@connect_identifier`) を指定します。CONNECT コマンドは、次の形式で入力します。

```
SQL> connect username/password@connect_identifier
```

パスワードを非表示にするには、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。

3. または、デフォルト以外のデータベースに接続された SQL\*Plus を起動し、次の形式で SQL\*Plus コマンドを入力します。

```
c:¥> sqlplus username/password@connect_identifier
```

パスワードを非表示にするには、次の形式で SQL\*Plus コマンドを入力します。

```
c:¥> sqlplus username@connect_identifier
```

パスワードを入力するように求められます。

## UNIX 用の SQL\*Plus コマンドライン・クイック・スタート

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は、次のとおりです。

1. UNIX 端末を起動します。
2. コマンドライン・プロンプトで、次の形式で SQL\*Plus コマンドを入力します。

```
$> sqlplus
```

3. プロンプトが表示されたら、Oracle9i のユーザー名およびパスワードを入力します。Oracle9i 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。
4. SQL\*Plus が起動し、デフォルト・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。

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は、次のとおりです。

1. SQL\*Plus を起動し、デフォルト・データベースに接続します。
2. CONNECT コマンドを使用して、接続するデータベースの Oracle Net データベース別名 (*@connect\_identifier*) を指定します。CONNECT コマンドは、次の形式で入力します。

```
SQL> connect username/password@connect_identifier
```

パスワードを非表示にするには、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。

3. または、デフォルト以外のデータベースに接続された SQL\*Plus を起動し、次の形式で SQL\*Plus コマンドを入力します。

```
$> sqlplus username/password@connect_identifier
```

パスワードを非表示にするには、次の形式で 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。

# 第1部

---

## SQL\*Plus スタート・ガイド

ここでは、SQL\*Plus の起動に必要な情報を示します。SQL\*Plus の概要、コマンドライン・インタフェースおよび *i*SQL\*Plus 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。

この部の内容は、次のとおりです。

- [SQL\\*Plus 概要](#)
- [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 概要

この章では、SQL\*Plus の概要を紹介します。ここでは、次の項目について説明します。

- [SQL\\*Plus について](#)
- [SQL\\*Plus のインストール](#)
- [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](#)
- [Oracle9i のサンプル表および SQL\\*Plus](#)

## SQL\*Plus について

SQL\*Plus は、Oracle Server または Client 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・インタフェースや、最近では iSQL\*Plus と呼ばれる Web ベース・ユーザー・インタフェースで使用できます。

Windows の場合、右クリックで表示されるコンテキスト・メニュー・オプションによって、ローカル・ファイルを iSQL\*Plus ユーザー・インタフェースに送ることができます。

SQL\*Plus には専用のコマンドおよび環境があり、Oracle RDBMS 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。

- SQL\*Plus コマンドの入力による SQL\*Plus 環境の構成
- SQL コマンドおよび PL/SQL ブロックの入力、編集、格納、取得および実行
- 問合せ結果の書式設定、計算の実行、格納および印刷
- エンド・ユーザーとの対話
- データベースの起動および停止
- データベースへの接続
- 変数の定義
- エラーの取得
- 表の列定義の表示
- データベースの管理

SQL\*Plus を使用して、対話形式でのレポート生成およびバッチ・プロセスでのレポート生成ができます。また、テキスト・ファイル、画面またはインターネットでの参照用の HTML ファイルに生成結果を出力することもできます。Web ページからスクリプトを実行するために、SQL\*Plus の HTML 出力機能とサーバー側の CGI スクリプトを組み合わせ使用したり、iSQL\*Plus の動的レポート機能を使用して、動的にレポートを生成できます。

たとえば、iSQL\*Plus を使用してデータベース・オブジェクトを表示するには、次のように入力します。



```
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)

データベースから関連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、列ヘッダーの名前が変更され、次の出力が表示されます。

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

## SQL\*Plus コマンドラインのアーキテクチャ

SQL\*Plus では、次の層で構成される 2 層モデルが使用されます。

- クライアント層 (コマンドライン・ユーザー・インタフェース)
- データベース層 (Oracle9i)

2つの層は、同じマシン上に存在する必要はありません。

### コマンドライン・ユーザー・インタフェース

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。

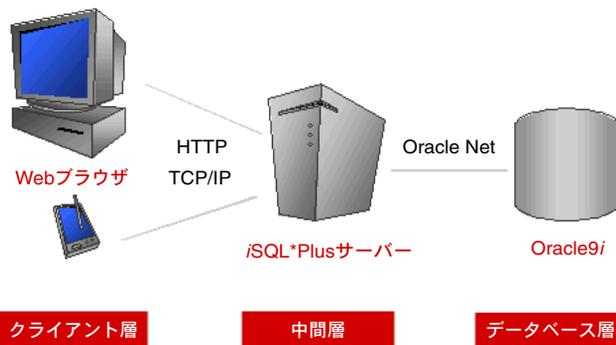
### Oracle9i

Oracle Net コンポーネントによって、SQL\*Plus クライアントと Oracle9i の間の通信が可能です。

## iSQL\*Plus アーキテクチャ

iSQL\*Plus は、ブラウザベースの高速インタフェースです。iSQL\*Plus では、次の層で構成される 3 層モデルで SQL\*Plus 処理エンジンが使用されます。

- クライアント層 (iSQL\*Plus ユーザー・インタフェース、通常は Web ブラウザ)
- 中間層 (iSQL\*Plus サーバー、Oracle Net および Oracle HTTP Server)
- データベース層 (Oracle9i)



iSQL\*Plus サーバーは、Oracle HTTP Server と同じマシン上に存在する必要があります。クライアント層は、同じマシン上に存在する必要はありません。中間層は、クライアント層とデータベース層との間の対話およびリソースを調整します。データベース層は Oracle9i データベースで、Oracle Net を介してアクセスします。

## Web ブラウザ

iSQL\*Plus ユーザー・インタフェースは、インターネットまたはイントラネットに接続している Web ブラウザで実行されます。iSQL\*Plus ユーザー・インタフェースを、インストールまたは構成する必要はありません。必要な情報は、Oracle9i データベースにアクセスするための Oracle HTTP Server の URL のみです。

## iSQL\*Plus サーバー

iSQL\*Plus サーバーは、Oracle9i のインストール時に Oracle HTTP Server とともにインストールされます。

中間層には、Oracle HTTP Server および iSQL\*Plus サーバーが含まれます。iSQL\*Plus サーバーを使用すると、iSQL\*Plus ユーザー・インタフェースと Oracle9i 間の通信および認証が可能になります。iSQL\*Plus の各セッションは一意に識別されるため、Oracle9i に対して複数のセッションを同時に開くことができます。

## Oracle9i

Oracle Net コンポーネントを使用すると、Oracle9i クライアント・サーバーのインストールと同じ方法で、iSQL\*Plus サーバーと Oracle9i 間の通信が可能になります。

# SQL\*Plus のインストール

SQL\*Plus は、Oracle データベースのコンポーネントです。SQL\*Plus および iSQL\*Plus と呼ばれる Web ベースのユーザー・インタフェースは、Oracle データベース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 および SQL\*Plus には、ホスト・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 インストール・ガイドおよび管理者リファレンスを参照してください。

参照用に、Oracle インストール・ガイドおよび管理者リファレンスを準備しておいてください。

# 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 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 コマンド言語は、簡単に使用できるように設計されています。たとえば、データベース内の LAST\_NAME というラベルが付いた列に、「Family Name」という、さらに明確なヘッダーを付けるには、次のコマンドを入力します。

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

同様に、EMPLOYEES という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。

- SQL\*Plus のオンライン・ヘルプ  
コマンドラインおよび iSQL\*Plus のオンライン・ヘルプ
- このマニュアルの SQL\*Plus の使用方法の説明  
このマニュアルの第 II 部の第 5 章～第 12 章

## 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。ただし、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 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。これらのオペレーティング・システム固有のマニュアルを、このマニュアルとあわせてお読みください。

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。サンプル表については、1-6 ページの「[Oracle9i のサンプル表および SQL\\*Plus](#)」を参照してください。

このマニュアルにおけるコマンドの使用方法については、xix ページの「[コード例中の表記規則](#)」を参照してください。

## Oracle9i のサンプル表および SQL\*Plus

Oracle9i には、多くのサンプル・スキーマが含まれています。このマニュアルのチュートリアルおよび例には、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。HR サンプル・スキーマを使用すると、このマニュアルの概念および操作を理解できるようになります。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。このマニュアルの例は、その会社の人事部長になったつもりで実行してください。

サンプル表の日付は、4 桁の年号を使用しています。SQL\*Plus のデフォルトの日付書式が YY-MM-DD の場合、日付は、2 桁の年号で表示されます。日付の表示方法を制御するには、SELECT 文で、SQL の TO\_CHAR ファンクションを使用してください。

Oracle9i に含まれるサンプル・スキーマの詳細は、『Oracle9i サンプル・スキーマ』を参照してください。図 1-1 に、EMP\_DETAILS\_VIEW ビューの内容を示します。

図 1-1 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)

## 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 Oracle9i 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 ユーザー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 表およびユーザーのロックを解除する必要があります。HR 表およびユーザーのロックを解除するには、SYSTEM ユーザーとしてログインし、次のコマンドを入力します。



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

HR スキーマの HR 表およびユーザーのロック解除の詳細は、『Oracle9i サンプル・スキーマ』を参照してください。HR スキーマの HR ユーザーは、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。このマニュアルの例を試すときに、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 表をコピーする場合は、DBA に連絡するか、または『Oracle9i サンプル・スキーマ』を参照してください。また、UNIX の場合は、次のサブ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、次のサブ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. Oracle9i アカウントのユーザー名およびパスワードについては、DBA に問い合せてください。
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%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、同じ手順を実行します。

サンプル表の詳細は、『Oracle9i サンプル・スキーマ』を参照してください。

---

## 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、iSQL\*Plus ユーザー・インタフェースおよび iSQL\*Plus Extension for Windows 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。

- [SQL\\*Plus コマンドライン・ユーザー・インタフェース](#)
- [iSQL\\*Plus ユーザー・インタフェース](#)
- [iSQL\\*Plus Extension for Windows](#)

## SQL\*Plus コマンドライン・ユーザー・インタフェース

SQL\*Plus のコマンドライン・ユーザー・インタフェースを使用して、次の操作のために SQL\*Plus、SQL および PL/SQL コマンドを書き込むことができます。

- SQL コマンドおよび PL/SQL ブロックの入力、編集、格納、取得および保存
- 問合せ結果の書式設定、計算、格納および印刷
- 表の列定義の表示
- エンド・ユーザーへのメッセージの送信およびエンド・ユーザーからの応答の受信
- データベースの管理

### コマンドラインの画面

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。次に、X 端末で実行されている SQL\*Plus を示します。



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>
```

## iSQL\*Plus ユーザー・インタフェース

iSQL\*Plus は、Oracle9i に対する Web ブラウザ・ベースのインタフェースであり、SQL\*Plus 製品のコンポーネントの 1 つです。

iSQL\*Plus によって、Web ブラウザを使用して Oracle9i データベースに接続し、コマンドラインで SQL\*Plus を使用した場合と同じ作業を実行できます。

使用する Web ブラウザおよび Web ブラウザ・ウィンドウのサイズによって、iSQL\*Plus 画面の外観およびレイアウトが異なる場合があります。

## iSQL\*Plus ナビゲーション

iSQL\*Plus には、様々なナビゲート方法があります。

- グローバル・ナビゲーション・アイコンの使用
- 軌跡リンクの使用

### アイコン

グローバル・ナビゲーション・アイコンは、各画面に表示されます。アイコンには、次の 3 つの状態があります。

- 背景が白:機能が使用できる場合。
- 背景が薄茶:機能が使用できない場合。
- 背景が青:アクティブな場合、つまり、その画面にナビゲートした場合。

iSQL\*Plus の画面には、次のアイコンが表示されます (オンライン・ヘルプも含みます)。

 **Logout** iSQL\*Plus セッションからログアウトし、「Login」画面に戻ります。

 **New Session** 別の Web ブラウザ・ウィンドウで新しい iSQL\*Plus セッションを開始します。新しいセッションへは、同一ユーザーまたは別の有効なユーザーとしてログインできます。

 **History** iSQL\*Plus の「History」画面を開きます。以前に実行されたスクリプトを 1 つ以上選択し、入力領域に再ロードするか、または「History」リストから削除できます。

 **Preferences** iSQL\*Plus の「Preferences」画面が開きます。この画面では、インタフェース・オプションの設定、システム変数の設定およびパスワードの変更ができます。

 **Help** 別の Web ブラウザ・ウィンドウで iSQL\*Plus ヘルプを開きます。

- ▶ **Next** iSQL\*Plus ヘルプで次のページを開きます。
- ◀ **Previous** iSQL\*Plus ヘルプで前のページを開きます。
- 📖 **Contents** iSQL\*Plus ヘルプの目次を開きます。
- 🔍 **Index** iSQL\*Plus ヘルプの索引を開きます。

## 軌跡リンク

軌跡リンクは、すでにナビゲートしてきたパスを示すリンクです。これは、「iSQL\*Plus」ロゴの下にテキストのリンクとして表示されます。表示されている任意の画面に戻るには、リンクをクリックします。たとえば、「Preferences」アイコンを使用して「Preferences」画面にナビゲートすると、「Work Screen」画面へのリンクが次のように表示されます。

[Work Screen](#) > Preferences

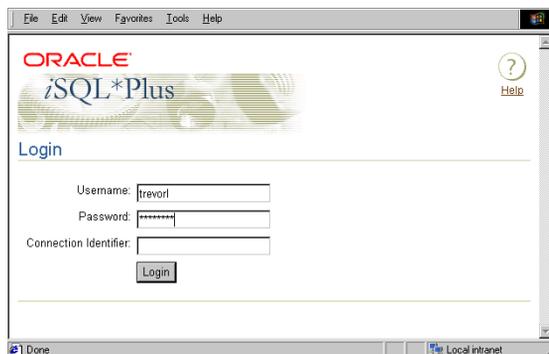
「Work Screen」リンクをクリックすると、「Work Screen」画面に戻ります。

## iSQL\*Plus の「Login」画面

「Login」画面へは、次の形式の URL で Web ブラウザから接続します。

`http://machine_name.domain:port/isqlplus`

「Login」画面が表示されます。



**Username:** 有効なユーザー名を入力して、Oracle9i に接続します。

**Password:** ユーザー名に対する有効なパスワードを入力します。

**Connection Identifier:** デフォルトの Oracle データベースを使用する（存在する場合）には、このフィールドを空のままにします。それ以外の場合は、Oracle Net 別名を入力して、接続するリモート・データベースを指定します。Oracle Net 別名を使用する場合は、iSQL\*Plus サーバーを実行しているマシン上で定義する必要があります。このマシンは、Web ブラウザを実行するマシンと異なる場合があります。

詳細な接続識別子も使用できます。たとえば、次のように入力できます。

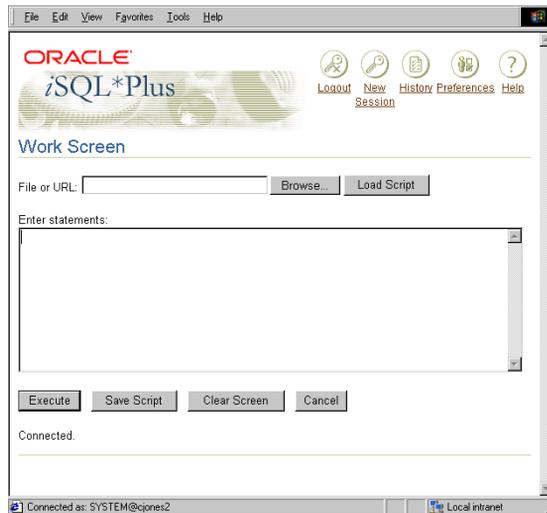
```
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=www.oracle.com)
(PORT=1521))) (CONNECT_DATA=((SERVICE_NAME=orashop.us.acme.com)))
```

iSQL\*Plus は、特定のデータベースへの接続を制限するように構成できます。データベース・アクセス制限が有効な場合、「Connection Identifier」テキスト・フィールドのかわりに、ドロップダウン・リストに使用可能なデータベースが表示されます。これによって、ホスト環境での iSQL\*Plus サーバーのセキュリティを向上させることができます。

**Login** 「Login」 ボタンをクリックすると、入力したユーザー名、パスワードおよび接続識別子を使用して iSQL\*Plus にログインできます。

## iSQL\*Plus の「Work Screen」画面

正常にログインすると、「Work Screen」画面が表示されます。



**File or URL:** 入力領域にロードして編集または実行するファイルのパスと名前または URL を入力します。

**Browse...** 「Browse...」 ボタンをクリックすると、ロードして編集または実行するスクリプト・ファイルを検索できます。ファイルを選択すると、そのパスと名前が「File or URL:」フィールドに入力されます。

**Load Script** 「Load Script」 ボタンをクリックすると、「File or URL:」フィールドで指定したスクリプトを iSQL\*Plus の入力領域にロードして編集または実行できます。

**Enter statements:** SQL 文、PL/SQL ブロックまたは iSQL\*Plus コマンドを入力します。この領域は、入力領域とも呼ばれます。「Preferences」画面からアクセスする「Interface Options」画面で、入力領域のサイズを変更できます。

**Execute** 「Execute」 ボタンをクリックすると、入力領域の内容を実行できます。設定項目の設定に応じて、実行結果が「Work Screen」画面または新しい Web ブラウザ・ウィンドウに表示されるか、あるいはファイルに保存されます。

**Save Script** 「Save Script」 ボタンをクリックすると、入力領域の内容をファイルに保存できます。ファイル名の入力を求めるプロンプトが表示されます。任意のファイル拡張子を選択できます。拡張子を .SQL にすると、スクリプトの識別に有効です。

**Clear Screen** 「Clear Screen」 ボタンをクリックすると、入力領域内の文、および表示されている出力をすべて消去できます。

**Cancel** 「Cancel」 ボタンをクリックすると、現在実行中のスクリプトを中断できます。

## iSQL\*Plus の「History」画面

「History」アイコンをクリックすると、「History」画面が開きます。「History」画面では、同一セッションですでに実行されたスクリプトを再ロードできます。

最後に実行されたスクリプトとは異なるスクリプトを「Work Screen」画面で実行すると、履歴エントリが作成されます。「History」画面には、スクリプトの先頭から 80 文字が表示されます。

「History」画面からスクリプトをロードすると、そのスクリプトは履歴リストの最上部に移ります。履歴リストが一杯になると、古いスクリプトは削除されます。セッションを終了すると、履歴は失われます。つまり、セッション間では履歴は共有されません。

履歴リストに保存されるデフォルトのエントリ数は、「Preferences」画面からアクセスする「Interface Options」画面で変更できます。



**Script** 履歴内のスクリプトの現行のリストを表示します。スクリプトは、最後に実行されたものから順に表示されます。最後に実行されたスクリプトがリストの最上部に表示されません。スクリプトのテキストをクリックすると、入力領域にロードされます。

スクリプトは逐語的に（入力したとおりに）表示されるため、パスワードを含む CONNECT コマンドのような項目が含まれているスクリプトには注意してください。

**Load** 選択したスクリプトを、「Work Screen」画面の入力領域にロードします。

**Delete** 選択したスクリプトを履歴から削除します。

**Cancel** 変更またはスクリプトのロードを履歴から実行せずに、「History」画面を閉じます。

## iSQL\*Plus の「Preferences」画面

「Preferences」画面では、インタフェース・オプションの設定、システム変数の設定およびパスワードの変更ができます。



**Set interface options** 「Set interface options」または隣りにある「Go」ボタンをクリックすると、「Interface Options」画面にアクセスできます。「Interface Options」画面は、次の項目を設定する場合に使用します。

- 入力領域のサイズ
- 「Work Screen」、ファイルまたは別の Web ブラウザ・ウィンドウから出力先を選択
- 履歴のサイズ

[「iSQL\\*Plus ユーザー・インタフェース・オプション」](#)を参照してください。

**Set system variables** 「Set system variables」または隣りにある「Go」ボタンをクリックすると、「System Variables」画面にアクセスできます。「System Variables」画面は、iSQL\*Plus のシステム変数を設定する場合に使用します。[「データベースへの変更の自動保存」](#)を参照してください。

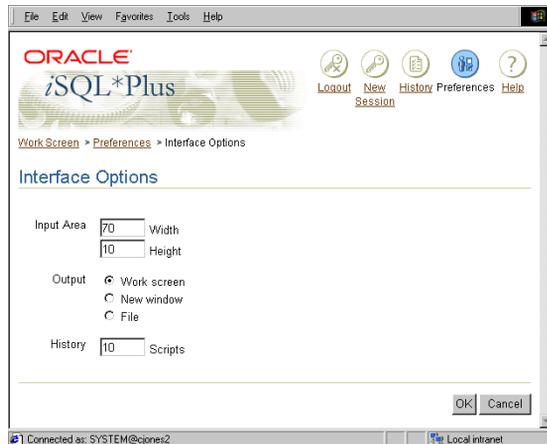
**Change your password** 「Change your password」または隣りにある「Go」ボタンをクリックすると、「Change Password」画面にアクセスできます。4-18 ページの [「iSQL\\*Plus でのパスワードの変更」](#)を参照してください。

**Go** リンクまたは「Go」ボタンをクリックすると、関連付けられた画面にナビゲートできます。

**OK/Cancel** 「OK」または「Cancel」ボタンをクリックすると、iSQL\*Plus の「Work Screen」画面に戻ります。

## iSQL\*Plus ユーザー・インタフェース・オプション

「Interface Options」画面では、入力領域のサイズ、スクリプトの結果の出力場所および履歴に保持するスクリプトの数を設定できます。「Interface Options」画面には、「Preferences」画面からアクセスします。



**Input Area** 入力領域の幅（文字数）および高さ（行数）を入力します。デフォルトの幅は70文字、高さは10行です。

**Output** セッションから生成されたスクリプト結果の出力先を、次の3つのオプションから選択します。

- **Work screen:** 入力領域の内容が実行されると、結果が画面の入力領域の下に表示されません。このオプションがデフォルトです。
- **New window:** 入力領域の内容が実行されると、新しく別の Web ブラウザ・ウィンドウが開き、結果が表示されます。
- **File:** 入力領域の内容が実行されると、結果がファイルに保存されます。ファイル名の入力を求めるプロンプトが表示されます。結果は HTML 形式で出力されるため、保存する出力ファイルの拡張子は .HTM または .HTML にすると便利です。

「Output」の設定は、新しいセッションが開始されると、常にデフォルトの「Work screen」に戻ります。

**History** 履歴に格納するスクリプトの数を入力します。設定可能な最小値は0、最大値は100、デフォルトは10です。無効な値を設定した場合は、前回の有効値が使用されます。

## iSQL\*Plus DBA の「Login」画面

SYSDBA または SYSOPER 権限で iSQL\*Plus にログインすると、iSQL\*Plus を介してデータベースの管理および DBA コマンドの実行ができます。この場合、Oracle HTTP Server で認証され、Oracle インスタンスからの Oracle SYSDBA または SYSOPER 権限を持つ必要があります。

SYSDBA 権限または SYSOPER 権限で接続するには、ユーザー名およびパスワードを Oracle HTTP Server 認証ファイルに追加する必要があります。10-11 ページの「DBA アクセスの有効化」を参照してください。

HTTP ネットワークのタイムアウトが発生する可能性があるため、DBA 操作を長時間実行する場合は、SQL\*Plus コマンドラインを使用することをお勧めします。

SYSDBA または SYSOPER 権限でログインするには、Web ブラウザの「場所」または「アドレス」フィールドに、iSQL\*Plus DBA の URL を入力する必要があります。iSQL\*Plus DBA の URL 形式は、次のとおりです。

```
http://machine_name.domain:port/islplusdba
```

次のダイアログ・ボックスが表示され、Oracle HTTP Server 認証のユーザー名およびパスワードの入力を要求されます。



**User Name** 有効な Oracle HTTP Server 認証のユーザー名を入力します。このユーザー名は、Oracle9i データベースのユーザー名と同じにする必要はありません。

**Password** ユーザー名に対する有効な Oracle HTTP Server 認証のパスワードを入力します。このパスワードは、Oracle9i データベースのパスワードと同じにする必要はありません。

正常に認証されると、iSQL\*Plus DBA の「Login」画面が表示されます。



**Username:** 有効なユーザー名を入力して、Oracle9i に接続します。

**Password:** ユーザー名に対する有効なパスワードを入力します。パスワードは表示されません。「Password」フィールドに入力した文字は、アスタリスクで表示されます。

**Connection Identifier:** デフォルトの Oracle データベースを使用する（存在する場合は）、このフィールドを空のままにします。それ以外の場合は、Oracle Net 接続識別子を入力して、接続するリモート・データベースを指定します。詳細な接続識別子も使用できます。たとえば、次のように入力できます。

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=www.oracle.com)
(PORT=1521))) (CONNECT_DATA=((SERVICE_NAME=orashop.us.acme.com)))
```

オプションで、(SERVICE\_NAME=name) 句のかわりに、(INSTANCE\_NAME=instance) 句を使用できます。Oracle8 以下のデータベースに接続する場合は、(SID=name) 句を使用します。また、Oracle Net 別名も使用できます。Oracle Net 別名を使用する場合は、iSQL\*Plus サーバーを実行しているマシン上で指定する必要があります。このマシンは、Web ブラウザを実行するマシンと異なる場合があります。

iSQL\*Plus は、特定のデータベースへの接続を制限するように構成できます。データベース・アクセス制限が有効な場合、「Connection Identifier」テキスト・フィールドのかわりに、ドロップダウン・リストに使用可能なデータベースが表示されます。これによって、ホスト環境での iSQL\*Plus サーバーのセキュリティを向上させることができます。

**Privilege:** 「Privilege」 ドロップダウン・リストには、次の2つのオプションが含まれています。

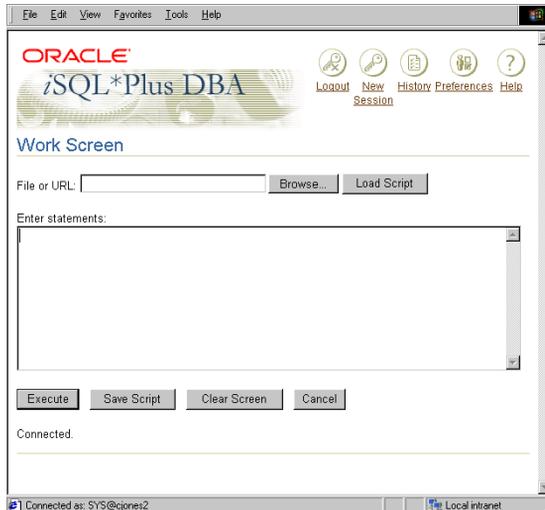
- **AS SYSDBA:** 指定されたデータベースに SYSDBA 権限で接続します。
- **AS SYSOPER:** 指定されたデータベースに SYSOPER 権限で接続します。

**Login** 「Login」 ボタンをクリックすると、入力したユーザー名、パスワード、接続識別子および DBA 権限で iSQL\*Plus にログインできます。

## iSQL\*Plus DBA の「Work Screen」画面

SYSDBA 権限または SYSOPER 権限でログインすると、特権付き接続であることを確認する iSQL\*Plus DBA の「Work Screen」画面が表示されます。

「Work Screen」画面に入力する場合と同様に入力します。



## iSQL\*Plus サーバーの統計

iSQL\*Plus サーバーの統計では、iSQL\*Plus セッションについての動的な情報および静的な環境情報が提供されます。動的な情報には、iSQL\*Plus サーバーを最大限に活用するための有効なフィードバックなどがあります。iSQL\*Plus サーバーのチューニングの詳細は、9-13 ページの「アクティブ統計の解釈」を参照してください。iSQL\*Plus サーバーの統計を含むレポートを出力するには、次のとおり入力します。

```
http://machine_name.domain:port/isqlplusdba?statistics={active|full}
[&refresh=number]
```

*machine\_name.domain* には、iSQL\*Plus サーバーの統計を生成する Oracle HTTP Server の URL を指定します。

*port* には、iSQL\*Plus サーバーで使用されるポート番号を指定します。

?statistics={active|full} では、レポートの詳細レベルを指定します。

指定する内容は次のとおりです。

full を指定すると、収集可能なすべての統計がレポートされます。これがデフォルトです。

active を指定すると、iSQL\*Plus サーバーで動的に変更されるセッション統計がレポートされます。この統計は、完全 (full) レポートの最後にも追加されます。

[&refresh=number] には、統計レポートが自動リフレッシュされるまでの時間を秒単位で指定します (オプション)。最小値は 10 秒です。

iSQL\*Plus DBA の URL にアクセスするには、Oracle HTTP Server 認証が必要ですが、データベースには接続しないため、Oracle9i データベースにログインする必要はありません。

iSQL\*Plus サーバーの統計レポートには、次の項目が含まれます。

## サーバーの詳細

この項には、iSQL\*Plus サーバーについての次の情報が表示されます。

- ホスト名
- ホスト・アドレス
- Oracle HTTP Server ポート
- ホスト・システム
- iSQL\*Plus サーバーの起動時刻
- iSQL\*Plus のバージョン
- iSQL\*Plus のログ・ファイル

## サーバー環境

この項には、次の iSQL\*Plus サーバー環境変数の設定が表示されます。

- ORACLE\_HOME
- ORACLE\_SID
- TNS\_ADMIN
- NLS\_LANG
- NLS\_NCHAR

## 構成パラメータ

この項には、次の iSQL\*Plus サーバー・パラメータの設定が表示されます。

- iSQLPlusNumberOfThreads
- iSQLPlusTimeOutInterval (分)
- iSQLPlusLogLevel
- iSQLPlusConnectIdList
- iSQLPlusHashTableSize

## クライアントの詳細

この項には、iSQL\*Plus クライアント (HTTP クライアント) によって使用される Web ブラウザの種類が表示されます。

## アクティブ統計

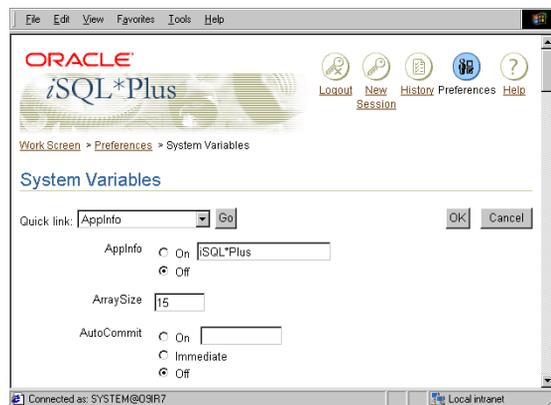
アクティブ統計の詳細は、9-11 ページの「[iSQL\\*Plus サーバーの統計](#)」を参照してください。

## iSQL\*Plus でのシステム変数の設定

「System Variables」画面で、iSQL\*Plus セッションのシステム変数を設定できます。

「System Variables」画面には、「Preferences」画面からアクセスできます。

システム変数の設定は、新しいセッションが開始されるたびに、デフォルト値に戻ります。



**Quick link:** ドロップダウン・リストから、設定するシステム変数を選択します。

**Go** 「Go」 ボタンをクリックすると、「Quick link:」で選択したシステム変数にナビゲートできます。

**システム変数およびオプション** 各システム変数には、必要に応じてチェックボックス、ラジオ・ボタンおよびテキスト入力フィールドがあります。システム変数は、アルファベット順に表示されます。

CONSTRAINT、ROLE および TRANSACTION の3つを除くシステム変数の現在の状態が表示されます。CONSTRAINT、ROLE および TRANSACTION の値は、関連付けられたテキスト・フィールドを使用して入力および設定できますが、現在の状態は表示されません。

**OK** 「OK」 ボタンをクリックすると、変更したすべてのシステム変数が設定されます。

**Cancel** 「Cancel」 ボタンをクリックすると、変更を行わずに「Preferences」画面に戻ります。

## iSQL\*Plus での SET コマンドと COLUMN コマンドの使用

iSQL\*Plus では、現在の SET コマンド変数の状態が次の実行時に継承されます。Web ブラウザの「戻る」ボタンを使用してスクリプトを再実行するか、または「History」画面からスクリプトを再実行すると、前回のスクリプトの実行後に実行されたコマンドによっては、前回と異なる結果が iSQL\*Plus に表示される場合があります。たとえば、新しい SET PAGESIZE コマンドによって、列ヘッダーが別の位置に設定される場合があります。

COLUMN、SET LINESIZE および SET WRAP コマンドは、iSQL\*Plus で使用するバッファ・サイズに影響します。スクリプト出力内の文字列に予期しない空白が含まれているか、文字列が切り捨てられている場合は、列のサイズが文字列の長さに対して十分な大きさであること、およびワード・ラップがオンに設定されていることを確認してください。

## iSQL\*Plus Extension for Windows

iSQL\*Plus Extension for Windows では、iSQL\*Plus でローカルの SQL スクリプトをロードまたは実行するためのコンテキスト・メニューが提供されています。iSQL\*Plus Extension for Windows は、.SQL ファイルで機能します。アクティブにするには、Windows エクスプローラで .SQL ファイルを右クリックします。

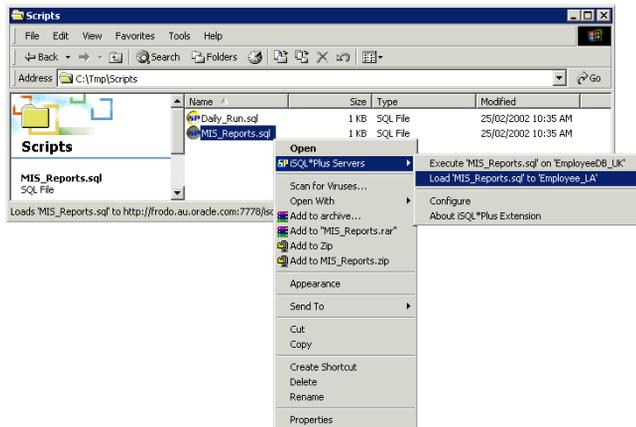
このユーティリティは、Oracle9i データベースおよび Oracle9i クライアント (Windows 版) のインストール時にインストールされます。インストール後、1つ以上の iSQL\*Plus サーバーへのアクセスを構成できます。

### コンテキスト・メニュー

iSQL\*Plus サーバーを構成すると、コンテキスト・メニューを使用して、次のことができます。

- ワークステーションからアクセス可能なローカル・スクリプトの実行またはロード
- 使用するサーバーを構成済 iSQL\*Plus サーバーのリストから選択
- ダブルクリックした際に .SQL ファイルに関連付けられるファイルを選択
- 使用する言語を構成済言語のリストから選択

次に、構成済 iSQL\*Plus サーバーでのコンテキスト・メニューの一般的な表示を示します。



ハイライト表示された iSQL\*Plus サーバーは、ブラウザで iSQL\*Plus を起動し、Employee\_LA という iSQL\*Plus サーバーの入力領域に MIS\_Reports.sql をロードするように構成されています。

iSQL\*Plus での iSQL\*Plus サーバーの構成の詳細は、3-16 ページの「[iSQL\\*Plus Extension for Windows の構成](#)」を参照してください。

# 3

---

## SQL\*Plus の構成

この章では、SQL\*Plus のコマンドライン環境および iSQL\*Plus 環境の構成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。

- SQL\*Plus の構成
- iSQL\*Plus の構成
- iSQL\*Plus Extension for Windows の構成

## SQL\*Plus の構成

SQL\*Plus 環境を特定の方法（たとえば、現在の時刻を SQL\*Plus のコマンド・プロンプトの一部として表示するなど）で設定し、その設定をそれぞれのセッションで再利用することもできます。この方法は、2つのホスト・オペレーティング・システム・ファイル（サイト全体の設定用のサイト・プロファイルおよびユーザー固有の設定用のユーザー・プロファイル）を使用して実行できます。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。正確な名前については、ご使用のオペレーティング・システム固有の Oracle インストール・ガイドおよび管理者リファレンスを参照してください。

### サイト・プロファイル

SQL\*Plus では、グローバル・サイト・プロファイル（データベース管理者が作成した SQL\*Plus スクリプト）がサポートされます。通常、このファイルは `glogin.sql` と呼ばれます。SQL\*Plus では、ユーザーが SQL\*Plus を起動して SQL\*Plus と Oracle との接続を確立すると、必ずこのスクリプトが実行されます。グローバル・サイト・プロファイルによって、DBA は、特定サイトのすべてのユーザー用に、デフォルトの SQL\*Plus 環境を設定できます。ユーザーが直接サイト・プロファイルにアクセスすることはできません。

サイト・プロファイル・ファイルは、`$ORACLE_HOME/sqlplus/admin/glogin.sql` です。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・ファイルも削除されます。

### ユーザー・プロファイル

SQL\*Plus では、サイト・プロファイルの後に実行されるユーザー・プロファイルもサポートされます。通常、このファイルは `login.sql` です。SQL\*Plus では、カレント・ディレクトリのユーザー・プロファイル、次に、環境変数 `SQLPATH` で指定されたディレクトリが検索されます。SQL\*Plus では、このコロンの区切られたディレクトリのリストが、指定した順に検索されます。

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。

ユーザー・プロファイルは、iSQL\*Plus ユーザー・インタフェースのコンテキストでは無効なため、個々の iSQL\*Plus セッションの初期設定または動作の制御には使用できません。

## 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。LOGIN ファイルには、次のようなコマンドを追加できます。

SET LINESIZE	数値を後に続けると、問合せ結果のページ幅の文字数が設定されます。
SET NUMFORMAT	数値書式 (\$99,999 など) を後に続けると、問合せ結果で数値を表示するためのデフォルト書式が設定されます。
SET PAGESIZE	数値を後に続けると、1 ページごとの行数が設定されます。
SET PAUSE	ON を後に続けると、出力の各ページの冒頭で SQL*Plus が一時停止されます ([Return] を押すと、スクロールが続行されます)。テキストを後に続けると、SQL*Plus の一時停止時に、常にそのテキストが表示されるように設定できます (PAUSE を ON に設定しておく必要もあります)。
SET SQLPROMPT	次の形式の接続情報変数を後に続けると、SQL*Plus のコマンドライン・プロンプトが変更され、接続中のデータベースの SID が表示されます。  SET SQLPROMPT '&_CONNECT_IDENTIFIER > '
SET TIME	ON を後に続けると、それぞれのコマンド・プロンプトの前に現在の時刻が表示されます。

SQL\*Plus の LOGIN ファイル内で設定できる各種の SET コマンド変数の詳細は、13-92 ページの「SET」コマンドを参照してください。

## 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、現行の SQL\*Plus のシステム変数 (SET) をホスト・オペレーティング・システム・ファイル (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。

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
```

デフォルトでは、SQL\*Plus はファイル名に拡張子 SQL を追加します。別のファイル拡張子を使用する場合は、ファイル名の終わりにピリオドを入力し、その後に拡張子を指定します。または、SET SUFFIX 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。

## 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 SUFFIX** 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。

### 例 3-1 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、次のように入力します。



```
STORE SET plusenv
```



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

## iSQL\*Plus の構成

次に、iSQL\*Plus の構成について項目で説明します。

### クライアント層の構成

この項では、iSQL\*Plus を使用するための Web ブラウザ構成オプションについて説明します。

#### Web ブラウザ設定

Cookie を有効にするには、Web ブラウザを構成する必要があります。iSQL\*Plus は、Java スクリプトが有効でない場合も使用できますが、Java スクリプトを有効にするとパフォーマンスが向上します。

#### MIME タイプの追加

一部の Web ブラウザでは、スクリプトを iSQL\*Plus にロードするため、MIME タイプ定義または .SQL 拡張子を持つファイルに対するアプリケーションの関連付けの削除または作成が必要な場合があります。

一部のブラウザでは、ローカル・マシンにスクリプトを保存するために、MIME タイプの設定が必要な場合があります。iSQL\*Plus の「Work Screen」画面の「Save Script」をクリックした際に、スクリプトを保存するためのファイル名入力のプロンプトが表示されずに新しいウィンドウが開く場合は、次の MIME タイプを設定します。

```
application/vnd.oracle-isqlplus.script
```

これによって、スクリプトをディスクに保存できるように構成されます。さらに、次を追加します。

```
application/vnd.oracle-isqlplus.output
```

これによって、iSQL\*Plus の「Interface Options」画面から「Save to file」オプションを使用できるようになります。

たとえば、Netscape Navigator 4.7 for Windows NT で、SQL 拡張子を持つファイルにアプリケーションの関連付けを設定する手順は、次のとおりです。

1. 「編集」メニューから「設定」を選択します。
2. 「Navigator」メニュー・ツリーから「アプリケーション」を選択します。

3. 「新しいタイプ」 ボタンをクリックします。表示されたフォームで、次のように設定します。
  - タイプの説明 : SQL ファイル
  - ファイルの拡張子 : SQL
  - MIME タイプ : text/plain
  - 開くアプリケーション : notepad.exe
  - 「上記の拡張子のついたファイルを送信する際の既定値にする」 : オフにします。

Web ブラウザをこのように設定していない場合、iSQL\*Plus がテキスト・ファイルとして認識しないスクリプトをロードしようとする、エラーが発生する場合があります。

### プロキシ・サーバー例外の追加

プロキシ・サーバーの構成によっては、iSQL\*Plus サーバーに接続するための iSQL\*Plus ユーザー・インタフェースの機能が影響を受ける場合があります。iSQL\*Plus サーバーに接続できない場合、iSQL\*Plus の「Login」画面をロードしようとする、ブラウザに「Document contains no data」というアラートが表示されます。Oracle HTTP Server の起動後にこのようなエラーが発生する場合は、プロキシ・サーバーを再構成するか、iSQL\*Plus を実行している Oracle HTTP Server 用のプロキシ例外をブラウザに作成する必要があります。次に、プロキシ例外の 2 つの設定例を示します。

Netscape Navigator 4.7 for Windows にプロキシ例外設定を構成する手順は、次のとおりです。

1. 「編集」メニューから「設定」を選択します。
2. 「詳細」カテゴリから「プロキシ」を選択します。
3. 「手動でプロキシを設定する」ラジオ・ボタンを選択します。
4. 「表示」ボタンをクリックします。「手動でプロキシを設定」ダイアログ・ボックスが表示されます。
5. 「例外」ペインに、プロキシを使用しない Oracle HTTP Server ドメインを入力します。

Microsoft Internet Explorer 5.0 にプロキシ例外設定を構成する手順は、次のとおりです。

1. 「ツール」メニューから「インターネットオプション」を選択します。
2. 「接続」タブの「ローカルエリア ネットワーク (LAN) の設定」の「LAN の設定」ボタンをクリックします。
3. 「プロキシサーバー」ペインの「詳細」ボタンをクリックします。この手順は、「プロキシサーバーを使用する」チェックボックスを選択している場合にのみ有効です。
4. 「例外」ペインに、プロキシを使用しない Oracle HTTP Server ドメインを「\*。」に続けて入力します。たとえば、Oracle HTTP Server ドメインの *machine\_name.domain* に対する例外を入力するには、*\*.machine\_name.domain* と入力する必要があります。

## 保持されるセッション設定

次に同じワークステーションから iSQL\*Plus へログインする場合、次の特定セッションの設定は保持または自動的に入力されます。

- 入力領域のサイズ
- 履歴エントリの数

ユーザー名、パスワードおよび出力先の設定は、iSQL\*Plus では保存されません。

## 中間層の構成 : iSQL\*Plus サーバー

iSQL\*Plus サーバーは、Oracle9i データベースのインストール時に中間層にインストールされます。iSQL\*Plus サーバーで、一部のパラメータおよび設定を構成できます。

### セッションの整合性

iSQL\*Plus の各ログインは一意に識別されるため、次のことが可能です。

- 同じマシンからの複数接続
- 異なるマシンからの複数接続

iSQL\*Plus では、セッションのコンテキスト情報を Oracle HTTP Server に格納することで、このステートフルな動作がサポートされます。Oracle HTTP Server で、HTTP 要求が常に同じサーバーにルート指定されていることを確認する必要があります。同じサーバーにルート指定されていない場合、セッションのコンテキストは見つかりません。ただし、複数の Oracle HTTP Server を起動すると、複数のサーバー間でのユーザー負荷の分散に有効な場合があります。

### iSQL\*Plus 構成ファイルの編集

isqlplus.conf 構成ファイルを直接編集して、次の iSQL\*Plus サーバー・パラメータ設定を変更できます。

- iSQLPlusNumberOfThreads
- iSQLPlusLogLevel
- iSQLPlusTimeOutInterval
- iSQLPlusHashTableSize
- iSQLPlusConnectIdList
- -idle-timeout
- iSQLPlusAllowUserEntMap

Windows の場合、isqlplus.conf ファイルは、次の場所にあります。

```
%ORACLE_HOME%\sqlplus\admin\isqlplus.conf
```

UNIX の場合、isqlplus.conf ファイルは、次の場所にあります。

```
$ORACLE_HOME/sqlplus/admin/isqlplus.conf
```

iSQL\*Plus 初期化パラメータの値を変更するには、パラメータに関連付けられた行を編集します。次の各項で、構成ファイル内で編集する行の構文について説明します。

## iSQL\*Plus スレッド・パラメータの設定

iSQLPlusNumberOfThreads パラメータを使用すると、iSQL\*Plus サーバーで使用可能なスレッドの数を設定できます。各スレッドで HTTP 要求が処理されるため、この値を使用して、iSQL\*Plus サーバーで処理できる同時 HTTP 要求の最大数を設定できます。

多数のユーザーが長時間問合せを実行している場合は、iSQLPlusNumberOfThreads の値を増やすことでパフォーマンスが向上する場合があります。iSQL\*Plus の構成ファイル isqlplus.conf を編集して、スレッド数を変更できます。構成ファイル内で変更する行の構文は、次のとおりです。

```
FastCgiServer ... -initial-env iSQLPlusNumberOfThreads=n
```

*n* の最小値は 1 で、最大値はマシンのリソースによって決定されます。値が未設定の場合、iSQLPlusNumberOfThreads は、デフォルト値の 20 になります。

## iSQL\*Plus ログ・レベルの設定

オラクル社カスタマ・サポート・センターから指示がないかぎり、警告またはデバッグのために iSQLPlusLogLevel を設定する必要はありません。

iSQLPlusLogLevel パラメータを使用すると、iSQL\*Plus サーバー・メッセージを記録するためにロギングを有効にするかどうか、およびロギングのレベルを決定できます。ロギングは、ユーザーの問題解決に有効です。ロギングは、off、warn レベルまたは debug レベルに設定できます。debug レベルで、最も詳細な情報が提供されます。

iSQLPlusLogLevel が無効な値に設定されている場合、または設定されていない場合は、デフォルトで warn レベルに設定されます。ログ・ファイルを開くことまたは書き込むことができない場合は、off に設定されます。ログ・ファイルのデフォルト位置は、Windows の場合は %ORACLE\_HOME%\sqlplus\log\isqlplus\log.xml、UNIX の場合は \$ORACLE\_HOME/sqlplus/log/isqlplus/log.xml です。サイズが 64KB に到達すると、新しいログ・ファイルが開始されます。古いファイルの名前は、log1006130580.xml のような一意の名前に変更されます。最大 10 個のログ・ファイルが保存され、古いファイルから自動的に削除されます。

ログインが有効なときは、iSQL\*Plus によってログ・データが既存ログ・ファイルに追加されるか、または既存ログ・ファイルがない場合は新しいファイルが作成されます。ログ・ファイルへの書き込みを妨げる書き込みエラーが発生した場合、iSQL\*Plus は起動されますが、ログ・ファイルは作成されず、メッセージは記録されません。

ログ・ディレクトリは、iSQL\*Plus での書き込みが可能である必要があります。ログ・ディレクトリには、所有者書き込み権限が含まれます。iSQL\*Plus が nobody として実行されているためログ・ファイルに書き込みができない場合は、iSQL\*Plus の管理者がログ・ファイルへの書き込み権限を iSQL\*Plus に手動で付与する必要があります。一時的にテストまたはトレースを行う場合を除いて、ファイルへのワールド書き込み権限の付与はお勧めしません。

Oracle HTTP Server ログ・ファイルに異常なエラーが書き込まれる場合があります。これらのエラーには、iSQL\*Plus サーバーを起動できない場合に生成されるエラーが含まれます。これらのエラーは、iSQLPlusLogLevel が OFF の場合でも書き込まれます。

iSQLPlusLogLevel が debug に設定されている場合は、プロセス・トレース情報がログに記録されます。

iSQL\*Plus の構成ファイル isqlplus.conf を編集して、ログ・レベルを変更できます。構成ファイル内で変更する行の構文は、次のとおりです。

```
FastCgiServer ... -initial-env iSQLPlusLogLevel=[off|warn|debug]
```

## iSQL\*Plus タイムアウト・パラメータの設定

iSQL\*Plus セッションをタイムアウトさせることによって、マシン負荷を削減し、リソースを有効に活用できます。タイムアウト時間は、iSQLPlusTimeInterval 初期化パラメータを使用して設定します。このパラメータは、セッションが時間切れになるまでのアイドル時間を定義します。iSQL\*Plus の構成ファイル isqlplus.conf を編集して、タイムアウト時間を変更できます。構成ファイル内で変更する行の構文は、次のとおりです。

```
FastCgiServer ... -initial-env iSQLPlusTimeInterval=n
```

*n* には、セッションがタイムアウトになるまでのアイドル時間 (分) を指定します。iSQLPlusTimeInterval のデフォルト値は 30 分です。1 ~ 1440 分までの任意の値を設定できます。ただし、ユーザーがスクリプトを入力できないほど小さい値は設定しないでください。

タイムアウトした iSQL\*Plus セッションをユーザーが使用しようとする時、「Login」画面が表示されて再度ログインするように要求され、次のエラーが表示されます。

SP2-00864: セッションが時間切れになりました。もう一度ログインしてください。

## iSQL\*Plus ハッシュ表サイズ・パラメータの設定

ハッシュ表のエントリ数は、iSQLPlusHashTableSize 初期化パラメータを使用して定義します。各ユーザー・セッションでは、アイドル状態の場合でも、ハッシュ表の 1 エントリが使用されます。デフォルト値は iSQLPlusNumberOfThreads パラメータの値から導出されます。ただし、iSQLPlusHashTableSize は、システムのチューニングのために単独で設定することもできます。

iSQL\*Plus 構成ファイル isqlplus.conf を編集して、ハッシュ表サイズを変更できます。構成ファイル内で変更する行の構文は、次のとおりです。

```
FastCgiServer ... -initial-env iSQLPlusHashTableSize=n
```

*n* には、ハッシュ表のエントリ数を指定します。

## データベース・アクセス制限の有効化または無効化

データベース制限パラメータを使用すると、ユーザーが iSQL\*Plus でアクセスできるデータベースを制限できます。データベース制限パラメータが有効な場合、使用可能なデータベースのドロップダウン・リストが「Login」画面の「Connection Identifier」テキスト・フィールドに表示されます。これによって、ホスト環境での iSQL\*Plus サーバーのセキュリティを向上させることができます。接続識別子は、iSQLPlusConnectIdList に定義されている順に表示されます。

iSQL\*Plus の構成ファイル isqlplus.conf を編集して、データベース・アクセスを制限できます。構成ファイル内で変更する行の構文は、次のとおりです。

```
FastCgiServer ... -initial-env "iSQLPlusConnectIdList=SID1, SID2,..."
```

*SID1, SID2, ...* は、アクセス可能なデータベースを指定する Oracle Net 接続識別子をカンマで区切ったリストです。次に例を示します。

```
FastCgiServer ... -initial-env "iSQLPlusConnectIdList=ABC1, PROD2, DEV3"
```

例中の引用符の使用法に注意してください。開始引用符は、iSQLPlusConnectIdList パラメータ名の前に置き、終了引用符は SID のリストの終わりに置く必要があります。

データベース・アクセス制限の詳細は、10-10 ページの「[データベース・アクセス制限の有効化または無効化](#)」を参照してください。

## iSQL\*Plus アイドル・タイムアウトの設定

FastCGI タイムアウト・パラメータまたはアイドル・タイムアウトでは、iSQL\*Plus からの結果に対する Oracle HTTP Server の待機時間を設定します。Web ブラウザの前に iSQL\*Plus がタイムアウトしない値に設定します。多くの長時間問合せでは、iSQL\*Plus がタイムアウトする前に結果が戻されます。

アイドル・タイムアウトを、ユーザー・セッションの存続期間を管理する `iSQLPlusTimeOutInterval` と混同しないでください。

iSQL\*Plus 構成ファイル `isqlplus.conf` を編集して、アイドル・タイムアウトの値を変更します。構成ファイル内で変更する行の構文は、次のとおりです。

```
FastCgiServer ... -initial-env -idle-timeout=n
```

`n` には、iSQL\*Plus からの結果に対する Oracle HTTP Server の待機秒数を指定します。デフォルト値は 3600 秒です。

## エンティティのマッピングの使用可能化

HTML では、エンティティのマッピングによって、特殊文字が印刷可能な文字に置き換えられます。エンティティのマッピングは、デフォルトでは使用可能ですが、iSQL\*Plus の出力ではユーザー定義の HTML を使用できません。エンティティのマッピングのパラメータによって、iSQL\*Plus サーバーでエンティティのマッピングの設定が変更可能かどうかを制御します。

iSQL\*Plus の構成ファイル `isqlplus.conf` を編集して、`iSQLPlusAllowUserEntMap` の値を変更できます。構成ファイルで変更する行の構文は、次のとおりです。

```
FastCgiServer ... -initial-env iSQLPlusAllowUserEntMap=all|none
```

パラメータが `none` に設定されている場合、あるいはパラメータが存在しない場合または値が無効な場合に、`SET MARKUP HTML ENTMAP` および `COLUMN ENTMAP` を使用すると、エラーが発生します。「System Variables」画面の「MARKUP ENTMAP」ラジオ・ボタンを OFF に変更すると、その値は無視され、ON に戻ります。`iSQLPlusAllowUserEntMap` のデフォルト値は、`none` です。

パラメータが `all` に設定されている場合、`SET MARKUP HTML ENTMAP` コマンドまたは `COLUMN ENTMAP` コマンドを実行して、iSQL\*Plus セッションまたはレポート列に対するエンティティのマッピングのステータスを変更できます。これによって、カスタム HTML を iSQL\*Plus のレポート出力に含めることが可能になります。

## カスケード・スタイルシートの変更

iSQL\*Plus では、カスケード・スタイルシートを使用して、ユーザー・インタフェースの形式を制御します。デフォルトのスタイルシートを独自のスタイルシートに置き換えることができます。デフォルトのスタイルシートは、必要なときに使用できるように、削除せずに名前を変更して残しておくことをお勧めします。

たとえば、Windows でカスケード・スタイルシートを置き換える手順は、次のとおりです。

1. %ORACLE\_HOME%\sqlplus\admin\iplus ディレクトリにナビゲートします。
2. 既存のスタイルシートの名前を iplus.css から別の名前に変更します。
3. 新しいスタイルシートを %ORACLE\_HOME%\sqlplus\admin\iplus ディレクトリにコピーします。
4. 新しいスタイルシートの名前を iplus.css に変更します。
5. 新しいスタイルシートを表示するために、Web ブラウザ表示のリフレッシュが必要な場合があります。

## ユーザーのサーバー認証の有効化

デフォルトでは、Oracle HTTP Server 認証は、SYSDBA および SYSOPER 権限を持つユーザーにのみ必要です。通常のユーザー・ログインに対して Oracle HTTP Server 認証を要求することによって、iSQL\*Plus にアクセスできるユーザーを制限する場合があります。詳細は、10-10 ページの「[ユーザーのサーバー認証の有効化](#)」を参照してください。

## DBA アクセスの有効化

SYSDBA または SYSOPER 権限で接続するか、あるいは iSQL\*Plus サーバーの統計レポートを生成するには、ユーザー名およびパスワードを Oracle HTTP Server の iSQL\*Plus 認証ファイルに追加する必要があります。詳細は、10-11 ページの「[DBA アクセスの有効化](#)」を参照してください。

## 中間層の構成 : Oracle HTTP Server

Oracle HTTP Server は、Oracle9i データベースのインストール時に中間層にインストールされます。次の操作が可能です。

- [Oracle HTTP Server の構成ファイルのテスト](#)
- [Oracle HTTP Server の起動と停止](#)
- [iSQL\\*Plus の有効化または無効化](#)

iSQL\*Plus の追加構成情報を、iSQL\*Plus サーバーの Oracle HTTP Server 構成ファイル httpd.conf に挿入する必要があります。

構成ファイルは、次の 2 層にネストされます。

- httpd.conf には、Oracle9i の構成ファイル oracle\_apache.conf が含まれます。
- oracle\_apache.conf には、iSQL\*Plus の構成ファイル isqlplus.conf が含まれます。

通常は、次の場合にのみ変更を行います。

- isqlplus.conf を変更して iSQL\*Plus サーバーを構成する場合
- oracle\_apache.conf を変更して iSQL\*Plus を無効にする場合

.conf ファイルを変更した後、エラーがないことを確認し、Oracle HTTP Server を停止してから起動して変更を実装します。

## Oracle HTTP Server の構成ファイルのテスト

Oracle HTTP Server の構成ファイル httpd.conf および挿入した任意の構成ファイルについてエラーを確認する手順は、次のとおりです。Windows の場合は、次の手順に従います。

1. Windows のコマンド・プロンプトを開きます。
2. 次のように入力して、Oracle HTTP Server の構成ディレクトリに移動します。

```
cd %ORACLE_HOME%\Apache\Apache\conf
```

3. 次のように入力して、Oracle HTTP Server の構成ファイル httpd.conf を解析します。

```
apache -t
```

構成ファイル内のエラーが表示されます。エラーがある場合は、挿入した構成ファイル oracle\_apache.conf または isqlplus.conf を編集してエラーを修正し、再度テストします。エラーがない場合は、メッセージ「Syntax OK」が表示されます。

UNIX の場合は、次の手順に従います。

1. 端末を起動します。
2. 次のように入力して、Oracle HTTP Server の構成ディレクトリに移動します。

```
cd $ORACLE_HOME/Apache/Apache/conf
```

3. 次のように入力して、Oracle HTTP Server の構成ファイル httpd.conf を解析します。

```
apachectl configtest
```

構成ファイル内のエラーが表示されます。エラーがある場合は、挿入した構成ファイル oracle\_apache.conf または isqlplus.conf を編集してエラーを修正し、再度テストします。エラーがない場合は、メッセージ「Syntax OK」が表示されます。

## Oracle HTTP Server の起動と停止

Oracle HTTP Server の構成ファイル `httpd.conf` および挿入した構成ファイルへの変更を有効にするには、Oracle HTTP Server 実行中の場合は、停止してから再起動する必要があります。サーバーに現在アクセスしているユーザーの数を調べる方法はないため、サーバーの停止は、できるだけ短時間にする必要があります。構成ファイルを変更する場合は、構成ファイル `httpd.conf` を解析し、エラーをレポートしてから、Oracle HTTP Server を停止および起動します。

Oracle HTTP Server を停止して起動する手順は、次のとおりです。Windows の場合は、次の手順に従います。

1. Windows のコマンド・プロンプトを開きます。
2. 次のように入力して、Oracle HTTP Server のホーム・ディレクトリに移動します。

```
cd %ORACLE_HOME%\Apache\Apache
```

3. 次のように入力して、Oracle HTTP Server の実行を停止します。

```
apache -k shutdown
```

4. 次のように入力して、Oracle HTTP Server を起動します。

```
apache -k start
```

UNIX の場合は、次の手順に従います。

1. 端末を起動します。
2. 次のように入力して、Oracle HTTP Server ホームの `bin` ディレクトリに移動します。

```
cd $ORACLE_HOME/Apache/Apache/bin
```

3. 次のように入力して、Oracle HTTP Server の実行を停止します。

```
apachectl stop
```

4. 次のように入力して、Oracle HTTP Server を起動します。

```
apachectl start
```

---

---

**警告：** Oracle HTTP Server を、UNIX の場合は dba または oper グループのメンバー、Windows の場合は ORA\_DBA、ORA\_OPER、ORA\_SID\_DBA または ORA\_SID\_OPER グループのメンバーで起動した場合、iSQL\*Plus DBA の URL は、オペレーティング・システムによって自動的に Oracle9i で認証されます。この問題を回避するには、Oracle HTTP Server を、これらのオペレーティング・システム・グループのメンバーでないユーザーとして起動します。

iSQL\*Plus DBA の URL のユーザーであれば、「/ as sysdba」または「/ as sysoper」を使用して、Oracle9i に対する認証が可能です。

---

---

### iSQL\*Plus の有効化または無効化

Oracle HTTP Server の構成ファイルを編集して、iSQL\*Plus を無効にすることができます。

iSQL\*Plus を無効にする手順は、次のとおりです。

1. Oracle HTTP Server を停止します。
2. 次のように入力して、Oracle HTTP Server の構成ディレクトリに移動します。

Windows の場合：

```
cd %ORACLE_HOME%\Apache\Apache\conf
```

UNIX の場合：

```
cd $ORACLE_HOME/Apache/Apache/conf
```

3. oracle\_apache.conf 構成ファイルを開きます。
4. 次のように、isqlplus.conf の include 行の最初に # を挿入して、コメント行にします。

Windows の場合：

```
# include "ORACLE_HOME\sqlplus\admin\isqlplus.conf"
```

UNIX の場合：

```
# include "ORACLE_HOME/sqlplus/admin/isqlplus.conf"
```

ORACLE\_HOME には、Oracle ホーム・ディレクトリのパスを指定します。

5. oracle\_apache.conf ファイルを保存します。
6. 編集した構成ファイルを解析して、編集内容をチェックします。「[Oracle HTTP Server の構成ファイルのテスト](#)」を参照してください。
7. 次回 Oracle HTTP Server を起動すると、iSQL\*Plus は無効になります。

iSQL\*Plus を再度有効にするには、編集内容を元に戻して oracle\_apache.conf の include 行のコメント指定を解除し、Oracle HTTP Server を再起動します。

## 中間層の構成 : Oracle Net

リモート・データベースに接続する場合は、iSQL\*Plus サーバーが実行されている中間層に Oracle Net がインストールおよび構成されていることを確認する必要があります。Oracle Net のインストールおよび構成の詳細は、Oracle9i のマニュアルを参照してください。

## データベース層の構成

Oracle9i はデータベース層にインストールされます。データベース層は、物理的に中間層とは別のマシンに存在する場合（Oracle Net を使用してアクセスします）、または物理的に中間層と同じマシンに存在する場合があります。Oracle Net を使用してデータベースにアクセスする場合は、データベース・リスナーが構成されていて実行中であることを確認します。Oracle9i のインストールおよび構成の詳細は、Oracle9i のマニュアルを参照してください。

## iSQL\*Plus Extension for Windows の構成

iSQL\*Plus Extension for Windows へは、Windows プラットフォームの拡張メニューからアクセスします。iSQL\*Plus Extension for Windows の拡張メニューへは、.SQL ファイル拡張子を持つファイルを右クリックしてアクセスします。

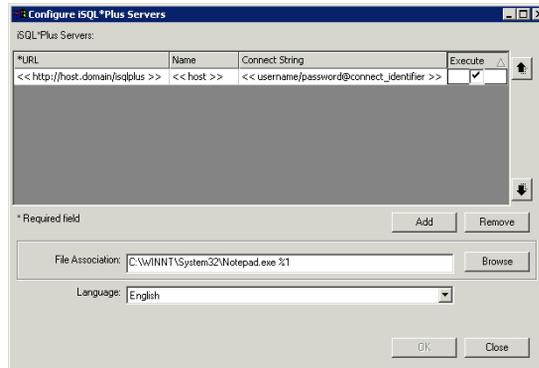
Oracle9i サーバーをインストールすると、デフォルトで、Oracle HTTP Server に対する iSQL\*Plus サーバーのエントリが作成されます。Oracle9i クライアントをインストールした場合、iSQL\*Plus サーバーのエントリは作成されません。「Configuration」ダイアログ・ボックスを使用して、使用可能な iSQL\*Plus サーバーを構成できます。

## 「Configuration」ダイアログ・ボックス

Windows エクスプローラで .SQL ファイルを右クリックして「Configuration」ダイアログ・ボックスを開き、iSQL\*Plus サーバーを選択し、「Configure」を選択できます。「Configure iSQL\*Plus Servers」ダイアログ・ボックスが表示されます。

フィールドが列に収まらない場合は、フィールドの全テキストがツール・ヒントに表示されます。ただし、フィールドを編集またはコピーするには、列が完全なフィールドを表示できる幅であることを確認してください。ヘッダーの列セパレータを選択してドラッグし、列幅を広げることができます。

フィールド名の列ヘッダーをクリックすると、iSQL\*Plus サーバーのリストを昇順または降順でソートできます。また、iSQL\*Plus サーバーのリストの右側にある上下矢印ボタンを使用すると、サーバー定義を手動でソートすることもできます。



**URL** 有効な iSQL\*Plus サーバーの URL を、次の形式で入力します。

```
http://host.domain/isqlplus
```

このフィールドは必須です。

**Name** この iSQL\*Plus サーバーの名前を入力します。名前を入力すると、この名前がコンテキスト・メニューで使用されます。入力しない場合は、iSQL\*Plus サーバーの詳細な URL が使用されます。

**Connect String** ユーザー名とパスワード、および接続するデータベースを指定するための接続文字列を入力します。ユーザー名およびパスワードを省略した場合は、iSQL\*Plus を起動すると、それらの入力を求めるプロンプトが表示されます。接続識別子または Oracle Net 別名を省略した場合は、デフォルトのデータベースに接続されます。

**Execute** 選択したファイルに対して実行される動作を変更するには、「Execute」チェックボックスのチェックを外します。デフォルトの動作は「Execute」です。もう 1 つの選択肢は「Load」です。「Execute」を選択すると、選択したファイルが iSQL\*Plus サーバーで実行され、結果がブラウザに表示されます。「Load」を選択すると、iSQL\*Plus サーバーが開き、選択されたファイルが入力領域にロードされます。

**Add** 「Add」をクリックすると、新しい iSQL\*Plus サーバーが追加されます。

**Remove** 「Remove」をクリックすると、選択した iSQL\*Plus サーバーの定義が削除されます。

**File Association** ダブルクリックしたときに、.SQL ファイル拡張子を持つファイルに関連付けられるアプリケーションのパスおよび引数を入力します。デフォルトのアプリケーションは、Windows のメモ帳です。この定義は、iSQL\*Plus サーバーの定義には影響しません。

**Browse** .SQL ファイル拡張子を持つファイルをダブルクリックしたときに起動されるアプリケーションを検索するには、「Browse」をクリックします。

**Language** iSQL\*Plus Extension で使用する言語を、使用可能な言語のドロップダウン・リストから選択します。

- ポルトガル語 (ブラジル)
- 英語
- フランス語 / フランス語 (カナダ)
- ドイツ語
- イタリア語
- 日本語
- 韓国語
- 簡体字中国語
- スペイン語 / スペイン語 (南米)
- 繁体字中国語

この言語設定は、iSQL\*Plus Extension にのみ有効で、iSQL\*Plus には有効ではありません。

**OK** 「OK」をクリックすると、変更が保存され、「Configure iSQL\*Plus Servers」ダイアログ・ボックスが閉じます。

**Cancel** 「Cancel」をクリックすると、変更が終了され、iSQL\*Plus Extension が閉じます。変更を取り消すかどうか、再度確認を求められます。

初めて「Configure iSQL\*Plus Servers」ダイアログ・ボックスを開くと、「OK」ボタンは使用不可で、「Cancel」ボタンは「Close」と表示されています。変更を行うと、「OK」が使用可能になり、「Close」の表示は「Cancel」に変わります。

## サーバーの追加

iSQL\*Plus Server Extension を使用するには、事前に 1 つ以上の iSQL\*Plus サーバーを構成する必要があります。iSQL\*Plus サーバーを追加する手順は、次のとおりです。

1. Windows エクスプローラで、.SQL ファイルを右クリックします。
2. iSQL\*Plus サーバーを選択して、コンテキスト・メニューのオプションを表示します。最初に、次のオプションが表示されます。
  - No iSQL\*Plus Servers configured  
Oracle9i サーバーがインストールされている場合は、標準 URL 用および DBA の URL 用の 2 つの構成済 iSQL\*Plus サーバーが表示されます。
  - Configure
  - About iSQL\*Plus Extension
3. 「Configure」をクリックします。「Configure iSQL\*Plus Servers」ダイアログ・ボックスが表示されます。
4. 「iSQL\*Plus Servers:」領域に iSQL\*Plus サーバーの定義を入力します。最初にこのダイアログ・ボックスを開くと、次の構文モデルが表示され、入力するべきフィールドの内容が示されます。

URL: << http://host.domain/isqlplus >>

Name: << host >>

Connect String: << username/password@connect\_identifier >>

iSQL\*Plus サーバーの URL は入力する必要があります。他のフィールドはオプションです。Oracle9i サーバーのインストール中に iSQL\*Plus サーバーを構成した場合は、表示が異なる場合があります。

5. 「OK」をクリックして、新しいサーバー定義を保存します。

新しい iSQL\*Plus サーバー名がコンテキスト・メニューに表示されます。コンテキスト・メニューからサーバーを選択すると、iSQL\*Plus サーバーの詳細な URL がステータス・バーに表示されます。

## サーバーの変更

iSQL\*Plus サーバーを変更する手順は、次のとおりです。

1. Windows エクスプローラで、.SQL ファイルを右クリックします。
2. iSQL\*Plus サーバーを選択して、コンテキスト・メニューのオプションを表示します。
3. 「Configure」をクリックします。「Configure iSQL\*Plus Servers」ダイアログ・ボックスが表示されます。
4. 使用可能なサーバーのリストから、変更するサーバーを選択します。
5. テキスト・フィールド内でサーバー定義を直接編集し、各フィールドの編集の終了後 [Enter] キーを押します。フィールドが列に収まらない場合にフィールドを編集またはコピーするには、ヘッダーの列セパレータを選択し、ドラッグして列幅を広げる必要があります。
6. 「OK」をクリックして、変更を保存します。

変更された iSQL\*Plus サーバー名がコンテキスト・メニューに表示されます。

## サーバーの削除

使用可能なサーバーのリストから、削除するサーバーを選択します。サーバーの詳細が、後に続くテキスト・ボックスに表示されます。「Remove」ボタンをクリックすると、使用可能なサーバーのリストからサーバーが削除されます。

iSQL\*Plus サーバーを削除する手順は、次のとおりです。

1. Windows エクスプローラで、.SQL ファイルを右クリックします。
2. iSQL\*Plus サーバーを選択して、コンテキスト・メニューのオプションを表示します。
3. 「Configure」をクリックします。「Configure iSQL\*Plus Servers」ダイアログ・ボックスが表示されます。
4. 使用可能なサーバーのリストから、削除するサーバーを選択します。
5. 「Remove」をクリックします。
6. 「OK」をクリックして、変更を保存します。

iSQL\*Plus サーバーがコンテキスト・メニューから削除されます。

# 4

---

---

## 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。

- SQL\*Plus コマンドラインの起動
- SQLPLUS コマンド
- コマンドライン・ヘルプの使用方法
- SQL\*Plus コマンドラインの終了
- iSQL\*Plus の起動
- iSQL\*Plus でのヘルプの使用方法
- iSQL\*Plus の終了
- iSQL\*Plus の起動
- ユーザー名およびパスワードのログイン
- データベース接続識別子

## SQL\*Plus コマンドラインの起動

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。

### 例 4-1 SQL\*Plus の起動

ここでは、SQL\*Plus の起動方法を示します。次の手順に従います。

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. ホスト・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。

---

---

**注意：** 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 9.2.0.1.0 - Production on Thu Aug 1 16:29:01 2002  
(c) Copyright 1986, 2002 Oracle Corporation. All rights reserved.  
Enter user-name:
```

4. ユーザー名を入力して [Return] を押します。「Enter password:」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 のリリースおよび PL/SQL などの使用可能なツール製品のリリースが表示されます。

続いて、次のような SQL\*Plus のコマンド・プロンプトが表示されます。

```
SQL>
```

このコマンド・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。このマニュアルでは、左側の余白にキーボードのアイコン () が示されている場合は、コマンド・プロンプト行に情報の入力が必要です。コンピュータ画面のアイコン () が示されている場合は、画面に表示される内容が示されています。

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。

## SQL\*Plus コマンドラインの起動ショートカット

SQL\*Plus の起動時に、SQLPLUS コマンドに続けてユーザー名とパスワードをスラッシュ (/) で区切って入力できます。たとえば、次のように入力できます。



```
SQLPLUS HR/your_password
```

次に、[Return] を押します。ホスト・オペレーティング・システムへのログイン時に、自動的に SQL\*Plus にログインするような設定も可能です。詳細は、ご使用のオペレーティング・システム固有の Oracle インストール・ガイドおよび管理者リファレンスを参照してください。

## SQLPLUS コマンド

SQL\*Plus 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 [ [Option] [Logon] [Start] ]
```

各項目の内容は、次のとおりです。

**Option** 構文は次のとおりです。詳細は、4-4 ページを参照してください。

```
-H[ELP] | -V[ERSION]
| [ [-L[OGON]] [-M[ARKUP] markup_option] [-R[ESTRICT] {1|2|3}]
[-S[ILENT]] ]
```

また、*markup\_option* の構文は、次のとおりです。

```
HHTML [ON|OFF] [HEAD text] [BODY text] [TABLE text]
[ENIMAP {ON|OFF}] [SPOOL {ON|OFF}] [PRE[FORMAT] {ON|OFF}]
```

**Logon** 構文は次のとおりです。詳細は、4-10 ページを参照してください。

```
{username[/password] [@connect_identifier | / ]
[AS {SYSOPER|SYSDBA}] | /NOLOG
```

**Start** 構文は次のとおりです。詳細は、4-11 ページを参照してください。

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

*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。

次の項で、SQLPLUS コマンドの項目について説明します。

## Option

### 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。

### VERSION オプション

-V [ERSION]

SQL\*Plus の現在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。

### 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。このオプションは、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。このオプションは、SQL\*Plus を実行可能な Windows の GUI では使用できません。

### MARKUP オプション

-M [ARKUP]

MARKUP オプションを使用すると、問合せまたはスクリプトから完全なスタンドアロン Web ページを生成することができます。MARKUP オプションが現在サポートしているバージョンは、HTML 4.0 Transitional です。

SQLPLUS -MARKUP HTML ON または SET 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 タグが閉じられます。

-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。

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出力には、<HTML> または <BODY> タグは含まれません。

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。

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 コマンドと機能的に同等なオプションがあります。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、13-92 ページの「SET」コマンドを参照してください。

---

---

**注意：** ご使用のオペレーティング・システムによっては、SQLPLUS 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 [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。SET MARKUP を使用する場合は、現行のセッションで、前回 SET MARKUP HTML コマンドで設定した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。デフォルトは OFF です。

MARKUP オプションの SPOOL ON を使用すると HTML 出力の書出しが可能になり、その後 SQL\*Plus コマンドの SPOOL *filename* を使用して、スプール・ファイルへの出力の書出しを開始できます。詳細は、後述の「SPOOL [ON|OFF]」および 13-127 ページの「SPOOL」コマンドを参照してください。

#### 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* には、次の内容が挿入されます。

```
<TITLE>SQL*Plus Report</TITLE>
```

*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、HTML 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> タグの内容を指定できます。デフォルトでは、属性は付いていません。テキスト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、HTML 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。

テキスト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、HTML 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「&lt;」、「&gt;」、「&quot;」および「&amp;」で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>SELECT DEPARTMENT_ID, CITY
      1 FROM EMP_DETAILS_VIEW
      2 WHERE SALARY = 12000;
```

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; SELECT DEPARTMENT_ID, CITY
      2 FROM EMP_DETAILS_VIEW
      3 WHERE SALARY = 12000;
```

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。

<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。

---

---

**注意：** ENTMAP は、HTML オプションが ON であるか、または SPOOL オプションが ON である場合のみ有効です。出力でのエンティティの使用方法の詳細は、13-29 ページの「**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。

---

---

**注意：** 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。

---

---

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。書き込まれるタグおよびそれらのデフォルトの内容は、次のとおりです。

```
<HTML>
<HEAD>
<TITLE>SQL*Plus Report</TITLE>
<META name="generator" content="SQL*Plus 9.0.1">
</HEAD>
<BODY>
```

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、ファイルが閉じられます。

```
</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。

---

---

**注意：** 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"  
SET MARKUP HTML ON PREFORMAT ON
```

---

---

### MARKUP の使用方法

PREFORMAT ON を明示的に設定していない既存のスクリプトでは、HTML 表に出力が生成されます。HTML の <PRE> タグに出力するには、PREFORMAT ON を設定する必要があります。

一部の SQL\*Plus コマンドでは、出力が HTML 表に送られた場合の動作が異なります。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、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 および RECSEP のみが、HTML レポート出力を生成します。

## 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 4-1 に、各制限レベルで使用禁止にされるコマンドを示します。

表 4-1 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

## 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、プロンプトは表示されませんが、SQL\*Plus によって入力が求められます。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。

## Logon

*username* [/*password*]

SQL\*Plus を起動し、Oracle 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。パスワードを、SQLPLUS コマンドの一部として、次の形式でコマンドラインに入力するとします。

```
sqlplus username [/password]
```

このパスワードは、システム上のすべてのユーザーが参照できます。一部のオペレーティング・システムには、すべてのコマンドおよびその引数を表示する監視ツールが備えられています。

*username* および *password* を省略すると、入力を求めるプロンプトが表示されます。*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。プロンプトは表示されますが、SQL\*Plus では、*password* が端末画面に表示されることはありません。SILENT モードでは、ユーザー名およびパスワードのプロンプトは表示されません。ユーザー名は、入力したとおりに表示されますが、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、Oracle インストールで使用する Oracle Net 通信プロトコルによって異なります。詳細は、ご使用のプロトコル用の Oracle Net のマニュアルを参照するか、DBA に問い合せてください。

/

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、`connect_identifier`は指定できません。デフォルト・ログオンでは、SQL\*Plus はユーザー名 `OPPS$name` を使用してユーザーをログインさせようとしています。`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「OPPS\$」は、テキストの他の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` に変更する場合があります。オペレーティング・システム認証の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

AS {SYSOPER|SYSDBA}

AS 句を使用すると、SYSOPER または SYSDBA 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、「/」および「/NOLOG」を使用しても、特権付き接続は可能です。

このオプションを使用する場合、ほとんどのオペレーティング・システムで、コマンド引数を引用符で囲む必要があります。次に例を示します。

```
SQLPLUS "/ AS SYSDBA"
SQLPLUS "SYSTEM/your_password AS SYSOPER"
```

/NOLOG

Oracle 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。

## Start

```
@{url/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が使用されます。デフォルトの拡張子は、.sql です。

詳細は、13-129 ページの「START」コマンドを参照してください。

## コマンドライン・ヘルプの使用方法

SQL\*Plus コマンドのオンライン・ヘルプにアクセスするには、SQL コマンド・プロンプトで `HELP` と入力し、続けてコマンド名を入力します。たとえば、次のように入力します。



```
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 1 画面内に収まるように 4 列に表示されます。たとえば、次のように入力します。



```
HELP INDEX
```

ヘルプが使用できないことを示す応答があった場合は、データベース管理者に連絡してください。詳細は、13-66 ページの「[HELP](#)」コマンドを参照してください。

## コマンドライン・ヘルプのインストール

データベース管理者は、SQL\*Plus ヘルプ表を作成し、それらを SQL\*Plus ヘルプ・データに移入します。SQL\*Plus ヘルプをインストールする前に、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。インストールされていない場合は、ヘルプ表を作成およびロードできない。
- SYSTEM ユーザーのデフォルト表領域が、ヘルプ・システムに十分対応できる大きさである。128KB 以上の空き領域が必要。
- SQL\*Plus ヘルプ・スクリプト・ファイルが、次のディレクトリで使用可能である。

```
$ORACLE_HOME/SQLPLUS/ADMIN/HELP/
```

ヘルプ・スクリプト・ファイルには、次のものがあります。

- `HLPBLD.SQL`: 新しいヘルプ表の削除および作成
- `HELPUS.SQL`: ヘルプ・データへのヘルプ表の移入
- `HELPDROP.SQL`: 既存の SQL\*Plus ヘルプ表の削除

SQL\*Plus ヘルプをインストールするには、次の手順を実行します。

1. SYSTEM ユーザーとして次のように入力し、SQL\*Plus を実行します



```
SQLPLUS SYSTEM/your_password
```

`your_password` では、SYSTEM ユーザーに対して定義したパスワードを指定します。

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



```
@$ORACLE_HOME/SQLPLUS/ADMIN/HELP/HLPBLD.SQL HELPUS.SQL
```

## SQL\*Plus コマンドラインの終了

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 のコマンド・プロンプトで EXIT コマンドを入力します。

### 例 4-2 SQL\*Plus の終了

SQL\*Plus を終了するには、SQL\*Plus のコマンド・プロンプトで次のように EXIT コマンドを入力します。



```
EXIT
```

切断した Oracle 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。しばらくすると、オペレーティング・システムのプロンプトが表示されます。

この章の先へ進む前に、例 4-1 の手順 3 ~ 5 に従って、再度 SQL\*Plus を起動します。または、「SQL\*Plus コマンドラインの起動ショートカット」で説明したショートカットを使用してログインします。

## iSQL\*Plus の起動

iSQL\*Plus を実行する前に、Oracle HTTP Server および iSQL\*Plus サーバーが中間層で実行されている必要があります。3-14 ページの「Oracle HTTP Server の起動と停止」を参照してください。

iSQL\*Plus を実行するには、次の手順に従います。

1. Web ブラウザの「場所」フィールドに、iSQL\*Plus の URL を入力します。たとえば、次のように入力します。

```
http://machine_name.domain:port/isqlplus
```

*machine\_name.domain* には URL を指定し、*port* には使用する Oracle HTTP Server のポート番号を指定します。iSQL\*Plus の「Login」画面が表示されます。

正常にログインすると、各ログインが一意に識別されるため、同じマシンまたは複数のクライアント・マシンから、複数の iSQL\*Plus セッションを実行できます。

2. ユーザー名、パスワードおよび接続識別子を入力します。詳細は、「ユーザー名およびパスワードのログイン」および「データベース接続識別子」を参照してください。
3. 「Login」ボタンをクリックします。iSQL\*Plus の「Work Screen」画面が表示されます。

SYSDBA または SYSOPER 権限で iSQL\*Plus を実行するには、次の iSQL\*Plus DBA の URL を使用します。

```
http://machine_name.domain:port/isqlplusdba
```

iSQL\*Plus DBA の URL から接続している場合、Oracle HTTP Server 認証によって DBA の「Login」画面または CONNECT コマンドを介して AS SYSDBA 接続または AS SYSOPER 接続が可能です。ただし、Oracle9i のユーザー名およびパスワード認証のため、アクセスできない場合があります。

## iSQL\*Plus でのヘルプの使用方法

iSQL\*Plus のオンライン・ヘルプには、iSQL\*Plus 固有のヘルプおよび構文が提供されています。iSQL\*Plus ヘルプにアクセスするには、「Help」アイコンをクリックします。ヘルプ画面が新しいブラウザ・ウィンドウに表示されます。iSQL\*Plus のヘルプは、次の 8 か国語で使用できます。

- ポルトガル語 (ブラジル)
- フランス語
- ドイツ語
- イタリア語
- 日本語
- 韓国語
- 簡体字中国語
- スペイン語

デフォルトは、英語に設定されています。SQL\*Plus での言語サポートの詳細は、[第 12 章「SQL\\*Plus グローバリゼーション・サポート」](#)を参照してください。

## iSQL\*Plus の終了

iSQL\*Plus を終了するには、「Logout」アイコンをクリックします。常に「Logout」アイコンを使用して iSQL\*Plus を終了し、システムおよびサーバー・リソースを解放することをお勧めします。

iSQL\*Plus では、EXIT コマンドを使用すると iSQL\*Plus セッションは終了せず、現在実行中のスクリプトが一時停止します。

## URL からの iSQL\*Plus の起動

URL から要求を送信して、iSQL\*Plus を起動し、URL 変数、SQL スクリプトおよび置換変数を渡すことができます。

SQL スクリプトは、HTTP、HTTPS または FTP から使用可能か、または URL 変数として iSQL\*Plus に渡される必要があります。iSQL\*Plus でスクリプトを実行し、その結果を Web ブラウザ・ウィンドウに戻すか、またはスクリプトを「Work Screen」画面にロードします。

iSQL\*Plus は、通常のユーザーとして、または SYSDBA か SYSOPER 権限で起動することができます。iSQL\*Plus サーバーの設定および統計についてのレポートを生成する場合も、iSQL\*Plus を起動できます。

通常のユーザーとして iSQL\*Plus を起動するために、Web ブラウザの「場所」または「アドレス」フィールドに入力する構文は、次のとおりです。

```
http://machine_name.domain:port/isqlplus[?UserOpts]
```

SYSDBA または SYSOPER 権限で iSQL\*Plus を起動するには、次の構文を使用します。

```
http://machine_name.domain:port/isqlplusdba[?DBAOpts]
```

iSQL\*Plus を起動して、iSQL\*Plus サーバーの統計レポートを生成するには、次の構文を使用します。

```
http://machine_name.domain:port/isqlplusdba[?Statistics]
```

各項目の内容は、次のとおりです。

*machine\_name.domain* には、Oracle HTTP Server の URL を指定します。

*port* には、Oracle HTTP Server で使用されるポート番号を指定します。

*UserOpts* には、`UserLogin|Script|UserLogin&Script` を指定します。

*DBAOpts* には、`DBALogin|Script|DBALogin&Script` を指定します。

*Statistics* には、`statistics={active|full}[&refresh=number]` を指定します。

さらに、次のように指定します。

*UserLogin* には、`userid=username[/password][@connect_identifier]` を指定します。

*DBALogin* には、`userid={username[/password][@connect_identifier] | / } AS {SYSDBA | SYSOPER}` を指定します。

*Script* には、`script=text[&type={url|text}][&action={execute|load}][&variable=ovalue. . .]` を指定します。

URL パラメータ *userid* が存在しない場合、またはこのパラメータの情報が不完全な場合は、iSQL\*Plus によって「Login」画面が表示されます。URL パラメータが完全で、ログイン情報が有効な場合、iSQL\*Plus は接続され、要求は続行されます。

SQL スクリプトのパラメータは、任意の順序で指定できます。スクリプト・パラメータは、`script`、`userid` などの予約済キーワードで始まる場合、リテラル・パラメータではなくコマンドとして、*iSQL\*Plus* で解釈される可能性があります。

URL パラメータの `type` が `url` か、または指定されていない場合、スクリプト・パラメータは SQL スクリプトの URL とみなされます。

URL パラメータの `type` が `text` の場合、スクリプト・パラメータの `text` は SQL スクリプト自身の内容とみなされます。この方法で渡されたスクリプトの場合、HTML キャラクタ・セットに制限が加えられる可能性があります。

URL パラメータの `action` が `execute` か、または指定されていない場合、SQL スクリプトは *iSQL\*Plus* で実行されます。

URL パラメータの `action` が `load` の場合、スクリプトは「Work Screen」画面の入力領域にロードされますが、実行はされません。Web ブラウザでは、大きいスクリプトを「Work Screen」画面に表示できない場合があります。その場合、スクリプトが切り捨てられる可能性があります。

Statistics の構文および機能の詳細は、2-12 ページの「[iSQL\\*Plus サーバーの統計](#)」を参照してください。

### 例

ユーザー名 `hr` およびパスワード `your_password` で *iSQL\*Plus* にログインするには、次のように入力します。

```
http://machine_name.domain/isqlplus?userid=hr/your_password
```

*iSQL\*Plus* で、URL `http://machine_name2.domain/myscript.sql` に置かれているスクリプトを実行し、ユーザー名およびパスワードを求めるプロンプトを表示させるには、次のように入力します。

```
http://machine_name.domain:port/isqlplus?script=http://machine_name2.domain/myscript.sql
```

URL に配置されているスクリプトを実行し、ユーザー名およびパスワードを渡し、データベースにログインし、スクリプトにパラメータを渡して置換変数の値を指定するには、次のように入力します。

```
http://machine_name.domain:port/isqlplusdba?userid=hr/your_password@oracle9i%20as%20sysdba&script=ftp://machine_name2.domain/script.sql&name=*&salary=12000
```

*iSQL\*Plus* DBA の URL を使用するため、Oracle HTTP Server 認証も必要です。空白はサポートされていないため、この例では `%20` としてエンコードされています。

ユーザー名およびパスワードを渡さずに、スクリプトを iSQL\*Plus にロードするには、次のように入力します。

```
http://machine_name.domain:port/isqlplus?script="select * from  
employee_details_view;"&type=text&action=load
```

## ユーザー名およびパスワードのログイン

SQL\*Plus を起動する場合は、認証された Oracle ユーザーとして識別されるユーザー名およびそのユーザー名の正当な所有者であることを証明するパスワードが必要です。

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。

Oracle9i 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は、次のとおりです。

- SYS
- SYSTEM

パスワードの変更については、次の「[パスワードの変更](#)」を参照してください。デフォルト・ログインの詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

一度ログインすると、SQL\*Plus の 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・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/FOX
```

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プロンプトが表示されます。CONNECT コマンドの後にユーザー名のみを入力し、パスワードを省略する方法もあります (この場合は、パスワードの入力を求める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時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。

SQL\*Plus の DISCONNECT コマンドを入力すると、SQL\*Plus を終了せずに、現在 Oracle に接続しているユーザーを切断できます。

## パスワードの変更

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。13-73 ページの「PASSWORD」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 `your_password` を使用してユーザー名 `HR` でデフォルト・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/your_password
```

DISCONNECT コマンドを使用すると、SQL\*Plus または iSQL\*Plus を終了せずに、現在 Oracle に接続しているユーザーを切断できます。

## iSQL\*Plus でのパスワードの変更

「Change Password」画面で Oracle データベース・アカウントのパスワードを変更できます。DBA 権限でログインしている場合は、他のユーザーのパスワードを変更できます。「Change Password」画面には、「Preferences」画面からアクセスできます。



**Username:** Oracle データベース・アカウントのユーザー名を入力します。

**Old password:** Oracle データベース・アカウントの現在のパスワードを入力します。

**New password:** 新規パスワードを入力します。

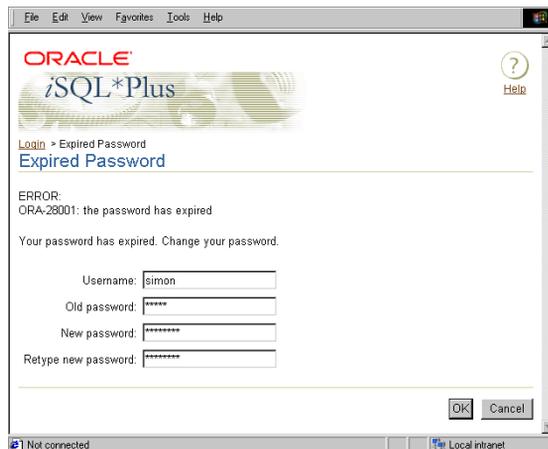
**Retype new password:** 正しく入力されていることを確認するため、新規のパスワードを再入力します。

**OK** Oracle データベース・アカウントのパスワードを変更するには、「OK」ボタンをクリックします。

**Cancel** パスワードを変更せずに「Preferences」画面に戻るには、「Cancel」ボタンをクリックします。

## iSQL\*Plus の「Expired Password」画面

パスワードが期限切れになっている場合は、iSQL\*Plus へのログイン時に、自動的に「Expired Password」画面が表示されます。「Change Password」画面に入力する場合と同様に、「Expired Password」画面のフィールドに入力します。



## マルチユーザー・システム

各ユーザーに、オペレーティング・システムにアクセスするためのユーザー名およびパスワードが必要です。それらは、SQL\*Plus に使用するものと同じである必要はありません。

## シングルユーザー・システム

一度に 1 人のユーザーのみがコンピュータを使用する場合、そのユーザーが自分で DBA の役割を果たす必要がある場合もあります。ユーザー名およびパスワードを独自に定義する場合は、『Oracle9i SQL リファレンス』を参照してください。

## データベース接続識別子

SQL\*Plus を使用して Oracle データベースのデータにアクセスするには、まずそのデータベースに接続しておく必要があります。SQL\*Plus の起動時に、接続識別子を使用して、接続するデータベースを指定します。接続識別子は、コマンドライン接続の開始に使用する SQLPLUS コマンドに対する引数として指定するか、または iSQL\*Plus の「Login」画面の「Connection Identifier」フィールドに入力します。

接続識別子を省略すると、デフォルトのデータベースに接続されます。デフォルトのデータベース以外のデータベースを指定するには、接続識別子を使用する必要があります。

DBA には、データベースを作成して、それらのデータベースへの接続を可能にする接続識別子を提供する責任があります。iSQL\*Plus のユーザー・インタフェースでは、データベース管理者は、接続識別子の制限リストを指定できます。これらの識別子には、通常の「Login」画面の「Connection Identifier」フィールドに代わるドロップダウン・リストからアクセスできます。

デフォルトのデータベースは、オペレーティング・システムの環境変数やシンボルを設定するか、または Oracle の特定の構成ファイルを編集することによって、オペレーティング・システム・レベルで構成されます。詳細は、ご使用のオペレーティング・システム固有の Oracle マニュアルを参照してください。

コマンドライン・ユーザー・インタフェースでは、SQLPLUS コマンドの NOLOG 引数を使用して 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、トランスポート・スクリプトの書込み、あるいは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。

他のコンピュータ上にあるデータベース、またはデフォルトのデータベース以外のホスト・コンピュータ上にあるデータベースは、リモート・データベースと呼ばれます。リモート・データベースに Oacle Net が構成されていて、両方のデータベースに互換性のあるネットワーク・ドライバが備えられている場合は、リモート・データベースに接続できます。リモート・データベースには、次の 2 つの方法で接続できます。

- SQL\*Plus の起動時に接続
- CONNECT コマンドを使用して接続

## SQL\*Plus の起動時にリモート・データベースに接続

iSQL\*Plus の起動時にリモート・データベースに接続するには、「Login」画面の「Connection Identifier」フィールドに Oracle Net サービス名を入力します。デフォルトの Oracle データベースを使用する（存在する場合）には、このフィールドを空のままにします。それ以外の場合は、Oracle Net 別名を入力して、接続するリモート・データベースを指定します。Oracle Net 別名を使用する場合は、iSQL\*Plus サーバーを実行しているマシン上で定義する必要があります。このマシンは、Web ブラウザを実行するマシンと異なる場合があります。

SQL\*Plus コマンドラインの起動時にリモート・データベースに接続するには、次の形式で SQLPLUS コマンドに Oracle Net サービス名を含めます。

```
SQLPLUS HR/your_password@connect_identifier
```

詳細な接続識別子も使用できます。たとえば、次のように入力できます。

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=www.oracle.com)
(PORT=1521)))(CONNECT_DATA=((SERVICE_NAME=orashop.us.acme.com)))
```

リモート・データベースに対して有効なユーザー名、パスワードおよび適切なサービス名を指定する必要があります。SQL\*Plus コマンドラインでは、必要に応じてユーザー名およびパスワードを求めるプロンプトが表示されます。接続したデータベースは、別のデータベースに CONNECT するか、DISCONNECT するか、または SQL\*Plus を終了するまで使用されます。

データベースへの接続時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。

## CONNECT コマンドを使用してリモート・データベースに接続

CONNECT コマンドを使用してリモート・データベースに接続するには、次の形式で CONNECT コマンドに Oracle Net 接続識別子を含めます。

```
CONNECT HR/your_password@connect_identifier
```

詳細は、使用しているプロトコル用の Oracle Net のマニュアルを参照するか、DBA に問い合わせてください。

SQL\*Plus コマンドラインでは、必要に応じてパスワードを求めるプロンプトが表示され、指定したデータベースに接続できます。iSQL\*Plus では、プロンプトは表示されませんが、ユーザー名およびパスワードを CONNECT コマンドに指定する必要があります。

データベースへの接続時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。

この方法でリモート・データベースに接続した場合、そのデータベースに対してすべての SQL コマンド、SQL\*Plus コマンドおよび PL/SQL ブロックが使用できます。

サービス名の指定のために入力する正確な文字列は、コンピュータで使用する Oracle Net プロトコルによって異なります。詳細は、13-44 ページの「CONNECT」を参照するか、または DBA に問い合わせてください。

---

---

**注意：** 接続識別子の @ 記号を、スクリプトの実行に使用する @ コマンドと混同しないでください。

---

---

# 第 II 部

---

## SQL\*Plus の使用方法

ここでは、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説明します。

この部の内容は、次のとおりです。

- 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 セキュリティ
- SQL\*Plus を使用したデータベース管理
- SQL\*Plus グローバリゼーション・サポート



# 5

---

---

## SQL\*Plus の基本

この章では、SQL\*Plus の基本的な使用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。

- コマンドの入力と実行
- SQL コマンドの実行
- PL/SQL ブロックの実行
- 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
- データベースへの変更の自動保存
- 実行中のコマンドの停止
- ホスト・オペレーティング・システムのコマンドの実行
- ヘルプの使用方法
- 表定義の表示
- PL/SQL 定義の表示
- 表示の制御
- エラー・メッセージの解釈

## コマンドの入力と実行

特に指定がないかぎり、通常、コマンドの使用方法についての説明はコマンドライン・ユーザー・インタフェースおよび *iSQL\*Plus* ユーザー・インタフェースの両方に適用できます。

*SQL\*Plus* コマンドラインでは、*SQL\*Plus* プロンプトにコマンドを入力します。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。

コマンドライン・ユーザー・インタフェースまたは *iSQL\*Plus* ユーザー・インタフェースのいずれかで、次の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、定型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。

*SQL\*Plus* コマンドラインおよび *iSQL\*Plus* の両方で、[¥] キーおよび [Delete] キーを使用できます。*iSQL\*Plus* では、Web ブラウザの編集キーを使用して、入力領域の文の編集に切り取りおよび貼付けを実行できます。また、テキスト・エディタなどの他のアプリケーションからスクリプトまたは文の切り取りまたはコピーを実行して、入力フィールドに直接貼り付けることもできます。

*iSQL\*Plus* では、「Save Script」ボタンを使用して、スクリプトをテキスト・ファイルに保存できます。「Load Script」ボタンを使用して、スクリプトをロードすることもできます。スクリプトを保存およびロードしておくと、編集およびテストを行う際に有効です。

## SQL バッファ

入力した最新の SQL コマンドまたは PL/SQL ブロック (*SQL\*Plus* コマンドは異なる) が格納される領域は、SQL バッファと呼ばれます。そのコマンドまたはブロックは、別のコマンドまたはブロックを入力するまで保持されます。現行の SQL コマンドまたは PL/SQL ブロックを編集または再実行する場合、それらを再入力する必要はありません。バッファ内に格納されたコマンドやブロックの編集または再実行の詳細は、6-13 ページの「スクリプトの実行」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。

## コマンドの実行

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。

iSQL\*Plus では、コマンドまたはスクリプトを入力領域に入力し、「Execute」ボタンをクリックして入力領域の内容を実行します。デフォルトでは、スクリプトの結果が入力領域の下に表示されます。現行のセッションですでに実行されているコマンドにアクセスし、再実行するには、「History」画面を使用します。

iSQL\*Plus では、入力領域の最後にある SQL 文または PL/SQL 文が実行されます。文が不完全、あるいは最後の「;」または「/」が欠落している場合も実行されます。iSQL\*Plus のスクリプトを SQL\*Plus コマンドラインで実行する場合は、「;」または「/」で文が終了していることを確認してください。

iSQL\*Plus では、現在のシステム変数およびその他のオプションの状態が、次の実行時に継承されます。「History」画面を使用してスクリプトを再実行すると、現在のシステム変数値によっては、前に取得した結果と異なる場合があります。

一部の SQL\*Plus コマンドは、iSQL\*Plus では、論理的に意味をなさない場合や、適用できない場合もあります。iSQL\*Plus でサポートされない SQL\*Plus コマンドのリストについては、[付録 D 「iSQL\\*Plus では未サポートのコマンド」](#)を参照してください。

## 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 コマンドの詳細は、『Oracle9i SQL リファレンス』を参照してください。

### 例 5-1 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。

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] を押します。SQL\*Plus でコマンドが処理され、結果が次のように表示されます。



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、再度コマンド・プロンプトが表示されます。操作を誤ったために前述の結果が得られなかった場合は、コマンドを再入力してください。

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。LINESIZE というシステム変数を調整して、文字の出力幅を設定する必要があります。詳細は、7-30 ページの「ページ・サイズの設定」を参照してください。通常、このマニュアルの例では、70 文字に設定されています。問合せ出力をこのマニュアルと同じように表示するには、SET LINESIZE を 70 に設定します。取得したレコード件数に関するメッセージが表示されるかどうかは、FEEDBACK というシステム変数の設定によって異なります。システム変数の詳細は、「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。このマニュアルの残りの例では、選択されたレコードの数を示しません。

## 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。

**1 つの SQL コマンドを複数行に分割** SQL コマンドは、個々の単語を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 5-1 「SQL コマンドの入力」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 5-1 「SQL コマンドの入力」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。

**SQL コマンドの終了** 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 5-1 「SQL コマンドの入力」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] を押します。SQL\*Plus によって、コマンドが処理されて SQL バッファに格納されます (詳細は、後述の「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] を押します。

---

---

**注意：** 同じ行でセミコロンの後にコメントを入力することはできません。コメントの挿入の詳細は、6-9 ページの「スクリプトへのコメントの挿入」を参照してください。

---

---

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することができます。コマンドの最終行の末尾で [Return] を押します。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] を押します。SQL\*Plus で、コマンドが処理されて SQL バッファに格納されます (詳細は、後述の「SQL バッファ」を参照してください)。

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。プロンプトに次の行の行番号が表示されます。

---

---

**注意：** SET SQLBLANKLINES コマンドを使用して、SQL 文内での空白行の表示方法および処理方法を変更できます。空白行の動作の変更については、13-92 ページの「SET」コマンドを参照してください。

---

---

再度 [Return] キーを押すと、SQL\*Plus のコマンド・プロンプトが表示されます。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。

**ストアド・プロシージャの作成** 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、SQL の CREATE コマンドを使用します。ストアド・プロシージャの作成に使用する SQL の CREATE コマンドを次に示します。

- 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](#)」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 コマンドを実行してストアド・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 (;) では、これらの CREATE コマンドを実行できません。

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。たとえば、次のように入力します。



```
SHOW ERRORS PROCEDURE ASSIGNVL
```

**SHOW** コマンドの詳細は、[第 13 章「SQL\\*Plus コマンド・リファレンス」](#)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、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。たとえば、次のように入力します。



```
EXECUTE :ID := EMPLOYEE_MANAGEMENT.GET_ID('BLAKE')
```

**EXECUTE** コマンドの詳細は、[第 13 章「SQL\\*Plus コマンド・リファレンス」](#)を参照してください。

## コマンド・プロンプトからの現行 SQL コマンドまたは PL/SQL ブロックの実行

コマンド・プロンプトで RUN コマンドまたは / (スラッシュ) コマンドを入力して、現行の SQL コマンドまたは PL/SQL ブロックを実行 (または再実行) できます。RUN コマンドを使用すると、コマンドまたはブロックを実行する前にバッファ内にある SQL コマンドまたは PL/SQL ブロックが表示されます。/ (スラッシュ) コマンドを使用すると、SQL コマンドまたは PL/SQL ブロックがそのまま実行されます。

## PL/SQL ブロックの実行

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 コマンドの詳細は、『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。

SQL\*Plus に PL/SQL サブプログラムを入力するには、PL/SQL モードに入る必要があります。PL/SQL モードに入る方法は、次のとおりです。

- SQL\*Plus コマンド・プロンプトで、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 (たとえば、CREATE FUNCTION 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。

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、セミコロン (;) または空白行でブロックを終了および実行することはできません。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。

SQL\*Plus では、SQL\*Plus コマンド・プロンプトから入力したサブプログラムは、SQL バッファに格納されます。現行のサブプログラムを実行するには、RUN コマンドまたはスラッシュ (/) を実行します。同様に、SQL の CREATE コマンドを実行してストアド・プロシージャを作成する場合にも、RUN またはスラッシュ (/) を入力する必要があります。これらの SQL コマンドは、他の SQL コマンドのようにセミコロン (;) では実行できません。

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 へ送信されます。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。

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;
```

サブプログラムの実行時に、サブプログラム内の SQL コマンドが、サブプログラムの外部にある場合とは異なる処理を行う場合があります。PL/SQL 言語の詳細は、『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。

## 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。個々の SQL\*Plus コマンドの詳細は、[第 13 章「SQL\\*Plus コマンド・リファレンス」](#)を参照してください。

コマンド入力を高速化するため、多くの SQL\*Plus コマンドに、1 文字または数文字の略称を使用できます。一部の SQL\*Plus コマンドの略称は、コマンドの詳細とともに [第 3 章「SQL\\*Plus の構成」](#)、[第 5 章「SQL\\*Plus の基本」](#) および [第 6 章「SQL\\*Plus でのスクリプトの使用」](#) で説明されています。すべての SQL\*Plus コマンドの略称については、[第 13 章「SQL\\*Plus コマンド・リファレンス」](#)を参照してください。

### 例 5-2 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。

1. コマンドラインで、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。行の入力終了後、[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. RUN コマンドを入力して、最新の（例 2-3 からの）問合せを再実行します。



```
RUN
```



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.
```

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。その後、RUN コマンドによって、バッファ内に格納されていた例 5-1 「SQL コマンドの入力」の問合せが再実行されます。SQL\*Plus では、SQL\*Plus コマンドは SQL バッファに格納されません。

## 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。

**長い SQL\*Plus コマンドを次の行に続ける方法** 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 キーワードが指定の位置にありません。
```

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。

**SQL\*Plus コマンドの終了** 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] を押します。必要な場合は、SQL\*Plus コマンドの末尾にセミコロンを入力できます。

## 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の設定は、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。システム変数は、SET 変数とも呼ばれます。

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、13-92 ページの「[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、3-2 ページの「[SQL\\*Plus の構成](#)」および 4-3 ページの「[SQLPLUS コマンド](#)」を参照してください。

SET コマンド変数の現行の設定を表示するには、コマンド・プロンプトで 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、13-120 ページの「[SHOW](#)」コマンドを参照してください。

## データベースへの変更の自動保存

SQL の DML コマンド (UPDATE、INSERT および DELETE) は、単独かまたは PL/SQL ブロック内で使用できます。これらのコマンドを使用して、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。それらの変更は、SQL の COMMIT コマンド、DCL または 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 SET コマンドの AUTOCOMMIT 変数で指定します。

### 例 5-3 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。



```
SET AUTOCOMMIT IMMEDIATE
```

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。



```
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ごとにカウントされ、10 コマンドごとに変更がコミットされます。

---

---

**注意：** 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、13-92 ページの「[SET](#)」コマンドの AUTOCOMMIT 変数を参照してください。

## 実行中のコマンドの停止

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、「Cancel」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、コマンド・プロンプトに戻ります。

iSQL\*Plus では、「Cancel」ボタンをクリックします。

---

---

**注意：**「Cancel」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（問合せ結果の出力については、[第 7 章「SQL\\*Plus レポートの書式設定」](#)を参照してください）。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ラクション・ガイドおよび管理者リファレンスを参照してください。

---

---

## ホスト・オペレーティング・システムのコマンドの実行

SQL\*Plus のコマンド・プロンプトからホスト・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、既存のホスト・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。

ホスト・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、ホスト・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、次の SQL\*Plus コマンドで、ホスト・コマンドの DIRECTORY \*.SQL を実行できます。



```
HOST DIRECTORY *.SQL
```

ホスト・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。

**注意：** 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は、第 10 章「SQL\*Plus セキュリティ」を参照してください。

## ヘルプの使用方法

SQL\*Plus の使用中に、表の列定義のリストを表示するか、スクロール表示を開始または停止する必要がある場合があります。また、コマンドを誤って入力したか、Oracle または SQL\*Plus に問題が発生したときに、受け取ったエラー・メッセージを解釈する必要がある場合もあります。次の各項では、それらの状態に関するヘルプの使用方法について説明します。

## 表定義の表示

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。

### 例 5-4 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;
```



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)

---

**注意：** DESCRIBE は、Oracle 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、『Oracle9i SQL リファレンス』を参照してください。

---

## PL/SQL 定義の表示

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 5-5 DESCRIBE コマンドの使用

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



```
DESCRIBE afunc
```



```
FUNCTION afunc RETURNS NUMBER
Argument Name      Type      In/Out  Default?
-----
F1                  CHAR      IN
F2                  NUMBER    IN
```

## 表示の制御

長いレポートまたは列数が多い表の定義の表示中、画面を停止して、表示内容を詳しく調べるとします。表示内容を調べている間、表示が一時停止します。続行するには、「Resume」を押します。

必要な場合は、SQL\*Plus の SET コマンドの PAUSE 変数を使用して、問合せまたはレポートの表示を 1 画面ごとに一時停止できます。詳細は、13-92 ページの「SET」コマンドを参照してください。

## エラー・メッセージの解釈

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、第 14 章「SQL\*Plus エラー・メッセージ」を参照してください。

### 例 5-6 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 EMPLOYEE.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。



```
SP2-00310: ファイル "employee.sql" をオープンできません。
```

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、次のいずれかの手順を実行します。

- 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、このマニュアルの第 14 章「SQL\*Plus エラー・メッセージ」にある「SQL\*Plus のエラー・メッセージおよびコード」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、このマニュアルの第 14 章「SQL\*Plus エラー・メッセージ」にある「SQL\*Plus COPY コマンドのメッセージ」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、『Oracle9i データベース・エラー・メッセージ』、またはご使用のオペレーティング・システム固有の Oracle インストレーション・ガイドおよび管理者リファレンスで、その Oracle 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このマニュアルの第 13 章「SQL\*Plus コマンド・リファレンス」、SQL コマンドについては『Oracle9i SQL リファレンス』、または PL/SQL ブロックについては『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。それ以外の場合は、DBA に問い合わせてください。



---

## 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。

- スクリプトの編集
- スクリプトの変更
- SQL\*Plus でのスクリプトの編集
- スクリプトへのコメントの挿入
- スクリプトの実行
- スクリプトのネスト
- リターン・コードの受信
- リターン・コードを伴うスクリプトの終了
- 対話型コマンドの作成
- バインド変数の使用方法

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、第 1 章「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。

## スクリプトの編集

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。コマンドライン上でスクリプトを取り出したり編集することが可能で、コマンドラインまたは iSQL\*Plus ユーザー・インタフェースのどちらでも実行できます。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。

すべての有効な SQL スクリプトは、使用しているワークステーションからアクセス可能な iSQL\*Plus にロードできます。一部の Web ブラウザでは、iSQL\*Plus にスクリプトをロードするために、MIME タイプまたは .SQL 拡張子を持つファイルに対するアプリケーションの関連付けを作成する必要があります。MIME タイプまたはアプリケーションの関連付けの作成方法については、3-5 ページの「[MIME タイプの追加](#)」を参照してください。

## 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 を終了せずに、ホスト・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、3-2 ページの「[SQL\\*Plus の構成](#)」および 13-48 ページの「[DEFINE コマンド](#)」および 13-59 ページの「[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。

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 ブロックの後にピリオド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 ブロックを挿入できます）。

### 例 6-1 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、ファイルを作成してスクリプトを保持する必要があります。



```
EDIT SALES
```

SQL\*Plus、SQL および PL/SQL の文とコマンドを入力できます。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 列の書式モデルに 0 (ゼロ) を指定すると、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、[第 5 章「SQL\\*Plus の基本」](#) および『Oracle9i SQL リファレンス』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。

## スクリプトの変更

EDIT コマンドを使用して、既存のスクリプトを変更できます。既存のスクリプトを EDIT コマンドで編集するには、EDIT の後にファイル名を続けて入力します。

たとえば、拡張子が SQL の PROFIT という既存のファイルを編集するには、次のコマンドを入力します。



```
EDIT PROFIT
```

EDIT で、ファイル名のみを指定した場合、そのファイルの拡張子は SQL であるとみなされます。

## SQL\*Plus でのスクリプトの編集

SQL\*Plus では、SQL\*Plus コマンドがバッファに格納されないため、コマンド・プロンプトで直接入力した SQL\*Plus コマンドを編集するには、[Back Space] を使用するか、またはコマンドを再入力します。

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。または、ホスト・オペレーティング・システムのエディタを使用して、バッファの内容を編集することもできます。

表 6-1 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 6-1 SQL\*Plus 編集コマンド**

コマンド	略称	用途
APPEND <i>text</i>	A <i>text</i>	行末にテキストを追加します。
CHANGE / <i>old/new</i>	C / <i>old/new</i>	行内の <i>old</i> を <i>new</i> に変更します。
CHANGE / <i>text</i>	C / <i>text</i>	テキストを行から削除します。
CLEAR BUFFER	CL BUFFER	すべての行を消去します。
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。
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> ~ <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	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> または <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> ~ <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。

## 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。

### 例 6-2 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。LIST コマンドで、次のように指定します (例 5-1 「SQL コマンドの入力」の手順に従った後、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、その例の手順を再実行してください)。

LIST

```
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 バッファには格納されません。そのため、セミコロンを削除しなくてもバッファの末尾に新しい行を追加でき、編集に便利です。

## 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。

### 例 6-3 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。



```
SELECT EMPLOYEE_ID, LAST_NAME, JO_ID, SALARY
*
```

```
ERROR at line 1:
ORA-00904: string: 無効な識別子です。
```

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、または文字 C
- 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。行を再入力するには、行番号を入力し、その後に空白および新しいテキストを続けて入力して、**[Return]** を押します。

#### 例 6-4 エラーの修正

**JOB\_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。



```
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 5-2 「SQL\*Plus コマンドの入力」 で指定した書式が保持されます（例 5-2 「SQL\*Plus コマンドの入力」 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、13-29 ページの「COLUMN」 コマンドを参照してください。

## 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、行 1 になります。



```
0 SELECT EMPLOYEE_ID
```

### 例 6-5 行の追加

例 6-4 「エラーの修正」 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（I と省略できます）、[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。



```
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

## 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 6-6 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

## 行の削除

バッファ内の行を削除するには、次のように DEL 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、削除する行を表示します。
2. DEL コマンドおよびオプションの句を入力します。

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、バッファに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、13-50 ページの「[DEL](#)」コマンドを参照してください。

## スクリプトへのコメントの挿入

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
- 複数行の1つ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

スクリプトでのコメントの使用方法については、「[コメント挿入時の注意](#)」を参照してください。

## REMARK コマンドの使用法

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'
```

## /\*...\*/ の使用方法

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。

コメントの始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。入力しない場合は、SQL\*Plus バッファ内のコマンドの実行時に、コメントはコマンドとして処理され、スラッシュは実行コマンドとして解釈されます。

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'
```

SQL コメントをコマンド・プロンプトで直接入力する場合、そのコメントはバッファ内に格納されません。

## 「--」 の使用方法

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することができます。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
```

## コメント挿入時の注意

通常、SQL\*Plus では、コメントとして識別される入力は解析または実行されません。

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。SQL\*Plus では、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する方法を示します。

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');
```

Warning: Procedure created with compilation errors.

例に示す位置にコメントを挿入すると、このコマンドが PL/SQL のコマンドであることが SQL\*Plus で認識されません。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: 文字が無効です。
```

SQL\*Plus では、文の終了記号の後で同じ行にテキストを記述することはできないため、コメントとして認識されません。

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 キーワードが指定の位置にありません。
```

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;
```

&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.

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、変数の値を求めるプロンプトが表示されます。SET DEFINE OFF を使用して、置換文字をスキップ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、13-92 ページの「SET」コマンドの DEFINE、SQLTERMINATOR および SQLBLANKLINES を参照してください。

## 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
```

ファイルの拡張子が SQL の場合、ファイル名にピリオドおよび拡張子 SQL を追加する必要はありません。

### 例 6-7 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。

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

SQL\*Plus でコマンドが入力される様子を表示する場合は、SET コマンドの ECHO 変数を ON に設定します。ECHO 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。

@ (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
```

指定したスクリプト内のコマンドを、@ コマンドで表示および実行する方法は、START の場合と同じです。SET ECHO での指定は、START コマンドと同様に、@ コマンドでも有効です。

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。

### 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、パスワード、空白、@ およびスクリプト名を指定します。



```
SQLPLUS HR/your_password @SALES
```

SQL\*Plus が起動し、スクリプトが実行されます。

- SQLPLUS コマンドの後にユーザー名、空白、@ およびスクリプト名を指定します。



```
SQLPLUS HR @SALES
```

パスワードの入力を求めるプロンプトが表示され、SQL\*Plus が起動し、スクリプトが実行されます。

- 実行するスクリプトの 1 行目に、ユーザー名を挿入します。SQLPLUS コマンドの後に @ およびスクリプト名を指定します。パスワードの入力を求めるプロンプトが表示され、SQL\*Plus が起動し、スクリプトが実行されます。
- 実行するスクリプトの 1 行目に、ユーザー名、スラッシュ (/) およびパスワードを挿入しておきます。SQLPLUS コマンドの後に @ およびスクリプト名を指定します。SQL\*Plus が起動し、スクリプトが実行されます。この方法を使用する前に、ファイルにパスワードを公開することに対するセキュリティ・リスクを考慮してください。

## 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
START YRENDSLS
```

---

**注意：** この例では、@@ コマンドが有効な場合があります。詳細は、[第13章「SQL\\*Plus コマンド・リファレンス」](#)の「@@ (二重アットマーク) コマンド」を参照してください。

---

## リターン・コードの受信

無効なユーザー名またはパスワードを指定したか、あるいはその他のエラーが発生したために SQL\*Plus へのログインに失敗した場合は、EXIT FAILURE コマンドと同じエラー状態が戻されます。詳細は、この章の「EXIT コマンド」を参照してください。

### 例 6-8 SQL\*Plus の起動

ユーザー名 HR およびパスワード `your_password` で SQL\*Plus を起動するには、次のように入力します。



```
SQLPLUS HR/your_password
```

SQL\*Plus を前述のように起動し、POLICY をデフォルトのデータベースにする (POLICY が有効な Oracle Net データベース接続識別子になります) には、次のように入力します。



```
SQLPLUS HR/your_password@POLICY
```

ユーザー名 HR およびパスワード `your_password` で SQL\*Plus を起動し、名前が STARTUP で拡張子が SQL のスクリプトを実行するには、次のように入力します。



```
SQLPLUS HR/your_password @STARTUP
```

HR および @STARTUP の間には、空白を入れます。

HTML ON で SQL\*Plus を起動して、出力をファイルに取得し、Web ブラウザで表示できるようにするには、次のように入力します。



```
SQLPLUS -M "HTML ON" HR/your_password
```

セッション中、EDIT および HOST コマンドへアクセスせずに SQL\*Plus を起動するには、次のように入力します。



```
SQLPLUS -R 1 HR/your_password
```

### 例 6-9 SQLPLUS 構文の表示

SQLPLUS コマンドの構文を表示するには、次のように入力します。



```
SQLPLUS -H
```



```
Usage: SQLPLUS [ [<option>] [<logon>] [<start>] ]
where <option> ::= -H | -V | [[-L] [-M <o>] [-R <n>] [-S] ]
```

```
<logon> ::= <username>[/<password>] [@<connect_string>] | / |/NOLOG
<start> ::= @<URL>|<filename>[.<ext>] [<parameter> ...]
-H displays the SQL*Plus version banner and usage syntax
-V displays the SQL*Plus version banner
-L attempts logon just once
-M <o> uses HTML markup options <o>
-R <n> uses restricted mode <n>
-S uses silent mode
```

## リターン・コードを伴うスクリプトの終了

ホスト・オペレーティング・システム上のバッチ・ファイルから実行中のスクリプトで SQL エラーが発生した場合、そのスクリプトはリターン・コードを付けて強制終了できます。この操作には、SQL\*Plus コマンドの **WHENEVER SQLERROR** を使用します。詳細は、13-150 ページの「**WHENEVER SQLERROR** コマンド」を参照してください。

同様に、オペレーティング・システム・エラーが発生した場合に終了するには、**WHENEVER OSERROR** コマンドを使用します。詳細は、13-148 ページの「**WHENEVER OSERROR**」コマンドを参照してください。

## 対話型コマンドの作成

SQL\*Plus の次の機能を使用すると、エンド・ユーザーによる入力可能なスクリプトを設定できます。

- ユーザー変数の定義
- コマンドでの値の置換
- START コマンドを使用した値の指定
- 値の入力を求めるプロンプトの表示

## ユーザー変数の定義

SQL\*Plus の **DEFINE** コマンドを使用すると、ユーザー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ユーザー変数も（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。

### 例 6-10 ユーザー変数の定義

ユーザー変数 **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
```



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

すべてのユーザー変数定義を表示するには、コマンド・プロンプトに単独で DEFINE を入力します。DEFINE を使用して明示的に定義するユーザー変数には、CHAR 値のみ指定できません（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 データ型のユーザー変数を暗黙的に定義できます。ACCEPT コマンドについては、この章の後半で詳しく説明します。

ユーザー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。

## 置換変数の使用

SALES での問合せ（例 6-1 「システム・エディタでの SQL スクリプトの作成」を参照）と同様の問合せを作成し、職種が SA\_MAN の従業員のみでなく、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 CHAR 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 句の値 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、ユーザー変数名の前に 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;
```

## 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、コマンド・プロンプトで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せん。

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、プロンプトを表示する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へ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。

予期しない結果が発生する場合があります。置換変数に渡す値と同じ名前の置換変数は作成しないでください。置換変数に指定した値が変数名と一致する場合、指定した値のかわりに、一致する変数の内容が使用されます。

### 例 6-11 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。



```
CLEAR BUFFER
INPUT
  SELECT &GROUP_COL, MAX(&NUMBER_COL) MAXIMUM
  FROM &TABLE
  GROUP BY &GROUP_COL
.
```

```
SAVE STATS
```



```
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';
```

### 値の入力を求める不要なプロンプトの回避

スクリプト 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 6-11 では、値の入力を求めるプロンプトが 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。ユーザーが変数を定義 (DEFINE) した場合、SQL\*Plus で (「&」変数または「&&」変数の形で) 変数を参照するそれぞれの置換変数の変数値が置換されます。SQL\*Plus では、ユーザーが変数を定義解除 (UNDEFINE) するまで、そのセッションでの変数値の入力を求めるプロンプトは表示されません。

#### 例 6-12 二重アンパサンドの使用法

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  
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,  
I
```



```

3i MIN (&&NUMBER_COL) MINIMUM,
4i SUM (&&NUMBER_COL) TOTAL,
5i AVG (&&NUMBER_COL) AVERAGE
6i
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
```

### 制限事項

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、SQL\*Plus のコメント (REMARK、`/*...*/` または `--`) などの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。

### 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。

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	置換文字の前で使用でき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 VERIFY ON	置換の前後にスクリプトの各行を表示します。
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。

システム変数の詳細は、13-92 ページの「[SET](#)」コマンドを参照してください。

## iSQL\*Plus の「Substitution Variables」画面

SET コマンドで指定したシステム変数は、iSQL\*Plus の動作に影響を与えます。SET DEFINE、SET ESCAPE、SET VERIFY ON および SET CONCAT は、変数の置換動作に影響を与えるため、スクリプトを実行する前に iSQL\*Plus で設定する必要があります。SET オプションの詳細は、13-92 ページの「SET」コマンドを参照してください。

iSQL\*Plus では、「&」および「&&」変数のスクリプトが前処理されます。また、このスクリプトが SQL\*Plus エンジンに送信され、実行される前に、これらの変数の値の入力を求めるプロンプトが表示されます。変数の置換は、次のいずれかのオプションと同期させる必要があります。

- SET DEFINE ON を入力して、後続スクリプトの実行前に、置換変数の入力を求めるプロンプトが常に表示されるように iSQL\*Plus を設定します。「Execute」ボタンをクリックしてコマンドを実行します。

変数の接頭辞として「&」および「&&」を使用し、スクリプトを入力します。DEFINE または UNDEFINE は使用しないでください。「Execute」ボタンをクリックしてスクリプトを実行します。iSQL\*Plus では、スクリプト内のすべての置換変数に対して値の入力を求めるプロンプトが表示されます。スクリプトの実行の終了時、スクリプト内の二重アンパサンド値 (&&) は定義されたままです。したがって、これらの変数値の定義が解除されるか、または iSQL\*Plus をログオフするまで、変数値の再入力を求めるプロンプトは表示されません。このプロンプト表示が必要な場合は、スクリプト内の変数の接頭辞として 1 つのアンパサンド (&) を使用します。これによって、常に、1 つのアンパサンドを接頭辞として使用した変数の置換値の入力を求めるプロンプトが表示されます。DEFINE を使用して、このモードでスクリプトに変数値を定義すると、「Substitution Variables」画面で入力された値がこれらの値で上書きされます。

- SET DEFINE OFF を入力して、後続スクリプトの実行前に置換変数の入力を求めるプロンプトが表示されないように iSQL\*Plus を設定します。「Execute」ボタンをクリックしてコマンドを実行します。

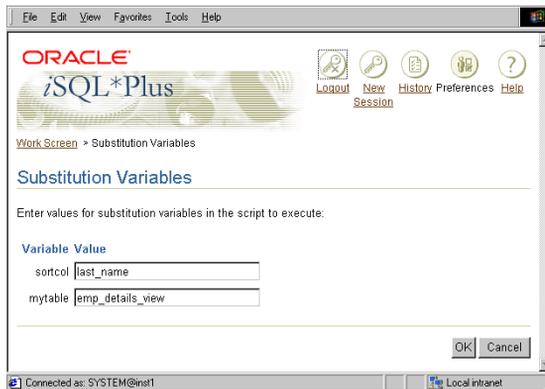
スクリプトを入力します。スクリプトの再実行時に、iSQL\*Plus で、変数値の入力を求めるプロンプトが表示されないようにするには、必ず SET DEFINE OFF をスクリプトの最後のコマンドとして指定してください。SET DEFINE ON を入力し、すべての変数を参照する前に「&」または「&&」で定義します。また、COLUMN ... *new\_value* または COLUMN ... *old\_value* を使用して、置換変数のインスタンス化もできます。「Execute」ボタンをクリックしてスクリプトを実行します。iSQL\*Plus では、置換変数の値の入力を求めるプロンプトは表示されません。この場合、iSQL\*Plus では、スクリプト内で定義 (DEFINE) されていない変数は NULL 値に置換されることに注意してください。

## iSQL\*Plus の「Substitution Variables」画面

置換変数が含まれたスクリプトを iSQL\*Plus で実行すると、「Substitution Variables」画面が表示されます。たとえば、次のように入力したとします。

```
BREAK ON &SORTCOL  
SELECT &SORTCOL, SALARY  
FROM &MYTABLE  
WHERE SALARY > 12000  
ORDER BY &SORTCOL
```

iSQL\*Plus に次のように表示されます。



**Variable** 置換変数名が表示されます。

**Value** 各置換変数の値を入力します。たとえば、前述のスクリプト例の場合は、「sortcol」フィールドに LAST\_NAME と入力し、「mytable」フィールドに EMP\_DETAILS\_VIEW と入力します。

**OK** 入力した置換変数の値を使用して入力領域のスクリプトを実行するには、「OK」ボタンをクリックします。

**Cancel** 入力領域のスクリプトを実行せずに「Work Screen」画面に戻るには、「Cancel」ボタンをクリックします。

## 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 (&) およびその後続けて数値を指定します。このスクリプトを実行するごとに、START によって、ファイル内の各 &1 が START *filenames* 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。

### 例 6-13 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
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
Cambrault	\$11,000	0.30
Zlotkey	\$10,500	0.20

パラメータは、1つのスクリプト内でいくつでも使用できます。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。

---

**注意：** RUN またはスラッシュ (/) を使用してコマンドを実行する場合は、パラメータを使用できません。コマンドをスクリプトに格納し、START または @ を使用して実行する必要があります。

---

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
```

## ユーザーとの対話

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。

### ユーザー変数値のプロンプトおよびアクセプト

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するコマンドで、ユーザーに指示または情報を与えるために使用します。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ユーザー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。

**例 6-14 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消去します。



```
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。



```
INPUT
```



```
4 PROMPT Enter a title of up to 30 characters
5 ACCEPT MYTITLE PROMPT 'Title: '
6 TTITLE LEFT MYTITLE SKIP 2
7 SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY
8 FROM EMP_DETAILS_VIEW
9 WHERE JOB_ID='SA_MAN'
10
```



```
SAVE PROMPT1
```



```
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、7-22 ページの「[ページおよびレポートのタイトルとサイズの定義](#)」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
```

## 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。

### 例 6-15 PROMPT および ACCEPT を置換変数と組み合わせて使用する方法

例 6-14 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消去します。



```
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。



```
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
```

SP2-00425: "one" は有効な数値ではありません。

## 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] を入力する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.
```

## 画面の消去

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
```

# バインド変数の使用方法

SQL\*Plus で PL/SQL サブプログラムに使用する変数を表示できるようにするか、または複数のサブプログラムに同じ変数を使用できるようにするとします。PL/SQL サブプログラムの中で変数を宣言した場合、その変数は SQL\*Plus では表示できません。その変数に SQL\*Plus からアクセスするには、PL/SQL でバインド変数を使用します。

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 からアクセスできます。バインド変数は、リターン・コードの格納、PL/SQL サブプログラムのデバッグなどに使用できます。

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。

### 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、13-142 ページの「[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）。

### 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。たとえば、次のように入力します。



```
VARIABLE ret_val NUMBER
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。

### 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。システム変数の表示については、13-75 ページの「**PRINT** コマンド」を参照してください。

## 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。

---

**注意：** ストアド・ファンクションの戻り値を REFCURSOR 変数に割り当てるには、Oracle7 リリース 7.3 以上を使用する必要があります。

---

### 例 6-16 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。



```
VARIABLE employee_info REFCURSOR
```

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、『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。

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;
```



```
/
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。

### 例 6-17 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 cv_types AS
TYPE EmpInfoTyp is REF CURSOR RETURN emp%ROWTYPE;
END cv_types;

```



/

Package created.

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



```

CREATE OR REPLACE PROCEDURE EmpInfo_rpt
(emp_cv IN OUT cv_types.EmpInfoTyp) AS
BEGIN
OPEN emp_cv FOR SELECT EMPLOYEE_ID, SALARY FROM EMP_DETAILS_VIEW -
WHERE JOB_ID='SA_MAN' ;
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 odcv REFCURSOR
EXECUTE EmpInfo_rpt (:odcv)
PL/SQL procedure successfully completed.

```



この時点でバインド変数を出力します。



```

PRINT odcv

```



```

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
```

### 例 6-18 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;
/
```



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 REF CURSOR 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.
```



```
PRINT rc
```



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

---

---

## 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
- 間隔の設定およびサマリー行によるレポートの明確化
- ページおよびレポートのタイトルとサイズの定義
- 問合せ結果の格納および印刷

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、[第 1 章「SQL\\*Plus 概要」](#)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。

## 列の書式設定

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。

### 列ヘッダーの変更

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、13-29 ページの「COLUMN」コマンドを参照してください。

#### 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。

#### 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
```

#### 例 7-1 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'
```



```
LAST_NAME          MONTHLY SALARY COMMISSION
-----
Russell            14000          .4
Partners           13500          .3
Errazuriz          12000          .3
Cambrault          11000          .3
Zlotkey            10500          .2
```



---

**注意：** 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、13-92 ページの「SET」コマンドを参照してください。

### 例 7-2 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。



```
/
```



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

### 例 7-3 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 =
/
```



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 '-'
```

---

---

**注意：** 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。

---

---

## NUMBER 列の書式設定

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。

### 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。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。

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。

### 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）、または先行 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 7-4 を参照)。書式モデルのすべてのリストおよび詳細は、13-29 ページの「COLUMN」コマンドを参照してください。

#### 例 7-4 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。



```
/
```



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、前述のように書式モデル内で 0 (ゼロ) を使用します。

## データ型の書式設定

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

- CLOB
- NCLOB
- XMLType

### 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、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 のうちの小さい方の値です。

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および書式は、有効な NLS パラメータから導出されます。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、13-29 ページの「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。

### 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、13-29 ページの「COLUMN」コマンドを参照してください。

#### 例 7-5 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  
/
```



```
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 7-5 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、13-29 ページの「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。

---



---

**注意：** 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
```

### 例 7-6 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
```



```
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、『Oracle9i XML API リファレンス -XDK および Oracle XML DB』 および 『Oracle9i ケース・スタディ -XML アプリケーション』を参照してください。

## 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 句を使用すると、入力するコマンドを減らすことができます。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。

### 例 7-7 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。

```
/
```



```
LAST          MONTHLY
NAME          SALARY    BONUS
-----
Russell      $14,000    $0
Partners     $13,500    $0
```

Errazuriz	\$12,000	\$0
Cambrault	\$11,000	\$0
Zlotkey	\$10,500	\$0

## 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
```

### 例 7-8 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
```



```
columns cleared
```

## 列の表示属性の非表示および表示

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
```

## 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
```

### 例 7-9 折り返した列の値の後で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。



```
/
```



```
LAST_NAME                JOB_TITLE                CITY
-----
King                     President                 Seattle
Kochhar                  Administration Vice      Seattle
                        President
-----
De Haan                  Administration Vice      Seattle
                        President
```

```
-----
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、各行の後に（前述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
```

## 間隔の設定およびサマリー行によるレポートの明確化

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（あるいはその両方）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** 値の重複を抑制できます。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（または、平均、最大、最小、標準偏差、平方偏差、行数）を含むサマリー行を計算して出力できます。

## ブレイ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制できます。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
```

---

---

**注意：** **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イクが発生します。

---

---

### 例 7-10 ブレイク列内の値の重複の抑止

前述の問合せ結果で重複した部門番号の表示を抑制するには、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;
```



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

```
6 rows selected.
```

## ブレイク列の値が変更されたときの空白の挿入

ブレイ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
```

### 例 7-11 ブレイ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。

```
/
```





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

6 rows selected.

## 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
```

---

**注意：** 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。

---

## 複数の間隔設定技法の使用法

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。

### 例 7-12 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。



```

                                     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
                AD_VP     Kochhar        17000
                AD_VP     De Haan        17000

                                     Page: 3
DEPARTMENT_ID JOB_ID      LAST_NAME      SALARY
-----
                90 AD_PRES   King           24000
                AD_VP     Kochhar        17000
                AD_VP     De Haan        17000

6 rows selected.
```

## ブレイ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、現行のブレイ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、現行のブレイ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間合せに、前回の BREAK コマンドが影響しないように設定できます。

## ブレイ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、複数のブレイク列、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 の後に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イ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。COMPUTE コマンドの詳細は、第 13 章「SQL\*Plus コマンド・リファレンス」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 7-1 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、13-39 ページの「[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。

### 例 7-13 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。



```
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、現行の間合せを実行します。



```
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_PRES      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、次のように入力します。



```
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 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、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 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.

SALARY 列の書式は、個々の SALARY 値のみでなく、SALARY の合計を表示する場合にも影響します。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。

## 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
```

### 例 7-14 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';
```



```
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
```

## 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。

### 例 7-15 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 7-14 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
  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。

### 例 7-16 同じブレイ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;
```



```
DEPARTMENT_ID LAST_NAME                SALARY
-----
                30 Colmenares                2500
                   Himuro                2600
                   Tobias                 2800
                   Baida                 2900
                   Khoo                  3100
                   Raphaelly            11000
*****
avg                                4150
sum                               24900
```

6 rows selected.

## 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
```

### 例 7-17 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
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。

## ページおよびレポートのタイトルとサイズの定義

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。

## 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、[第 13 章「SQL\\*Plus コマンド・リファレンス」](#)の「TTITLE」コマンドおよび「REPHEADER」コマンドを参照してください。

### 例 7-18 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、次のように現行の間合せを実行します。



/



```

                                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.

### 例 7-19 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。



```

                                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.

定義を変更せずにレポート・ヘッダーを非表示にするには、次のように入力します。



```
REPHEADER OFF
```

### 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。

#### 例 7-20 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  
/  

```



```
                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。前述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、この章の後半で説明します。

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、前述の例で、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'
```

## 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、または 0 (ゼロ) 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、空白 14 個分インデントされます。

### 例 7-21 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
```

この時点で現行の間合せを実行し、結果を表示します。



```
/
```



```
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.

## 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ユーザー変数に格納します。



```
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
```

## 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、13-138 ページの「TTITLE」コマンドを参照してください。

### 例 7-22 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、現行の間合せを再実行します。



```
/
```



```
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。

### 例 7-23 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。



/



```
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.

## 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び表示できます。



```
TTITLE ON  
BTITLE ON
```

## タイトル内に列の値を表示する方法

マスター / デティール・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。

### 例 7-24 マスター / デティール・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。前述の例で入力した 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;
```



```
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イ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・ブレイ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。

## タイトル内に現在の日付を表示する方法

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。

変数（この例では \_DATE）を作成するには、SQL\*Plus の LOGIN ファイルに次のコマンドを追加します。



```
BREAK ON TODAY
COLUMN TODAY NEW_VALUE _DATE
SELECT TO_CHAR(SYSDATE, 'fmMonth DD, YYYY') TODAY
FROM DUAL;
CLEAR BREAKS
```

SQL\*Plus の起動時に、これらのコマンドは SYSDATE（現在の日付）の値を \_DATE という名前の変数に格納します。現在の日付を表示するには、他の変数の場合と同様に、タイトル内で \_DATE を参照します。

LOGIN ファイル内の SELECT コマンド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、『Oracle9i SQL リファレンス』を参照してください。LOGIN ファイルの詳細は、3-3 ページの「[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、対話形式でコマンド・プロンプトからも入力できます。詳細は、13-29 ページの「[COLUMN](#)」コマンドを参照してください。

## 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:24
- 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
```

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 LINESIZE 変数を使用して変更できます。

```
SET LINESIZE number_of_characters
```

### 例 7-25 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消去（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 LINESIZE 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 -
CENTER '01-JAN-2001' SKIP 2
```

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;
```



```

                                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 LINESIZE をデフォルト値にリセットします。



```
SET PAGESIZE 24
SET NEWPAGE 1
SET LINESIZE 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 LINESIZE
```

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。

## 問合せ結果の格納および印刷

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、メモ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。詳細は、ご使用のオペレーティング・システム固有の Oracle インストール・ガイドおよび管理者リファレンス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
```

### 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、Oracle8 から Oracle9i ヘッダを移行するには、SQL\*Loader で使用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、13-92 ページの「SET」コマンドを参照してください。

## 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。

## 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ホスト・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。

### 例 7-26 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、ホスト・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。



## 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	\$14,000
	Partners	\$13,500
*****		-----
sum		\$27,500
90	King	\$24,000
	Kochhar	\$17,000
	De Haan	\$17,000
*****		-----
sum		\$58,000
		-----
sum		\$98,500

COMPANY CONFIDENTIAL

6 rows selected.



---

## SQL\*Plus からの HTML レポートの生成

この章では、問合せ結果を含む HTML レポートの生成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。

- SQL\*Plus コマンドラインを使用したレポートの作成
- iSQL\*Plus を使用したレポートの作成

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、第 1 章「SQL\*Plus 概要」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。

## SQL\*Plus コマンドラインを使用したレポートの作成

プレーン・テキスト出力および SQL\*Plus コマンドライン・ユーザー・インタフェースを使用すると、完全な Web ページまたは Web ページに埋め込まれる HTML 出力を生成できます。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。

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、4-4 ページの SQLPLUS コマンドの「[MARKUP オプション](#)」および 13-109 ページの「[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 タグが閉じられます。

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。

## 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き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、<BODY> 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。

### 例 8-1 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 PFORMAT OFF ENMAP ON -
HEAD "<TITLE>Department Report</TITLE> " -
<STYLE type='text/css' > -
<!-- BODY {background: #FFFFFF} --> -
</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' ENMAP 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` を開くと、次のように表示されます。



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「&quot;」、「&amp;」、「&lt;」および「&gt;」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。

**例 8-2 <html>**

```
<head>
<TITLE>Department Report</TITLE> <STYLE type="text/css"> <!-- BODY {background:
#FFFFFFC6} --> </STYLE>
<meta name="generator" content="SQL*Plus 9.2.0.1.0">
</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&gt; spool off
<br>
</body>
</html>
Creating a Report using the SQLPLUS Command
```



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。

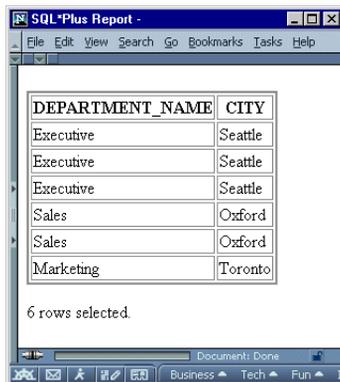
```
SQLPLUS -S -M "HTML ON TABLE 'BORDER=2'" HR/your_password@Ora9i
@depart.sql>depart.html
```

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、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完全な 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、「[CGI スクリプトを使用したレポートの作成](#)」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。



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。

## CGI スクリプトを使用したレポートの作成

SET MARKUP コマンドを対話形式で使用するかわりに、SQLPLUS-MARKUP コマンドを使用すると、「マークアップ・モード」で SQL\*Plus セッションを開始できます。これによって、Common Gateway Interface (CGI) スクリプトが内部に埋め込まれた SQL\*Plus のセッションを実行できます。このようにして作成されたファイルは Web ブラウザに直接表示できます。UNIX のシェル・スクリプト、Windows のバッチ・ファイル、Java、Java スクリプト、Perl スクリプトなどの、Web サーバーでサポートされているスクリプト言語を使用して、SQL\*Plus をコールできます。

この埋込み型の方法で、既存の SQL\*Plus スクリプトを変更せずに使用して、Web ページを作成できます。この方法は、Web ベースのレポートを動的に作成する場合に有効です。

### 例 8-3 CGI スクリプトからのレポートの作成

CGI スクリプトを使用して SQL\*Plus を実行し、SQL スクリプトから Web レポートを生成できます。そのためには、次の 3 つの要素が必要です。

- パラメータをコールし、CGI スクリプトへ渡す Web ページ
- 入力を収集し、SQL\*Plus を実行する CGI スクリプト
- SQL\*Plus セッションによって実行される SQL スクリプト

### CGI を使用した Web ページの例

次の例で示す Web ページは、ユーザー名とパスワード、データベース接続文字列および実行する SQL スクリプトの名前を入力するフィールドが含まれている HTML 形式です。

---

---

**注意：** ログイン情報をスクリプト・ファイルに埋め込む場合、または CGI スクリプトを使用してログイン情報のプロンプトを表示し、入力された情報を SQLPLUS コマンドに渡す場合は、サーバーのセキュリティについて十分に注意してください。

デフォルト値を使用するのではなく、初期条件を設定してください。たとえば、ENTMAP のデフォルトが ON であっても、明示的に ENTMAP ON を指定するようにします。

---

---

次のスクリプトを含む HTML ファイルを作成し、そのファイルを plus.html という名前で保存します。



```
<html>
<head><title>SQL*Plus CGI Report</title></head>
<body bgcolor="#ffffff">

<h1>SQL*Plus CGI Report</h1>
```

```

<!-- Change the URL here.  On Windows NT you may need to use
http://machine_name.domain:port/cgi-bin/perl?plus.pl if your web server is not
configured to identify the script as a Perl program -->

<form method=post action="http://machine_name.domain:port/cgi-bin/plus.pl">
<table border=0 summary="">

<tr>
  <td>Username:</td>
  <td><input type="text" name="username" size="10" align="left"></td>
</tr>
<tr>
  <td>Password:</td>
  <td><input type="password" name="password" size="10" align="left"></td>
</tr>
<tr>
  <td>Connect string: </td>
  <td><input type="text" name="db" size="10" align="left"></td>
</tr>
<tr>
  <td>Report to run: </td>
  <td><input type="text" name="script" value="employee.sql" size=40></td>
</tr>
<tr>
  <td><input type="submit" value="Run it">&nbsp;<input type="reset" value="Reset
Form"></td>
  <td>&nbsp;</td>
</tr>
</table>
</form>
</body>
</html>

```

### CGI を使用した Perl スクリプトの例

次の例で示す CGI スクリプトは Perl スクリプトですが、シェル・スクリプト、Java クラスまたは Web サーバーでサポートされているその他の言語でも記述できます。次のような Perl CGI スクリプトを作成し、ご使用の Web サーバーの cgi-bin ディレクトリに plus.pl という名前で保存します。



```

#!/usr/local/bin/perl -w
# Copyright (c) Oracle Corporation 1999, 2002. All Rights Reserved.
# NAME
#   plus.pl
# DESCRIPTION
#   This is a demonstration program to run a SQL*Plus report via CGI.
#   It is provided as is with no warranty implied or expressed.
#   Users are strongly recommended to understand error handling and

```

```
# security issues before implementing CGI scripts.
#
# NOTES
# This demonstration requires that SQL*Plus 9.2.0 (or later) is
# installed on your web server, and the web server is configured to
# run CGI programs. The database may be on another machine, but
# must have Oracle Net access configured.
#
# This demonstration consists of three files:
# plus.html - Sample HTML form that you open in your web
# browser. It calls plus.pl to run employee.sql
# plus.pl - Sample CGI program to run SQL*Plus
# employee.sql - Sample SQL script to generate a report from
# the HR sample schema.
# These scripts need to be customized for your site.
#
# INSTALLATION INSTRUCTIONS:
# 1. Put plus.pl (this file) in the cgi-bin directory of your
# web server and edit the environment variable section at the top
# of the file. Make the program executable, for example on UNIX,
# chmod +x plus.pl
# You may need to customize the top line of this script to point
# to the Perl installation on your machine, and in the syntax
# required for your operating system.
# 2. Put employee.sql in the cgi-bin directory too.
# 3. Put plus.html in a directory you can access from the web.
# Edit plus.html to change the form URL to that of your web server.
# 4. Open plus.html in your browser and enter the fields. As
# this demonstration uses the view, EMP_DETAILS_VIEW, from the HR
# sample schema, enter the associated username, HR, and password.
# If your database is not the default, or is on another machine,
# enter a valid network alias, or full connection identifier in
# the Connect Identifier field, otherwise leave it blank. If
# employee.sql is in your cgi-bin directory, you will probably
# not need to specify a path, otherwise specify a machine path
# and filename.

$debug = 0; # Set this to 1 to see the form fields values entered.
# !!! Customize these environment variables and the executable name.
# !!! On Windows use "$ENV{'ORACLE_HOME'}\bin\sqlplus" for the executable.

# Set up the SQL*Plus environment
$ENV{'ORACLE_SID'} = "Ora9i"; # Your SID goes here
$ENV{'ORACLE_HOME'} = "/oracle"; # Your Oracle Home directory goes here
# $ENV{'TNS_ADMIN'} = "/var/opt/oracle";
$plusexe = "$ENV{'ORACLE_HOME'}/bin/sqlplus";
```

```

# Extract parameters and values from data entered through web browser
$i=<>;
@in = split(/[&;]/,$i);
foreach $i (0 .. $#in)
{ ($key,$val) = split(=/,$in[$i],2);
# Change encoding to machine character set
  $key =~ s/%([A-Za-f0-9]{2})/pack("c",hex($1))/ge;
  $val =~ s/%([A-Za-f0-9]{2})/pack("c",hex($1))/ge;
# Store the value
  $in{"$key"} = $val;
}

# Construct the connection string from values passed in
$connstr = $in{'username'}."/".$in{'password'};
$connstr = $connstr."@".$in{'db'} if ($in{'db'});

# Construct the SQL script to be run
$script = "@".$in{'script'};

# Force output to be flushed
$| = 1;

# Print mime type
print "Content-Type: text/html¥n¥n";

if ($debug)
{ print "<html><body>¥n";
  print "$$plusexe:$connstr:$script:¥n";
  print "</body></html>¥n";
  exit;
}

# Call SQL*Plus with the parameters entered from the web browser
system ("$$plusexe -r 3 -s -m ¥"html on¥" $connstr $script");
exit;

```

### CGI を使用した SQL スクリプトの例

テキスト・エディタで次のような SQL\*Plus スクリプトを作成し、ご使用の Web サーバーの cgi-bin ディレクトリに employee.sql という名前で保存します。

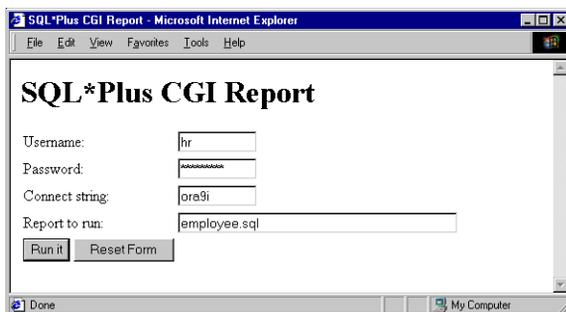


```

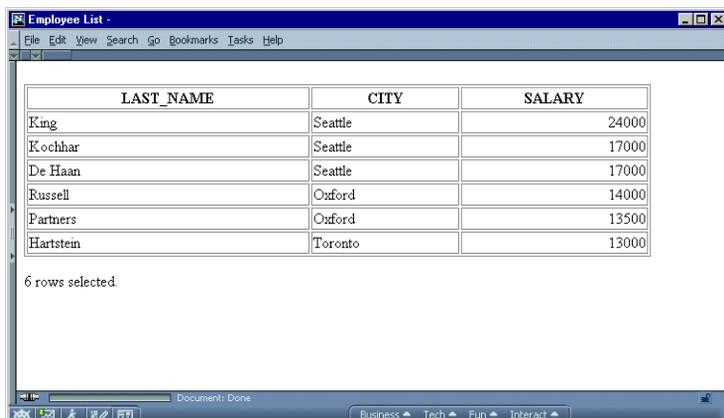
SELECT LAST_NAME, CITY, SALARY
FROM EMP_DETAILS_VIEW;
EXIT;

```

Web ブラウザを起動し、適切な URL を入力して plus.html を開きます。



「Run it」をクリックし、シェル・スクリプト plus.pl を実行すると、SQL\*Plus が起動され、employee.sql スクリプトが実行されます。問合せ結果は、直接 Web ブラウザに表示されます。



LAST_NAME	CITY	SALARY
King	Seattle	24000
Kochhar	Seattle	17000
De Haan	Seattle	17000
Russell	Oxford	14000
Partners	Oxford	13500
Hartstein	Toronto	13000

6 rows selected.

## レポートでの SQL\*Plus コマンドの非表示

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。

## 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。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 8-1 対応する HTML エンティティ**

文字	HTML エンティティ	意味
<	&lt;	HTML タグ・ラベルの開始
>	&gt;	HTML タグ・ラベルの終了
"	&quot;	二重引用符
&	&amp;	アンバサンド

Web ブラウザには「>」が表示されますが、エンコードされた実際の HTML ファイルでの HTML エンティティは、「&gt;」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。

## iSQL\*Plus を使用したレポートの作成

スクリプトを実行する要求を URL から iSQL\*Plus に送信すると、動的レポートを作成し、変数をスクリプトに渡すことができます。スクリプトは、HTTP POST プロトコルを使用してダウンロードし、HTTP、HTTPS または FTP を介して使用できるようにする必要があります。iSQL\*Plus では、HTML 形式のフィールド値がパラメータとして使用されてスクリプトが実行され、その結果が新しい Web ブラウザ・ウィンドウに戻されます。

未定義の変数の入力を求めるプロンプトは表示されません。スクリプト内に未定義の変数がないように、またはパラメータとして明示的に渡される変数がないように注意する必要があります。

要求には、ユーザー名およびパスワードの情報を含めることもできます。HTML ファイルにユーザー名およびパスワードを含める場合は、セキュリティについて慎重に検討してください。ユーザー名またはパスワードを含めない場合、iSQL\*Plus では、スクリプトの実行時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。

SET MARKUP コマンドを使用して、レポート用の HEAD または BODY オプションを変更する場合は、出力を生成する最初のコマンドの前に SET MARKUP コマンドを配置します。

#### 例 8-4 動的レポートの作成

次のスクリプトを作成し、Oracle HTTP Server 上にある `script.sql` という名前のファイルに保存します。

```
SET PAGESIZE 200
SELECT *
FROM EMP_DETAILS_VIEW
ORDER BY LAST_NAME, EMPLOYEE_ID
/
```

次の内容を含む HTML ファイルを作成します。

```
<HTML>
<HEAD>
<TITLE>iSQL*Plus Dynamic Report</TITLE>
</HEAD>
<BODY>
<H1><em>i</em>SQL*Plus Report</H1>
<A
  HREF="http://machine_name.domain:port/isqlplus?script=http://machine_name.domain:port/script.sql">
  Run Employee Report</A>
</BODY>
</HTML>
```

`machine_name.domain` をホスト名およびドメイン名で置き換え、`port` を Oracle HTTP Server のポート番号で置き換えます。この HTML ファイルを Oracle HTTP Server に保存します。

Web ブラウザに HTML ファイルをロードし、「Run Employee Report」をクリックします。iSQL\*Plus で、ユーザー名およびパスワードが要求されます。iSQL\*Plus にログインします。iSQL\*Plus でスクリプトが実行され、その結果が Web ブラウザに表示されます。

#### 例 8-5 パラメータを含む動的レポートの作成

次のスクリプトを作成し、Oracle HTTP Server 上にある `employee_name.sql` という名前のファイルに保存します。

```
SET VERIFY OFF
SET PAGESIZE 200
SET FEEDBACK OFF
SET MARKUP HTML ENTMAP OFF
PROMPT <H1>Employee Details for Employee(s) with Last Name like &last_name%</H1>
SET MARKUP HTML ENTMAP ON
SELECT *
FROM EMPLOYEES
WHERE UPPER(last_name) LIKE UPPER('&last_name%')
/
```

次の内容を含む HTML ファイルを作成します。

```
<HTML>
<HEAD>
<TITLE><em>i</em>SQL*Plus Dynamic Report</TITLE>
</HEAD>
<BODY>
<H1><em>i</em>SQL*Plus Report</H1>
<H2>Query by Last Name</H2>
<FORM METHOD=get ACTION="http://machine_name.domain:port/isqlplus">
<INPUT TYPE="hidden" NAME="script"
VALUE="http://machine_name.domain:port/employee_name.sql">
Enter last name of employee: <INPUT TYPE="text" NAME="last_name" SIZE="20">
<INPUT TYPE="submit" VALUE="Run Report">
</FORM>
</BODY>
</HTML>
```

INPUT TYPE の名前は、スクリプト内の列または置換変数と同じにします。次に例を示します。

```
<INPUT TYPE="text" NAME="last_name" SIZE="20">
```

この例では、employee\_name.sql スクリプト内の置換変数 &last\_name にマップします。

machine\_name.domain をホスト名およびドメイン名で置き換え、port を Oracle HTTP Server のポート番号で置き換えます。この HTML ファイルを Oracle HTTP Server に保存します。

HTML ファイルを Web ブラウザにロードします。テキスト・フィールドに名前または名前の一部を入力します。たとえば、「Fay」と入力します。「Run Report」ボタンをクリックします。iSQL\*Plus でスクリプトが実行され、その結果が Web ブラウザに表示されます。

#### 例 8-6 パラメータおよびログインの詳細を含む動的レポートの作成

次のスクリプトを作成し、Oracle HTTP Server 上にある employee\_id.sql という名前のファイルに保存します。

```
SET VERIFY OFF
SET PAGESIZE 200
SET MARKUP HTML ENIMAP OFF
PROMPT <H1>Employee Details for Employee Number &eid</H1>
SET MARKUP HTML ENIMAP ON
SELECT *
FROM EMPLOYEES
WHERE EMPLOYEE_ID = &eid
/
```

次の内容を含む HTML ファイルを作成します。

```
<HTML>
<HEAD>
<TITLE>iSQL*Plus Dynamic Report</TITLE>
</HEAD>
<BODY>
<H1><em>i</em>SQL*Plus Report</H1>
<H2>Query by Employee ID</H2>
<FORM METHOD=get ACTION="http://machine_name.domain:port/isqlplus">
<INPUT TYPE="hidden" NAME="userid" VALUE="hr/your_password">
<INPUT TYPE="hidden" NAME="script"
VALUE="http://machine_name.domain:port/employee_id.sql">
Enter employee identification number: <INPUT TYPE="text" NAME="eid" SIZE="10">
<INPUT TYPE="submit" VALUE="Run Report">
</FORM>
</BODY>
</HTML>
```

*machine\_name.domain* をホスト名およびドメイン名で置き換え、*port* を Oracle HTTP Server のポート番頭で置き換え、*hr/your\_password* を有効なユーザー ID およびパスワードで置き換えます。この HTML ファイルを Oracle HTTP Server に保存します。

HTML ファイルを Web ブラウザにロードします。テキスト・フィールドに従業員識別番号を入力します。たとえば、「105」のように入力します。「Run Report」ボタンをクリックします。iSQL\*Plus でスクリプトが実行され、その結果が Web ブラウザに表示されます。



---

---

## 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。ここでは、次の項目について説明します。

- 文のトレース
- タイミング統計の収集
- パラレル問合せおよび分散問合せのトレース
- システム変数による SQL\*Plus のパフォーマンスへの影響
- iSQL\*Plus サーバーの統計
- iSQL\*Plus のチューニングに関する注意事項

Oracle9i のチューニングの詳細は、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』および『Oracle9i データベース・パフォーマンス・プランニング』を参照してください。

## 文のトレース

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。

## 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。

SET AUTOTRACE OFF	AUTOTRACE レポートは生成されません。これがデフォルトです。
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 表の作成方法の詳細は、『Oracle9i SQL 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、『Oracle9i SQL リファレンス』および 13-92 ページの「SET」コマンドの AUTOTRACE 変数を参照してください。

### 例 9-1 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/your_password
@${ORACLE_HOME}/RDBMS/ADMIN/UTLXPLAN.SQL
```



```
Table created.
```

**例 9-2 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 9-3 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.
```

**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
```

デフォルトの書式は、サイト・プロファイル（たとえば、glogin.sql）で定義されています。

実行計画の出力は、EXPLAIN PLAN コマンドを使用して生成します。実行計画の出力の解釈については、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

## 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。

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。

結果には、次の統計が含まれます。データベース統計の完全なリストについては、『Oracle9i データベース・リファレンス』を参照してください。統計およびその解釈方法については、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

データベース統計名	説明
recursive calls	ユーザーおよびシステム・レベルで生成された再帰的コールの数。Oracle 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ある場合、Oracle 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み込み一貫性が要求された回数。
physical reads	ディスクからのデータ・ブロック読み込みの合計数。「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（バイト単位）。
bytes sent via SQL*Net 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 via SQL*Net 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。

データベース統計名	説明
SQL*Net roundtrips 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)	完全にメモリー内で実行され、ディスク書込みを必要としなかったソート操作の数。
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の数。
rows processed	操作中に処理された行の数。

#### 例 9-4 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 パ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
/
```



```
LAST_NAME                SALARY JOB_TITLE
-----
King                      24000 President
Kochhar                   17000 Administration Vice President
De Haan                   17000 Administration Vice President
Russell                   14000 Sales Manager
Partners                  13500 Sales Manager
Hartstein                 13000 Marketing Manager
```

6 rows selected.

#### Execution Plan

```
-----
0      SELECT STATEMENT Optimizer=CHOOSE
1      0    TABLE ACCESS (BY INDEX ROWID) OF 'EMPLOYEES'
2      1      NESTED LOOPS
3      2        TABLE ACCESS (FULL) OF 'JOBS'
4      2        INDEX (RANGE SCAN) OF 'EMP_JOB_IX' (NON-UNIQUE)
```

#### Statistics

```
-----
0      recursive calls
2      db block gets
```

```

34 consistent gets
0 physical reads
0 redo size
848 bytes sent via SQL*Net to client
503 bytes received via SQL*Net from client
4 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
6 rows processed

```

### 例 9-5 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
/

```



```

6 rows selected.

```

#### Execution Plan

```

-----
0      SELECT STATEMENT Optimizer=CHOOSE
1  0    TABLE ACCESS (BY INDEX ROWID) OF 'EMPLOYEES'
2    1      NESTED LOOPS
3      2        TABLE ACCESS (FULL) OF 'JOBS'
4      2          INDEX (RANGE SCAN) OF 'EMP_JOB_IX' (NON-UNIQUE)

```

#### Statistics

```

-----
0 recursive calls
2 db block gets
34 consistent gets
0 physical reads
0 redo size
848 bytes sent via SQL*Net to client
503 bytes received via SQL*Net from client
4 SQL*Net 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。

### 例 9-6 データベース・リンクを使用した文のトレース

データベース・リンクを使用して文をトレースするには、次のように入力します。



```
SET AUTOTRACE TRACEONLY EXPLAIN
SELECT * FROM EMPLOYEES@MY_LINK;
```



Execution Plan

```
-----
0      SELECT STATEMENT (REMOTE) Optimizer=CHOOSE
1    0  TABLE ACCESS (FULL) OF 'EMPLOYEES' MY_LINK.DB_DOMAIN
```

この実行計画は、行 1 でアクセスしている表がデータベース・リンク MY\_LINK.DB\_DOMAIN を経由していることを示しています。

## タイミング統計の収集

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。

詳細は、13-136 ページの「TIMING」コマンド、およびご使用のオペレーティング・システム固有の Oracle インストレーション・ガイドと管理者リファレンスを参照してください。統計を収集する AUTOTRACE の使用については、「文のトレース」も参照してください。

すべてのタイマーを削除するには、コマンド・プロンプトで CLEAR TIMING を入力します。

## 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する場合、実行計画には、コストベースの最適化の行数見積り（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、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』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	Real Application Clusters またはリモート・データベースについて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。デフォルトの書式は、サイト・プロファイル（たとえば、glogin.sql）で定義されています。

### 例 9-7 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、次のようにします。



```
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 LONGCHUNKSIZE 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
```

```
-----
0      SELECT STATEMENT Optimizer=CHOOSE (Cost=1 Card=1 Bytes=26)
1      0      SORT (AGGREGATE)
2      1      SORT* (AGGREGATE)                               :Q2000
3      2      NESTED LOOPS* (Cost=1 Card=41 Bytes=1066)      :Q2000
4      3      TABLE ACCESS* (FULL) OF 'D2_T2' (Cost=1 Card=41 Byte :Q2000
```

```

s=533)

5      3      INDEX* (UNIQUE SCAN) OF 'D2_I_UNIQUE1' (UNIQUE)      :Q2000

2 PARALLEL_TO_SERIAL      SELECT /*+ PIV_SSF */ SYS_OP_MSR(COUNT(A1.C0
                           )) FROM (SELECT /*+ ORDERED NO_EXPAND USE_NL
                           (A3) INDEX(A3 "D2_I_UNIQUE1") */ A2.C0 C0,A3
                           .ROWID C1,A3."UNIQUE1" C2 FROM (SELECT /*+ N
                           O_EXPAND ROWID(A4) */ A4."UNIQUE1" C0 FROM "
                           D2_T2" PX_GRANULE(0, BLOCK_RANGE, DYNAMIC) A4)
                           A2, "D2_T1" A3 WHERE A2.C0=A3."UNIQUE1")A1

3 PARALLEL_COMBINED_WITH_PARENT
4 PARALLEL_COMBINED_WITH_PARENT
5 PARALLEL_COMBINED_WITH_PARENT

```

この実行計画の行 0 は、コストベースのオブティマイザの見積りでは、行数が 1 で 26 バイトであることを示しています。この文の合計コストは 1 です。

行 2、3、4 および 5 には、パラレル操作であることを示すアスタリスクが付いています。たとえば、行 3 の NESTED LOOPS ステップは、PARALLEL\_TO\_SERIAL 操作です。PARALLEL\_TO\_SERIAL 操作では、出力を順次に生成するために SQL 文が実行されます。また、行 2 はパラレル問合せサーバーの識別子が Q2000 であったことも示しています。

パラレル・レポートの行を表す数字は、親レポートの行と相互に参照しています。たとえば、前述の例の次の行を見てください。

```
4 PARALLEL_COMBINED_WITH_PARENT
```

この行の 4 は、親レポートの「43 TABLE ACCESS\*...」の行を参照しています。

### 例 9-8 ディスク読み込みおよびバッファ取得の監視



```
SET AUTOTRACE ON TRACEONLY STATISTICS
```

結果は次のようになります。



```
Statistics
```

```
-----
70 recursive calls
 0 db block gets
591 consistent gets
404 physical reads
 0 redo size
315 bytes sent via SQL*Net to client
850 bytes received via SQL*Net from client
```

```
3 SQL*Net roundtrips to/from client
3 sorts (memory)
0 sorts (disk)
0 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。

---

---

**注意：** V\$SQL または TKPROF を使用して、ディスク読みおよびバッファ取得を監視することもできます。

---

---

## システム変数による SQL\*Plus のパフォーマンスへの影響

次のシステム変数は、SQL\*Plus のパフォーマンスに影響を与える場合があります。

### 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、スクリプトのパフォーマンスおよびリソース使用率の登録および監視が使用禁止になります。これによって、オーバーヘッドが減少し、パフォーマンスが向上します。

### SET ARRAYSIZE

SQL\*Plus で、一度にデータベースからフェッチされるバッチと呼ばれる行の数を設定します。有効な値は 1 ~ 5000 です。値を大きくすると、多数の行がフェッチされ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 操作の結果に対する影響はありません。

### SET DEFINE OFF

SQL\*Plus で、置換変数用にスクリプトを解析するかどうかを制御します。DEFINE が OFF の場合、SQL\*Plus では、置換変数用のスクリプトは解析されません。スクリプトで置換変数が使用されない場合、DEFINE に OFF を設定すると、パフォーマンスがいくらか向上します。

## SET FLUSH OFF

SET FLUSH は、iSQL\*Plus ではサポートされていません。

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 を設定すると、ホスト・オペレーティング・システムで、出力がバッファに格納されます。これによって、プログラムによる入出力が削減され、パフォーマンスが向上します。

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。

## SET SERVEROUTPUT

SQL\*Plus で、DBMS 出力を確認および表示するかどうかを制御します。SERVEROUTPUT が OFF の場合、SQL\*Plus では DBMS 出力は確認されません。また、適用可能な SQL または PL/SQL 文の実行後に出力結果は表示されません。このように、出力結果の確認および表示を行わないことによって、パフォーマンスが向上します。

## SET TRIMOUT ON

SET TRIMOUT は、iSQL\*Plus ではサポートされていません。

SQL\*Plus で、それぞれの表示行の終わりに後続空白を入れるかどうかを判断します。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。TRIMOUT ON は、スプール出力には影響を与えません。

## SET TRIMSPool ON

SET TRIMSPool は、iSQL\*Plus ではサポートされていません。

SQL\*Plus で、それぞれのスプール行の終わりに後続空白を入れるかどうかを判断します。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。TRIMSPool ON は、端末出力には影響を与えません。

# iSQL\*Plus サーバーの統計

iSQL\*Plus サーバーの統計レポートのアクティブ統計では、iSQL\*Plus サーバーのチューニングに対して、有効なフィードバックが提供されます。

レポートを実行するには、iSQL\*Plus DBA の URL にアクセスするための Oracle HTTP Server 認証が必要ですが、データベースには接続しないため、Oracle9i にログインする必要はありません。iSQL\*Plus サーバーの統計レポートを実行するための URL の構文は、次のとおりです。

```
http://machine_name.domain:port/isqlplusdba?statistics={active|full}
[&refresh=number]
```

リソースの可用性を最大にするには、iSQL\*Plus の各ユーザーに、制限が適切に定義されたデータベース・スキーマ・プロファイルを設定することをお勧めします。

iSQL\*Plus サーバーの完全な統計レポートの完全な構文および詳細は、2-12 ページの「[iSQL\\*Plus サーバーの統計](#)」を参照してください。次に、アクティブ統計について説明します。

## アクティブ統計

アクティブ統計レポートでは、次の項目の現在の値が示されます。

Sessions active	同時アクティブ・セッションの数または現在 iSQL*Plus にログインしているセッションの数。
Sessions since startup	iSQL*Plus サーバーの起動後に確立されたセッションの累計。
Maximum concurrent sessions	iSQL*Plus サーバーの起動後の同時セッションの最大数またはピーク時の数。
Sessions expired since startup	iSQL*Plus サーバーの起動後、動作が停止したためにタイムアウトが発生したセッションの累計。
Requests active	アクティブな同時 HTTP 要求の数。各要求は、ボタンのクリックなどのユーザーの動作、および iSQL*Plus によるその要求の処理に対応します。アクティブな要求の最大数は、iSQLPlusNumberOfThreads によって設定されます。アクティブな要求の数がこの制限に達したときに、ユーザー応答時間が十分でない場合は、iSQLPlusNumberOfThreads の値を増やすことで、応答時間を延長できます。
Requests since startup	iSQL*Plus サーバーの起動後のアクティブな HTTP 要求の累計。
Next expiry operation (minutes)	次の期限切れプロセスまでの時間 (分)。端数は切り捨てます。
Expiry operations since startup	iSQL*Plus サーバーの起動後に期限切れプロセスが実行された回数。
Hash table collisions	現在、ハッシュ表で衝突が発生しているアクティブ・セッションの数。衝突に関する問題の有無を確認するには、この値と Sessions active の値を比較します。
Hash table collisions since startup	iSQL*Plus サーバーの起動後に、ハッシュ表で衝突が発生したセッションの累計。衝突に関する問題の有無を確認するには、この値と Sessions since startup の値を比較します。

## アクティブ統計の解釈

次に、アクティブ統計の解釈について説明します。

### スレッド数の増加

ユーザーのアイドル時間がアクティブ時間より長い場合、`iSQLPlusNumberOfThreads` の値を増やす必要があります。各スレッドで処理できるユーザー要求は1つです。

要求は、ユーザーがボタンをクリックしたとき、または `iSQL*Plus` でコマンド・リンクをたどったときに開始し、すべての結果がユーザーに戻されたときに終了します。アクティブ時間とは、ユーザー要求が処理中で、そのユーザー要求を処理するためのスレッドが消費される時間です。アイドル時間とは、ユーザー要求が処理され、そのユーザー要求を処理するための関連スレッドが、別の `iSQL*Plus` セッションで使用可能である時間です。

`isqlplus.conf` に `iSQLPlusHashTableSize` が指定されていない場合、`iSQLPlusNumberOfThreads` の値を増やすと、`iSQLPlusHashTableSize` の値も増えることに注意してください。

### ハッシュ表のサイズの増加

通常、ユーザーのアイドル時間がアクティブ時間より長い場合、`iSQLPlusNumberOfThreads` の特定の値に対応するように、`iSQLPlusHashTableSize` の値を増やす必要があります。各ユーザー・セッションでは、アイドルの状態でもハッシュ表の1エントリが使用されます。

### タイムアウト時間の短縮

多数のセッションでタイムアウトが発生している場合、ユーザーが正常にログアウトされておらず、セッションがアイドル状態であることを表します。このとき、`iSQL*Plus` サーバーの負荷が高い場合は、`iSQLPlusTimeOutInterval` の値を減らして、セッションのタイムアウトをさらに多く発生させることを検討します。

### アイドル・タイムアウト

アイドル・タイムアウトとは、Oracle HTTP Server が `iSQL*Plus` から結果が戻されるまで待機する時間です。FastCGI タイムアウト・パラメータの値 `-idle-timeout` は、3600 秒に設定されています。この値によって、Web サーバーの前に `iSQL*Plus` がタイムアウトすることを回避できます。実行時間の長い多くの問合せで、`iSQL*Plus` がタイムアウトする前に結果が戻されるようにするには、十分な値です。

アイドル・タイムアウトを、アイドル状態のユーザー・セッションで使用するリソースの継続期間を管理する `iSQLPlusTimeOutInterval` と混同しないでください。

## iSQL\*Plus のチューニングに関する注意事項

ORACLE\_HOME/Apache/Apache/logs/error\_log のログに、頻繁に次の iSQL\*Plus エラーが表示される場合があります。

```
[Mon Feb 14 17:08:09 2005] [error] [client 148.87.9.44]
(146)Connection refused: FastCGI: failed to connect to server
"/oracle/sqlplus/bin/isqlplus": connect() failed
```

または、Oracle HTTP Server によって、次のエラー・ページが生成される場合があります。

```
Internal Server Error
The server encountered an internal error or misconfiguration and was unable to
complete your request.
```

これは、iSQL\*Plus で要求の負荷に対処できなかったことを示します。通常、iSQLPlusNumberOfThreads の値が小さすぎるのが原因です。

# 10

---

---

## SQL\*Plus 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メソッ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。

- `PRODUCT_USER_PROFILE` 表
- 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
- `SQLPLUS -RESTRICT` でのコマンドの使用禁止
- iSQL\*Plus のセキュリティ

## PRODUCT\_USER\_PROFILE 表

各種の Oracle 製品で、SYSTEM アカウント内の表の 1 つである PRODUCT\_USER\_PROFILE (PUP) 表を使用した製品レベルのセキュリティが提供されています。このセキュリティは、SQL の GRANT と REVOKE コマンドおよびユーザー・ロールによるユーザー・レベルのセキュリティを補います。

SQL\*Plus 環境では、DBA は PUP 表を使用して、特定の SQL および SQL\*Plus コマンドをユーザー単位で使用禁止にできます。Oracle ではなく SQL\*Plus でこのセキュリティが実行されます。DBA は、GRANT、REVOKE および SET ROLE コマンドのアクセスを制限して、ユーザーによる各自のデータベース権限の変更を制御することもできます。

SQL\*Plus では、ユーザーが SQL\*Plus にログインすると、PUP 表から制限が読み込まれ、セッションが終了するまでその制限が保持されます。PUP 表の変更は、次に対象ユーザーが SQL\*Plus にログインするときに有効になります。

SYSTEM、SYS、または SYSDBA または SYSOPER 権限で認証されたユーザーが接続またはログインする場合、SQL\*Plus では PUP 表は読み込まれません。そのため、これらのユーザーに制限は適用されません。

PUP 表は、ローカル・データベースのみに適用されます。データベース・リンク経由でリモート・データベース上のオブジェクトにアクセスする場合、リモート・データベース用の PUP 表は適用されません。リモート・データベースは、データベース・リンクからユーザー名およびパスワードを抽出して、ユーザー・プロファイルおよびユーザー権限を維持することはできません。

## PUP 表の作成

PUP 表を作成するには、拡張子が SQL の PUPBLD という名前のスクリプトを、SYSTEM で実行します。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。詳細は、ご使用のオペレーティング・システム固有の Oracle インストール・ガイドおよび管理者リファレンスを参照するか、または DBA に問い合わせてください。

---

---

**注意：** 表の作成に誤りがある場合、権限が付与されたユーザー以外のすべてのユーザーが Oracle に接続するときに、PUP 表の情報がロードされていないという警告が表示されます。

---

---

## PUP 表の構造

PUP 表には、次の列が含まれています。

PRODUCT	NOT NULL CHAR (30)
USERID	CHAR (30)
ATTRIBUTE	CHAR (240)
SCOPE	CHAR (240)
NUMERIC_VALUE	NUMBER (15, 2)
CHAR_VALUE	CHAR (240)
DATE_VALUE	DATE
LONG_VALUE	LONG

## PUP 列の説明および使用方法

次に、PUP 表の各列について説明します。

Product	製品名（この場合は SQL*Plus）が含まれている必要があります。この列には、ワイルド・カードまたは NULL は入力できません。
Userid	<p>コマンドを使用禁止にする対象ユーザーのユーザー名（大文字）が含まれている必要があります。複数のユーザーのコマンドを使用禁止にするには、SQL ワイルド・カード (%) を使用するか、または複数のエントリを入力します。したがって、次のエントリはすべて有効です。</p> <ul style="list-style-type: none"> <li>■ HR</li> <li>■ CLASS1</li> <li>■ CLASS%（名前が CLASS で始まるすべてのユーザー）</li> <li>■ %（すべてのユーザー）</li> </ul>
Attribute	<p>使用禁止にする SQL、SQL*Plus または PL/SQL コマンドの名前（大文字）が含まれている必要があります（たとえば、RUN など）。ルールを使用禁止にする場合、文字列 ROLES が含まれている必要があります。ワイルド・カードは入力できません。使用禁止にできる SQL および SQL*Plus コマンドのリストについては、「<a href="#">PUP 表の管理</a>」を参照してください。ルールを使用禁止にする方法の詳細は、「<a href="#">ロールの作成および制御</a>」を参照してください。</p>
Scope	<p>使用されません。NULL を入力してください。他の製品では、特定のファイル制限またはその他のデータがこの列に格納されることがあります。</p>

Numeric_Value	使用されません。NULL を入力してください。他の製品では、この列に数値が格納されることがあります。
Char_Value	SQL、SQL*Plus または PL/SQL コマンドを使用禁止にするには、文字列 DISABLED が含まれている必要があります。ロールを使用禁止にする場合は、使用禁止にするロールの名前が含まれている必要があります。ワイルド・カードは入力できません。ロールを使用禁止にする方法の詳細は、「 <a href="#">SQLPLUS -RESTRICT でのコマンドの使用禁止</a> 」を参照してください。
Date_Value	使用されません。NULL を入力してください。他の製品では、この列に DATE 値が格納されることがあります。
Long_Value	使用されません。NULL を入力してください。他の製品では、この列に LONG 値が格納されることがあります。

## PUP 表の管理

ユーザー名 SYSTEM の DBA は、PUP 表を所有し、それに関するすべての権限を持っています。他の Oracle ユーザー名では、この表に対しては SELECT 権限のみを持つようになります。このアクセス権限を使用すると、ユーザー名に設定された制限および PUBLIC に設定された制限を表示できます。スクリプト PUPBLD.SQL を実行すると、PUP 表に対する SELECT 権限が PUBLIC に付与されます。

## SQL\*Plus、SQL および PL/SQL コマンドの使用禁止

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
-----	-----	-----	-----	-----	-----	-----
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。

次に、使用禁止にできる SQL\*Plus コマンドを示します。

COPY	HOST	SET
EDIT	PASSWORD	SPOOL
EXECUTE	QUIT	START
EXIT	RUN	
GET	SAVE	

---

---

**注意：**

- 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（VMS 環境では \$、UNIX 環境では ! など）も使用禁止になります。
  - 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 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。
- 
- 

次に、使用禁止にできる SQL コマンドを示します。

ALTER	GRANT	SET CONSTRAINTS
ANALYZE	INSERT	SET ROLE
AUDIT	LOCK	SET TRANSACTION
CONNECT	NOAUDIT	TRUNCATE
CREATE	RENAME	UPDATE
DELETE	REVOKE	
DROP	SELECT	

次に、使用禁止にできる PL/SQL コマンドを示します。

BEGIN	DECLARE
-------	---------

---

---

**注意：** BEGIN および DECLARE コマンドを使用禁止にしても、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。

---

---

### 例 10-1 PUP 表の制限の設定

次に、PUP 表に、ユーザー HR の SELECT 文の使用を制限する行を挿入する例を示します。

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/your_password
```

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/your_password;  
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。



```
SP2-00544: コマンド: SELECT が無効です。
```

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 SYSDBA 権限を持つユーザーとして再接続し、次のように入力します。



```
DELETE FROM PRODUCT_USER_PROFILE WHERE USERID = 'HR';
```

## ロールの作成および制御

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。

ロールの詳細は、『Oracle9i SQL リファレンス』、『Oracle9i データベース管理者ガイド』および『Oracle9i データベース概要』を参照してください。

## SET ROLE の使用禁止

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、『Oracle9i SQL リファレンス』および『Oracle9i データベース管理者ガイド』を参照してください。

## ユーザー・ロールの使用禁止

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。

---

**注意：** 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 を入れます。次に例を示します。

PRODUCT	USERID	ATTRIBUTE	SCOPE	NUMERIC VALUE	CHAR VALUE	DATE 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。詳細は、「SET ROLE の使用禁止」を参照してください。

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。

## SQLPLUS -RESTRICT でのコマンドの使用禁止

PUP 表と同様に、RESTRICT オプションで、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。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

### 注意：

- 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（VMS 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、4-9 ページの「[RESTRICT オプション](#)」を参照してください。

## iSQL\*Plus のセキュリティ

iSQL\*Plus の使用時に、セキュリティとユーザー認証を考慮する必要があるのは、主に次の 2 つの場合です。

- Web ブラウザと Oracle HTTP Server 間の HTTP プロトコル接続
- iSQL\*Plus サーバー (Oracle HTTP Server 内) と Oracle9i データベース間の Oracle Net 接続

iSQL\*Plus では、中間層のオペレーティング・システムにアクセスして、オペレーティング・システムへのアクセスを伴う HOST、EDIT、SPOOL などのコマンドを実行することはできません。

iSQL\*Plus では、Web ブラウザと Oracle HTTP Server 間の接続に対するセキュリティは、Oracle で完全にサポートされている標準の HTTPS によって行われます。このセキュリティによって、Secure Sockets Layer (SSL) を介して、Oracle 提供の暗号化メカニズムを使用した保護リスナー接続が可能になります。このセキュリティは、Oracle HTTP Server のインストール時に `mod_ssl` モジュールをインストールすることで実装できます。Oracle における HTTPS セキュリティの実装については、『Oracle Advanced Security 管理者ガイド』を参照してください。

iSQL\*Plus サーバーと Oracle9i データベース間の Oracle Net 接続では、従来のクライアント / サーバー・アーキテクチャと同じセキュリティが提供されます。Oracle Net 接続のセキュリティの詳細は、『Oracle9i Net Services 管理者ガイド』および『Oracle Advanced Security 管理者ガイド』を参照してください。

## 管理権限

iSQL\*Plus へのアクセスには、次の 3 つのモードがあります。

- 通常のユーザーとしての接続  
iSQL\*Plus の「Login」画面で、Oracle データベース・アカウントのユーザー名とパスワードを入力する必要があります。
- SYSDBA または SYSOPER 権限を持つユーザーとしての接続  
iSQL\*Plus DBA の「Login」画面で Oracle データベース・アカウントのユーザー名とパスワードを、また別のダイアログ・ボックスで Oracle HTTP Server 認証のユーザー名とパスワードを入力する必要があります。
- iSQL\*Plus サーバーの統計レポートの生成  
Oracle HTTP Server 認証のユーザー名とパスワードが必要です。

## ユーザーのサーバー認証の有効化

デフォルトでは、Oracle HTTP Server 認証は、SYSDBA および SYSOPER 権限を持つユーザーにのみ有効です。通常のユーザー・ログインに対する Oracle HTTP Server 認証を有効にすることによって、iSQL\*Plus にアクセスできるユーザーを制限する場合があります。この場合、`isqlplus.conf` ファイルの次の行を編集すると、通常のユーザー接続に対する Oracle HTTP Server 認証が有効になります。

```
<Location /isqlplus>
  SetHandler fastcgi-script
  Order deny,allow
  Allow from all
</Location>
```

次のように変更します。

```
<Location /isqlplus>
  SetHandler fastcgi-script
  Order deny,allow
  AuthType Basic
  AuthName 'iSQL*Plus'
  # The following line for UNIX, comment out the Windows line.
  # AuthUserFile $ORACLE_HOME/sqlplus/admin/iplus.pw
  # The following line for Windows, comment out the UNIX line.
  AuthUserFile %ORACLE_HOME%\sqlplus\admin\iplus.pw
  Require valid-user
</Location>
```

`isqlplus.conf` ファイルの編集の詳細は、3-7 ページの「[セッションの整合性](#)」を参照してください。

ユーザー接続に対する Oracle HTTP Server 認証のユーザー名およびパスワードを含めるファイルとして、`iplus.pw` を使用することをお勧めします。これによって、ユーザー接続の要求時に、ユーザーは、Oracle9i のユーザー名とパスワード以外に、Oracle HTTP Server 認証のユーザー名とパスワードの入力も要求されます。

## データベース・アクセス制限の有効化または無効化

ユーザーが iSQL\*Plus でアクセスできるデータベースを、制限リストに含まれるデータベースに制限することができます。データベース・アクセス制限が有効な場合、使用可能なデータベースのドロップダウン・リストが「Login」画面の「Connection Identifier」テキスト・フィールドに表示されます。これによって、ホスト環境での iSQL\*Plus サーバーのセキュリティを向上させることができます。接続識別子は、`iSQLPlusConnectIdList` に定義されている順に表示されます。

isqlplus.conf ファイルの次の行を編集すると、データベース・アクセス制限を有効にできます。

```
FastCgiServer ... -initial-env "iSQLPlusConnectIdList=SID1, SID2, ..."
```

SID1, SID2, ... は、アクセス可能なデータベースを指定する Oracle Net 接続識別子をカンマで区切ったリストです。次に例を示します。

```
FastCgiServer ... -initial-env "iSQLPlusConnectIdList=ABC1, PROD2, DEV3"
```

isqlplus.conf ファイルの編集の詳細は、3-7 ページの「[セッションの整合性](#)」を参照してください。

前述のとおり、接続識別子に引用符や空白を使用することはできませんが、iSQLPlusConnectIdList= 引数を引用符で囲む必要があります。接続識別子では、大 / 小文字が区別されます。また、引数にリストされた各接続識別子は、tnsnames.ora ファイルの別名と同じである必要があります。

isqlplus.conf ファイル内のある文には、いくつかの初期化パラメータが設定されています。既存の要素は変更しないでください。

設定後は、「Login」画面で行われるすべての接続、すべての動的レポートおよび CONNECT コマンドで試行されるすべての接続は、接続先が制限リストに含まれるデータベース以外の場合は拒否されます。

同様に、SET INSTANCE が使用されると、定義済の接続識別子は iSQLPlusConnectIdList 内のエントリに一致する必要があります。一致しない場合、接続は拒否されます。

接続識別子が未指定の場合、または指定された識別子が iSQLPlusConnectIdList 内のエントリに一致しない場合は、データベース接続が拒否され、次のエラーが発生します。

```
SP2-00884: データベース database_name への接続は許可されません。
```

## DBA アクセスの有効化

iSQL\*Plus のユーザーは、自動的に SYSDBA および SYSOPER 権限を持つユーザーとして接続することはできないので、システム管理者が設定する必要があります。SYSDBA または SYSOPER 権限で接続するか、あるいは iSQL\*Plus サーバーの統計レポートを生成するには、ユーザー名とパスワードを Oracle HTTP Server の iSQL\*Plus 認証ファイルに追加する必要があります。たとえば、Windows でのインストールでは、認証ファイルはユーザー・エントリを含まずに %ORACLE\_HOME%\sqlplus\admin\iplusdba.pw にインストールされます。認証ファイルで使用するユーザー名とパスワードは、Oracle9i のユーザー名とパスワードとは無関係です。

---

---

**警告：** Oracle HTTP Server を、UNIX の場合は dba または oper グループのメンバー、Windows の場合は ORA\_DBA、ORA\_OPER、ORA\_SID\_DBA または ORA\_SID\_OPER グループのメンバーで起動した場合、iSQL\*Plus DBA の URL は、オペレーティング・システムによって自動的に Oracle9i で認証されます。この問題を回避するには、Oracle HTTP Server を、これらのオペレーティング・システム・グループのメンバーでないユーザーとして起動します。

iSQL\*Plus DBA の URL のユーザーであれば、「/ as sysdba」または「/ as sysoper」を使用して、Oracle9i に対する認証が可能です。

---

---

## Oracle HTTP Server 認証ファイルへのエントリの追加

ユーザー接続に対して Oracle HTTP Server 認証を有効にした場合は、ユーザー・レベル接続に必要なユーザー名とパスワードを格納する認証ファイルを別に作成する必要があります。ユーザー・レベルの Oracle HTTP Server 認証の有効化については、「[ユーザーのサーバー認証の有効化](#)」を参照してください。

Windows で Oracle HTTP Server 認証ファイルに新規のエントリを作成する手順は、次のとおりです。

1. Oracle HTTP Server を実行しているマシンに Oracle HTTP Server 管理者としてログインします。
2. コマンド・プロンプトを起動します。
3. `htpasswd` ユーティリティを実行して、ユーザーを認証ファイルに追加します。通常、`htpasswd` は、`%ORACLE_HOME%\Apache\Apache\bin` に含まれています。SYSDBA ユーザーまたは SYSOPER ユーザーの場合は、次の形式を使用します。

```
htpasswd %ORACLE_HOME%\sqlplus\admin\iplusdba.pw username
```

ユーザー接続の場合、`iplus.pw` は空の認証ファイルとして作成されており、次の形式を使用します。

```
htpasswd %ORACLE_HOME%\sqlplus\admin\iplus.pw username
```

いずれの場合も、対応するパスワードの入力を要求されます。`htpasswd` の詳細は、Oracle HTTP Server のマニュアルを参照してください。

UNIX で Oracle HTTP Server 認証ファイルに新規のエントリを作成する手順は、次のとおりです。

1. Oracle HTTP Server を実行しているマシンに Oracle HTTP Server 管理者としてログインします。
2. 端末を起動します。
3. `htpasswd` ユーティリティを実行して、ユーザーを認証ファイルに追加します。通常、`htpasswd` は、`$ORACLE_HOME/Apache/Apache/bin` に含まれています。SYSDBA ユーザーまたは SYSOPER ユーザーの場合は、次の形式を使用します。

```
htpasswd $ORACLE_HOME/sqlplus/admin/iplusdba.pw username
```

ユーザー接続の場合、`iplus.pw` は空の認証ファイルとして作成されており、次の形式を使用します。

```
htpasswd $ORACLE_HOME/sqlplus/admin/iplus.pw username
```

いずれの場合も、対応するパスワードの入力を要求されます。`htpasswd` の詳細は、Oracle HTTP Server のマニュアルを参照してください。

## セキュリティの使用方法

iSQL\*Plus を理解して構成するために、次の注意事項を参考にしてください。

- SYSDBA 権限または SYSOPER 権限で正常にログインし、Oracle HTTP Server 認証のユーザー名とパスワードが認証されると、ブラウザを再起動するまで Oracle HTTP Server への再認証を要求されない場合があります。ただし、Oracle9i データベースのユーザー名とパスワードはログイン時に要求されます。
- Product User Profile (PUP) 表はユーザー単位です。SQL\*Plus クライアント・サーバーのインストールではデータベース単位です。PUP 表の詳細は、[「PRODUCT\\_USER\\_PROFILE 表」](#)を参照してください。
- クライアント・サーバーのインストールでは、グローバル構成ファイル `glogin.sql` は中間層マシンから読み込まれます。ユーザーの `login.sql` ファイルは読み込まれません。



---

## SQL\*Plus を使用したデータベース管理

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。この章で説明されているコマンドの機能を利用するには、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、『Oracle9i データベース概要』を参照してください。

## 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、[第 10 章「SQL\\*Plus セキュリティ」](#)を参照してください。

## 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 データベース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 データベースの現在の状態を制御できません。

Oracle7 サーバーのインスタンスを起動または停止する場合、STARTUP または SHUTDOWN コマンドは使用できません。

## データベースの起動

データベースの起動には、次の 3 つの手順が含まれています。

### 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 データベース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。

データベースの起動の詳細は、『Oracle9i データベース概要』を参照してください。  
STARTUP コマンドの詳細は、[第 13 章「SQL\\*Plus コマンド・リファレンス」](#)を参照してください。

### 例 11-1 インスタンスの起動

データベースをマウントせずに Oracle 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 11-2 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 11-3 データベースのオープン

Oracle9i サーバーのパラメータ・ファイル INITSALE.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排他モードでマウントおよびオープンして、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=INITSALE.ORA EXCLUSIVE RESTRICT
```

SALES は、INITSALE.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。

## データベースの停止

データベースの停止には、次の 3 つの手順が含まれます。

### 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. データベースのディスマウント

データベースをディ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 インスタンスを構成するバックグラウンドの Oracle プロセスが終了します。

データベースの停止の詳細は、『Oracle9i データベース概要』を参照してください。

SHUTDOWN コマンドの詳細は、第 13 章「SQL\*Plus コマンド・リファレンス」を参照してください。

#### 例 11-4 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
Database closed.
Database dismounted.
ORACLE instance shut down.
```

## REDO ログ・ファイル

各 Oracle データベースには、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 ログ自体を含む障害を防ぐため、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。

## 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、ARCHIVE LOG 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、『Oracle9i データベース概要』を参照してください。ARCHIVE LOG コマンドの詳細は、[第 13 章「SQL\\*Plus コマンド・リファレンス」](#)を参照してください。

自動的にアーカイブを開始するには、次のように入力します。



```
ARCHIVE LOG START
```

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

## 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。RECOVER コマンドの詳細は、[第 13 章「SQL\\*Plus コマンド・リファレンス」](#)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。



---

## SQL\*Plus グローバリゼーション・サポート

この章では、SQL\*Plus におけるグローバリゼーション・サポート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。

- [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](#)
- [iSQL\\*Plus でのグローバリゼーション・サポートの構成](#)

## 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

SQL\*Plus に別の言語を表示するには、SQL\*Plus を起動する前に、Oracle9i データベースおよびクライアント環境を構成する必要があります。クライアント環境の構成では、環境変数 NLS\_LANG を設定します。SQL\*Plus の起動時に、NLS\_LANG が読み込まれます。

### SQL\*Plus

UTF-8 文字コードを使用して、日本語キャラクタ・セットでデータを表示および入力する手順は、次のとおりです。

UNIX の場合は、次の手順に従います。

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
```

Windows の場合は、次の手順に従います。

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. 「コントロールパネル」から「システム」を開きます。
3. 新規の環境変数 NLS\_LANG を作成し、値は Japanese\_Japan.UTF8 を設定します。
4. Windows を再起動します。

NLS\_LANG パラメータには 3 つのコンポーネントがあり、各コンポーネントは、グローバリゼーション機能のサブセットを制御します。選択したキャラクタ・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 およびサポート・ソフトウェアの詳細は、『Oracle9i Database グローバリゼーション・サポート・ガイド』を参照してください。

### Oracle9i データベース

UTF-8 文字コードを使用して、日本語キャラクタ・セットでデータを格納する手順は、次のとおりです。

Oracle9i データベースが、UTF8 キャラクタ・セットで作成されていることを確認します。US7ASCII 以外のキャラクタ・セットでデータベースを作成する方法については、Oracle9i のインストレーション・ガイドを参照してください。

## iSQL\*Plus でのグローバリゼーション・サポートの構成

iSQL\*Plus に別の言語を表示するには、3 つの層すべてを構成する必要があります。次に、Microsoft Windows の Unicode (UTF-8) キャラクタ・セットを使用した日本語の例を示します。

日本語表示のために UTF-8 文字コードを使用する手順は、次のとおりです。

### Web ブラウザ

1. Unicode キャラクタ・セットをサポートするフォントが使用可能であることを確認します。

Windows オペレーティング・システムにグローバリゼーション・サポートを提供する Language Pack とその更新版は、Microsoft 社の Web サイトで入手できます。

2. Web ブラウザの文字コードを変更します。

Microsoft Internet Explorer 5.0 の場合

- a. 「表示」メニューから「エンコード」を選択します。
- b. 「Unicode」(UTF-8) を選択します。

Netscape Navigator 4.7 の場合

- a. 「表示」メニューから「文字コードセット」を選択します。
- b. 「Unicode」(UTF-8) をクリックします。

3. UTF-8 文字コードに対応するフォントを設定します。

Microsoft Internet Explorer 5.0 の場合

- a. 「ツール」メニューから「インターネットオプション...」を選択します。
- b. 「インターネットオプション」ダイアログ・ボックスの「全般」タブを選択し、「フォント」をクリックします。
- c. 言語スクリプト、その Web ページの (プロポーショナル) フォントおよび使用するプレーン・テキストの (固定幅) フォントを選択します。
- d. 「ツール」メニューの「Windows Update」を選択して、追加言語のフォントおよびサポートをインストールできます。

Netscape Navigator 4.7 の場合

- a. 「編集」メニューから「設定」を選択します。
- b. 「表示」の下にある「フォント」を選択します。
- c. 「文字コードセット」ドロップダウン・リストから「Unicode」を選択します。
- d. 「プロポーショナルフォント」および「固定ピッチフォント」ドロップダウン・リストから、Unicode キャラクタ・セットをサポートするプロポーショナル・フォントと固定ピッチ・フォントを選択します。

## Oracle HTTP Server

Oracle HTTP Server オペレーティング・システムで使用するグローバリゼーション・サポート変数を設定します。すべての iSQL\*Plus セッションは、この環境を継承します。

Windows の場合は、次の手順に従います。

1. Oracle HTTP Server を停止します。
2. 「コントロールパネル」から「システム」を開きます。
3. 新規の環境変数 NLS\_LANG を作成し、値は Japanese\_Japan.UTF8 を設定します。
4. Oracle HTTP Server を再起動します。

UNIX の場合は、次の手順に従います。

1. Oracle HTTP Server を停止します。
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
```

3. Oracle HTTP Server を再起動します。

Oracle HTTP Server は、NLS\_LANG 変数の設定に使用した端末から起動する必要があります。これは、変数が起動時に読み込まれることを確認するためです。または、ログイン初期化ファイルまたはコマンド・シェル初期化ファイルで NLS\_LANG を設定することもできます。

## Oracle9i

UTF8 キャラクタ・セットを使用して Oracle9i データベースを作成します。

# 第III部

---

## SQL\*Plus リファレンス

ここでは、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。

この部の内容は、次のとおりです。

- [SQL\\*Plus コマンド・リファレンス](#)
- [SQL\\*Plus エラー・メッセージ](#)



---

## SQL\*Plus コマンド・リファレンス

この章では、コマンドラインおよび iSQL\*Plus インタフェースで利用できる 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。

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。コマンドの表記法については、xix ページの「コード例中の表記規則」を参照してください。
項	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。
使用方法	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。

SQL\*Plus の各コマンドを説明する前に、すべてのコマンドをリストに示し、それぞれを簡単に紹介する一覧表を記載しています。

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。ハイフンの前に空白文字を入力してもかまいません。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。

## SQL\*Plus コマンド一覧

コマンド	ページ	説明
@	13-5	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。
@@	13-7	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を検索する追加機能があるため、ネストしたスクリプトの実行に便利です。
/（スラッシュ）	13-9	SQL コマンドまたは PL/SQL ブロックを実行します。
ACCEPT	13-10	1 行の入力を読み込み、所定のユーザー変数に格納します。
APPEND	13-12	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	13-10	オンライン REDO ログ・ファイルの自動アーカイブの開始または終了、指定した REDO ログ・ファイルの手動による（明示的な）アーカイブ、あるいは REDO ログ・ファイルの情報の表示を行います。
ATTRIBUTE	13-16	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	13-18	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	13-23	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	13-24	バッファ内のカレント行のテキストを変更します。
CLEAR	13-27	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	13-29	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	13-39	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	13-44	指定したユーザー名で Oracle に接続します。
COPY	13-47	問合せから、ローカル・データベースまたはリモート・データベース内の表に結果をコピーします。
DEFINE	13-48	ユーザー変数を指定し、それに 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	13-50	バッファの 1 つ以上の行を削除します。

コマンド	ページ	説明
DESCRIBE	13-52	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	13-58	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 からログオフします。ただし、SQL*Plus は終了しません。
EDIT	13-59	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、ホスト・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	13-61	1 つの PL/SQL 文を実行します。
EXIT	13-62	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	13-64	ホスト・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	13-66	SQL*Plus ヘルプ・システムにアクセスします。
HOST	13-67	SQL*Plus を終了せずに、ホスト・オペレーティング・システム・コマンドを実行します。
INPUT	13-69	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	13-71	バッファ内の 1 つ以上の行を表示します。
PASSWORD	13-73	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	13-74	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	13-75	バインド変数の現行の値を表示します。
PROMPT	13-76	指定したメッセージをユーザーの画面に送信します。
QUIT	13-62	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	13-77	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。
REMARK	13-83	スクリプト内でコメントを開始します。
REPFOOTER	13-84	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	13-86	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。

コマンド	ページ	説明
RUN	13-89	現在、バッファに格納されている SQL コマンドまたは PL/SQL ブロックを表示し、実行します。
SAVE	13-90	バッファの内容を、ホスト・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	13-92	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	13-120	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	13-125	現在実行中の Oracle インスタンスを停止します。
SPOOL	13-127	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	13-129	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、iSQL*Plus の url からのみコールできます。SQL*Plus コマンドラインでは、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。
STARTUP	13-131	Oracle 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	13-135	現行の SQL*Plus 環境の属性を、ホスト・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。
TIMING	13-136	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。
TTITLE	13-138	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。
UNDEFINE	13-141	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 コマンドでの引数によって）定義した 1 つ以上のユーザー変数を削除します。
VARIABLE	13-142	PL/SQL で参照できるバインド変数を宣言します。
WHENEVER OSERROR	13-148	オペレーティング・システム・コマンドでエラーが発生した場合に、SQL*Plus を終了したり、指定された操作を実行します。
WHENEVER SQLERROR	13-150	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、SQL*Plus を終了したり、指定された操作を実行します。

## @ (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。iSQL\*Plus では、url 形式のみがサポートされています。

### 項

項または句の詳細は、次の説明を参照してください。

*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。

*file\_name[.ext]*

実行するスクリプトを指定します。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の変更については、この章の「SET」コマンドの SUFFIX 変数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・システム環境固有の情報については、ご使用のオペレーティング・システム固有の Oracle インストール・ガイドおよび管理者リファレンス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。

パラメータの使用方法については、6-23 ページの「対話型コマンドの作成」の「iSQL\*Plus の「Substitution Variables」画面」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。

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。

@ コマンドは、START と同様の働きをします。

START コマンドが使用禁止にされる（10-4 ページの「SQL\*Plus、SQL および PL/SQL コマンドの使用禁止」を参照）と、同時に @ コマンドも使用禁止になります。START コマンドの詳細は、13-129 ページの「START」コマンド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、この章の「SET」コマンドの SQLTERMINATOR 変数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。



```
@WKRPRT.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
```

## @@ (二重アットマーク)

### 構文

```
@@url|file_name[.ext]
```

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。このコマンドがコールされたスクリプトと同じパス内または同じ *url* で、指定されたスクリプトを検索する追加機能があるため、ネストしたスクリプトを実行する場合に便利です。iSQL\*Plus では、*url* 形式のみがサポートされています。

### 項

項または句の詳細は、次の説明を参照してください。

*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。

*file\_name[.ext]*

実行するネストしたスクリプトを指定します。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の変更については、この章の「SET」コマンドの SUFFIX 変数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・システム固有の Oracle インストール・ガイドおよび管理者リファレンス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。

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。

**@@** コマンドは、**START** と同様の働きをします。

**START** コマンドが使用禁止にされる（10-4 ページの「[SQL\\*Plus、SQL および PL/SQL コマンドの使用禁止](#)」を参照）と、同時に **@@** コマンドも使用禁止になります。詳細は、13-127 ページの「[SPOOL](#)」コマンド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、13-117 ページの「[SET](#)」コマンドの **SQLTERMINATOR** 変数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** という名前の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。**@** コマンドの部分では、現行のローカルの作業ディレクトリ内で **EMPRPT** という名前の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。

## / (スラッシュ)

### 構文

/ (スラッシュ)

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・プロンプトに入力することも、複数行コマンドの行番号プロンプトに入力することもできます。

スラッシュ・コマンドは、RUN と同様の働きをしますが、バッファ内のコマンドを画面に表示しません。

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、コマンド・プロンプトにスラッシュ (/) を入力します。



/



CITY	COUNTRY_NAME
-----	-----
Seattle	United States of America
Oxford	United Kingdom
Seattle	United States of America

## ACCEPT

iSQL\*Plus では使用できません。

### 構文

```
ACC[EPT] variable [NUM[BER] | CHAR | DATE] [FOR [MAT] format]  
[DEF[AULT] default] [PROMPT text | NOPR[OMPT]] [HIDE]
```

1 行の入力を読み込み、所定のユーザー変数に格納します。

### 項

項または句の詳細は、次の説明を参照してください。

*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* のデータ型が 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* のデータ型が CHAR データ型に設定されます。CHAR の最大長制限は 240 バイトです。マルチバイト・キャラクタ・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。

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、再度応答を求めます。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については、この章の「[CHANGE](#)」コマンド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 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 Oracle NLS\_DATE\_FORMAT になります。Oracle 日付書式の詳細は、『Oracle9i データベース管理者ガイド』および『Oracle9i SQL リファレンス』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 *text* 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、この章の「[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 01/01/2001 である hired という名前の DATE 変数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 '01/01/2001'-
PROMPT 'Enter date hired: '
```

## APPEND

*iSQL\*Plus* では使用できません。

### 構文

```
A [PPEND] text
```

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。

### 項

項または句の詳細は、次の説明を参照してください。

*text*

追加するテキストを指定します。*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 *text* 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。

## ARCHIVE LOG

### 構文

```
ARCHIVE LOG {LIST|STOP} | {START|NEXT|ALL|integer } [TO destination]
```

オンライン REDO ログ・ファイルの自動アーカイブの開始または終了、指定した REDO ログ・ファイルの手動による（明示的な）アーカイブ、あるいは REDO ログ・ファイルの情報の表示を行います。

### 項

項または句の詳細は、次の説明を参照してください。

#### 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、「Next log sequence to archive」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。

#### STOP

自動アーカイブを無効にします。インスタンスが ARCHIVELOG モードで、すべての REDO ログ・ファイル・グループが一杯の場合、データベース処理は REDO ログ・ファイルがアーカイブされるまで（たとえば、ARCHIVE LOG NEXT または ARCHIVE LOG ALL コマンドを入力するまで）中断されます。

#### START

自動アーカイブを有効にします。バックグラウンド・プロセス ARCH を起動し、要求に応じて自動アーカイブを実行します。ARCH が起動され、ファイル名が指定されると、そのファイル名が新しいアーカイブの標準の出力先になります。初期化パラメータ LOG\_ARCHIVE\_START が TRUE に設定されている場合、インスタンス起動時に ARCH が自動的に起動します。

#### NEXT

アーカイブされていない次のオンライン REDO ログ・ファイル・グループを手動でアーカイブします。

#### ALL

アーカイブされていないすべてのオンライン REDO ログ・ファイル・グループを、手動でアーカイブします。

#### *integer*

ログ順序番号 *n* で、オンライン REDO ログ・ファイル・グループのアーカイブを指定します。オンラインの REDO ログ・ファイル・グループであれば、どれでも指定できます。ログ・ファイルがオンラインに見つからない、または順序番号が無効の場合は、エラーが発生します。このオプションは、ログ・ファイル・グループの再アーカイブに使用します。

#### *destination*

オペレーティング・システムでの出力先デバイスまたはディレクトリを指定します。アーカイブ先デバイスは、インストール時に指定します。アーカイブ先を指定する方法については、ご使用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。オペレーティング・システムの多くは、複数のログ・ファイルを同一のテープにスプールできます。

アーカイブ先がコマンドラインで指定されていない場合は、初期化パラメータ LOG\_ARCHIVE\_DEST の指定に従います。コマンド ARCHIVE LOG START の *destination* で指定されたデバイスまたはディレクトリは、今後すべての自動または手動アーカイブの新しい標準出力先になります。他のオプションで出力先が指定された場合は、一時的に、現行の（手動の）アーカイブのみで有効な出力先になります。その後の自動アーカイブでは、標準出力先を変更しません。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 マニュアルを参照してください。

## 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、この章の「[CONNECT](#)」コマンドを参照してください。

オンライン REDO ログ・ファイル・グループが一杯で、再利用可能なものがない場合は、データベース処理が中断されます。ログ・ファイル・グループをアーカイブすると、データベース処理が再開されます。

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 マニュアルを参照してください。

---



---

**注意：** このコマンドを使用できるのは、現行のインスタンスのみです。Real Application Clusters 内の異なるインスタンスまたはすべてのインスタンスに対してアーカイブを指定するには、SQL コマンドの ALTER SYSTEM を使用します。SQL コマンドの使用方法については、『Oracle9i SQL リファレンス』を参照してください。

---



---

## 例

アーカイバ・プロセスを起動し、LOG\_ARCHIVE\_DEST で指定したアーカイブ先を使用して自動アーカイブを開始するには、次のように入力します。



```
ARCHIVE LOG START
```

自動アーカイブを停止するには、次のように入力します。



```
ARCHIVE LOG STOP
```

順序番号 1001 を使用して、指定した出力先へアーカイブ・ログ・ファイル・グループをアーカイブするには、次のように入力します。



```
ARCHIVE LOG 1001 '/vobs/oracle/dbs/arch'
```

arch は、出力先デバイス上のファイル名に接頭辞を指定します。残りのファイル名は、初期化パラメータ LOG\_ARCHIVE\_FORMAT によって、アーカイブ REDO ログ・ファイルのファイル名の書式を指定します。

# ATTRIBUTE

## 構文

```
ATTRIBUTE [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。

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。

## 項

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。

項または句の詳細は、次の説明を参照してください。

*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 コマンドをいく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
```

# BREAK

## 構文

```
BRE[AK] [ON report_element [action [action]]] ...
```

各項目の意味は、次のとおりです。

`report_eleme` 構文は次のとおりです。

```
nt
      {column|expr|ROW|REPORT}
```

`action` 構文は次のとおりです。

```
[SKI[P] n|[SKI[P]]
PAGE] [NODUP[LICATES]|DUP[LICATES]]
```

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**」コマンドも参照）

現行の **BREAK** 定義を表示するには、句を指定しないで **BREAK** のみを入力します。

## 項

項または句の詳細は、次の説明を参照してください。

```
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 column** 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発生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* を使用するときは、必ず SQL SELECT コマンドの中で ORDER BY 句も使用します。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。

**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 が実行するレポート内の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 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、COLUMN 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。

前述の ON *column* で指定した情報は、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 ブレークは、BREAK コマンド内のどこに指定してあっても、最も内側のブレークになります。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 ブレークは、BREAK コマンド内のどこに指定してあっても、最も外側のブレークになります。

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。

項または句の詳細は、次の説明を参照してください。

**SKI [P] *n***

ブレー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 に SET ON が指定されている場合を除き、SET MARKUP HTML ON モードおよび iSQL\*Plus では動作しません。

**[SKI [P]] PAGE**

ブレー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、NEWPAGE 0 も同時に指定しない場合、SKIP PAGE によって必ずしも物理的なページ・ブレークが行われません。

**NODUP [L ICATES]**

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 [L ICATES]**

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

職種の重複値を出力し、JOB\_ID の値が変更されたときに SALARY の平均値を出力して空白行を 1 行挿入し、さらに、DEPARTMENT\_ID の値が変更されたときに SALARY の合計値を出力して、再度空白行を挿入するレポートを生成するには、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 SKIP 1 ON JOB_ID SKIP 1 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
```

## BREAK

---

```
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.
```

## BTITLE

### 構文

```
BTI[TLE] [printspec [text|variable] ...] [ON|OFF]
```

*printspec* には、*text* 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                CE[ENTER]
S[KIP] [n]          R[IGHT]
TAB n                BOLD
LE[FT]                FORMAT text
```

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。

旧形式の BTITLE の詳細は、[付録 C 「廃止された SQL\\*Plus コマンド」](#) を参照してください。

### 項

BTITLE コマンド構文の中の項と句の詳細は、この章の「TTITLE」コマンドを参照してください。

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。

### 使用方法

*text* の前に、*printspec* 句を入力しない場合は左揃えになります。コマンド名の直後に有効な *printspec* 句 (LEFT、SKIP、COL など) が続いている場合、SQL\*Plus は新形式で BTITLE を解釈します。

### 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'
```

## CHANGE

iSQL\*Plus では使用できません。

### 構文

```
C[HANGE] 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

項または句の詳細は、次の説明を参照してください。

*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。たとえば、次のように入力したとします。

```
CHANGE /aq/aw
```

この場合、「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。途中で「...」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 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 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3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。たとえば、次のように入力したとします。



```
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/
```

# 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

項または句の詳細は、次の説明を参照してください。

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消去されます。複数のバッファを使用している場合以外、CLEAR BUFFER の働きは CLEAR SQL と同じです (付録 C 「[廃止された SQL\\*Plus コマンド](#)」の「[SET BUFFER](#)」コマンドを参照)。

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]

画面が消去されます。

iSQL\*Plus では使用できません。

SQL

SQL バッファからテキストが消去されます。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 (付録 C 「廃止された SQL\*Plus コマンド」の「SET BUFFER」コマンドを参照。)

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

ブレイクを消去するには、次のように入力します。



```
CLEAR BREAKS
```

列定義を消去するには、次のように入力します。



```
CLEAR COLUMNS
```

## 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] | C[ENTRE] | 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_WAPPED] | 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

さらに、1 つまたはすべての列の現行の表示属性を表示します。

## 項

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。

項または句の詳細は、次の説明を参照してください。

{*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、それぞれエンティティ「&lt;」、「&gt;」、「&quot;」および「&amp;」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、4-4 ページの「[MARKUP オプション](#)」およびこの章の後半の「[SET](#)」コマンドを参照してください。

#### FOLD\_A [FFTER]

列ヘッダーの後および列内の各行の後に、改行が挿入されます。SQL\*Plus では、SELECT 構文のリスト内の最後の列の後は余分な改行は挿入されません。

#### FOLD\_B [EFORE]

列ヘッダーの前および列内のそれぞれの行の前に、改行が挿入されます。SQL\*Plus は、SELECT リスト内の最初の列の前には余分な改行を挿入しません。

#### FOR [MAT] *format*

列の表示書式を指定します。指定する書式は、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。

**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 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。

データ型の幅を、FORMAT *An* (A は英数字) を使用して *n* に変更します。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。

**DATE 列** SQL\*Plus 内の書式未設定の DATE 列のデフォルトの幅と書式は、有効な NLS パラメータから導出されます。その他の場合、デフォルトの幅は A9 です。Oracle9i では、NLS パラメータは、データベース・パラメータ・ファイル内で設定されることもあり、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あることもあります。また、NLS パラメータは、ALTER SESSION コマンドによって各セッションごとに指定することもできます (NLS パラメータの詳細は、Oracle9i のマニュアル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 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。

DATE 列の幅を  $n$  に変更するには、COLUMN コマンドで FORMAT  $An$  を指定します。列ヘッダーより短い幅を指定した場合、ヘッダーは切り捨てられます。

**NUMBER 列** NUMBER 列の幅を変更するには、FORMAT の後に、表 13-1 に示す要素の 1 つを指定します。

**表 13-1 数値書式**

要素	例	説明
9	9999	9 の数は、戻される有効桁数を指定します。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。
0	0999 9990	先行 0 (ゼロ) またはこの桁の 0 (ゼロ) の値は、0 と表示されません。
\$	\$9999	値の前にドル記号が付けられます。
B	B9999	書式モデル内の「0」に関係なく、0 (ゼロ) の値が空白で表示されます。
MI	9999MI	負の値の後に「-」が表示されます。正の値の場合は、後続空白が表示されます。
S	S9999	正の値の場合は「+」、負の値の場合は、「-」が、この桁に戻されます。
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。
D	99D99	この桁には、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。
G	9G999	この桁には、桁グループ・セパレータが表示されます。
C	C999	この桁には、ISO 通貨記号が表示されます。
L	L999	この桁には、各国通貨記号が表示されます。
, (カンマ)	9,999	この桁には、カンマが表示されます。
. (ピリオド)	99.99	この桁には、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。
V	999V99	値に $10^n$ を掛けます。 $n$ は、「V」の後の「9」の数です。
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)。
RN または rn	RN	大文字または小文字のローマ数字が表示されます。値は 1 ~ 3999 の整数です。
DATE	DATE	値が MM/DD/YY 書式の日付として表示されます。これは、ユリウス日付を表す NUMBER 列を書式設定するために使用されます。

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、桁グループ・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 FORMAT を使用しない場合、列の幅は、必ず 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合わせて四捨五入する場合があります。

値が列幅に収まらない場合、SQL\*Plus は、その幅に相当する各桁にシャープ記号 (#) を表示することによって、オーバーフローを示します。

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。

たとえば、次のように入力したとします。

```
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 文字の変更の詳細は、この章の「SET」コマンドの HEADSEP 変数を参照してください。

**JUS**[TIFY] {L[EFT] | C[ENTER] | C[ENTRE] | 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。

**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 / デイテール・レポートに使用すると便利です。マスター / デイテール・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。

下部タイトルに列値を表示する方法の詳細は、[COLUMN OLD\\_VALUE](#) 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、この章の終わりにある「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の詳細は、「FORMAT」コマンドの COLUMN 句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、列の印刷が行われます。

**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 / デティール・レポートに使用すると便利です。マスター / デティール・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。

上部タイトルに列値を表示する方法の詳細は、[COLUMN NEW\\_VALUE](#) 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、この章の終わりにある「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ずに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。

## 使用方法

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。したがって、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。

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 s
                    hipped by air freight
                    t 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。この場合、これまで使用していた 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、7-29 ページ「ページおよびレポートのタイトルとサイズの定義」の「タイトル内に現在の日付を表示する方法」を参照してください)。



```
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;
```

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。



```

                                     Job Report                                     04/19/01

Job:      MK_MAN

LAST
NAME          MANAGER_ID HIRE_DATE          SALARY DEPARTMENT_ID
-----
Hartstein          100 17-FEB-96          $13,000.00          20
                                     -----
                                     $13,000.00
```

```

                                     Job Report                                     04/19/01

Job:      SA_MAN

LAST
NAME          MANAGER_ID HIRE_DATE          SALARY DEPARTMENT_ID
-----
Errazuriz          100 10-MAR-97          $12,000.00          80
Zlotkey            100 29-JAN-00          $10,500.00          80
Cambrault          100 15-OCT-99          $11,000.00          80
Russell            100 01-OCT-96          $14,000.00          80
Partners           100 05-JAN-97          $13,500.00          80
```

-----  
Job Report

04/19/01

Job: SA\_MAN

```

LAST
NAME                MANAGER_ID HIRE_DATE                SALARY DEPARTMENT_ID
-----
                                $12,200.00
    
```

6 rows selected.

DATE 列のデフォルトの書式を 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 Report

04/19/01

Job: SA\_MAN

```

HIRE_DATE
-----
1994-06-07
    
```

ALTER SESSION コマンドの詳細は、『Oracle9i SQL リファレンス』を参照してください。

SELECT 文では、一部の SQL 計算または関数 (TO\_CHAR など) の結果として、列幅が非常に広くなることがあります。このような場合は、FORMAT オプションを使用して列幅を変更します。

# COMPUTE

## 構文

```
COMP[UTE] [function [LAB[EL] text] ...
          OF {expr|column|alias} ...
          ON {expr|column|alias|REPORT|ROW} ...]
```

選択した行のサブセットについて、様々な標準計算方法を使用してサマリー行を計算し出力します。また、すべての **COMPUTE** 定義を表示します。サマリーの作成方法の詳細は、7-11 ページの「[間隔の設定およびサマリー行によるレポートの明確化](#)」を参照してください。

## 項

項または句の詳細は、次の説明を参照してください。

*function* ...

表 13-2 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 13-2 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
COU[NT]	NULL 以外の値の数	すべての型
MIN[IMUM]	最小値	NUMBER、CHAR、NCHAR、 VARCHAR2 (VARCHAR)、 NVARCHAR2 (NCHAR VARYING)
MAX[IMUM]	最大値	NUMBER、CHAR、NCHAR、 VARCHAR2 (VARCHAR)、 NVARCHAR2 (NCHAR VARYING)
NUM[BER]	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
VAR[IANCE]	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わ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 は警告を発行し、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} ...

SQL\*Plus がブレイクとして使用するイベントを指定します。(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。) このイベントが発生すると (式の値が変化するか、新しい ROW がフェッチされるか、レポートの終わりに達すると)、COMPUTE は計算済の値を出力し、計算を再開します。

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
```

## COMPUTE

---

```
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.

# CONNECT

## 構文

```
CONN[ECT] { logon / / } [AS {SYSOPER|SYSDBA}]
```

*logon* は、次の構文に従ってください。

```
username/password[@connect_identifier]
```

指定したユーザー名で Oracle に接続します。iSQL\*Plus では、パスワードの入力を求めるプロンプトは表示されないため、CONNECT コマンドにユーザー名とパスワードを含める必要があります。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とは表示されません。

## 項

項または句の詳細は、次の説明を参照してください。

*username/password*

Oracle 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。*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に対してスラッシュ (/) を入力するか、そのまま [Return] を押した場合、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（後述の「/ (スラッシュ)」を参照）。

*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。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。パスワードを変更する方法の詳細は、SQL\*Plus の場合は [PASSWORD](#) コマンドを、iSQL\*Plus の場合は 4-18 ページの「[パスワードの変更](#)」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 インストールで使用する Oracle Net 通信プロトコルによって異なります。詳細は、ご使用のプロトコルに該当する Oracle Net のマニュアルを参照するか、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。

/ (スラッシュ)

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` を指定できません。デフォルト・ログオンでは、SQL\*Plus はユーザー名 `OP$S$name` を使用してユーザーをログインさせようとしています。`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、『Oracle9i データベース管理者ガイド』を参照してください。

AS {SYSOPER|SYSDBA}

AS 句は、SYSOPER または SYSDBA システム権限を付与されているユーザーに対して、特権を付けて接続を許可します。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。システム権限の詳細は、『Oracle9i データベース管理者ガイド』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 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せん。

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、『Oracle9i SQL 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 およびパスワード 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/your_password@FLEETDB
```

ユーザー名 HR を使用して接続するときに、SQL\*Plus でパスワードの入力を求めるプロンプトを表示させるには、次のように入力します。



```
CONNECT HR
```

パスワード・ファイルの設定の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

パスワード・ファイルを使用して、現行ノードのインスタンスに、ユーザー名 **HR** およびパスワード **HR** の権限で接続するには、次のように入力します。



```
CONNECT HR/your_password 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
```

最後の2つの例では、デフォルト・スキーマは **SYS** になります。

## COPY

COPY コマンドでは、Oracle8 以上に導入されたデータ型または機能を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、将来のリリースでは廃止される予定です。COPY コマンドの詳細および構文については、[付録 B 「SQL\\*Plus の COPY コマンド」](#)を参照してください。

## DEFINE

### 構文

```
DEF[INE] [variable] |[variable = text]
```

ユーザー変数を指定し、それに 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

項または句の詳細は、次の説明を参照してください。

*variable*

どのユーザー変数の値を割り当てるか（または表示するか）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*

ユーザー変数を定義（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。すべてのユーザー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、次のいずれかのイベントが発生する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、後続の SQL SELECT コマンドでその列が参照される
- SQL\*Plus の終了

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、ユーザーが変数を UNDEFINE するまで、そのセッションでの変数値の入力を求めるプロンプトを表示しません。

DEFINE を使用して変数 \_EDITOR を定義できます。この変数は、SQL\*Plus の EDIT コマンドによって起動されるホスト・システム・エディタを確立します。

定義した変数の値を複数の行に継続させた場合（SQL\*Plus のコマンド継続文字を使用）、SQL\*Plus は、結果の変数の中で、入力された各継続文字およびキャリッジ・リターン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 に対する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。

## DEL

iSQL\*Plus では使用できません。

## 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

項または句の詳細は、次の説明を参照してください。

<code>n</code>	行 <code>n</code> が削除されます。
<code>n m</code>	行 <code>n</code> ~ <code>m</code> が削除されます。
<code>n *</code>	行 <code>n</code> からカレント行までが削除されます。
<code>n LAST</code>	行 <code>n</code> から最終行までが削除されます。
<code>*</code>	カレント行が削除されます。
<code>* n</code>	カレント行から行 <code>n</code> までが削除されます。
<code>* LAST</code>	カレント行から最終行までが削除されます。
<code>LAST</code>	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。

---

---

**注意：** DEL は SQL\*Plus コマンドで、DELETE は SQL 文です。SQL の DELETE 文の詳細は、『Oracle9i SQL リファレンス』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
```

## DESCRIBE

### 構文

```
DESC [RIBE] { [schema.] object [@connect_Identifier] }
```

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

項または句の詳細は、次の説明を参照してください。

*schema*

*object* が設定されているスキーマを指定します。 *schema* を省略した場合、SQL\*Plus は、コマンド実行者自身の *object* が指定されたとみなします。

*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。

*@connect\_Identifier*

*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、『Oracle9i SQL 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) または XMLType)
- 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、この章の「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、この章の「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）

## 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
```



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
```

属性 `STREET` および `CITY` 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		
-----		
MAP MEMBER FUNCTION RATIONAL_ORDER RETURNS NUMBER		

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)
```

CREATE TYPE コマンドの詳細は、『Oracle9i SQL 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については、この章の「SET」コマンドおよび「SHOW」コマンドを参照してください。

## 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、現行のユーザーを Oracle 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 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。Oracle からログアウトしてホスト・コンピュータのオペレーティング・システムに制御を戻すには、iSQL\*Plus の場合は Logout を使用し、SQL\*Plus コマンドラインの場合は、EXIT または QUIT を使用します。

### 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/your_password
SELECT LAST_NAME, DEPARTMENT_NAME FROM EMP_DETAILS_VIEW;
DISCONNECT
SET INSTANCE FIN2
CONNECT HR2/your_password
```

## EDIT

iSQL\*Plus では使用できません。

## 構文

```
ED[IT] [file_name[.ext]]
```

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、ホスト・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。

## 項

項または句の詳細は、次の説明を参照してください。

```
file_name[.ext]
```

編集するファイル（通常はスクリプト）を指定します。

SQL バッファの内容をホスト・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、ファイル名を指定せずに 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、この章の「SET」コマンドの SUFFIX 変数を参照してください。

ファイル名を指定すると、SQL\*Plus は、そのディレクトリを見つけるために現在の作業ディレクトリを検索します。現行の作業ディレクトリの中でファイルが見つからないと、SQL\*Plus は、指定した名前のファイルを作成します。

ユーザー変数 `_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ています。`_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。ユーザー変数の値の変更については、この章の「COPY」コマンドを参照してください。`_EDITOR` が未定義であれば、EDIT は、デフォルトのホスト・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 を単独で指定すると、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、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、この章の「SET」コマンドの EDITFILE 変数を参照してください。

---

---

**注意：** 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。

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。

## 例

ホスト・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
```

# EXECUTE

## 構文

```
EXEC[UTE] statement
```

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、『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。

## 項

項または句の詳細は、次の説明を参照してください。

```
statement
```

PL/SQL 文を指定します。

## 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。

## 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については、この章の「[VARIABLE](#)」コマンドを参照してください。

# 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 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。

iSQL\*Plus の場合、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、現在の iSQL\*Plus スクリプトの処理を停止して、入力領域にフォーカスを戻します。iSQL\*Plus では、リターン・コードにアクセスする方法はありません。iSQL\*Plus では、「Logout」ボタンをクリックして Oracle をログアウトします。

## 項

項または句の詳細は、次の説明を参照してください。

{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

警告を示すリターン・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 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。詳細は、ご使用のオペレーティング・システム固有の Oracle インストール・ガイドおよび管理者リファレンスを参照してください。

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 ~ 255 に制限されません。

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、この章の「[WHENEVER SQLERROR](#)」コマンドおよび「[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
```

## GET

iSQL\*Plus では使用できません。iSQL\*Plus の場合は、Load Script を使用します。

### 構文

```
GET [FILE] file_name[.ext] [LIS[T] | NOL[IST]]
```

ホスト・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

項または句の詳細は、次の説明を参照してください。

#### 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、この名前の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。

#### LIS[T]

ロード後、ファイルの内容が表示されます。これがデフォルトです。

#### NOL[IST]

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の変更の詳細は、この章の「SET」コマンドの SUFFIX 変数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、この章の「[SAVE](#)」コマンド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
```

## HELP

### 構文

HELP [*topic*]

SQL\*Plus コマンドラインのヘルプにアクセスします。項目を表示する場合は、HELP INDEX を入力します。

### 項

項または句の詳細は、次の説明を参照してください。

*topic*

SQL\*Plus ヘルプ・トピックを指定します。たとえば、COLUMN を指定できます。

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 のみを入力します。

### 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 コマンドの構文も表示します。

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、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
```

---

# HOST

iSQL\*Plus では使用できません。

## 構文

HO[ST] [*command*]

SQL\*Plus を終了せずに、ホスト・オペレーティング・システム・コマンドを実行します。

## 項

項または句の詳細は、次の説明を参照してください。

*command*

ホスト・オペレーティング・システムのコマンドを指定します。

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。複数のオペレーティング・システムの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・システム固有の Oracle 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。

---

**注意：** 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、引き続き起動する SQL\*Plus セッションのみに影響を与えます。

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は、第 10 章「SQL\*Plus 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 \$ (VMS)、! (UNIX) またはその他の文字を使用できます。詳細は、ご使用のオペレーティング・システム固有の Oracle 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。

ご使用のオペレーティング・システムによっては、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・システム固有の Oracle 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、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。SQLTERMINATOR の詳細は、この章の「SET」コマンドの [SQLTERMINATOR](#) 変数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
```

# INPUT

iSQL\*Plus では使用できません。

## 構文

```
I [INPUT] [text]
```

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。

## 項

項または句の詳細は、次の説明を参照してください。

*text*

追加するテキストを指定します。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 *text* 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、*text* 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
```

# 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。

## 項

項または句の詳細は、次の説明を参照してください。

<code>n</code>	行 <code>n</code> が表示されます。
<code>n m</code>	行 <code>n</code> から行 <code>m</code> が表示されます。
<code>n *</code>	行 <code>n</code> からカレント行までが表示されます。
<code>n LAST</code>	行 <code>n</code> から最終行までが表示されます。
<code>*</code>	カレント行が表示されます。
<code>* n</code>	カレント行から行 <code>n</code> までが表示されます。
<code>* LAST</code>	カレント行から最終行までが表示されます。
<code>LAST</code>	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** のみを入力します。表示された最終行または表示された 1 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
```

次のような形式で、バッファ内のすべての行が表示されます。



```
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
```

## PASSWORD

iSQL\*Plus では使用できません。iSQL\*Plus の場合、「Password」画面を使用します。

### 構文

```
PASSW[ORD] [username]
```

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。

### 項

項または句の詳細は、次の説明を参照してください。

*username*

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。

### 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。パスワード変更の詳細は、この章の「[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。

## PAUSE

iSQL\*Plus では使用できません。

### 構文

```
PAU[SE] [text]
```

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。

### 項

項または句の詳細は、次の説明を参照してください。

*text*

表示するテキストを指定します。

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込みます（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、この章「**SET**」コマンドの **PAUSE** 変数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 ...
```

# PRINT

## 構文

```
PRI[NT] [variable ...]
```

バインド変数の現行の値を表示します。バインド変数の詳細は、『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。

## 項

項または句の詳細は、次の説明を参照してください。

```
variable ...
```

値を表示するバインド変数の名前を指定します。

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。詳細および例については、この章の「[VARIABLE](#)」コマンドを参照してください。

出力の間合せと同じように、PRINT 出力の書式を制御できます。詳細は、[第7章「SQL\\*Plus レポートの書式設定」](#)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、この章の「[SET](#)」コマンド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
```

# PROMPT

## 構文

```
PRO[MPT] [text]
```

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。

## 項

項または句の詳細は、次の説明を参照してください。

*text*

表示するメッセージのテキストを指定します。*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 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 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せん。



```
PROMPT
PROMPT Please enter a valid department
PROMPT For example: 10, 20, 30, 40
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。



```
@YEAREND.SQL VAL1 VAL2.SQL VAL1 VAL2x
```



```
Please enter a valid department
For example: 10, 20, 30, 40
Department ID?>
```

プロンプト Department ID?> が表示されたら、部門番号を入力できます。デフォルトでは、SQL\*Plus は置換の前後で & NEWDEPT を含む行を表示します。また、Department ID?> のプロンプトで入力された番号に対応する部門名を表示します。SET VERIFY OFF を使用して、この動作を抑止できます。

# RECOVER

## 構文

```
RECOVER {general | managed | 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 }
   [TEST | ALLOW integer CORRUPTION ]...]|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}
 [UNTIL {CANCEL | TIME date | CHANGE integer}
   | USING BACKUP CONTROLFILE]...]
```

*partial\_database\_recovery* 句の構文は次のとおりです。

```
{TABLESPACE tablespace [, tablespace]... | DATAFILE datafilename
 [, datafilename]...| STANDBY {TABLESPACE tablespace
 [, tablespace]... | DATAFILE datafilename [, datafilename]...}
 UNTIL [CONSISTENT] [WITH] CONTROLFILE }
```

*managed* 句の構文は次のとおりです。

```
MANAGED STANDBY DATABASE
 [ {NODELAY | [TIMEOUT] integer | CANCEL [IMMEDIATE] [NOWAIT]}
   | [DISCONNECT [FROM SESSION] ] [FINISH [NOWAIT] ] ]
```

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メディア・リカバリを実行します。RECOVER コマンドの詳細は、『Oracle9i データベース管理者ガイド』、『Oracle9i SQL リファレンス』の ALTER DATABASE RECOVER 文、『Oracle9i バックアップおよびリカバリ概要』および『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。

ネットワークのタイムアウトが発生する可能性があるため、RECOVER のような DBA 操作を長時間実行する場合は、SQL\*Plus コマンドラインを使用することをお勧めします。

iSQL\*Plus で RECOVER コマンドを使用するには、AUTORECOVERY ON を設定する必要があります。AUTORECOVERY OFF でデータベースをリカバリしようとする、エラーが発生します。

SP2-00872 iSQL\*Plus では、SET AUTORECOVERY を使用してください。

## 項

項および句の詳細は、次の説明を参照してください。

### 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 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 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 句を使用して時間を削減できます。

### FROM location

アーカイブ REDO ログ・ファイル・グループが読み込まれた位置を指定します。位置の値は、オペレーティング・システムの表記規則に従って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。

### 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 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。通常のリカバリ処理では、integer に 2 以上は指定できません。

### 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 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 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。

**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。

**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、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。

**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。

**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・プロセスは指定された時間、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、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。

#### CANCEL (managed 句)

管理リカバリでは、CANCEL は、現行のアーカイブ REDO ファイルを適用後に管理スタンバイのリカバリ処理を終了します。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・ファイルが読み込まれた後で、管理リカバリ処理を終了します。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 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 NOWAIT

次の REDO ログ・ファイルの後で管理リカバリ処理を終了し、すぐに制御をセッションに戻します。

#### DISCONNECT FROM SESSION

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。

#### FINISH

スタンバイ・データベースの現行のログのスタンバイ・ログ・ファイルをリカバリします。プライマリ・データベースの障害時に使用します。この句は、アーカイブ・ログに指定された遅延を無視して、Oracle ログがすぐに適用されます。

#### NOWAIT

リカバリ処理の完了後ではなく、ただちに制御を戻します。

## 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て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。**SETLOGSOURCE** 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'
```

## 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 および 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、6-9 ページの「[スクリプトへのコメントの挿入](#)」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;
```

## REPFOOTER

### 構文

```
REPFOOTER [PAGE] [printspec [text|variable] ...] | [ON|OFF]
```

*printspec* には、*text* 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。

### 項

REPFOOTER コマンド構文の中の項および句に関する追加情報については、この章の「[REPHEADER](#)」コマンドを参照してください。

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。

### 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。

---

---

**注意：** 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
```

Page: 2

END EMPLOYEE LISTING REPORT

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
```

## REPHEADER

### 構文

```
REP[EAD]ER [PAGE] [printspec [text|variable] ...] | [ON|OFF]
```

*printspec* には、*text* 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。

### 項

項または句の詳細は、次の説明を参照してください。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*

ユーザー変数、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 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 句で *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（ゼロ）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、または次の 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 *text* 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、この章の「COLUMN」コマンドの **FORMAT** 句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。

デフォルトの書式の詳細は、この章の「COLUMN」コマンドの **FORMAT** 句を参照してください。

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。

## 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
```

# 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、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.
```

## SAVE

iSQL\*Plus では使用できません。iSQL\*Plus の場合は、Save Script を使用します。

### 構文

```
SAV[E] [FILE] file_name [.ext] [CRE[ATE] | REP[LACE] | APP[END]]
```

SQL バッファの内容を、ホスト・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

項または句の詳細は、次の説明を参照してください。

#### 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、この名前で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。これが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。このデフォルト拡張子の変更については、この章の「SET」コマンドの SUFFIX 変数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
```

## 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、次のものがあります。

- データの表示幅
- HTML 書式設定のカスタマイズ
- 列ヘッダーの出力の有効または無効
- 1 ページごとの行数

システム変数の設定には、iSQL\*Plus の「System Variables」画面も使用できます。

## 構文

```
SET system_variable value
```

*system\_variable* および *value* について、次の [SET システム変数の一覧](#) に示します。

## 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。

SET ROLE および SET TRANSACTION は、SQL コマンドです (詳細は、『Oracle9i SQL リファレンス』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。

## SET システム変数の一覧

システム変数	ページ	説明
SET APPINFO {ON OFF text}	13-96	DBMS_APPLICATION_INFO パッケージを使用した、スクリプトの自動登録を設定します。
SET ARRAYSIZE {15 n}	13-97	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。
SET AUTOCOMMIT {ON OFF IMMEDIATE n}	13-98	データベースに対する保留中の変更を、Oracle がコミットする時点を制御します。
SET AUTOPRINT {ON OFF}	13-98	バインド変数の自動出力を設定します。
SET AUTORECOVERY [ON OFF]	13-98	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。
SET AUTOTRACE {ON OFF TRACE [ONLY]} [EXPLAIN] [STATISTICS]	13-99	正常に実行された SQL DML 文 (SELECT、INSERT、UPDATE または DELETE) のレポートを表示します。
SET BLOCKTERMINATOR {. c}	13-99	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。
SET CMDSEP {; c ON OFF}	13-100	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。
SET COLSEP {_ text}	13-101	SQL*Plus では、SET COLSEP によって、<PRE> タグ内部にレンダリングされる列出力と列出力の間に出力される列セパレータ文字を指定します。 選択された列の間に出力するテキストを設定します。
SET COMPATIBILITY {V7 V8 NATIVE}	13-101	現行の接続先である Oracle のバージョンを指定します。
SET CONCAT {. c ON OFF}	13-102	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。
SET COPYCOMMIT {0 n}	13-102	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。
SET COPYTYPECHECK {ON OFF}	13-102	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。
SET DEFINE {& c ON OFF}	13-102	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。
SET DESCRIBE [DEPTH {1 n ALL}] [LINENUM {ON OFF}] [INDENT {ON OFF}]	13-102	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。

システム変数	ページ	説明
SET ECHO {ON OFF}	13-104	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。
*SET EDITFILE <i>file_name</i> [.ext]	13-104	EDIT コマンド用のデフォルトのファイル名を設定します。
SET EMBEDDED {ON OFF}	13-104	ページのどこから各レポートが始まるかを制御します。
SET ESCAPE {¥ c ON OFF}	13-104	エスケープ文字として使用する文字を定義します。
SET FEEDBACK {6 n ON OFF}	13-105	問合せによって <i>n</i> 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。
SET FLAGGER {OFF ENTRY INTERMEDIATE FULL}	13-105	SQL 文が ANSI/ISO SQL92 標準に準拠しているかどうかを調べます。
*SET FLUSH {ON OFF}	13-105	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。
SET HEADING {ON OFF}	13-106	レポートへの列ヘッダーの出力を制御します。
SET HEADSEP {  c ON OFF}	13-106	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。
SET INSTANCE [ <i>instance_path</i>  LOCAL]	13-106	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 LINESIZE {80 n}	13-107	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。
SET LOBOFFSET {n 1}	13-107	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。
SET LOGSOURCE [ <i>pathname</i> ]	13-108	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。
SET LONG {80 n}	13-108	LONG、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。
SET LONGCHUNKSIZE {80 n}	13-108	SQL*Plus が LONG、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。
SET MARKUP HTML [ON OFF] [HEAD <i>text</i> ] [BODY <i>text</i> ] [TABLE <i>text</i> ] [ENTMAP {ON OFF}] [SPOOL {ON OFF}] [PREFORMAT {ON OFF}]	13-109	iSQL*Plus が使用する出力である HTML マークアップ・テキストを出力します。
*SET NEWPAGE {1 n NONE}	13-110	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。

システム変数	ページ	説明
SET NULL <i>text</i>	13-110	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。
SET NUMFORMAT <i>format</i>	13-111	数値を表示するためのデフォルトの書式を設定します。
SET NUMWIDTH {10  <i>n</i> }	13-111	数値を表示するためのデフォルトの幅を設定します。
SET PAGESIZE {24  <i>n</i> }	13-111	各ページの行数を設定します。
*SET PAUSE {ON OFF  <i>text</i> }	13-111	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。
SET RECSEP {WRAPPED EACH OFF}	13-111	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。
SET RECSEPCHAR {_  <i>c</i> }	13-111	レコード・セパレータを表示または出力します。
SET SERVEROUTPUT {ON OFF} [SIZE <i>n</i> ] [FORMAT {WRAPPED WORD_WRAPPED TRUNCATED}]	13-112	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。
*SET SHIFTOUT {VISIBLE INVISIBLE}	13-113	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 SHOWMODE {ON OFF}	13-114	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。
*SET SQLBLANKLINES {ON OFF}	13-114	SQL*Plus が SQL コマンドまたはスクリプト内の空白行を許可するかを制御します。
SET SQLCASE {MIXED LOWER UPPER}	13-115	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。
*SET SQLCONTINUE {>   <i>text</i> }	13-115	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シーケンスを設定します。
*SET SQLNUMBER {ON OFF}	13-116	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。
SET SQLPLUSCOMPATIBILITY { <i>x.y.z</i> }	13-116	VARIABLE の動作または出力書式を <i>x.y.z</i> 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。
*SET SQLPREFIX {#  <i>c</i> }	13-116	SQL*Plus の接頭文字を設定します。
*SET SQLPROMPT {SQL>  <i>text</i> }	13-116	SQL*Plus コマンド・プロンプトを設定します。
SET SQLTERMINATOR {;  <i>c</i>  ON OFF}	13-117	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、 <i>c</i> に設定します。
*SET SUFFIX {SQL  <i>text</i> }	13-117	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。

システム変数	ページ	説明
*SET TAB {ON OFF}	13-118	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。
*SET TERMOUT {ON OFF}	13-118	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。
*SET TIME {ON OFF}	13-118	現在の時刻表示を制御します。
SET TIMING {ON OFF}	13-118	タイミング統計の表示を制御します。
*SET TRIMOUT {ON OFF}	13-118	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。
*SET TRIMSPOOL {ON OFF}	13-119	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。
SET UNDERLINE {- c ON OFF}	13-119	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。
SET VERIFY {ON OFF}	13-119	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。
SET WRAP {ON OFF}	13-119	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。

\*SET コマンドは、iSQL\*Plus では使用できません。

## SET APPI[NFO]{ON|OFF|text}

DBMS\_APPLICATION\_INFO パッケージを使用した、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、登録名を読み込むこともできます。

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使用禁止になります。そのかわりに、text 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。

---

**注意：** この機能を使用するには、DBMS\_APPLICATION\_INFO パッケージへのアクセス権限が必要です。DBMS\_APPLICATION\_INFO パッケージを作成するには、DBMSUTIL.SQL（この名前はご使用のオペレーティング・システムによって異なります）を SYS で実行します。DBMSUTIL.SQL は、Oracle9i データベース・サーバー製品の一部です。

---

DBMS\_APPLICATION\_INFO パッケージの詳細は、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

## 例

APPINFO の設定（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 APPI OFF
```

## 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の数（バッチ）を設定します。有効な値は 1 ~ 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。

## SET AUTO[COMMIT]{ON|OFF|IMM[EDIATE]}*n*

データベースに対する保留中の変更を、Oracle がコミットする時点を制御します。ON を指定すると、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。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 オプションと同じ効果が得られます。*n* を指定すると、*n* 個の SQL INSERT、UPDATE、DELETE 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）より大きく、2,000,000,000 より小さい値である必要があります。文カウンタは、次の処理が正常に完了した後で、0 にリセットされます。*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、文字カウンタは 0（ゼロ）にリセットされます。

---

---

**注意：** 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せん。

---

---

## 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、この章の「[PRINT](#)」コマンドを参照してください。

## 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。AUTORECOVERY に ON を指定すると、予期できる名前でも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。AUTORECOVERY が ON の場合に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。

iSQL\*Plus で RECOVER コマンドを使用するには、AUTORECOVERY ON を設定する必要があります。AUTORECOVERY OFF でデータベースをリカバリしようとするすると、エラーが発生します。

SP2-00872 iSQL\*Plus では、SET AUTORECOVERY ON を使用してください。

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、この章の「[RECOVER](#)」コマンド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
```

## 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 または DELETE) 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。

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せんが、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。実行計画の詳細は、『Oracle9i SQL リファレンス』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。

実行計画および統計表示については、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

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。

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、9-2 ページの「文のトレース」を参照してください。

## 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。

## SET CMDS[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コロン (;) に設定されません。

### 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.

## SET COLSEP {*|text|*}

iSQL\*Plus では、SET COLSEP によって、<PRE> タグ内部にレンダリングされる列出力と列出力の間に出力される列セパレータ文字を指定します。HTML 表出力がデフォルトです。事前に書式設定された出力を生成するには、SET MARKUP HTML PREFORMAT ON コマンドを使用して PREFORMAT ON を設定する必要があります。

選択された列の間に出力するテキストを設定します。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、この章の「[SET RECSEP](#)」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 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

## SET COM[PATIBILITY]{V7|V8|NATIVE}

使用する Oracle SQL 構文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。COMPATIBILITY を、接続先のデータベースで使用する Oracle SQL 構文のバージョンにあわせて正しく設定する必要があります。この設定に誤りがあると、SQL コマンドを実行できません。

### 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、Oracle9i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。

## 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。

## SET COPYC[OMMIT] {Q|*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バッチの数を制御します。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 ~ 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のみコミットを実行します。

## SET COPYTYPECHECK {Q|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。

## 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、C-6 ページの「SET SCAN」コマンドを参照してください。

## 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 ~ 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。

オブジェクトの定義を表示する方法の詳細は、この章の「[DESCRIBE](#)」コマンドを参照してください。

## 例

深さレベル 2 までビュー EMP\_DETAILS\_VIEW の定義を表示して、行数も表示しながら出力をインデントするには、最初に次のようにビュー定義を表示します。



```
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)

出力でインデントおよび行数が表示されるように EMP\_DETAILS\_VIEW を書式設定するには、SET DESCRIBE コマンドを使用して次のように入力します。



```
SET DESCRIBE DEPTH 2 LINENUM ON INDENT ON
```

以前の設定を表示するには、次のように入力します。



```
DESCRIBE EMP_DETAILS_VIEW
```



	Name	Null?	Type
1	EMPLOYEE_ID	NOT NULL	NUMBER (6)
2	JOB_ID	NOT NULL	VARCHAR2 (10)
3	MANAGER_ID		NUMBER (6)
4	DEPARTMENT_ID		NUMBER (4)
5	LOCATION_ID		NUMBER (4)
6	COUNTRY_ID		CHAR (2)
7	FIRST_NAME		VARCHAR2 (20)
8	LAST_NAME	NOT NULL	VARCHAR2 (25)

9	SALARY		NUMBER(8,2)
10	COMMISSION_PCT		NUMBER(2,2)
11	DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
12	JOB_TITLE	NOT NULL	VARCHAR2(35)
13	CITY	NOT NULL	VARCHAR2(30)
14	STATE_PROVINCE		VARCHAR2(25)
15	COUNTRY_NAME		VARCHAR2(40)
16	REGION_NAME		VARCHAR2(25)

## SET ECHO {ON|OFF}

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。ON を指定すると、コマンドが表示されます。OFF を指定すると、表示されません。

## SET EDITF[ILE] *file\_name*.[*ext*]

SET EDITFILE は、iSQL\*Plus ではサポートされていません。

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、この章の「EDIT」コマンドを参照してください。

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。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。

## SET EMB[EMDED] {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。

## 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
```

## SET FEED[BACK] {6|n|ON|OFF}

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。

## 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り、Oracle 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警告を表示します。

## SET FLU[SH] {ON|OFF}

SET FLUSH は、iSQL\*Plus ではサポートされていません。

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、ホスト・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。

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。

## 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
```

## SET HEADS[EP] {||c|ON|OFF}

ヘッダーのセパレータ文字として使用する文字を定義します。ヘッダー区切り文字に、英数字または空白は使用できません。ヘッダー区切り文字を、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、ヘッダー区切り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。

## 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 LOCAL** を入力します。初期デフォルト・インスタンスの設定方法の詳細は、ご使用のオペレーティング・システム固有の Oracle マニュアルを参照してください。

一部のインスタンスに現在接続されていない場合は、インスタンスのみを変更できます。つまり、現行のインスタンスから切断したことを最初に確認する必要があります。インスタンスを設定または変更し、使用可能にして新しく設定するには、インスタンスを再接続します。

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 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。

## SET LIN[ESIZE] {80|n}

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。詳細は、ご使用のオペレーティング・システム固有の Oracle インストール・ガイドおよび管理者リファレンスを参照してください。

## SET LOBOF[FSET] {n|1}

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。

## 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 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。パス名を指定せずに 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/oracle90/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/oracle90/dbs/arch"  
RECOVER DATABASE
```

## SET LONG {*80|n*}

LONG、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。*n* の最大値は、2GB 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。

## SET LONGC[HUNKSIZE] {*80|n*}

SQL\*Plus が LONG、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。

## 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。

## SET MARK[UP] HTML [ON|OFF] [HEAD *text*] [BODY *text*] [TABLE *text*] [ENTMAP {ON|OFF}][SPOOL {ON|OFF}] [PRE[FORMAT] {ON|OFF}]

iSQL\*Plus が使用する出力である HTML マークアップ・テキストを出力します。iSQL\*Plus では無効な HTML 出力を生成するオプションを使用しないように注意してください。ブラウザ画面が壊れる可能性があります。HEAD オプションおよび BODY オプションは、動的レポートおよびローカル・ファイルに保存されるレポートに対して有効です。

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、4-4 ページの「[MARKUP オプション](#)」を参照してください。使用方法の例については、13-109 ページの「[SET MARKUP](#)」および 8-1 ページの「[SQL\\*Plus からの HTML 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。

---



---

**注意：** 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ます。

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.

## SET NEWP[AGE] {1|n|NONE}

**SET NEWPAGE** は、**iSQL\*Plus** ではサポートされていません。

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。

## SET NULL text

**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。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。

## SET NUMF[ORMAT] *format*

数値を表示するためのデフォルトの書式を設定します。*format* に、数値書式を入力します。数値書式の詳細は、この章の「**COLUMN**」コマンドの **FORMAT** 句を参照してください。

## 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、この章の「**COLUMN**」コマンドの **FORMAT** 句を参照してください。

## SET PAGES[IZE] {24|*n*}

各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。

## SET PAU[SE] {ON|OFF|*text*}

SET PAUSE は、iSQL\*Plus ではサポートされていません。

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。ON を指定すると、SQL\*Plus はレポートの各ページの出力開始時に一時停止します。それぞれの一時停止の後で、[Return] を押す必要があります。入力する *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。複数の語を入力する場合、*text* 全体を一重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。

## 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。

## 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。

## SET SERVEROUTPUT [PUT] {ON|OFF} [SIZE *n*] [FOR[MAT] {WRA[PPED]|WOR[D\_WRAPPED]|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。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。

SIZE には、Oracle8i または Oracle9i データベース・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。*n* のデフォルト値は 2000 です。*n* は、2000 以上 1000000 未満である必要があります。

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。

各 FORMAT について、各サーバー出力行は新しい出力行から開始されます。

DBMS\_OUTPUT.PUT\_LINE の詳細は、『Oracle9i アプリケーション開発者ガイド - 基礎編』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ジャ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 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 ('TEXT');
```



```
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
```

## SET SHIFT[INOUT] {VIS[IBLE]|INV[ISIBLE]}

SET SHIFTINOUT は、iSQL\*Plus ではサポートされていません。

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 SHIFTOINOUT 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キャラクタ・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 SHIFTTINOUT VISIBLE
SELECT LAST_NAME, JOB_ID FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000;
```



```
LAST_NAME      JOB_ID
-----
:JJOO:         :AABBCC:
:AA:abc        :DDEE:e
```

「:」は、参照できるシフト文字です。

大文字は、マルチバイト・キャラクタを示します。

小文字は、シングルバイト・キャラクタを示します。

## SET SHOW[MODE] {ON|OFF}

SET SHOWMODE は、iSQL\*Plus ではサポートされていません。

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 を指定すると、設定値が表示されます。OFF を指定すると、非表示になります。SHOWMODE ON の動作は、すでに廃止された SHOWMODE BOTH と同じです。

## SET SQLBL[ANKLINES] {ON|OFF}

SET SQLBLANKLINES は、iSQL\*Plus ではサポートされていません。

SQL\*Plus が SQL コマンドまたはスクリプト内の空白行を許可するかを制御します。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
```

## 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。

## SET SQLCO[NTINUE] {**>** | *text*}

SET SQLCONTINUE は、iSQL\*Plus ではサポートされていません。

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シーケンスを設定します。

## 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。

## SET SQLN[UMBER] {ON|OFF}

SET SQLNUMBER は、iSQL\*Plus ではサポートされていません。

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。

## SET SQLPLUSCOMPAT[IBILITY] {x.y[.z]}

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。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、8.1.7、9.0.1 または 9.2.0 です。新しいリリースでは、SQLPLUSCOMPATIBILITY は VARIABLE 以外の機能に影響を与える場合があります。

SQLPLUSCOMPATIBILITY の値を 9.0.1 より前のバージョンに設定すると、NCHAR または NVARCHAR2 データ型の VARIABLE 定義が、Oracle8i の動作で処理され、変数のサイズが選択された各国語キャラクタ・セットによってバイト単位または文字単位となります。

標準の glogin.sql ファイルには、SET SQLPLUSCOMPAT 8.1.7 が含まれます。SET SQLPLUSCOMPAT 9.2.0 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。

## SET SQLPRE[FIX] {#|c}

SET SQLPREFIX は、iSQL\*Plus ではサポートされていません。

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。

## SET SQLP[ROMPT] {SQL>|text}

SET SQLPROMPT は、iSQL\*Plus ではサポートされていません。

SQL\*Plus コマンド・プロンプトを設定します。

## 例

次の例のスクリプトを正常に実行するには、SELECT ANY TABLE 権限が必要です。

ユーザー名および SID 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 > '
```

SID 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 TERMOUT OFF
COLUMN X NEW_VALUE Y
SELECT RTRIM(INSTANCE, CHR(0)) X FROM V$THREAD;
SET SQLPROMPT '&Y SQL>'
SET TERMOUT ON
```

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 TERMOUT OFF
COLUMN D22 NEW_VALUE VAR
SELECT USERNAME D22 FROM USER_USERS;
SET SQLPROMPT '&&VAR>'
SET TERMOUT ON
```

これらの設定は動的ではありません。接続コマンドを使用して別のインスタンスにログインする場合など、インスタンスを変更する場合は、それらをリセットする必要があります。

## SET SQLT[ERMINATOR] {;|c|ON|OFF}

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。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、空白行を入力します。SQLBLANKLINES を ON に設定すると、SQL コマンドの終了には BLOCKTERMINATOR を使用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。

## SET SUFFIX {SQL|text}

SET SUFFIX は、iSQL\*Plus ではサポートされていません。

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 .UFI に設定するには、次のように入力します。



```
SET SUFFIX UFI
```

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.UFI という名前のファイルを検索します。

## SET TAB {ON|OFF}

SET TAB は、*iSQL\*Plus* ではサポートされていません。

*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。

## SET TERM[OUT] {ON|OFF}

SET TERMOUT は、*iSQL\*Plus* ではサポートされていません。

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。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。ON を指定すると、出力が表示されます。TERMOUT OFF は、対話形式で入力するコマンドからの出力には影響を与えません。

## SET TI[ME] {ON|OFF}

SET TIME は、*iSQL\*Plus* ではサポートされていません。

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。

## 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。SET TIMING ON で表示されるデータの詳細は、ご使用のオペレーティング・システム固有の Oracle 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。複数のコマンドのタイミング情報については、この章の「[TIMING](#)」コマンドを参照してください。

## SET TRIM[OUT] {ON|OFF}

SET TRIMOUT は、*iSQL\*Plus* ではサポートされていません。

*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。

## SET TRIMS[POOL] {ON|OFF}

SET TRIMSPOOL は、iSQL\*Plus ではサポートされていません。

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。

## 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、*c* に設定します。*c* 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。

## SET VER[IFY] {ON|OFF}

置換変数を値と置き換える前後で、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。ON を指定すると、テキストが表示されます。OFF を指定すると、非表示になります。

## 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。

# SHOW

## 構文

SHO[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]

ERR[ORS] [ { FUNCTION | PROCEDURE | PACKAGE | PACKAGE BODY  
                  | TRIGGER | VIEW | TYPE | TYPE BODY | DIMENSION  
                  | JAVA CLASS } [*schema.*]*name*]

LNO

PARAMETERS [*parameter\_name*]

PNO

REL[EASE]

REPF[OOTER]

REPH[EADER]

SGA

SPOOL[L]

SQLCODE

TTI [TLE]

USER

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

項または句の詳細は、次の説明を参照してください。

*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。

BTI [TLE]

現行の BTITLE 定義が表示されます。

```
ERR[ORS] [{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（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在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。これらの幅は、COLUMN コマンドを使用して変更できます。

LNO

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。

PARAMETERS [parameter\_name]

1 つ以上の初期化パラメータに対して、現在の値を表示します。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 データベース・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: 表またはビューが存在しません。

PNO

現在のページ番号が表示されます。

REL [EASE]

SQL\*Plus がアクセスしている Oracle のリリース番号が表示されます。

REPF [OOTER]

現在の REPFOOTER 定義が表示されます。

REPH [EADER]

現在の REPHEADER 定義が表示されます。

SPOO [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

現在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。SGA 句を使用するには、SELECT ON V\_\$SGA オブジェクト権限が必要です。権限がない場合は、次のメッセージが表示されます。

ORA-00942: 表またはビューが存在しません。

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。

TTI [TLE]

現在の 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"

## 例

現行の LINESIZE を表示するには、次のように入力します。



```
SHOW LINESIZE
```

現行の行サイズが 80 文字の場合、SQL\*Plus から次の応答があります。



```
LINESIZE 80
```

次の例は、ストアド・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;
/
```



警告： 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。



```
SHOW ERRORS PROCEDURE PROC1
NO ERRORS.
SHOW ERRORS PROCEDURE HR.PROC1
Errors for PROCEDURE HR.PROC1:
LINE/COL ERROR
```

```
-----
3/3      PLS-00049: バインド変数 'P1' が正しくありません。
```

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
```

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"
```

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
```

# SHUTDOWN

## 構文

```
SHUTDOWN [ABORT | IMMEDIATE | NORMAL | TRANSACTIONAL [LOCAL]]
```

現在実行中の Oracle 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ィスマウントが実行されます。

## 項

項または句の詳細は、次の説明を参照してください。

### ABORT

トランザクション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すべてのユーザーは、暗黙的に切断されます。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

現行のトランザクション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およびディスマウントされ、インスタンスが停止されます。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ちます。

新しい接続は禁止されます。データベースはクローズおよびディスマウントされ、インスタンスが停止されます。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、クライアントからの要求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。

**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** 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。マルチスレッド・サーバーを介して接続できません。データベース接続の詳細は、この章の「[CONNECT](#)」コマンドを参照してください。

## 例

標準モードでデータベースを停止するには、次のように入力します。



```
SHUTDOWN
```



```
Database closed.  
Database dismounted.  
Oracle instance shut down.
```

## SPOOL

iSQL\*Plus では使用できません。ファイルへの出力には、Preferences の設定を使用してください。

### 構文

```
SPOOL [OL] [file_name[.ext] | 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

項または句の詳細は、次の説明を参照してください。

*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。

OFF

スプールを停止します。

OUT

スプールを停止して、ファイルをホスト・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。

**例**

デフォルトのファイル拡張子を使用して、ファイル **DLARY** に出力を記録するには、次のように入力します。



```
SPOOL DIARY
```

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。



```
SPOOL OUT
```

# 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、iSQL\*Plus では Web サーバーから、SQL\*Plus コマンドラインでは、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。通常の方法でスクリプト変数に値を渡します。

## 項

項または句の詳細は、次の説明を参照してください。

*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、FTP および Gopher プロトコルをサポートします。

*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。このデフォルト拡張子の変更については、この章の「SET」コマンドの SUFFIX 変数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・システム環境固有の情報については、ご使用のオペレーティング・システム固有の Oracle インストール・ガイドおよび管理者リファレンス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については、6-23 ページの「対話型コマンドの作成」の「iSQL\*Plus の「Substitution Variables」画面」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・プロファイルの中で、START コマンドを使用禁止にする (10-4 ページの「SQL\*Plus、SQL および PL/SQL コマンドの使用禁止」を参照) 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、この章の「@ (アットマーク)」コマンドおよび「@@ (二重アットマーク)」コマンド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, SALARY
FROM EMP_DETAILS_VIEW
WHERE 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。

# STARTUP

## 構文

STARTUP *options* | *migrate\_options*

*options* の構文は、次のとおりです。

```
[FORCE] [RESTRICT] [PFILE=filename] [QUIET] [ MOUNT [dbname] |  
[ OPEN [open_options] [dbname] ] |  
NOMOUNT ]
```

*open\_options* の構文は次のとおりです。

```
READ {ONLY | WRITE [RECOVER]} | RECOVER
```

*migrate\_options* の構文は、次のとおりです。

```
[PFILE=filename] MIGRATE [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 インスタンスを起動します。

## 項

項および句の詳細は、次の説明を参照してください。

### FORCE

再起動する前に、現行の Oracle 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 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のは、指定したパラメータ・ファイルになります。PFILE の指定を省略すると、デフォルトの STARTUP パラメータ・ファイルが使用されます。

**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** と同時には指定できません。

**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。適用する必要がある、次のログ・ファイルの提案された位置および名前がプロンプトに表示されます。

**MIGRATE**

**OPEN MIGRATE** 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またはダウングレード・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。**MIGRATE** は Oracle データベース・サーバーを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。

データベース・バージョンの移行、またはデータベース・リリースのアップグレードやダウングレードの準備、テストおよび実装の詳細は、『Oracle9i データベース移行ガイド』を参照してください。

アップグレード・スクリプトまたはダウングレード・スクリプトを実行すると、インストール済の Oracle データベースを、別のバージョンに変換します。たとえば、Oracle7 データベースを Oracle9i データベースに変換します。移行が完了したら、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。

## 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。マルチスレッド・サーバーを介して接続できません。

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インスタンスを起動し、デフォルト・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
```

現在稼働中のインスタンスを強制終了し、再起動後にデータベースをマウントします。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 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
```

パラメータ・ファイル myinit.ora を使用して現在稼働中のインスタンスを強制終了し、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
```

## STORE

iSQL\*Plus では使用できません。

### 構文

```
STORE SET file_name[.ext] [CRE[ATE] | REP[LACE] | APP[END]]
```

現行の SQL\*Plus 環境の属性を、ホスト・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

項または句の詳細は、次の説明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。

STORE コマンド構文のその他の項および句に関する情報については、この章の「[SAVE](#)」コマンドを参照してください。

### 使用方法

このコマンドを実行すると、[START](#) コマンド、[@](#) コマンドまたは [@@](#) 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
```

## 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

項または句の詳細は、次の説明を参照してください。

**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。

**TIMING** で表示されるデータの詳細は、ご使用のオペレーティング・システム固有の Oracle インストラクション・ガイドおよび管理者リファレンスを参照してください。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
```

## TTITLE

### 構文

```
TTI [TLE] [printspec [text|variable] ...] [ON|OFF]
```

*printspec* には、*text* 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                CE [NTER]
S [KIP] [n]         R [IGHT]
TAB n                BOLD
LE [FT]              FORMAT text
```

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。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、C-7 ページの「[TTITLE](#)」を参照してください。

### 項

項または句の詳細は、次の説明を参照してください。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、*text* を一重引用符で囲んで入力します。

*variable*

ユーザー変数、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 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。

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* をすぎてしまっている場合は、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 *text* 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**」コマンドの **FORMAT** 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を使用して、**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。

デフォルトの書式の詳細は、この章の「**COLUMN**」コマンドの **FORMAT** 句を参照してください。

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

最初に出現した *text* 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。コマンド名の直後に有効な *printspec* 句 (LEFT、SKIP、COL など) が続いている場合、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、この章の「[COLUMN NEW\\_VALUE](#)」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、こ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 2001' -  
RIGHT 'Page:' FORMAT 999 SQL.PNO SKIP CENTER -  
'Data in Thousands'
```



```
Monthly Analysis           01 Jan 2001           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
```

# UNDEFINE

## 構文

```
UNDEF[INE] variable ...
```

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 コマンドでの引数によって) 定義した 1 つ以上のユーザー変数を削除します。

## 項

項または句の詳細は、次の説明を参照してください。

```
variable
```

削除するユーザー変数の名前を指定します。同じコマンドで、1 つ以上のユーザー変数を削除できます。

## 例

POS という名前のユーザー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 つのユーザー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
```

## VARIABLE

### 構文

```
VAR [TABLE] [variable [NUMBER|CHAR|CHAR (n [CHAR|BYTE])|NCHAR|NCHAR (n)|VARCHAR2 (n [CHAR|BYTE])|NVARCHAR2 (n)|CLOB|NCLOB|REFCURSOR] ]
```

PL/SQL で参照できるバインド変数を宣言します。バインド変数の詳細は、6-29 ページの「[バインド変数の使用方法](#)」を参照してください。PL/SQL 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。

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。

### 項

項または句の詳細は、次の説明を参照してください。

*variable*

作成するバインド変数の名前を指定します。

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、選択したキャラクタ・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 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、選択したキャラクタ・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 9.2.0 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、選択した各国語キャラクタ・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 型で長さが *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 4000 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 VARCHAR2 変数 *n* の最大値は、選択したキャラクタ・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 4000 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 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 設定より優先されます。

**NVARCHAR2** (*n*)

NVARCHAR2 型で長さが *n* 文字までの変数が作成されます。*n* の最大値は、選択したキャラクタ・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 4000 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 9.2.0 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、選択した各国語キャラクタ・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 4000 バイトです。

**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。

## 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、この章の「[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、AUTOPRINT コマンドを使用します。詳細は、この章の「[SET AUTOPRINT](#)」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 は、バインド変数用の PRINT 文の処理が完了した後、または SQL\*Plus が終了するときに、LOB ロケータをクローズします。

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 カーソル変数の詳細は、『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。

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.

この例では、変数を表示するための 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、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、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 report 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 confidential information.
```

CLOB 列の作成については、『Oracle9i SQL リファレンス』を参照してください。

## 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（ファイル書き込みエラーなど）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。

iSQL\*Plus では、オペレーティング・システム・エラーが発生した場合、指定した操作（デフォルトでは現行のスクリプトを停止）を実行し、フォーカスを入力領域に戻します。

### 項

項または句の詳細は、次の説明を参照してください。

```
[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、この章の「EXIT」コマンド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\*Plus 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。

## 例

次のスクリプト内のコマンドは、出力ファイルからの読み込み時に障害が発生した場合、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.....
```

## 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。

iSQL\*Plus では、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、指定した操作（デフォルトでは現行のスクリプトを停止）を実行し、フォーカスを入力領域に戻します。

### 項

項または句の詳細は、次の説明を参照してください。

```
[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、この章の「EXIT」コマンド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

次のスクリプト内のコマンドは、iSQL\*Plus の現行のスクリプトの処理を中止して、「Work Screen」画面の入力領域にフォーカスを戻します。

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 が終了し、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 SQL コマンド・エラーが発生した場合、iSQL\*Plus は現行のスクリプトの処理を中止し、「Work Screen」画面の入力領域にフォーカスを戻します。



```
WHENEVER SQLERROR EXIT SQL.SQLCODE
select column_does_not_exist from dual;
```



```
select column_does_not_exist from dual
*
ERROR at line 1:
ORA-00904: string: 無効な識別子です。
```

Disconnected from Oracle.....

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 "HEADIING"
```



```
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: 行 2、列 10:  
PLS-00201: 識別子 COLUMN_DOES_NOT_EXIST を宣言してください。  
ORA-06550: 行 2、列 3:  
PL/SQL: SQL Statement ignored  
  
Disconnected from Oracle.....
```

---

---

## SQL\*Plus エラー・メッセージ

ここでは、SQL\*Plus および iSQL\*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。

次のエラーメッセージが含まれます。

- SQL\*Plus のエラー・メッセージ
- iSQL\*Plus のエラー・メッセージ
- COPY コマンドのメッセージ

Oracle が表示する、接頭辞 ORA-、TNS- および PLS- のエラー・メッセージについては、『Oracle9i データベース・エラー・メッセージ』を参照してください。

## 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、再実行してください。

**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。

**SP2-00028 内部 SQL\*Plus エラー - 無効なエラー (mode\_number)**

**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、オラクル社カスタマ・サポート・センターに連絡してください。

**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:** 入力したコマンドが長すぎました。

**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。

**SP2-00051 切替え値が switch\_value のため、正しく処理できません。**

**原因:** 内部エラーが発生しました。

**処置:** メッセージおよび番号を記録して、オラクル社カスタマ・サポート・センターに連絡してください。

**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。

**SP2-00053 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。

**SP2-00054 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、COMPUTES 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 関数の最大数を超過しています。

**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。

**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。

**SP2-00108 CREATE、REPLACE、APPEND という名前および略称は使用できません。**

**原因:** ファイル名に「file」という単語が指定されました。

**処置:** 名前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のですが、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のですが、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:** 変数または記号に値を指定してください。

**SP2-00138 DEFINE *variable* は追加されませんでした (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。

**SP2-00145 udalnk が 12345.. ではありません。リンク・エラーと思われます。**

**処置:** 実行可能な SQL\*Plus が正しくリンクされていません。

**処置:** メッセージおよび番号を記録し、システム管理者に連絡して、SQL\*Plus に再リンクしてください。

**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。

**SP2-00152 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記録して、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*"**

**使用方法:** SET SHIFT[INOUT] [VIS[IBLE] | INV[ISIBLE]]

**SET NEWP[AGE] [1 | n | NONE]**

**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 のオンライン・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:** データベース管理者に連絡して、オンライン・ヘルプをインストールしてください。

**SP2-00172 HELP は使用できません。**

**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。

**SP2-00176 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 変数の割当てで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 入力が高すぎます。 *number\_of\_characters* 文字以内である必要があります。**

原因: 入力値が高すぎました。

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: *number\_of\_attempts* 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り当てできません。**

**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 1 (*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 *value* が範囲 (*lower\_value* から *upper\_value*) 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。

**SP2-00270 *afiset (number)* 内の不明な FLAG が無視されました。**

**原因:** SET コマンドで、不明なエラーが発生しました。

**処置:** メッセージを記録して、オラクル社カスタマ・サポート・センターに連絡してください。

**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。

**SP2-00281 *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]]

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。

**SP2-00306 オプションが無効です。**

**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。

**SP2-00308 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*)。**

**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空間がありません。**

**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。

**SP2-00331 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:** 操作はできません。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バッファ (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。

**SP2-00395 使用方法: SQLPLUS [[<*option*>] [<*logon*>] [<*start*>]]**

ただし <*option*> ::= -H | -V | [[-L] [-M <*o*>] [-R <*n*>] [-S]]

-H は SQL\*Plus バージョン・バナーと使用方法の構文を表示します。

-V は SQL\*Plus バージョン・バナーを表示します。

-L 一度のみログオンしようとします。

-M <*o*> HTML マークアップ・オプションを使用します。<*o*>

-R <*n*> 限定モードを使用します。<*n*>

-S サイレント・モードを使用します。

<*logon*> ::= <*username*>[/<*password*>][@<*connect\_string*>] | / | /NOLOG

<*start*> ::= @<*URL*>|<*filename*>[.<*ext*>] [<*parameter*> ...]

**原因:** SQL\*Plus コマンド・オプションが無効です。

**処置:** 正しい使用方法については、4-3 ページの「SQLPLUS コマンド」の構文を確認してください。

**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 "value" は有効な数値ではありません。**

原因: ACCEPT コマンドに入力された値が数値ではありませんでした。

処置: 有効な数値を入力してください。

**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。

**SP2-00450 使用方法: WHENEVER SQLERROR.**

{ CONTINUE [COMMIT | ROLLBACK | NONE] | EXIT [SUCCESS | FAILURE | WARNING | n | <variable> | :<bindvariable>] [COMMIT | ROLLBACK] }

原因: WHENEVER SQLERROR 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。

**SP2-00453 使用方法: WHENEVER OSERROR**

{ CONTINUE [ COMMIT | ROLLBACK | NONE ] | EXIT [ SUCCESS | FAILURE | WARNING | n | <variable> | :<bindvariable> | OSCODE ] [ COMMIT | ROLLBACK ] }

原因: WHENEVER OSERROR 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。

**SP2-00486 FROM 句または TO 句が欠落している場合は現行の SQL\*Plus 接続を使用します。**

使用方法: COPY FROM <db> TO <db> <opt> <table> [(<cols>)] USING <sel>

<db>: scott/tiger@d:chicago-mktg などのデータベース文字列

<opt>: APPEND または CREATE、INSERT、REPLACE キーワードの内の 1 つ

<table>: 接続先表名

<cols>: 接続先列別名のカンマ区切りリスト。

<sel>: 有効な SQL SELECT 文

原因: COPY コマンドの使用が誤って指定されました。

処置: 有効なオプションを指定してください。

**SP2-00495 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 かっこ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 コマンド: *command\_name* が無効です。**

**原因:** データベースのスキーマを明示的に使用禁止にするコマンドを使用しようとした。

**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。

**SP2-00548 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]]**

**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。

**SP2-00550 使用方法: SHOW ERRORS [[FUNCTION | PROCEDURE | PACKAGE | PACKAGE BODY | TRIGGER | VIEW | TYPE | TYPE BODY | 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。

**SP2-00556 ファイル名が無効です。**

**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。

**SP2-0560 使用方法: DESCRIBE [ スキーマ.] オブジェクト [. サブオブジェクト | @db\_link] [ 列]**

**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 [SIZE n] [FOR[MAT] [WRA[PPED] | WOR[D\_WRAPPED] | TRU[NCATED]]]**

**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、SQL を使用する、Oracle 固有の 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。

**SP2-00577 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てしました。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\_bytesbytes*) を割り当てできません。**

**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:** データ型を修正して、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。

**SP2-0599** 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** 使用方法: STORE コマンドが不在です。

**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。

**SP2-00609 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記録して、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。**

**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 サーバーの現行のバージョンが古すぎます。

**処置:** Oracle サーバー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 ロールが使用可能かを確認してください。**

**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が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、『Oracle9i データベース・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なければなりません。**

**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。  
プロダクト・ユーザー・プロファイル内でロールを使用禁止中にエラー発生。

**原因:** このエラー・メッセージは、PRODUCT\_USER\_PROFILE 表が SYSTEM アカウン  
トで作成されていないことを示す警告です。

**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 のインストー  
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。**

**原因:** このメッセージの後に、Oracle サーバー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、Oracle7.2 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。

**SP2-00631 "string\_name" で始まる文字列は長すぎます。**

最大サイズは *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。**

PASSW[ORD] [*username*]

**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 データベースに接続してくださ  
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。

**SP2-00642 SQL\*Plus 内部エラー、状態: *error\_state*、コンテキスト: *error\_number***  
続行するのは危険です。

**原因:** 内部エラーが発生しました。

**処置:** メッセージを記録して、オラクル社カスタマ・サポート・センター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、書き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キャラクタ・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 キャラクタ・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記録して、オラクル社カスタマ・サポート・センター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 "*format*" 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。

**SP2-00691 SYSDBA または SYSOPER が入ります。"*command\_name*" ではありません。**

原因: CONNECT AS 構文を使用しようとし、SYSDBA または SYSOPER 以外を指定しました。

処置: 構文を修正し、CONNECT コマンドを再発行してください。

**SP2-00692 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、コマンド行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システム管理者に問い合せてください。
- 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 LINESIZE *line\_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み込み不能である可能性があります。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のみ発生するため、このメッセージ・ファイルからは読み込めません。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。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 つのみ読み込みます。たとえば、NLS\_LANG が JAPANESE\_JAPAN.JA16EUC の場合、sp1ja.msb が読み込まれます。NLS\_LANG が設定されていない場合、デフォルト（英語）の spplus.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、ファイルの読み込み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、spplus.msb ファイルを読み込め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 サーバー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。

**SP2-00753 STARTUP MIGRATE は、Oracle 9.2 以上で有効です。**

**原因:** リリース 9.2 より前の Oracle サーバー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変数を確認し、リリース 9.2 以上の Oracle サーバーに接続していることを確認してください。

## iSQL\*Plus のエラー・メッセージ

### SP2-00850 コマンド "*command\_name*" は iSQL\*Plus では使用不能です。

**原因:** コマンドが SQL\*Plus エンジンで認識されていないか、または iSQL\*Plus で使用禁止です。このエラーは、コマンドが iSQL\*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、セキュリティ上の理由から許可されていない場合、または廃止されたコマンドである場合に発生します。

**処置:** スクリプトからコマンドを削除してください。コマンドが使用禁止である場合は、かわりのコマンドをドキュメントで確認してください。

### SP2-00851 "*command\_name...*" で始まるコマンドは iSQL\*Plus では使用不能です。

**原因:** コマンドが認識されていないか、または iSQL\*Plus で使用禁止です。このエラーは、コマンドが iSQL\*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、セキュリティ上の理由から許可されていない場合、または廃止されたコマンドである場合に発生します。

**処置:** スクリプトからコマンドを削除してください。コマンドが使用禁止である場合は、かわりのコマンドをドキュメントで確認してください。

### SP2-00852 オプションは iSQL\*Plus では使用不能です。

**原因:** このコマンド・オプションは、iSQL\*Plus で使用できません。このエラーは、通常、SP2-00158 または SP2-00735 の後で発生します。一部のオプションは、Web ベースのコンテキストでは意味を持たないため、使用できません。これらを使用すると、正常な操作が行えないか、またはセキュリティ上の危険があります。

**処置:** コマンドからオプションを削除してください。

### SP2-00853 「Username」フィールドが空です。

**原因:** iSQL\*Plus の「Login」画面で「Username」フィールドが空です。

**処置:** ログインの前にユーザー名を入力してください。

### SP2-00854 パスワードは 2 回入力できません。

**原因:** iSQL\*Plus の「Login」画面で「Username」および「Password」フィールドを解析中に、エラーが発生しました。

**処置:** ユーザー名の一部としてパスワードを入力 (たとえば「scott/tiger」) していないか、同時に「Login」画面の「Password」フィールドにパスワードを入力していないかを確認してください。パスワードは、1 回のみ入力してください。このエラーは、iSQL\*Plus が、どのパスワードを使用するかを判断できない場合に発生します。

**SP2-00855 接続識別子は 2 回入力できません。**

**原因:** iSQL\*Plus の「Login」画面で「Connection Identifier」フィールドの接続識別子を読み込もうとして、エラーが発生しました。

**処置:** 「Username」フィールドで、完全な接続構文（たとえば、username/password@connect\_identifier）が使用されている場合、「Connection Identifier」フィールドは、空にしてください。

**SP2-00856 使用方法: CONN[ECT] { username/password[@connect\_identifier] | / } [AS {SYSOPER | SYSDBA}]**

**原因:** iSQL\*Plus が Oracle インスタンスに接続できなかったか、またはユーザー名およびパスワードが正しくありませんでした。iSQL\*Plus スクリプトで、不完全または不適切な CONNECT コマンドが指定されました。

**処置:** iSQL\*Plus では、CONNECT コマンドに、すべての認証情報が含まれている必要があります。ユーザー名およびパスワードの両方が指定されていることを確認してください。

**SP2-00857 iSQL\*Plus が再起動されました。**

**原因:** 要求を処理する Web サーバー (Oracle HTTP Server) がタイムアウトしているか、セッションが期限切れであるか、または Web サーバーが停止しているかのいずれかです。いずれの場合も、セッション情報（たとえば、現行のデータベース接続、SET 変数の値）が使用不可です。

**処置:** Web サーバーを再起動する（必要な場合）か、または iSQL\*Plus に再度ログインしてください。

**SP2-00858 使用方法: SET MARKUP HTML [HEAD text] [BODY text] [TABLE text] [ENTMAP {ON | OFF}] [PRE[FORMAT] {ON | OFF}]**

**原因:** iSQL\*Plus で、SET MARKUP には無効なオプションが指定されました。

**処置:** 無効なオプションを削除してください。

**SP2-00860 既知のコマンドのリストを表示するには、HELP と入力してください。**

**原因:** 無効なコマンドが入力されました。

**処置:** HELP コマンドを使用して、SQL\*Plus コマンドの構文を表示してください。

**SP2-00863 iSQL\*Plus の処理が完了しました**

**原因:** iSQL\*Plus の入力スクリプト内のすべてのコマンドが出力されずに実行されました。

**処置:** 処置は必要ありません。

**SP2-00864 セッションが時間切れになりました。もう一度ログインしてください。**

**原因:** iSQL\*Plus のセッションが長時間アイドル状態であるため、他の接続用にリソースを解放するために、コンテキストが削除されました。

**処置:** iSQL\*Plus に再接続してください。システム管理者は、タイムアウト間隔を設定してください。

**SP2-00865 セッションがブロックされました。もう一度ログインしてください。**

**原因:** 前回の問合せがまだ処理中のときに、iSQL\*Plus からの問合せを実行しようとした。このエラーは、問合せの処理中に、ブラウザの「戻る」ボタンまたは「中止」ボタンを押したときに発生します。

**処置:** iSQL\*Plus に再接続してください。

**SP2-00866 入力領域に文を入力してください。**

**原因:** 入力領域に、実行される文がないときに、「Execute」ボタンがクリックされました。

**処置:** 実行する文を入力領域に入力し、「Execute」ボタンを再度クリックしてください。

**SP2-00867 保存するスクリプトがありません。**

**原因:** 保存する文が入力領域にないときに、「Save Script」ボタンがクリックされました。

**処置:** 保存する文を入力領域に入力し、「Save Script」ボタンを再度クリックしてください。

**SP2-00868 実行するスクリプトがありません。**

**原因:** スクリプト内に実行する文がないときに、スクリプトを実行し、結果をファイルに出力しようとした。

**処置:** スクリプトに実行する文が含まれていることを確認し、スクリプトを再度実行してください。

**SP2-00869 無効なファイル内容。**

**原因:** Web サーバーが認識できないフォーマットで、スクリプトを入力領域にロードしようとした。

**処置:** スクリプトがテキスト・ファイルで、ブラウザがファイルを認識するために必要な MIME タイプが正しく設定されていることを確認してください。拡張子 .SQL のファイルをロードする場合、通常は、ブラウザが SQL MIME タイプ設定であることを確認してください。

**SP2-00870 正常に終了しました。**

**原因:** これは情報メッセージです。iSQL\*Plus は、正常に終了しました。

**処置:** 処置は必要ありません。

**SP2-00871 ロードするスクリプトがありません。**

**原因:** 「Load Script」をクリックしましたが、ファイル名が指定されていないか、指定した名前のファイルが存在していないか、または、存在しても空です。

**処置:** フィールドに、実在するファイル名を入力し、再試行してください。指定したファイルが存在するが、空である場合、処置は必要ありません。

**SP2-00872 iSQL\*Plus では、SET AUTORECOVERY ON を使用してください。**

**原因:** iSQL\*Plus で、RECOVER コマンドの対話モードを使用しようとしてしました。iSQL\*Plus では、自動リカバリ・モードを使用してください。

**処置:** SET AUTORECOVERY ON に切り替えて、RECOVER コマンドを再実行してください。

**SP2-00873 予期しない引用符が URL 引数で見つかりました。**

**原因:** キーワード / 値の組合せの値の部分に、引用符が見つかりました。

**処置:** 確認し、余分な引用符を削除してください。

**SP2-00874 URL 引数にキーワードが欠落しています。**

**原因:** URL 引数のキーワード / 値の組合せに、キーワードが見つかりません。

**処置:** 欠落したキーワードまたは欠落した等号を確認してください。

**SP2-00875 URL 引数にキーワードはありますが値がありません。**

**原因:** URL のキーワード / 値の組合せで、キーワードの値が欠落しています。

**処置:** 欠落した値または欠落した等号を確認してください。

**SP2-00876 URL 引数に終了引用符が欠落しています。**

**原因:** URL のキーワード / 値の組合せで、開始引用符に一致する終了引用符が見つかりません。

**処置:** 終了引用符を確認して挿入してください。

**SP2-00877 URL 引数に予期しない文字が見つかりました。**

**原因:** キーワード / 値の組合せで、値は引用符で囲まれています、URL 引数で不要な文字が見つかりました。

**処置:** 不要な文字を削除してください。

**SP2-00878 重複するキーワード "keyword" が指定されています。**

**原因:** URL 引数の前回のパラメータで、すでにキーワードが指定されています。

**処置:** キーワードのいずれか1つを削除してください。

**SP2-00879 実行する動的レポートに対してスクリプトを指定してください。**

**原因:** スクリプトのキーワードが、URL で指定されていません。

**処置:** URL 引数に、スクリプトのキーワード / 値の組合せを追加してください。

**SP2-00880 スクリプト "script\_url" を実行するには接続詳細を入力してください。**

**原因:** 動的レポートの URL 引数に、接続文字列が指定されていません。つまり、ユーザー ID キーワードが指定されていません。

**処置:** 対話形式のログイン・ページからログインしてください。

**SP2-00881 username[/password] [@connect\_identifier] | /**

**原因:** 接続文字列で、iSQL\*Plus ユーザー名に不適切な構文が使用されました。AS SYSDBA 句または AS SYSOPER 句が含まれていません。

**処置:** 構文を修正してください。特権付きの接続には、ログイン・ページの AS SYSDBA フィールドまたは AS SYSOPER フィールドを使用してください。

**SP2-00882 「Password」フィールドが空です。**

**原因:** iSQL\*Plus の「Login」画面の「Password」フィールドが空です。

**処置:** ログインの前にパスワードを入力してください。

**SP2-00883 指定した入力領域サイズが無効です。**

**原因:** 入力領域のサイズに指定された幅または高さが数値以外であるか、または1～999の範囲外です。

**処置:** 入力領域のサイズに、妥当な数値を再度指定してください。

**SP2-00884 データベース "connect\_identifier" への接続は許可されません。**

**原因:** SQL\*Plus が接続できるように構成されていないデータベースに接続しようとしてしました。

**処置:** iSQL\*Plus の CONNECT コマンドでこのエラーが発生した場合、*isqlplus.conf* ファイルの「iSQLPlusConnectIdList」構成パラメータに、使用した接続識別子が含まれていません。たとえば、コマンド「CONNECT username/password@connect\_identifier」と入力する場合、「connect\_identifier」は、パラメータ・リストの接続識別子と一致している必要があります。接続識別子では、大 / 小文字が区別されません。

**SP2-00885 有効なユーザー名または「/」のみがユーザー名フィールドで許可されます。**

**原因:** iSQL\*Plus の「Login」画面の「Username」フィールドに、ユーザー名または「/」以外の値を入力しようとしました。

**処置:** iSQL\*Plus の「Login」画面の「Username」フィールドには、ユーザー名のみ指定できます。ユーザー名または「/」は、iSQL\*Plus DBA の「Login」画面に指定できます。

**SP2-00886 履歴にスクリプトがありません。**

**原因:** このセッションではまだスクリプトが実行されていないため、履歴のリストには使用可能なスクリプトがありません。

**処置:** スクリプトは、このセッションで実行された後のみ履歴のリストで使用可能になります。

**SP2-00887 指定した履歴サイズが数値でないか、[0 - 100] の範囲外です。**

**原因:** 指定された履歴のサイズが数値以外であるか、または 0 ~ 100 の範囲外です。

**処置:** 履歴のサイズに、0 ~ 100 の範囲で数値を再度入力してください。

**SP2-00888 キーワード "keyword" は、iSQL\*Plus では使用不能です。**

**原因:** iSQL\*Plus の URL 引数に、iSQL\*Plus が認識しないか、または iSQL\*Plus の DBA 接続時のみ使用可能なキーワードまたは変数を使用しようとしました。

**処置:** キーワードまたは変数の引数を削除するか、または iSQL\*Plus URL のかわりに、iSQL\*Plus DBA の URL を使用してください。

**SP2-00889 キーワード型に指定される値は、URI または TEXT のいずれかである必要があります。**

**原因:** 指定したキーワードに対して入力した値が無効です。

**処置:** 指定したキーワードに対して有効な値を使用して、コマンドを再入力してください。キーワード「type」に対して有効な値は、「URL」または「TEXT」です。

**SP2-00890 インスタンスが設定されていません。または、接続識別子が指定されていません。**

**原因:** CONNECT コマンドまたは SET INSTANCE コマンドに対して、接続識別子またはデータベース・インスタンスが指定されていません。

**処置:** コマンドを再入力し、有効な接続識別子またはデータベース・インスタンスを指定してください。

**SP2-00891 スクリプトをロードするには接続詳細を入力してください。**

**原因:** URL の「userid」引数が指定されていないか、あるいは指定された情報が不十分または不適切であるため、iSQL\*Plus はサーバーにログインできませんでした。

**処置:** スクリプトを「Work Screen」画面にロードするために、「Login」画面からログインしてください。

**SP2-00892 session の期限切れ中にエラーが発生しました。**

**原因:** iSQL\*Plus は、ユーザー・セッション ID で識別される、タイムアウトしたセッションを閉じることができませんでした。

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00893 session が期限切れになりました。**

**原因:** iSQL\*Plus は、ユーザー・セッション ID で識別される、タイムアウトしたセッションを期限切れにしました。

**原因:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00894 username が URL から connect\_id へのログインに失敗しました。**

**原因:** iSQL\*Plus は、指定されたユーザー名、URL、接続識別子によるユーザーでのログインに失敗しました。

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00895 username に session として URL から connect\_id へログインしました。**

**原因:** iSQL\*Plus は、指定されたユーザー・セッション ID、ユーザー名、URL および接続識別子を使用したユーザーでログインしました。

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00896 username からのログアウトに失敗しました。**

**原因:** iSQL\*Plus は、指定したユーザー・セッション ID を使用したユーザーのログアウトに失敗しました。

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00897 username からログアウトしました。**

**原因:** iSQL\*Plus は、指定したユーザー・セッション ID を使用したユーザーをログアウトしました。

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00898 アクティブ・セッション数 number**

**原因:** アクティブな iSQL\*Plus セッション数が表示されます。

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00899 期限切れになるセッション数は number です。アクティブ・セッションは number です。**

**原因:** タイムアウトし、期限切れにされるアクティブ・セッション数を表示します。

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00900 ユーザー名 = *username*、パスワード = *password*、接続識別子 = *connect\_id*、権限 = *privilege***

**原因:** iSQL\*Plus ログイン詳細の表示: ユーザー名、パスワード、接続識別子および権限

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00901 ユーザー名 = *username*、パスワード = *password*、接続識別子 = *connect\_id***

**原因:** iSQL\*Plus ログイン詳細の表示: ユーザー名、パスワードおよび接続識別子

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00902 接続文字列 = *username/password@sid privilege***

**原因:** SQL\*Plus 接続 API によって作成された、iSQL\*Plus 接続文字列の完全な構文の表示: *username/password@sid [AS {SYSDBA|SYSOPER}]*

**処置:** 処置は必要ありません。これはデバック・メッセージです。

**SP2-00911 パスワードのみがパスワード・フィールドで許可されます。**

**原因:** iSQL\*Plus の「Login」画面の「Password」フィールドに、パスワード以外の値を入力しようとした。

**処置:** 「Password」フィールドには、有効なパスワードのみ入力できます。

**SP2-00912 接続識別子のみが接続識別子フィールドで許可されます。**

**原因:** iSQL\*Plus の「Login」画面の「Connection Identifier」フィールドに、有効な接続識別子以外の値を入力しようとした。

**処置:** 「Connection Identifier」フィールドには、有効な接続識別子または有効な別名のみ指定できます。

**SP2-00913 iSQL\*Plus でキーワードが認識されませんでした。**

**原因:** iSQL\*Plus から手動で URL を入力してサービスを要求しようとしたが、キーワードが認識されませんでした。

**処置:** 構文および URL に手動で入力できるサービスのタイプについては、iSQL\*Plus のドキュメントを参照してください。

**SP2-00914 iSQL\*Plus で値 "*value*"( キーワード "*keyword*") が認識されませんでした。**

**原因:** iSQL\*Plus から手動で URL を入力してサービスを要求しようとしたが、キーワードに指定した値が認識されませんでした。

**処置:** 構文および URL に手動で入力できるサービスのタイプについては、iSQL\*Plus のドキュメントを参照してください。

**SP2-00915 Cookies が使用不可である場合があります。**

**原因:** ログイン後、iSQL\*Plus は、Cookie がブラウザで使用可能な場合のみ後続の要求を処理できます。

**処置:** ブラウザで Cookie を有効にし、再度ログインしてください。

**SP2-00916 スキーム *scheme* はサポートされていません。**

**原因:** iSQL\*Plus は、HTTP、HTTPS および FTP のみサポートします。

**処置:** サポートされているスキームに変更してください。

**SP2-00917 ユーザーが取消しを要求しました。**

**原因:** 実行中の iSQL\*Plus スクリプトが、ユーザーによって取り消されました。「Cancel」ボタンのクリック、スクリプトの再実行または「Work Screen」画面の他のボタンやリンクのクリックによって取り消された可能性があります。

**処置:** 処置は必要ありません。

**SP2-00918 URL をオープンしようとして HTTP エラー *number* が発生しました。**

**原因:** URL のコンテンツをフェッチしようとして、HTTP エラーが発生しました。URL は、名前が変更されたか、削除または一時的に使用不可になっている可能性があります。

**処置:** URL の綴りが正しく、リクエスト・サーバーから使用可能であることを確認してください。

**SP2-00919 不明な OS です。**

**原因:** 取得した OS バージョン情報から、オペレーティング・システム・プラットフォームを判断できません。

**処置:** 処置は必要ありません。

**SP2-00920 URL をオープンしようとして HTTP エラー *message* が発生しました。**

**原因:** URL のコンテンツをフェッチしようとして、HTTP エラーが発生しました。URL は、名前が変更されたか、削除または一時的に使用不可になっている可能性があります。

**処置:** URL の綴りが正しく、リクエスト・サーバーから使用可能であることを確認してください。

**SP2-00921 キーワード操作に指定された値は、EXECUTE または LOAD のいずれかである必要があります。**

**原因:** 指定したキーワードに対して入力した値が無効でした。

**処置:** 指定したキーワードに対して有効な値を使用して、コマンドを再入力してください。キーワード「action」に対して有効な値は、「EXECUTE」または「LOAD」です。

**SP2-00922 Windows のバージョンを判別できません。**

**原因:** 取得したバージョンの API を使用して、Windows のバージョンを判別できません。

**処置:** 処置は必要ありません。

**SP2-00923 iSQL\*Plus の URL から AS SYSDBA または AS SYSOPER でログインすることはできません。**

**原因:** AS SYSDBA または AS SYSOPER 権限で、iSQL\*Plus URL からログインしようとした。

**処置:** AS SYSDBA 権限または AS SYSOPER 権限でのログインは、iSQL\*Plus DBA URL からのみ許可されています。AS SYSDBA または AS SYSOPER 引数を、ログイン要求から削除するか、または iSQL\*Plus DBA URL からログインしてください。

**SP2-00924 指定された接続文字列が重複しているか、キーワードが不適切です。**

**原因:** 接続文字列で指定されたログインの詳細に、重複するキーワード、不適切なキーワードまたはシーケンス・キーワードが含まれている可能性があります。

**処置:** 接続文字列の構文を確認し、不適切な内容を削除または修正してください。

**SP2-00925 ロードまたは削除するスクリプトが選択されていません。**

**原因:** 「Script History」 ページで、履歴リストからロードまたは削除するスクリプトが選択されていません。

**処置:** ロードまたは削除する各スクリプトのチェックボックスをクリックし、「Load」ボタンまたは「Delete」ボタンをクリックしてください。

**SP2-0926 iSQL\*Plus 内部エラー: 状態 = *number*、メッセージ = *message***

**原因:** iSQL\*Plus で内部エラーが発生しました。

**処置:** メッセージを記録して、オラクル社カスタマ・サポート・センターに連絡してください。

## COPY コマンドのメッセージ

**CPY00002 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：**オラクル社カスタマ・サポート・センターに連絡してください。

**CPY00003 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：**オラクル社カスタマ・サポート・センターに連絡してください。

**CPY00004 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。

**CPY00005 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：**COPY コマンドを指定しなおします。選択する項目のデータ型が宛先と一致していることを確認します。変換するには、TO\_DATE、TO\_CHAR および TO\_NUMBER を使用します。

**CPY00006 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。

**CPY00007 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。

**CPY00008 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。

**CPY00009 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。

# 第IV部

---

## SQL\*Plus の付録

この部の内容は、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
- iSQL\*Plus では未サポートのコマンド



# A

---

---

## SQL\*Plus の制限

この付録では、SQL\*Plus の一般的な制限について説明します。記載されている制限値は、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	30 バイト
ユーザー変数名の長さ	30 バイト
ユーザー変数値の長さ	240 文字
コマンドラインの長さ	2500 文字
SQL*Plus で入力する LONG 値の長さ	LINESIZE 値
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
変数置換後の行のサイズ	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	VMS の場合は 60,000 文字、その他の場合は 32,767 文字
最大 ARRAYSIZE	5,000 行
ネストされたスクリプトの最大数	VMS、CMS、UNIX の場合は 20、その他の場合は 5
最大ページ数	99,999
PL/SQL エラー・メッセージの最大サイズ	2KB
ACCEPT 文字列の最大長	240 バイト
DEFINE 変数の最大数	2048

---

---

---

## 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

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、[第1章「SQL\\*Plus 概要」](#)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廃止される予定です。COPY コマンドでは、この付録で示すデータ型はサポートされますが、今後、新しいデータ型はサポートされません。

## COPY コマンドの構文

### 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、ローカル・データベースまたはリモート・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

### 項

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。

**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。

**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 データベースを *username[/password]@connect\_identifier* で指定します。FROM 句または TO 句のいずれかのパスワードを指定しない場合、SQL\*Plus でその入力を求めるプロンプトが表示されます。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、Oracle Net 接続文字列を構成する *connect\_identifier* 句を含める必要があります。正確な構文は、Oracle インストールで使用する Oracle Net 通信プロトコルによって異なります。詳細は、ご使用のプロトコルに該当する Oracle Net のマニュアルを参照するか、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。

## 使用方法

Oracle データベースと 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 データベース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 データベース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。バッチのサイズは、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。

## 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/your_password@HQ TO JOHN/your_password@WEST -  
REPLACE WESTEMPLOYEES -  
USING SELECT * FROM EMPLOYEES
```

次のコマンドは、EMPLOYEES 表から選択された行を SQL\*Plus が接続しているデータベースにコピーします。SQL\*Plus は、コピー時に SALESMEN を作成します。SQL\*Plus では、EMPLOYEE\_ID および LAST\_NAME 列のみがコピーされ、コピー先でのこれらの列の名前は EMPLOYEE\_ID および SA\_MAN になります。



```
COPY FROM HR/your_password@ORACLE01 -  
CREATE SALESMEN (EMPLOYEE_ID, SA_MAN) -  
USING SELECT EMPLOYEE_ID, LAST_NAME FROM EMPLOYEES -  
WHERE JOB_ID='SA_MAN';
```

## 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

---

---

**注意：** 一般に、COPY コマンドは、Oracle データベースと 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 データベース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。

---

---

## 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/your_password@BOSTONDB -  
TO TODD/your_password@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 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わかりユーザーが入力したアクション句（前述の例では CREATE）によって異なります。詳細は、この章の後半の「[宛先の表の扱いを制御する方法](#)」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。

---

---

**注意：** Oracle データベースと 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 データベース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 データベース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/your_password@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。

## 宛先の表の扱いを制御する方法

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。

#### 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。

---



---

**注意：** 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/your_password@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 句が指定されていると、BOSTONDB で指定されたデータベースに、ユーザー名 HR、パスワード *your\_password* で接続します。

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 は問合せを含んでいても、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。

## COPY で表示されるメッセージの解釈

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 変数は、COPY が変更をデータベースにコミットした後のバッチの数を設定します (COPYCOMMIT を 0 (ゼロ) 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)。SET 変数の詳細は、13-92 ページの「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。

## 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 DEPT 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/your_password@BOSTONDB -  
CREATE EMPLOYEE_COPY2 -  
USING SELECT * FROM ADAMS.DEPT
```

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。

## 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

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とパスワードおよびローカル・データベースのサービス名を指定します。

```
COPY FROM HR/your_password@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、パスワードおよびサービス名を挿入します。

```
COPY FROM HR/your_password@BOSTONDB -  
TO HR/your_password@BOSTONDB -  
INSERT EMPLOYEE_COPY2 -  
USING SELECT * FROM EMPLOYEE_COPY
```



---

---

## 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。

## 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 (旧形式)	BTITLE (13-23 ページ)	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 (13-29 ページ)	列の表示属性をデフォルトにリセットします。
DOCUMENT	REMARK (13-83 ページ)	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 NEWPAGE (13-110 ページ)	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 (13-59 ページ)	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 CLOSECURSOR	なし	廃止
SET DOCUMENT	なし	廃止
SET MAXDATA	なし	廃止
SET SCAN	SET DEFINE (13-102 ページ)	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 (13-101 ページ)	SELECT によって選択された列の間に出力するテキストを設定します。
SET TRUNCATE	SET WRAP (13-119 ページ)	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	なし	廃止
TTITLE (旧形式)	TTITLE (13-138 ページ)	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。

## BTITLE (旧形式)

### 用途

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。

### 構文

```
BTI [TLE] text
```

### 使用方法

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、この付録の「[TTITLE \(旧形式\)](#)」を参照してください。

## COLUMN DEFAULT

### 用途

特定の列の表示属性をデフォルト値にリセットします。

### 構文

```
COL [UMN] {column|expr} DEF [AULT]
```

### 使用方法

COLUMN CLEAR と同じ働きをします。

## DOCUMENT

### 用途

スクリプト内のドキュメントのブロックを開始します。

### 構文

```
DOC [UMENT]
```

## 使用方法

スクリプトにコメントを挿入するための現行のメソッドの詳細は、6-9 ページの「[スクリプトへのコメントの挿入](#)」および 13-83 ページの「[REMARK](#)」コマンドを参照してください。

DOCUMENT と入力して [Return] を押すと、ドキュメントの作成が終了するまで、「SQL>」のかわりに「DOC>」というプロンプトが表示されます。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、この付録の「[SET DOCUMENT](#)」を参照してください。

## NEWPAGE

### 用途

$n$  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。

### 構文

```
NEWPAGE [1 |  $n$ ]
```

### 使用方法

スプール出力を進めるための現行のメソッドの詳細は、13-110 ページの「[SET](#)」コマンドの [NEWPAGE](#) 変数を参照してください。

## SET BUFFER

### 用途

指定したバッファを現行バッファにします。

### 構文

```
SET BUF[FER] {buffer | SQL}
```

### 使用方法

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。

## SET CLOSECURSOR

### 用途

カーソル使用動作を設定します。

### 構文

```
SET CLOSECUR[SOR] {ON|OFF}
```

### 使用方法

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。

## SET DOCUMENT

### 用途

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。

### 構文

```
SET DOC[UMENT] {ON|OFF}
```

### 使用方法

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 コマンドの詳細は、この付録の「[DOCUMENT](#)」を参照してください。

## SET MAXDATA

### 用途

SQL\*Plus が処理できる最大行幅を設定します。

### 構文

```
SET MAXD [ATA] n
```

### 使用方法

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。

## SET SCAN

### 用途

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。

### 構文

```
SET SCAN {ON|OFF}
```

### 使用方法

ON を指定すると、SET DEFINE ON と同じ機能が得られます。

## SET SPACE

### 用途

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。

### 構文

```
SET SPACE {1|n}
```

## 使用方法

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 を使用の方が便利です。

## SET TRUNCATE

### 用途

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。

### 構文

```
SET TRU[NCATE] {ON|OFF}
```

### 使用方法

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 を使用の方が便利です。

## SHOW LABEL

### 用途

現行セッションのセキュリティ・レベルを表示します。

### 構文

```
SHO[W] LABEL
```

## TTITLE (旧形式)

### 用途

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。

### 構文

```
TTI[TLE] text
```

## 使用方法

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。エスケープ文字の設定の詳細は、13-104 ページの「SET」コマンドの ESCAPE 変数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、13-138 ページの「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'
```

---

## iSQL\*Plus では未サポートのコマンド

ここでは、iSQL\*Plus のユーザー・インタフェースではサポートされない SQL\*Plus コマンドを示します。未サポートのコマンドまたはコマンド・オプションを使用すると、エラー・メッセージ SP2-00850 が発生します。

次のコマンドは、iSQL\*Plus ではコンテキストを持たないため、実装されていません。

ACCEPT	PASSWORD
CLEAR SCREEN	PAUSE

次の SET コマンド変数は、iSQL\*Plus ではコンテキストを持たないため、実装されていません。

SET EDITFILE	SET SQLBLANKLINES	SET TAB
SET FLUSH	SET SQLCONTINUE	SET TERMOUT
SET NEWPAGE	SET SQLNUMBER	SET TIME
SET PAUSE	SET SQLPREFIX	SET TRIMOUT
SET SHIFTOINOUT	SET SQLPROMPT	SET TRIMSPOOL
SET SHOWMODE	SET SUFFIX	

次のコマンドは、中間層でのセキュリティの問題があるため、実装されていません。

GET	SPOOL
HOST	STORE

次のコマンドは、iSQL\*Plus には関連しない SQL バッファの編集コマンドであるため、実装されていません。

APPEND	DEL	INPUT
CHANGE	EDIT	SAVE



---

# 用語集

## ARCHIVELOG

オンライン REDO ログ・ファイルを再利用する前にアーカイブする REDO ログ・モード。ARCHIVELOG モードにすると、インスタンスおよびディスク障害からデータベースを完全にリカバリできる。データベースがオープンされ使用できる状態になっているときに、バックアップを取ることもできる。ただし、管理操作を追加して、アーカイブ REDO ログをメンテナンスする必要がある。「[アーカイブ REDO ログ \(archived redo log\)](#)」を参照。

## ASCII

デジタル・データを使用して印字可能文字を表すための規則。ASCII は、American Standard Code for Information Interchange の頭文字。

## CGI (Common Gateway Interface)

Web サーバーの一部で、サーバー上で実行するプログラムを使用して、ユーザーとの対話（通常は Web ブラウザを介する）を可能にする。CGI スクリプトでは、対話形式で Web ページまたは Web ページの要素を動的に作成するか、または、Web ページに対して入力し、その入力内容に対して応答できる。ユーザーがオンラインで記入して送信するという対話形式での使用方法が一般的である。CGI スクリプトで一般的に使用される言語には、Perl、Java スクリプトおよび Java がある。

## CGI スクリプト (CGI script)

「[CGI \(Common Gateway Interface\)](#)」を参照。

### **CHAR データ型 (CHAR datatype)**

ANSI/ISO 互換性に提供される Oracle データ型。CHAR 列は、固定長の列である。また、いくつかの印刷可能文字 (A、3、&、空白など) を含むことができ、1 ~ 2000 バイトまでの文字または NULL を格納できる。CHAR データ型の詳細は、『Oracle9i SQL リファレンス』を参照。

### **CLOB データ型 (CLOB datatype)**

標準の Oracle データ型の 1 つ。CLOB データ型は、シングルバイトのキャラクタ・ラージ・オブジェクト・データを格納するために使用され、最大 2GB までの文字データを格納できる。

### **DATE データ型 (DATE datatype)**

日時のデータを格納するために使用される標準の Oracle データ型。デフォルトの日付書式は、YY-MM-DD (たとえば、98-11-23) である。DATE 列には、紀元前 4712 年 1 月 1 日から西暦 9999 年 12 月 31 日までの日時を指定できる。

### **DBA**

「データベース管理者 (database administrator: DBA)」を参照。

### **DCL**

「データ制御言語 (Data Control Language: DCL)」を参照。

### **DDL**

「データ定義言語 (Data Definition Language: DDL)」を参照。

### **DML**

「データ操作言語 (Data Manipulation Language: DML)」を参照。

### **DUAL 表 (DUAL table)**

1 行のみ含まれている DUAL という名前の標準の Oracle データベース表。DUAL 表は、小さいダミー表 (意味のないデータ) を要求して既知の結果 (真など) を保証するアプリケーションに役立つ。

### **HTML**

WWW 上のほとんどの文書を記述する際に使用する言語。Hypertext Markup Language (HTML) で記述すると、ほぼすべての Web ブラウザで表示でき、リンクを指定できる。HTML に関する勧告は、World Wide Web Consortium (W3) から公開されている。詳細は、<http://www.w3.org> を参照。

## **iSQL\*Plus**

SQL\*Plus に対するブラウザ・ベースのインタフェース。SQL\*Plus 製品のコンポーネントの 1 つ。

iSQL\*Plus によって、Web ブラウザを使用して Oracle9i データベースに接続し、コマンドラインで SQL\*Plus を使用した場合と同じ作業を実行できる。iSQL\*Plus を使用して SQL\*Plus、SQL および PL/SQL のコマンドを記述し、次の作業を実行できる。

- SQL コマンドおよび PL/SQL ブロックの入力、編集、実行および保存
- 問合せ結果の計算および出力
- 表の列定義の表示
- データベース間でのデータへのアクセスおよびデータのコピー
- データベースの管理

## **iSQL\*Plus サーバー (iSQL\*Plus Server)**

iSQL\*Plus の 3 層アーキテクチャの中間層に存在する。SQL\*Plus エンジンおよび iSQL\*Plus モジュールで構成される。Oracle HTTP Server で提供される Web サーバーと組み合わせて使用すると、Oracle9i で Web 対応の iSQL\*Plus インタフェースが使用可能になる。

## **LGWR**

「[ログ・ライター \(Log Writer: LGWR\)](#)」を参照。

## **LONG データ型 (LONG datatype)**

標準の Oracle データ型の 1 つ。LONG 列は、A、3、&、空白などの印刷可能文字を含むことができ、0 ~ 2GB までの長さになれる。

## **MARKUP**

SQL\*Plus 出力を HTML 形式で生成し、インターネットで配信可能にするための SET MARKUP 句または SQLPLUS -MARKUP 句。HTML で生成された SQL\*Plus 出力は、HTML 3.2 をサポートしている Web ブラウザで表示できる。

## **NCHAR データ型 (NCHAR datatype)**

Oracle9i 以上では、NCHAR データ型は、Unicode データ型のみになるように再定義される。NCHAR データ型は、固定幅の各国語キャラクタ・セット文字列を指定する。幅指定は文字数を示す。列サイズの最大値は、2000 バイト。NCHAR データ型の詳細は、『Oracle9i SQL リファレンス』を参照。

## **NCLOB データ型 (NCLOB datatype)**

標準の Oracle データ型の 1 つ。NCLOB データ型は、固定幅の各国語キャラクタ・セット文字 (NCHAR) データの格納に使用される。最大 4GB の文字テキスト・データを格納できる。

## Net8

「Oracle Net」を参照。

## NLS\_LENGTH\_SEMANTICS

SQL\*Plus クライアント・アプリケーションで使用される環境パラメータ。バイト長セマンティクスまたは文字長セマンティクスのいずれかを使用して CHAR、VARCHAR2 列および変数を作成できる。NCHAR、NVARCHAR2、CLOB および NCLOB 列は、常に文字ベースであるため、この変数の影響を受けない。セッションの起動時にこの変数が明示的に設定されていない場合、デフォルト値である BYTE (バイト長セマンティクス) が使用される。

NLS\_LENGTH\_SEMANTICS の値は、明示的に長さ指定しない CHAR または VARCHAR2 宣言の長さセマンティクスとして適用される。変数の表示時、または表、ビュー、シノニムまたはその他のオブジェクトの記述時にも使用される。サーバー側で、

NLS\_LENGTH\_SEMANTICS は、初期化パラメータとして設定され、SQL コマンド ALTER SESSION および ALTER SYSTEM を使用して動的に変更される。サーバー側での NLS\_LENGTH\_SEMANTICS 設定の詳細は、『Oracle9i Database グローバリゼーション・サポート・ガイド』を参照。NLS\_LENGTH\_SEMANTICS は、クライアントとサーバー間で異なるが、セッションの有効範囲を変更する ALTER SESSION SET NLS\_LENGTH\_SEMANTICS=value コマンドの発行は、SQL\*Plus セッションで反映されることに注意。

## NULL

値が使用できないまたは値が不明であることを意味する値。NULL は、特定のどの値 (NULL も含む) と同等しくない。NULL との比較は、常に偽になる。

## NULL 値 (NULL value)

値が存在しないこと。

## NUMBER データ型 (NUMBER datatype)

標準の Oracle データ型の 1 つ。NUMBER 列には数値を含むことができ、その数値に小数点および記号を付けることもできる。1 ～ 105 桁の 10 進数 (38 桁までが有効) が使用できる。

## NVARCHAR2 データ型 (NVARCHAR2 datatype)

Oracle9i 以上では、NVARCHAR2 データ型は、Unicode のみのデータ型として再定義される。NVARCHAR2 データ型は、可変幅の各国語キャラクタ・セット文字列を指定する。幅指定は文字数を示す。列サイズの最大値は、4000 バイト。NVARCHAR2 データ型の詳細は、『Oracle9i SQL リファレンス』を参照。

## Oracle Net

ネットワーク上でクライアント / サーバー通信とサーバー / サーバー通信の両方を可能にする Oracle のリモート・データ・アクセス・ソフトウェア。Oracle Net は、分散処理および分散データベース機能をサポートしている。Oracle Net は多数の通信プロトコルで動作し、それらを相互に接続する。Oracle Net は、Net8 および SQL\*Net バージョン 2 と下位互換がある。

## Oracle サーバー (Oracle Server)

オラクル社が販売しているリレーショナル・データベース管理システム (RDBMS)。RDBMS のコンポーネントには、DBA およびデータベース・ユーザーが使用するカーネルおよび様々なユーティリティが含まれている。

## PL/SQL

オラクル社が開発した SQL の 3GL 手続き型言語拡張機能。PL/SQL では、SQL が持つ使用しやすさと柔軟性と、構造化プログラミング言語の手続き型機能 (IF...THEN、WHILE、LOOP など) が結合されている。PL/SQL がデータベースに格納されていない場合でも、アプリケーションは、個々の SQL 文ではなく PL/SQL のブロックをデータベースに送る。そのため、ネットワークの通信量を軽減できる。

PL/SQL は実行時に解釈および解析されるので、コンパイルは必要ない。

## RAW データ型 (RAW datatype)

標準の Oracle データ型。RAW データ列は、バイナリを含む形式でデータを含む場合がある。RAW 列を使用して、バイナリ (非文字) ・データを格納できる。

## RDBMS (Relational Database Management System)

Oracle7 以下の用語。データベース内の実際のデータのみでなく、システムの作成およびメンテナンスに使用するソフトウェアを示す。「リレーショナル・データベース管理システム (Relational Database Management System: RDBMS)」、「サーバー (server)」および「Oracle サーバー (Oracle Server)」も参照。

## REDO ログ (redo log)

データに与えたすべての変更の連続したログ。REDO ログは、変更がディスクに書き込まれることを妨げる障害のイベント内で書き込まれたり使用される。REDO ログは、複数の REDO ログ・ファイルで構成される。

## REDO ログ・ファイル (redo log file)

データベースへのすべての変更のレコードを含むファイル。これらのファイルは、リカバリ操作のために使用される。「REDO ログ (redo log)」を参照。

## SELECT 構文のリスト (SELECT list)

問合せ内のキーワード SELECT に続く項目のリスト。これらの項目には、列名、SQL ファンクション、定数、疑似列、列の計算および別名が含まれる場合がある。問合せの結果に含まれる列の数は、SELECT 構文のリスト内の項目の数と一致する。

## SELECT 文 (SELECT statement)

複数の表またはビューからフェッチする行および列を指定する SQL 文。「SQL 文 (SQL statement)」を参照。

## SET コマンド変数 (SET command variable)

「システム変数 (system variable)」を参照。

## SGA

「システム・グローバル領域 (System Global Area: SGA)」を参照。

## SQL\*Loader

オペレーティング・システム・ファイルから Oracle データベース表にデータをロードするために使用する Oracle のツール製品。

## SQL\*Net

Net8 の前身。Oracle サーバーとともに動作し、Oracle RDBMS または Oracle のツール製品 (SQL\*Forms など) を実行する複数のコンピュータが、ネットワークを介してデータを交換できるようにする Oracle 製品。SQL\*Net は分散処理と分散データベース機能をサポートする。SQL\*Net は多くの通信プロトコルで動作し、それらを相互に接続する。

## SQL\*Plus

データ操作、データ定義および Oracle データベースに対するアクセス権の定義用の、SQL をベースとした対話型言語。多くの場合、エンド・ユーザーのレポート作成ツールとして使用される。

## SQL コマンド (SQL command)

「SQL 文 (SQL statement)」を参照。

## SQL スクリプト (SQL script)

SQL\*Plus で起動して、データベース管理を迅速および簡単に実行できる SQL 文を含むファイル。

## SQL バッファ (SQL buffer)

ユーザーが入力した最新の SQL コマンドまたは PL/SQL ブロックを含むデフォルト・バッファ。SQL\*Plus コマンドは、SQL バッファに格納されません。

## SQL 文 (SQL statement)

SQL 言語で記述された完全なコマンドまたは文。文 (statement) (SQL) と同義語。

## Structured Query Language (SQL)

リレーショナル・システム用に国際的に認められている規格。問合せのみでなく、データ定義、操作、セキュリティ、参照整合性などにも使用される。「データ操作言語 (Data Manipulation Language: DML)」、「データ定義言語 (Data Definition Language: DDL)」および「データ制御言語 (Data Control Language: DCL)」も参照。

## **SYSDBA**

Admin Option および SYSOPER システム権限とともに、すべてのシステム権限が含まれる権限。「[SYSOPER](#)」を参照。

## **SYSOPER**

DBA が STARTUP、SHUTDOWN、ARCHIVE LOG、RECOVER などの操作を実行できる権限。「[SYSDBA](#)」を参照。

## **SYSTEM ユーザー名 (system username)**

各データベースに自動的に作成される 2 つの標準 DBA ユーザー名のうちの 1 つ (もう 1 つは SYS)。Oracle ユーザー SYSTEM は、パスワード MANAGER とともに作成される。SYSTEM ユーザー名は、データベースをメンテナンスする DBA が優先的に使用するユーザー名である。

## **SYS ユーザー名 (SYS username)**

「[SYSTEM ユーザー名 \(system username\)](#)」を参照。

## **USERID**

オプションの Oracle Net アドレスを使用して、ユーザー名およびパスワードを指定できるコマンドライン引数。

## **VARCHAR**

Oracle データ型の 1 つ。このデータ型は、Oracle の VARCHAR2 データ型 (後述の定義を参照) と同じように機能する。ただし、将来的に VARCHAR の機能を変更する場合がありますので、VARCHAR のかわりに VARCHAR2 を使用することをお勧めする。

## **VARCHAR2**

Oracle データ型の 1 つ。可変長の英数字文字列。最大長は 4000 バイト。列型 VARCHAR2 に入力されたデータが 4000 バイト未満の場合、空白の埋込みは行われず、データは入力された長さで格納される。入力されたデータが 4000 バイトを超える場合、エラーが発生する。VARCHAR2 データ型の詳細は、『Oracle9i SQL リファレンス』を参照。

## **XMLType データ型 (XMLType datatype)**

標準の Oracle データ型の 1 つ。XML 文書を格納する場合に使用する。他の Oracle データ型と同様に操作および選択できる。最大 2GB の XML データを格納できる。

## **アーカイブ REDO ログ (archived redo log)**

再利用する前にオンライン REDO ログ・ファイルをアーカイブするリカバリ構造。

## アカウント (account)

オペレーティング・システムまたは製品 (Oracle データベース・サーバー、Oracle Forms など) の権限があるユーザー。オペレーティング・システムによっては、ID、ユーザー ID、ログインなどを指す。アカウントは、システム管理者により作成および制御される。

## 位置合せ (alignment)

データをフィールド内で位置付ける方法。定義されたフィールド幅で左揃え、右揃え、中央揃え、フラッシュ / 左揃え、フラッシュ / 右揃え、またはフラッシュ / 中央揃えに位置合せができる。

## インスタンス (instance)

Oracle データベースへの接続に必要なバックグラウンド・プロセスおよびメモリー領域。データベース・システムには、1つのインスタンスおよび1つのデータベースが必要である。Oracle データベース・サーバーは、SGA および Oracle データベース・サーバー・システムの一連のプロセスで構成される。

## インスタンス障害 (instance failure)

「データベース・インスタンス障害 (database instance failure)」を参照。

## インスタンス・リカバリ (instance recovery)

ソフトウェアまたはハードウェア障害のイベント内でのインスタンスのリカバリ。インスタンスをリカバリすると、再度データベースが使用可能になる。インスタンスが異常終了すると、次のインスタンス起動でインスタンス・リカバリが自動的に実行される。

## エディタ (editor)

ファイルの作成または変更を行うプログラム。

## エラー・メッセージ (error message)

コンピュータ・プログラム (SQL\*Plus など) からのメッセージ。プログラムまたはコマンドの実行を妨げる潜在的な問題をユーザーに通知する。

## エンド・ユーザー (end user)

システムの開発対象になるユーザー。たとえば、航空券予約システムのエンド・ユーザーは、航空券予約担当になる。「SQL\*Plus」を参照。

## オープン状態のデータベース (open database)

インスタンスによってマウントおよびオープンされ、ユーザーのアクセスが可能なデータベース。データベースがオープンしている場合、ユーザーは、データベース内の情報にアクセスできる。「マウントされたデータベース (mounted database)」を参照。

## オブジェクト (object)

オブジェクト型のインスタンス。Oracle9i では、オブジェクトは、持続的（たとえば、データベースに格納されている）である場合と、一時的（たとえば、PL/SQL または Oracle Call Interface (OCI) の変数）である場合がある。「[オブジェクト型 \(object type\)](#)」を参照。

## オブジェクト型 (object type)

オブジェクトの構造および動作をモデルとするユーザー定義型。異なるプログラミング言語内のクラス概念と等しい。Oracle9i のオブジェクト型には、属性およびメソッドを持つパブリック・インタフェースがある。オブジェクト型は、抽出データ型と呼ばれることもある。

## オブジェクト・リレーショナル・モデル (object-relational model)

単一のシステムへのリレーショナルおよびオブジェクト・データ・モデルのキーの外観を結合するデータベース・モデル。Oracle9i は、オブジェクト・リレーショナル・データベース・システムである。

## オペレーティング・システム (operating system)

コンピュータ・リソースを管理するシステム・ソフトウェア。メモリーの割当てなどの基本的な操作を実行し、コンピュータ・コンポーネントを通信できるようにする。

## 折返し (wrapping)

レポートまたは出力の機能の 1 つ。テキスト全体が 1 行で収まらない場合に、テキストの一部を新しい行に移動させる機能。

## オンライン REDO ログ (online redo log)

(1) アーカイブされていないが、データベース・アクティビティを記録するためのインスタンスに使用可能であるか、またはデータが一杯であるためにアーカイブまたは再利用を待機している REDO ログ・ファイル。(2) データベースに行われたすべてのコミット済変更を記録している複数のオンライン REDO ログ・ファイルのセット。

## 改ページ (form feed)

実行すると、プリンタが新しい用紙の上端へ移動する制御文字。SQL\*Plus で改ページを表示すると、ほとんどの端末では改ページによって画面が消去される。

## 拡張子 (extension)

オペレーティング・システムによっては、ファイル指定全体の 2 番目の部分。いくつかの標準のファイル拡張子は、ファイルの種類または用途を示すために使用される。たとえば、SQL、LOG、LIS、EXE、BAT、DIR などがそれにあたる。オペレーティング・システムによっては、ファイル・タイプとも呼ばれる。

## 型 (type)

列には、CHARACTER、DATE、NUMBER または LONG の 4 つの型の 1 つの情報が含まれる。ユーザーが列の内容に対して実行できる操作は、その列に入っている情報の型によって異なる。「書式 (format)」を参照。

## カレント行 (current line)

SQL\*Plus エディタなどのエディタで、編集コマンドが影響する現行バッファ内の行。

## 行 (row)

(1) レコードの同義語。複数の列に対する値を持つデータベース表内の 1 行になっているデータ。タプルと呼ばれることもある。(2) 問合せの出力に含まれる一連のフィールド値。「列 (column)」を参照。

## 切捨て (truncate)

故意にまたは誤って、値の先頭または終わりから 1 つ以上の文字を破棄または失うこと。

## 句 (clause)

SQL 文の一部。句自体は、SQL の全文にはならない。たとえば、WHERE 句がそれにあたる。

## クライアント (client)

他のアプリケーションまたはコンピュータ (サーバー) のサービス、データまたは処理が必要なユーザー、ソフトウェア・アプリケーションまたはコンピュータ。2 タスク環境では、クライアントはユーザー・プロセスとなる。ネットワーク環境では、クライアントはローカル・ユーザー・プロセスで、サーバーはローカルまたはリモートになる。

## 計算 (computation)

データベースからフェッチしたデータでランタイム計算を実行するために使用される。これらの計算は、SELECT 文で直接実行できる種類の計算のスーパーセット。「計算式列 (formula column)」を参照。

## 計算結果列 (computed column)

「計算 (computation)」を参照。

## 計算式列 (formula column)

PL/SQL プロシージャ、ファンクションまたは式、ユーザー・イグジット、SQL 文またはそれらの任意の組合せからデータを取得する手動で作成された列。

## 構成 (configuration)

Oracle Net で、ネットワーク通信を準備するための一連の指示。詳細は、Oracle Net のマニュアルを参照。

## 構成ファイル (configuration files)

ネットワークのコンポーネントの識別および特性設定をするために使用されるファイル。構成のほとんどは、ネットワーク・コンポーネントをネーミングし、コンポーネント間の関係を識別するプロセス。

## 構文 (syntax)

コマンド、修飾子およびパラメータを結合して、有効なコマンド文字列を形成するための規則系。

## コマンド (command)

特定のタスクを実行するための、プログラム、アプリケーション、オペレーティング・システムまたは他のソフトウェアの要求、またはそれらへの指示。コマンドは、1つの単語の場合と、その他の句、様々な引数、オプション、パラメータまたは修飾子を必要とする場合がある。文とは異なり、コマンドは入力するとすぐに実行される。SQL\*Plus コマンドとしては、ACCEPT、CLEAR および COPY コマンドがそれにあたる。

## コマンド・ファイル (command file)

「スクリプト (script)」を参照。

## コマンド・プロンプト (command prompt)

デフォルトでは SQL> というテキスト。これを使用して、SQL\*Plus は次のコマンドを要求する。

## コマンドライン (command-line)

キーボード入力したコマンドが表示されるコンピュータの画面上の行。たとえば、コンピュータの DOS プロンプトに続く領域がそれにあたる。「プロンプト (prompt)」を参照。

## コミット (commit)

データベース内のデータに永続的な変更（挿入、更新、削除）を加える。変更をコミットする前は、変更を格納するか、または、データを以前の状態にリストアできるように新旧両方のデータが存在する。

## コメント (comment)

プログラムに説明のテキストを挿入するための言語要素。コメントを実行しても影響はない。

## サーバー (server)

Oracle データベースに対して、同時に共有データ・アクセスする場合に必要な機能を処理する Oracle ソフトウェア。サーバーは、クライアント・アプリケーションから発行された SQL 文および PL/SQL 文を取得して処理する。サーバーを管理するコンピュータは、その負荷に合わせて最適化されている必要がある。

### **サマリー (summary)**

サマリーまたはサマリ一行を使用して、小計、総合計、実行合計および他のレポート内のデータの合計を計算する。

### **サマリ一行 (summary line)**

合計、平均、最大値または他の計算した値を含むレポート内の行。ユーザーは、BREAK および COMPUTE コマンドを使用してサマリ一行を作成する。

### **式 (expression)**

既存の値から新しい値を計算するために使用される SALARY + COMMISSION などの式。式は、列名、関数、演算子および定数で構成される。式は、コマンドまたは SQL 文にある。

### **システム・エディタ (system editor)**

オペレーティング・システムが提供するテキスト・エディタ。

### **システム管理者 (system administrator)**

コンピュータのオペレーティング・システムの運用およびメンテナンスに対する責任者。

### **システム・グローバル領域 (System Global Area: SGA)**

ユーザー・プロセスおよびバックグラウンド・プロセスが要求する情報 (1 つの Oracle インスタンスに対するデータや制御情報など) を含む共有記憶域。

SGA は、Oracle インスタンスが起動するときに割り当てられ、インスタンスが停止するときに割当てが解除される。

### **システム変数 (system variable)**

状態または環境を示す変数。Oracle または SQL\*Plus によってデフォルト値が提供される。たとえば、LINESIZE、PAGESIZE など。SQL\*Plus コマンドの SHOW および SET を使用して、システム変数の値を表示および変更する。

### **自動コミット (autocommit)**

SQL\*Plus 固有の機能。SQL コマンドまたは PL/SQL ブロックが正常に実行されるたびに、SQL\*Plus が自動的にデータベースへの変更をコミットできるようになる。SET コマンドの AUTOCOMMIT 変数を ON に設定すると、この機能が使用できるようになる。AUTOCOMMIT 変数を *n* に設定すると、INSERT、UPDATE または DELETE コマンド、あるいは PL/SQL ブロックが *n* 回正常に実行されるたびに、この機能が使用できるようになる。

### **出力 (output)**

レポートを実行した後の結果。出力は、画面に表示されたり、ファイルに格納されたり、用紙に印刷される。

## 出力ファイル (output file)

コンピュータからデータを転送する先のファイル。

## 小計 (subtotal)

レポートで、数値列内の値の合計。ブレイク・フィールド内の同じ値がある行のグループから取られる。「サマリー (summary)」を参照。

## 書式 (format)

情報を含む列の文字、数値、日付、ロングなどの型。ユーザーは、間合せてそれらの列から取得する情報の書式を設定できる。たとえば、日付型の情報は、23/11/98、Monday Twenty-third November 1998、またはその他の有効な日付書式で表示される。

## 書式モデル (format model)

レポート列の値の外観を制御する句の要素。COLUMN、TTITLE および BTITLE コマンドの FORMAT 句に事前定義済書式モデルを指定する。TO\_DATE などの SQL の日付変換関数の DATE 列に対して書式モデルを使用できる。

## スキーマ (schema)

データの論理構造 (スキーマ・オブジェクトの集合)。スキーマはデータベース・ユーザーによって所有され、そのユーザーと同じ名前を持つ。

## スクリプト (script)

対話形式で入力できる一連のコマンドを含むファイル。このファイルは保存されるため、利便性が良く再実行できる。スクリプトは、通常、オペレーティング・システムに固有の名前で呼ばれる。SQL\*Plus では、START、@ または @@ コマンドでスクリプトを実行できる。

## スプーリング (spooling)

出力をディスク記憶域に送信または保存する。通常は、ファイルの印刷または転送に使用される。SQL\*Plus の SPOOL コマンドは、スプーリングを制御する。

## セキュリティ・レベル (security level)

階層型の分類と情報の機密度を表す非階層型格納機構を組み合わせたもの。

## セッション (session)

ユーザー名が Oracle データベースに接続してから切断する前までの時間およびその時間内に発生したイベント。

## 接続 (connect)

データベースに接続するためにユーザーがユーザー名およびパスワードを入力して、Oracle に対して自分自身を識別させる。SQL\*Plus で CONNECT コマンドを使用すると、Oracle からログオフし、指定するユーザー名で再度ログインできる。

### **接続識別子 (connect identifier)**

Oracle Net がネットワーク上の特定の Oracle インスタンスに接続するために使用する一連のパラメータ。プロトコルも含まれる。

### **選択 (select)**

問合せ (SQL 文の SELECT) を使用して、複数のデータベース表から行をフェッチする。

### **タイトル (title)**

レポートの各ページの上端または下端に表示される複数の行。TTITLE および BTITLE コマンドを使用し、タイトルの設定および書式設定を行う。

### **タイマー (timer)**

TIMING コマンドが作成する内部記憶域。

### **置換変数 (substitution variable)**

SQL\*Plus で、前に 1 つまたは 2 つのアンパサンド (&) が付いた変数名または数字。置換変数をスクリプト内で使用して、スクリプトを実行するときに提供される値を表す。

### **注釈 (remark)**

SQL\*Plus で、REMARK コマンドを使用してスクリプトに挿入できるコメント。

### **データ型 (datatype)**

(1) データの標準型。Oracle のデータ型には、CHAR、NCHAR、VARCHAR2、NVARCHAR2、DATE、NUMBER、LONG、CLOB、NCLOB、RAW、LONG RAW および XMLType がある。ただし、Oracle データベース・サーバーは、他の標準データ型も認識し、変換する。(2) ある項目にプロパティとして対応付けることができる固定属性の名前付きのセット。データ型を指定すると、データの動作を定義できる。

### **データ制御言語 (Data Control Language: DCL)**

データおよびデータベースへの接続を制御する SQL 文のカテゴリ。GRANT および REVOKE 文がそれにあたる。DCL 文は、DML 文とともにグループ化される場合がある。

### **データ・セキュリティ (data security)**

オブジェクト・レベルでのデータベースの接続および使用を制御するメカニズム。たとえば、データ・セキュリティは、特定のスキーマ・オブジェクトへのアクセスおよびオブジェクト上でそれぞれのユーザーが許可されている特定のタイプのアクション (たとえば、ユーザー HR が SELECT および INSERT 文を発行できるが、EMP 表を使用して DELETE 文を発行することはできないなど) を含む。データ・セキュリティには、それぞれのスキーマ・オブジェクトに対して監査されるアクション (存在する場合) も含まれる。

### **データ操作言語 (Data Manipulation Language: DML)**

データベース・データへの問合せや更新を行う SQL 文のカテゴリ。一般的な DML 文として、SELECT、INSERT、UPDATE および DELETE がある。DCL 文は、DML 文とともにグループ化される場合がある。

### **データ定義言語 (Data Definition Language: DDL)**

表、ビューなどのデータベース・オブジェクトを定義または削除する SQL 文のカテゴリ。CREATE、ALTER および DROP 文がそれにあたる。

### **データ・ディクショナリ (data dictionary)**

Oracle データベース・サーバーによって自動的に作成および更新された表およびビューのセット。ユーザー、データ記憶域および権限に関する管理情報を含む。データ・ディクショナリは、Oracle を最初にインストールしたときにインストールされる。Oracle データベース・サーバー自体およびすべての Oracle ユーザーのための中心的な情報ソースになる。表は、Oracle によって自動的にメンテナンスされる。データ・ディクショナリは、カタログと呼ばれる場合がある。

### **データベース (database)**

1 つの単位として扱われるオペレーティング・システム・ファイルの集合。Oracle データベース・サーバーは、データベースに連データのディクショナリ表およびユーザー表を格納する。データベースには、データベース・ファイル、REDO ログ・ファイルおよび制御ファイルの 3 種類のファイルが必要である。

### **データベース・インスタンス障害 (database instance failure)**

Oracle データベース・インスタンス (SGA およびバックグラウンド・プロセス) の作業の継続を妨げる問題がある場合に発生する障害。インスタンス障害は、停電などのハードウェア問題、オペレーティング・システム・クラッシュなどのソフトウェア問題に起因する。インスタンス障害が発生した場合、SGA のバッファ内のデータはデータ・ファイルに書き込まれない。

### **データベース・オブジェクト (database object)**

データベース内に作成および格納される。データベース・オブジェクトとして、表、ビュー、シノニム、索引、順序、クラスタ、列などがある。

### **データベース管理者 (database administrator: DBA)**

(1) Oracle サーバー・データベースまたはデータベース・アプリケーションの運用およびメンテナンスの責任者。データベース管理者は、サーバーまたはデータベースをカスタマイズしてユーザーのローカル・コミュニティのニーズに応じるために使用状況を監視する。(2) DBA 権限を所有し、データベース管理操作を実行できる Oracle ユーザー名。通常、これら 2 つを同時に意味する。1 つのサイトに複数の DBA が存在する場合がある。

### **データベース・サーバー (database server)**

Oracle サーバーのカーネルを実行し、データベースを含める。

### **データベース指定 (database specification)**

データベースを識別する英数字コード。Oracle Net 操作でデータベースを指定したり、データベース・リンクを定義するために使用する。SQL\*Plus では、COPY、CONNECT または SQLPLUS コマンドにデータベース指定を設定できる。

### **データベース文字列 (database string)**

Oracle Net パラメータの文字列の 1 つ。ネットワーク接頭辞、接続するホスト・システム、およびホスト・システム上のデータベースのシステム ID を示すために使用される。

### **データベース・リンク (database link)**

ローカル・データベースに格納されているオブジェクト。リモート・データベース、リモート・データベースへの通信パス、および、場合によってはリモート・データベースに対するユーザー名およびパスワードを識別する。データベース・リンクを定義しておくと、そのリンクを使用し、リモート・データベースの表に問合せを実行できる。DB リンクとも呼ばれる。SQL\*Plus では、DESCRIBE コマンドまたは COPY コマンドにデータベース・リンクを設定できる。

### **ディスマウント・データベース (dismounted database)**

インスタンスによってマウントされていないデータベース。そのため、ディスマウント・データベースは、オープンできず、使用できない。

### **ディレクトリ (directory)**

一部のオペレーティング・システムで、ファイル・グループ用の名前付きの記憶域。実際のディレクトリは、特定のデバイス上にある一連のファイルをリストにした 1 つのファイルを示す。

### **テキスト・エディタ (text editor)**

ホスト・コンピュータのオペレーティング・システムで実行されるプログラムの 1 つ。ホスト・システム・ファイル、SQL コマンド、SQL\*Plus コマンドまたは PL/SQL ブロックを含む SQL\*Plus スクリプトの作成および編集に使用される。

### **デフォルト (default)**

ユーザーが指定しなかった場合に SQL\*Plus が使用する句またはオプションの値。

### **デフォルト・データベース (default database)**

「ローカル・データベース (local database)」を参照。

### **問合せ (query)**

任意の組合せ、式または順序でデータを取得する SQL SELECT 文。問合せは、データを変更しないで、データを取得するのみの読み専用操作。多くの場合、問合せは、DML 文とみなされる。

### **問合せ結果 (query results)**

問合せによって取得したデータ。

### **トランザクション (transaction)**

単一のユーザーが実行する複数の SQL 文を導出する論理作業単位。Oracle と互換性のある ANSI/ISO の SQL 規格によると、トランザクションはユーザーの最初の実行 SQL 文で開始される。ユーザーが明示的にコミットまたはロールバックすると、トランザクションは終了する。

### **ネットワーク (network)**

ハードウェアとソフトウェアによってリンクされ、データまたは周辺装置（あるいはその両方）を共有できる複数のコンピュータで構成されているグループ。

### **バイト (byte)**

8 つの連続したビットで構成されるグループ。英字、数値または記号（文字）を表す。コンピュータによってデータの単位として扱われる。

### **配列処理 (array processing)**

一度に 1 行ではなく複数のデータ行に対して実行される処理。SQL\*Plus、エクスポート / インポート、プリコンパイラなどの Oracle ユーティリティでは、ユーザーが配列のサイズを設定できる。配列のサイズを大きくすると、パフォーマンスが向上する場合がある。

### **バインド参照 (bind reference)**

PL/SQL 構造体または SQL SELECT 文内の単一のリテラル値（たとえば、文字列、数値、日付など）を変換するために使用されるパラメータへの参照。バインド参照では、パラメータ名の前にコロン (:) を付ける必要がある。

### **バインド変数 (bind variable)**

SQL 文の中の変数。SQL 文を正常に実行するには、バインド変数を有効な値または値のアドレスに置換する必要がある。

### **パスワード (password)**

ユーザー名に対応付けられた 2 次的な識別単語（または英数字の文字列）。データ・セキュリティに使用し、所有者のみが識別する。オペレーティング・システムまたはソフトウェア・アプリケーション（Oracle データベースなど）と接続する場合に、オペレーティング・システムのログイン ID、Oracle ユーザー名またはアカウント名と一緒にパスワードを入力する。ユーザー名または ID がパブリックであるのに対して、秘密パスワードはユーザー名の所有者のみがその名前を使用し、データにアクセスできることを保証する。

## バックグラウンド・プロセス (background process)

オペレーティング・システム環境で起動し、サービスまたはアクションを実行する非対話型処理。Oracle データベース・サーバー製品によっては、バックグラウンド・プロセスを異なるタスクに使用する場合があります。たとえば、データベースの同時ユーザーにかわって行うタスクの実行および調整、電子メッセージの処理および配信、印刷サービスなどがそれにあたります。

## パッケージ (packages)

データベース内の 1 つの単位として、関連するプロシージャ、ファンクションおよび他のパッケージ構造体をまとめてカプセル化または格納する方法。パッケージにより、データベース管理者やアプリケーション開発者の操作が効率的になるだけでなく、データベースの機能性やパフォーマンスも向上する。

## バッファ (buffer)

ユーザーの SQL 文または PL/SQL ブロックが一時的に格納される領域。SQL バッファが、デフォルト・バッファである。複数のバッファからコマンドを編集および実行できる。ただし、SQL\*Plus では、複数のバッファを使用する必要はない。

## 幅 (width)

列、パラメータまたはレイアウトのオブジェクトの幅。幅は文字で測定される。空白も 1 文字として扱われる。

## パラメータ (parameter)

アンパサンドの後に 1 つの数字を続けて構成した置換変数 (&1、&2 など)。スクリプト内でパラメータを使用する。START コマンドの引数を介してパラメータに値を渡す。

## パラメータ・ファイル (parameter file)

Oracle9i サーバーが使用するファイル。データベースの起動時に使用する特定の値および構成の設定を指定する。パラメータ・ファイルの機能の詳細は、『Oracle9i データベース管理者ガイド』を参照。

## 引数 (argument)

START コマンド内でスクリプト名に続くデータ項目。引数は、スクリプト内のパラメータに値を提供する。

## ビット (bit)

データの最小単位。ビットに使用される値は、0 と 1 のみである。ビットが 8 つ結合されると、バイトというグループにまとめられる。バイトはそれぞれ 1 つの文字を表す。「[バイト \(byte\)](#)」を参照。

## ビュー (view)

ビューは、1つまたは多くの表からデータを提示する「ストアド・クエリー」とみなすこともできる。ビューは、実際にはデータを含まない、またはデータを格納しないが、基礎となる表からデータを導出する。ビューに対しても、問合せ、更新、挿入および削除を実行できる。ビューでの操作は、そのビューの実表に影響を与える。

## 表 (table)

RDBMS (リレーショナル・データベース管理システム) の記憶域の基本単位。表はエン트리および関係を表し、同じ種類の値 (列) を含む複数の情報単位 (行) で構成される。各列には、列名、データ型 (CHAR、NCHAR、VARCHAR2、NVARCHAR2、DATE、NUMBER など) および幅 (DATE などのデータ型によって事前に定義されている場合がある) が備えられている。表を作成した後、有効なデータ行を挿入できる。その後、表の情報に対し、問合せ、削除または更新を実行できる。表のデータに対して定義されたビジネス・ルールを規定するために、整合性制約とトリガーを表に対して定義できる。

## 表示書式 (display format)

「書式 (format)」を参照。

## 表示幅 (display width)

出力フィールドの値を表示するために許容される文字または空白の数。

## 表の別名 (table alias)

表の一時的な名前。問合せで定義され、その問合せの間にも有効になる。使用する場合は、SELECT 文の FROM 句で別名を設定する。SELECT 構文のリスト内に表示される。「別名 (alias)」を参照。

## ファイル (file)

1つの単位として扱われるデータの集合。リスト、ドキュメント、索引、注、一連のプロシージャなどがそれにあたる。通常、磁気テープまたはディスクに格納されているデータを参照するために使用される。「ファイル名 (filename)」、「拡張子 (extension)」および「ファイル・タイプ (file type)」も参照。

## ファイル・タイプ (file type)

オペレーティング・システムによっては、通常、ファイルの使用または用途を表すファイル名の一部。「拡張子 (extension)」を参照。

## ファイル名 (filename)

ファイル指定の名前コンポーネント。ファイル名は、ファイル自体が作成されるときにユーザーまたはシステムのいずれかによって割り当てられる。「拡張子 (extension)」および「ファイル・タイプ (file type)」も参照。

### **ファンクション (function)**

操作を実行し、操作の完了時に値を戻す PL/SQL サブプログラム。ファンクションには、組み込み済のものとユーザーが名前を指定するものがある。「**プロシージャ (procedure)**」と対比。

### **ブレイク (break)**

SQL\*Plus が問合せまたはレポートを処理しているときに発生するイベント（式の値の変更など）。指定したブレイクが発生した場合、SQL\*Plus に指示して様々な操作（小計の出力など）を実行できる。

### **ブレイク階層 (break hierarchy)**

SQL\*Plus がブレイクの発生の有無を検査し、対応するブレイク操作をトリガーする順序。

### **ブレイク・グループ (break group)**

複数のブレイク列を含むグループ。

### **ブレイク順序 (break order)**

ブレイク列のデータを表示する順序。有効なオプションは、昇順および降順。

### **ブレイク列 (break column)**

ユーザーがブレイク操作を定義したレポート内の列。その列の値を変更すると、ブレイクが発生する。

### **ブレイク・レポート (break report)**

ブレイク列の共通値に基づいて、表の行をセットに分割したレポート。

### **プロシージャ (procedure)**

特定の作業を実行するために 1 つの実行可能単位としてまとめてグループ化した一連の SQL および PL/SQL 文。プロシージャおよびファンクションは、ほとんど同じである。相違点は、ファンクションは常に単一の値をコール元に戻すのに対し、プロシージャはコール元に値を戻さないこと。

### **プロセス (process)**

オペレーティング・システム内の制御のスレッド。一連の手順を実行できるオペレーティング・システム内のメカニズム。オペレーティング・システムによっては、ジョブまたはタスクという言葉を使用する場合もある。プロセスには、通常、実行するプライベート・メモリー領域がある。

### **ブロック (block)**

PL/SQL で、手続き型ロジックで関連付けられている SQL および PL/SQL コマンドのグループ。

### **プロンプト (prompt)**

(1) データの入力または他のアクションを実行するように指示するコンピュータ・プログラムからのメッセージ。(2) ユーザーの応答を支援するキューとしてシステムが使用するワード。このようなメッセージは、通常、ユーザーにフィールドに情報を応答として入力するように求める。「[コマンドライン \(command-line\)](#)」を参照。

### **文 (statement) (SQL)**

1 つの SQL 文。句とは異なり、完全な文に類似している。SQL 文またはコマンドのそれぞれの部分は、式、述語または句と呼ばれる。「[SQL 文 \(SQL statement\)](#)」を参照。

### **ページ (page)**

表示された 1 画面分のデータ、またはレポートに印刷された用紙 1 枚分のデータ。

### **ヘッダー (heading)**

SQL\*Plus で、列の上に表示される出力列の名前になるテキスト。「[列ヘッダー \(column heading\)](#)」を参照。

### **別名 (alias)**

SQL では、表、列または SQL 文内の値に割り当てられた一時的な名前。同じ文または対応する SQL\*Plus コマンドの中で、その項目を参照するために使用される。

### **変数 (variable)**

単一の値を保持する名前付きのオブジェクト。SQL\*Plus は、バインド変数、置換変数、システム変数およびユーザー変数を使用する。

### **ホスト・コンピュータ (host computer)**

SQL\*Plus の実行元となるコンピュータ。

### **本体 (body)**

レポートの大部分 (テキスト、図形、データおよび計算) が含まれているレポート領域。

### **マウントされたデータベース (mounted database)**

Oracle インスタンスに関連付けられたデータベース。データベースは、オープンまたはクローズされる。ユーザーが接続するには、データベースをマウントおよびオープンする必要がある。マウントされていてもオープンされないデータベースは、メンテナンスするために DBA によって接続される場合がある。

### **マルチスレッド・サーバー (multi-threaded server)**

マルチスレッド・サーバーを使用すると、サーバー・プロセスの数を最小限に抑え、使用可能なシステム・リソースの使用効率を最大化して、多数のユーザー・プロセスが少数のサーバー・プロセスを共有できる。

### 無名ブロック (anonymous block)

名前がない PL/SQL プログラム・ユニット。実行文を囲む BEGIN キーワードおよび END キーワードを明示的に指定する必要はない。

### 文字 (character)

コンピュータ・システム上で 1 つのアルファベット文字または数字を保持できる 1 つの位置。複数の文字がフィールド内に保持される。複数のフィールドが 1 つのレコードを形成し、複数のレコードはファイルまたはデータベース表内に保持される。

### 文字間調整 (justification)

「[位置合せ \(alignment\)](#)」を参照。

### 文字列 (character string)

連続した文字、数値または記号で構成されるグループ。通常は、単語、名前またはその一部で構成される。

### 文字列 (string)

行の中の単語または文字の並び。

### ユーザー変数 (user variable)

明示的に DEFINE コマンドを使用して、または暗黙的に START コマンドへの引数を使用して、ユーザーが定義および設定した変数。

### ユーザー名 (username)

Oracle データベース・サーバーおよび他のユーザーがユーザーを識別するための名前。すべてのユーザー名は、プライベート・パスワードに対応付けられる。Oracle データベースに接続する場合は、ユーザー名およびプライベート・パスワードを入力する必要がある。「[アカウント \(account\)](#)」を参照。

### ユリウス日付 (Julian date)

SQL ファンクション JDATE を使用して整数形式で日付を表すアルゴリズム。ユリウス日付を使用すると、日付に対して他の算術関数を実行できる。

### 予約語 (reserved word)

(1) 特定のソフトウェアまたはオペレーティング・システムで特殊な意味を持つ用語。(2) SQL では、SQL 文で使用するために予約される一連のワード。データベース・オブジェクトの名前として予約語は使用できない。

### ラベル (label)

COMPUTE コマンド内で計算された値に対して出力するラベルを定義する。COMPUTE ラベルの最大長は、500 文字。

### **リカバリ (recover)**

指定した REDO ログ・ファイルからデータベースの全部または一部をリストアする Oracle プロセス。

### **リモート・コンピュータ (remote computer)**

ローカル・コンピュータ以外のネットワーク上にあるコンピュータ。

### **リモート・データベース (remote database)**

デフォルト・データベース以外のデータベース。リモート・コンピュータ上に常駐し、CONNECT、COPY および SQLPLUS コマンドで指定できる。

### **リレーショナル・データベース管理システム (Relational Database Management System: RDBMS)**

Oracle7 以下の用語。共有データの格納と検索用に設計されたコンピュータ・プログラム。リレーショナル・システムでは、データは、それぞれ同じ列のセットを持つ 1 つ以上の行で構成された表に格納される。Oracle は、リレーショナル・データベース管理システムである。その他のタイプのデータベース・システムは、階層型またはネットワーク型データベース・システムに分類される。

### **レコード (record)**

複数の列に対する値を持つデータベース表内の 1 行になっているデータ。

### **列 (column)**

特定のデータ・ドメインを表すデータベース表内の垂直方向の領域。列には、列名および特定のデータ型がある。たとえば、従業員情報の表では、すべての従業員の採用年月日は 1 つの列で構成される。レコード・グループ列は、データベース列を表す。

### **列の式 (column expression)**

取得するデータベース列を定義した SELECT 文中の式。この式は、列名、または列名を参照する有効な SQL 式の場合がある。

### **列ヘッダー (column heading)**

レポート内に表示されるそれぞれの列に対して作成されるヘッダー。

### **レポート (report)**

(1) 問合せの結果。(2) 出力。特に、すぐに読める形式で設定されている出力。狭義では、SQL\*Plus からの出力。

### **ローカル・データベース (local database)**

SQL\*Plus の起動時に SQL\*Plus を接続するデータベース。通常は、ホスト・コンピュータ上のデータベース。デフォルト・データベースとも呼ばれる。「[リモート・データベース \(remote database\)](#)」を参照。

**ロール (roles)**

ユーザーまたは他のロールに付与される関連する権限の名前付きのグループ。

**ロールバック (rollback)**

SQL ROLLBACK コマンドを使用して、現行のトランザクションでデータへ加えた保留中の変更を破棄する。セーブポイントを指定して、トランザクションの一部をロールバックできる。

**ログイン (ログオン) (log in (log on))**

オペレーティング・システムとの通信を確立し、ユーザーの端末セッションのデフォルトの特性を設定するための一連のアクションを端末で実行すること。

**ログオフ (ログアウト) (log off (log out))**

オペレーティング・システムとの対話型通信および端末セッションを終了すること。

**ログオン文字列 (logon string)**

ローカル・データベースまたはリモート・データベースに接続するアプリケーションを実行するために使用するユーザー指定のコマンドライン。ログオン文字列は明示的に接続識別子を含めるか、暗黙的にデフォルトの接続識別子を使用する。

**ログ・ライター (Log Writer: LGWR)**

Oracle インスタンスが使用するバックグラウンド・プロセス。LGWR は、ディスクに REDO ログ・エントリを書き込む。REDO ログ・データは、システム・グローバル領域の REDO ログ・バッファ内に生成される。トランザクションがコミットし、ログ・バッファが一杯になると、LGWR はオンライン REDO ログ・ファイルに REDO ログ・エントリを書き込む。

## 記号

# シャープ記号, 13-33  
\$ 数値書式, 7-5  
&&, 6-20  
& (アンパサンド)  
置換変数, 6-17  
置換変数の無効化, 6-22  
\* (アスタリスク)  
LIST コマンド, 13-71  
DEL コマンド, 6-4, 13-50  
LIST コマンド, 6-4  
-- (コメント・デリミタ), 6-10  
~ 負の無限大記号, 13-33  
-idle-timeout パラメータ, 9-13  
-MARKUP, 4-4, 8-2  
SPOOL 句, 4-5  
SQLPLUS コマンド句, 4-5  
-SILENT オプション, 4-10, 8-11  
- (ハイフン)  
句, 4-4  
長い SQL\*Plus コマンドの継続, 5-9, 13-1  
. (ピリオド), 5-7  
/\*...\*/ (コメント・デリミタ), 6-10  
/NOLOG オプション, 4-11  
/ (スラッシュ) コマンド  
RUN との類似, 13-89  
デフォルト・ログオン, 13-45  
RUN との類似, 5-7, 13-9  
カレント PL/SQL ブロックの実行, 5-7  
現行の SQL コマンドの実行, 5-7  
コマンド・プロンプトでの入力, 5-7, 13-9  
使用方法, 13-9  
デフォルト・ログオン, 4-11  
バッファ行番号プロンプトでの入力, 5-5, 13-9

:BindVariable 句  
EXIT コマンド, 13-63  
: (コロン)  
バインド変数, 6-29  
;(セミコロン), 5-4  
@@ (二重アットマーク) コマンド, 3-4, 13-7  
START との類似, 13-8, 13-130  
スクリプト, 13-7  
@ (アットマーク)  
CONNECT コマンド, 13-44  
COPY コマンド, B-2, B-5  
SQLPLUS コマンド, 4-3, 4-21, 6-14  
START との類似, 6-14, 13-6, 13-130  
コマンド, 3-4, 6-13, 13-5  
コマンド指数, 13-5  
スクリプト, 6-14, 13-5  
スクリプトへのパラメータの受渡し, 13-5  
[Cancel] キー, 5-12  
\_EDITOR、EDIT コマンド, 6-2, 13-59  
~ 無限大記号, 13-33

## 数字

0、数値書式, 7-5  
9、数値書式, 7-5

## A

ACCEPT コマンド, 6-26, 13-10  
CHAR 句, 13-10  
DATE 句, 13-10  
DEFAULT 句, 13-11  
DEFINE コマンド, 13-48  
FORMAT 句, 13-10  
HIDE 句, 13-11

- NOPROMPT 句, 13-11
- NUMBER 句, 6-28
- PROMPT 句, 6-27, 13-11
  - 値の入力を求めるプロンプトのカスタマイズ, 6-28
- ALIAS 句, 13-30
  - ATTRIBUTE コマンド, 13-16
- ALL 句, 13-120
- ALTER コマンド
  - 使用禁止, 10-5
- ANALYZE コマンド
  - 使用禁止, 10-5
- APPEND 句
  - COPY コマンド, B-3, B-7
  - SAVE コマンド, 13-90
- APPEND コマンド, 6-4, 6-8, 13-12
- APPINFO 句, 9-10, 13-96
- ARCHIVE LOG
  - コマンド, 11-4, 13-13
  - モード, 11-4
- ARCH バックグラウンド・プロセス, 13-14
- ARRAYSIZE 変数, 9-10, 13-93, 13-97
  - COPY コマンドとの関連, B-4, B-8
- ATTRIBUTE コマンド, 13-16
  - ALIAS 句, 13-16
  - CLEAR COLUMN コマンド, 13-17
  - CLEAR 句, 13-16
  - FORMAT 句, 13-17
  - LIKE 句, 13-17
  - OFF 句, 13-17
  - ON 句, 13-17
    - 属性の表示特性の表示, 13-16
    - 表示特性, 13-16
    - 表示特性の制御, 13-17
    - 複数入力, 13-17
    - 列の消去, 13-27, 13-30
    - 列の表示属性の非表示, 13-17
    - 列の表示属性の表示, 13-17
- AUDIT コマンド
  - 使用禁止, 10-5
- AUTOCOMMIT 変数, 5-11, 13-93, 13-98
- AUTOMATIC 句, 13-78
- AUTOPRINT 変数, 13-93, 13-98
- AUTORECOVERY 変数, 13-93, 13-98
- AUTOTRACE 変数, 9-2, 13-93, 13-99
- AVG 関数, 7-16

## B

---

- BEGIN コマンド, 5-7
  - 使用禁止, 10-5
- BLOCKTERMINATOR, 13-93, 13-99, 13-114, 13-117
- BODY オプション, 4-6
- BODY 句, 4-6
- BOLD 句, 13-87, 13-139
- BREAKS 句, 7-15, 13-27
- BREAK コマンド, 7-11, 13-18
  - BREAKS の消去, 7-15
  - COMPUTE との併用, 7-15, 13-18, 13-20, 13-41
  - DUPLICATES 句, 13-21
  - ON column 句, 7-12, 13-18
  - ON expr 句, 13-19
  - ON REPORT 句, 7-19, 13-20
  - ON ROW 句, 7-13, 13-20
  - REFCURSOR 変数の書式設定に使用, 13-144
  - SET COLSEP との併用, 13-101
  - SKIP PAGE 句, 7-13, 13-21
  - SKIP 句, 7-13, 13-20
  - SQL の ORDER BY 句, 7-11, 7-12, 7-14, 13-19
    - 値の重複の抑止, 7-12
    - 各行の後への空白の挿入, 7-13
    - 現行のブレイク定義の表示, 7-15, 13-21
    - 総サマリーおよびサブサマリーの出力, 7-19
    - タイトルの変数に現在の日付を格納, 7-29
    - タイトルへの列値の表示, 7-28
    - 定義の取消し, 13-27
    - 複数のブレイク列の指定, 7-14, 13-18
    - ブレイク列が変更されたときの空白の挿入, 7-13
    - レポートの終わりでのサマリー行の出力, 7-19
- 「Browse」ボタン, 2-6, 2-15
- BTITLE 句, 13-121
- BTITLE コマンド, 7-22, 13-23
  - BOLD 句, 13-139
  - CENTER 句, 13-139
  - COL 句, 13-139
  - FORMAT 句, 13-139
  - LEFT 句, 13-139
  - OFF 句, 13-138
  - RIGHT 句, 13-139
  - SKIP 句, 13-139
  - TAB 句, 13-139
  - TTITLE コマンド, 13-23
    - 下部タイトルの前に空白行を出力, 7-25
  - 旧形式, C-3

現行の定義の非表示, 13-138  
タイトルのインデント, 13-139  
タイトル要素の整列, 13-139  
列値変数の参照, 13-35  
BUFFER 句, 6-3, 6-4, 13-27  
BUFFER 変数, C-4

## C

---

CANCEL 句, 13-79, 13-81  
「Cancel」ボタン, 2-8, 4-19, 6-24  
CENTER 句, 7-24, 13-87, 13-139  
「Change Password」画面, 4-18  
CHANGE コマンド, 6-4, 6-5, 13-24  
CHAR 句, 13-10  
    VARIABLE コマンド, 13-142  
CHAR 列  
    DESCRIBE による定義の表示, 13-52  
    書式の変更, 7-6, 13-31  
    デフォルトの書式, 7-6, 13-31  
「Clear Screen」ボタン, 2-6  
CLEAR 句, 7-9, 13-30  
    ATTRIBUTE コマンド, 13-16  
CLEAR コマンド, 13-27  
    BREAKS 句, 7-15, 13-27  
    BUFFER 句, 6-3, 6-4, 13-27  
    COLUMNS 句, 13-27  
    COMPUTES 句, 13-27  
    SCREEN 句, 6-29, 13-28  
    SQL 句, 13-28  
    TIMING 句, 13-28  
CLOB 句  
    VARIABLE コマンド, 13-143  
CLOB 列  
    最大幅の設定, 13-94, 13-108  
    取得位置の設定, 13-94, 13-107  
    取得サイズの設定, 13-94, 13-108  
    書式の変更, 7-6, 13-31  
    デフォルトの書式, 13-31  
CLOSECURSOR 変数, C-2, C-5  
CMDSEP 変数, 13-93, 13-100  
COLSEP 変数, 13-93, 13-101  
COLUMNS 句, 13-27  
COLUMN コマンド, 7-2, 13-29  
    ALIAS 句, 13-30  
    BREAK コマンド, 13-20  
    CLEAR 句, 7-9, 13-30  
    DEFAULT 句, C-3  
    DEFINE コマンド, 13-48  
    ENTMAP 句, 13-30  
    FOLD\_AFTER 句, 13-30, 13-31  
    FOLD\_BEFORE 句, 13-31  
    FORMAT 句, 7-4, 7-6, 13-31  
    HEADING 句, 7-2, 13-33  
    HEADSEP 文字, 13-33  
    JUSTIFY 句, 13-34  
    LIKE 句, 7-8, 13-34  
    NEW\_VALUE 句, 7-28, 7-29, 13-34  
    NEWLINE 句, 13-34  
    NOPRINT 句, 7-28, 13-34  
    NULL 句, 13-34  
    NUMBER 列の書式設定, 7-4, 13-32  
    OFF 句, 7-9, 13-35  
    OLD\_VALUE 句, 7-29, 13-35  
    ON 句, 7-9, 13-35  
    PRINT 句, 13-34  
    REFCURSOR 変数の書式設定, 13-144  
    TRUNCATED 句, 7-7, 13-35  
    WORD\_WRAPPED 句, 7-7, 7-10, 13-35  
    WRAPPED 句, 7-7, 13-35  
    下部タイトルへの列値の表示, 7-29, 13-35  
    上部タイトルへの列値の表示, 7-28, 13-34  
    タイトルの変数に現在の日付を格納, 7-29, 13-37  
    表示をデフォルトにリセット, 7-9, 13-30, C-2  
    複数入力, 13-35  
    列の書式設定, 7-6  
    列の表示属性の非表示, 7-9, 13-35  
    列の表示属性の表示, 7-9, 13-30, 13-35  
COL 句, 7-25, 13-87, 13-139  
COMMIT 句, 13-62  
    WHENEVER OSERROR, 13-148  
    WHENEVER SQLERROR, 13-150  
COMMIT コマンド, 5-11  
COMPATIBILITY 変数, 13-93, 13-101  
COMPUTES 句, 13-27  
COMPUTE コマンド, 7-11, 13-39  
    AVG 関数, 7-16  
    COUNT 関数, 7-16  
    LABEL 句, 7-16, 7-19, 13-40  
    MAXIMUM 関数, 7-16  
    MINIMUM 関数, 7-16  
    NUMBER 関数, 7-16  
    OF 句, 7-15  
    OF での SELECT 式の参照, 13-40

ON column 句, 7-15, 13-40  
ON expr 句, 13-40  
ON REPORT 句, 7-19, 13-40  
ON ROW 句, 13-40  
ON での SELECT 式の参照, 13-40  
REFCURSOR 変数の書式設定に使用, 13-144  
STD 関数, 7-16  
SUM 関数, 7-16  
VARIANCE 関数, 7-16  
同じ列についての複数のサマリーの出力, 7-20  
異なる列についてのサマリーの計算, 7-20  
最大 LABEL 長, 13-40  
すべての定義の表示, 7-21, 13-40  
総サマリーおよびサブサマリーの出力, 7-19  
定義の取消し, 7-21, 13-27  
ブレークでのサマリー行の出力, 7-15  
レポートの終わりでのサマリー行の出力, 7-19  
CONCAT 変数, 6-22, 13-93, 13-102  
Connection Identifier  
フィールド, 2-5, 2-11  
CONNECT コマンド, 4-17, 4-18, 4-21, 13-44  
@ (アットマーク), 13-44  
SYSDBA 句, 4-11, 13-45  
SYSOPER 句, 4-11, 13-45  
サービス名, 4-22  
パスワードの変更, 13-44, 13-45, 13-73  
ユーザー名 / パスワード, 13-44, 4-21  
CONNECT コマンド (SQL)  
使用禁止, 10-5  
CONTINUE 句  
WHENEVER OSERROR, 13-148  
WHENEVER SQLERROR, 13-150  
Cookie、iSQL\*Plus の構成, 3-5  
COPYCOMMIT 変数, 13-93, 13-102  
COPY コマンドとの関連, B-4, B-8  
COPYTYPECHECK 変数, 13-93, 13-102  
COPY コマンド, 13-47, B-2, B-5  
@ (アットマーク), B-2, B-5  
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-9  
APPEND 句, B-3, B-7  
ARRAYSIZE 変数, B-4, B-8  
COPYCOMMIT 変数, B-4, B-8  
CREATE 句, B-3, B-7  
FROM 句, B-5  
INSERT 句, B-3, B-7  
LONG 変数, B-4, B-8

REPLACE 句, B-3, B-6  
SELECT によるソース表の命名, B-3, B-6  
TO 句, B-5  
USING 句, B-3, B-6  
アクションの決定, B-6  
宛先の表, B-3, B-6  
宛先の列の指定, B-3, B-6  
エラー・メッセージ, 14-43  
コピーするデータの指定, B-3, B-6  
サービス名, B-5, B-7, B-9  
サンプル・コマンド, B-5, B-6  
使用禁止, 10-5  
ソースの行および列の決定, B-3, B-6  
データベース間のデータのコピー, B-5  
問合せ, B-3, B-6  
必須の接続識別子, B-3  
表へのデータの挿入, B-3, B-7  
表内のデータの置換, B-3, B-6  
表の作成, B-3, B-6, B-7  
別のユーザーの表への参照, B-8  
メッセージの解釈, B-8  
ユーザー名 / パスワード, B-2, B-5, B-7, B-9  
COUNT 関数, 7-16  
CREATE 句  
COPY コマンド, B-3, B-7  
CREATE コマンド  
PL/SQL の入力, 5-6  
使用禁止, 10-5  
CSS ユーザー・インタフェース, 3-12

## D

---

DATABASE 句, 13-79  
DATAFILE 句, 13-80  
DATE  
DESCRIBE による列定義の表示, 13-52  
DATE 句, 13-10  
DATE 列  
書式の変更, 7-6, 13-31, 13-38  
デフォルトの書式, 7-6  
DB2, 13-102  
DBA, 11-2  
iSQL\*Plus へのアクセス, 10-9  
権限, 2-10, 13-131  
モード, 13-131  
DBA の「Login」画面, 2-10  
DBA アクセス, 10-11

DBA の「Work Screen」画面, 2-12  
DBMS\_APPLICATION\_INFO パッケージ, 9-10,  
13-93, 13-96  
DECLARE コマンド、使用禁止, 10-5  
DECLARE コマンド (PL/SQL), 5-7  
DEFAULT 句, 13-11  
DEFINE コマンド, 6-16, 13-48  
    CHAR 値, 13-48  
    SET DEFINE ON|OFF, 13-93, 13-102  
    UNDEFINE コマンド, 6-17, 13-141  
    置換変数, 6-20, 13-48  
    ホスト・システム・エディタ, 6-2, 13-48  
DEFINE 変数, 6-22, 13-93, 13-102  
DELETE コマンド  
    使用禁止, 10-5  
DEL コマンド, 6-4, 6-9, 13-50  
    アスタリスクの使用, 6-4, 13-50  
DESCRIBE コマンド (SQL\*Plus), 5-13, 13-52  
    connect\_identifier, 13-52  
    表示できる PL/SQL プロパティ, 13-53  
    表示できる表プロパティ, 13-52  
DISABLED キーワード、使用禁止コマンド, 10-4  
DISCONNECT コマンド, 4-17, 4-18, 13-58  
DOCUMENT コマンド, C-2, C-3  
    新しいバージョンの REMARK, C-4  
DOCUMENT 変数, C-2, C-5  
DROP コマンド  
    使用禁止, 10-5  
DUPLICATES 句, 13-21

## E

---

ECHO 変数, 6-13, 13-94, 13-104  
EDITFILE 変数, 13-94, 13-104  
EDIT コマンド, 6-2, 13-59  
    \_EDITOR の定義, 13-59  
    使用禁止, 10-5  
    スクリプトの作成, 6-2  
    スクリプトの変更, 6-3, 13-59  
    デフォルトのファイル名の設定, 13-94, 13-104  
EMBEDDED 変数, 13-94, 13-104  
EMP 表, 1-7  
「Enter statements」フィールド, 2-6, 2-7  
ENTMAP, 4-6  
ENTMAP 句, 4-6, 8-12, 13-30  
ESCAPE 変数, 6-22, 13-94, 13-104  
EXECUTE コマンド, 13-61

    使用禁止, 10-5  
「Execute」ボタン, 2-6  
EXIT 句  
    WHENEVER OSERROR, 13-148  
    WHENEVER SQLERROR, 13-150  
EXIT コマンド, 4-13, 13-62  
    :BindVariable 句, 13-63  
    COMMIT 句, 13-62  
    FAILURE 句, 13-62  
    ROLLBACK 句, 13-63  
    SET MARKUP とともに使用, 8-3  
    WARNING 句, 13-62  
    使用禁止, 10-5  
    スクリプトで使用, 13-130  
「Expired Password」画面, 4-19  
Extension for Windows, 2-15

## F

---

FAILURE 句, 13-62  
FEEDBACK 変数, 13-94, 13-105  
FLAGGER 変数, 13-94, 13-105  
FLUSH 変数, 13-94, 13-105  
FOLD\_AFTER 句, 13-31  
FOLD\_BEFORE 句, 13-31  
FORCE 句, 13-131  
FORMAT 句, 13-10, 13-31  
    ATTRIBUTE コマンド, 13-17  
    COLUMN コマンド, 7-4, 7-6  
    REPHEADER および REPFOOTER コマンド, 13-87  
    TTITLE および BTITLE コマンド, 7-27, 13-139  
FROM 句, 13-78  
FROM 句 (SQL\*Plus), B-5

## G

---

GET コマンド, 13-64  
    LIST 句, 13-64  
    NOLIST 句, 13-64  
    使用禁止, 10-5  
    スクリプトの取得, 13-64  
GLOGIN.SQL, 3-2, 9-4, 9-8  
    「LOGIN.SQL」を参照  
glogin.sql, 10-13  
GRANT コマンド, 10-2  
    使用禁止, 10-5

## H

---

HEADING 句, 7-2, 13-33

HEADING 変数, 13-106

HEADSEP 変数, 13-94, 13-106

COLUMN コマンドでの使用, 7-3, 13-33

HEAD オプション, 4-5

HEAD 句, 4-5

HELP コマンド, 13-66

HIDE 句, 13-11

「History」画面, 2-6

HOST コマンド, 5-12, 13-67

使用禁止, 10-5

HTML, 8-2

エンティティ, 8-12

オプション, 4-5

句, 4-5

タグ (tag), 8-2

動的レポートの実行, 8-12

ファイルへのスプール, 4-7

htpasswd

ユーザー名 / パスワード認証ファイル・エントリの  
追加, 10-12

HTTP

HTTPS セキュリティ, 10-9

セキュリティ, 10-9

HTTP Server

DBA の有効化, 10-11

グローバルゼーション・サポートの構成, 12-4

構成, 3-12

構成ファイルのテスト, 3-13

認証, 2-10, 10-10

認証エントリの追加, 10-12

認証ユーザー名 / パスワード・エントリ, 3-12

HTTP Server 認証の有効化, 3-12, 10-10

httpd.conf, 3-13

httpd.conf の編集内容のテスト, 3-16

httpd.conf ファイルの解析, 3-16

## I

---

IMMEDIATE モード, 13-125

INIT.ORA ファイル

パラメータ・ファイル, 13-131

INPUT コマンド, 6-4, 6-7, 13-69

複数行の入力, 13-69

INSERT 句, B-3, B-7

INSERT コマンド

使用禁止, 10-5

INSTANCE 変数, 13-94, 13-106

「Interface Options」画面, 2-9

iplus.css, 3-12

iplus.pw, 10-10

iplusdba.pw, 10-11

iSQL\*Plus

「Change Password」画面, 4-18

Cookie の有効化, 3-5

DBA の「Login」画面, 2-10

DBA の「Work Screen」画面, 2-12

「Expired Password」画面, 4-19

Extension for Windows, 2-15

Extension for Windows の構成, 3-16

「History」画面, 2-6

HTTP Server 認証の有効化, 3-12

HTTP Server の構成, 3-12

「Interface Options」画面, 2-9

iSQLPlusHashTableSize パラメータ, 9-13

iSQLPlusNumberOfThreads パラメータ, 9-13

Java スクリプトの有効化, 3-5

「Login」画面, 2-4, 4-13

Oracle Net の構成, 3-16

「Preferences」画面, 2-8

「System Variables」画面, 2-14

「Work Screen」画面, 2-5

アクセス・モード, 10-9

オンライン・ヘルプ, 4-14

概要, 2-3

カスケード・スタイルシート, 3-12

起動, 4-13, 4-15

グローバルゼーション・サポート, 12-1

グローバルゼーション・サポートの構成, 12-3

言語サポート, 4-14

構成, 3-5

コマンドの入力, 2-6

サーバーの構成, 3-7

サーバーの統計, 2-12

サーバー・パラメータ, 2-14

3 層モデル, 1-4

システム変数の設定, 2-14

終了, 4-14

セキュリティ, 10-9

セッション, 1-5

未サポートのコマンド, D-1

有効化または無効化, 3-15

- ユーザー・アクセス, 10-9
- ユーザー・インタフェース, 1-4, 1-5, 2-3
- iSQL\*Plus, 2-3
- iSQL\*Plus サーバー
  - DBA アクセスの有効化, 10-11
  - httpd.conf の構成, 3-13
  - oracle\_apache.conf の構成, 3-13
  - アクティブ統計, 9-11
  - データベース・アクセス制限の有効化, 10-10
  - 認証, 10-10
- iSQL\*Plus での言語サポート, 4-14
- iSQL\*Plus のエラー・メッセージ, 14-33
- iSQL\*Plus の有効化, 3-15
- isqlplus.conf, 3-8, 3-9, 3-10, 3-11, 3-13, 10-11
- iSQLPlusConnectIdList, 10-10
- iSQLPlusHashTableSize パラメータ, 9-13
- iSQLPlusNumberOfThreads パラメータ, 9-13
- iSQLPlusTimeOutInterval パラメータ, 9-13

## J

---

- Java スクリプト、iSQL\*Plus の構成, 3-5
- JUSTIFY 句, 13-34

## L

---

- LABEL 変数
  - SHOW コマンド, C-2, C-7
- LEFT 句, 7-24, 13-87, 13-139
- LIKE 句, 7-8, 13-17, 13-34
- LINESIZE 変数, 7-24, 7-31, 13-94, 13-107
- LIST 句, 13-13, 13-64
- LIST コマンド, 6-4, 13-71
  - アスタリスクの使用, 6-4, 13-71
  - カレント行の判断, 6-5, 13-71
  - 最終行をカレント行にする, 6-5, 13-71
- LNO 句, 13-121
- 「Load Script」ボタン, 2-6
- LOBOFFSET 変数, 13-94, 13-107
- LOCK TABLE コマンド
  - 使用禁止, 10-5
- LOG\_ARCHIVE\_DEST パラメータ, 13-13
- LOGFILE 句, 13-78
- LOGIN.SQL, 3-2
  - 「GLOGIN.SQL」を参照
  - SET コマンドの挿入, 3-3
  - 挿入するサンプル・コマンド, 3-3

- タイトルの変数に現在の日付を格納, 7-29
- login.sql
  - 「glogin.sql」を参照
- LOGIN.SQL 内の TIME 句, 3-3
- 「Login」画面, 2-4, 2-10, 4-13, 10-10
- 「Login」ボタン, 2-5, 2-12
- LONG
  - DESCRIBE による列定義の表示, 13-52
- LONGCHUNKSIZE 変数, 7-6, 13-31, 13-94, 13-108, 13-109
- LONGRAW
  - DESCRIBE による列定義の表示, 13-52
- LONG 変数, 13-94, 13-108
  - COPY コマンドへの影響, B-4, B-8
- LONG 列
  - 最大幅の設定, 13-94, 13-108
  - 取得サイズの設定, 13-94, 13-108
  - 書式の変更, 7-6, 13-31
  - デフォルトの書式, 13-31

## M

---

- MARKUP, 4-4, 8-2, 13-109
  - BODY 句, 4-6
  - ENTMAP 句, 4-6
  - HEAD 句, 4-5
  - PREFORMAT 句, 4-8
  - SPOOL 句, 4-5
  - TABLE 句, 4-6
- MAXDATA 変数, C-2, C-6
- MAXIMUM 関数, 7-16
- Microsoft Internet Explorer
  - プロキシ例外の構成, 3-6
- MIME タイプ
  - 構成, 3-5
- MINIMUM 関数, 7-16
- mod\_ssl, 10-9
- MOUNT 句, 13-132

## N

---

- NCHAR 句
  - VARIABLE コマンド, 13-142
- NCHAR 列
  - 書式の変更, 7-6, 13-31
  - デフォルトの書式, 7-6, 13-31

NCLOB 句  
  VARIABLE コマンド, 13-143  
NCLOB 列  
  最大幅の設定, 13-94, 13-108  
  取得位置の設定, 13-94, 13-107  
  取得サイズの設定, 13-94, 13-108  
  書式の変更, 7-6, 13-31  
  デフォルトの書式, 13-31  
Netscape Navigator  
  プロキシ例外の構成, 3-6  
「New password」フィールド, 4-18  
NEW\_VALUE 句, 7-28, 13-34  
  タイトルの変数に現在の日付を格納, 7-29, 13-34  
NEWLINE 句, 13-34  
NEWPAGE コマンド, C-2, C-4  
NEWPAGE 変数, 7-30, 13-94, 13-110  
NEXT 句, 13-14  
NLS, 12-2, 12-3  
NLS\_DATE\_FORMAT, 13-10, 13-38  
NOAUDIT コマンド  
  使用禁止, 10-5  
NOLIST 句, 13-64  
NOLOG, 4-11  
NOMOUNT 句, 13-132  
NONE 句  
  WHENEVER OSERROR, 13-148  
  WHENEVER SQLERROR, 13-150  
NOPRINT 句, 7-16, 7-28, 13-34  
NOPROMPT 句, 13-11  
NORMAL モード, 13-125  
NULL 句, 13-34  
NULL 値  
  表示するテキストの設定, 13-34, 13-95, 13-110  
NULL 変数, 13-95, 13-110  
NUMBER  
  DESCRIBE による列定義の表示, 13-52  
NUMBER 関数, 7-16  
NUMBER 句, 6-28, 13-10  
  VARIABLE コマンド, 13-142  
NUMBER 列  
  書式の変更, 7-4, 13-32  
  デフォルトの書式, 7-4, 13-33  
NUMFORMAT 句  
  LOGIN.SQL, 3-3  
NUMFORMAT 変数, 13-95, 13-111  
NUMWIDTH 変数, 13-95, 13-111  
  NUMBER 列の書式への影響, 7-4, 13-33

NVARCHAR2 列  
  書式の変更, 7-6, 13-31  
  デフォルトの書式, 7-6, 13-31

## O

OFF 句, 13-35  
  ATTRIBUTE コマンド, 13-17  
  COLUMN コマンド, 7-9, 13-35  
  REPFOOTER コマンド, 13-87  
  REPHEADER コマンド, 13-87  
  SPOOL コマンド, 7-32, 13-127  
  TTITLE および BTITLE コマンド, 7-27, 13-138  
OF 句, 7-15  
「Old password」フィールド, 4-18  
OLD\_VALUE 句, 7-29, 13-35  
ON column 句  
  BREAK コマンド, 7-12, 13-18  
  COMPUTE コマンド, 7-15, 13-40  
ON expr 句  
  BREAK コマンド, 13-19  
  COMPUTE コマンド, 13-40  
ON REPORT 句  
  BREAK コマンド, 7-19, 13-20  
  COMPUTE コマンド, 7-19, 13-40  
ON ROW 句  
  BREAK コマンド, 7-13, 13-20  
  COMPUTE コマンド, 13-40  
ON 句  
  ATTRIBUTE コマンド, 13-17  
  COLUMN コマンド, 7-9, 13-35  
  TTITLE および BTITLE コマンド, 7-28  
OPEN 句, 13-132  
Oracle HTTP Server, 1-5  
  glogin.sql, 10-13  
  Product User Profile 表, 10-13  
  管理者, 10-12  
  起動および停止, 3-14  
  グローバリゼーション・サポートの構成, 12-4  
  構成, 3-7, 3-12  
  構成ファイルのテスト, 3-13  
  認証ユーザー名 / パスワード・エントリ, 3-12,  
  10-11  
Oracle HTTP Server の停止, 3-14  
Oracle Net, 1-4, 10-9  
  セキュリティ, 10-9  
  接続識別子, 13-44

プロトコル, 4-22  
Oracle Net、構成, 3-16  
oracle\_apache.conf, 3-13, 3-15  
Oracle9i, 1-4, 3-16  
グローバルゼーション・サポート, 12-2, 12-4  
ORDER BY 句  
出力で値を同時に表示, 7-11  
タイトルへの列値の表示, 7-28  
「Output」オプション  
ファイル, 2-9  
「Work Screen」画面, 2-9  
新しいウィンドウ, 2-9  
OUT 句, 7-33, 13-127

## P

---

PAGESIZE 句  
LOGIN.SQL, 3-3  
PAGESIZE 変数, 5-4, 7-30, 13-95, 13-111  
PAGE 句, 13-86  
PARAMETERS 句, 13-121  
Password  
フィールド, 2-4, 2-11  
PASSWORD コマンド, 13-44, 13-73  
使用禁止, 10-5  
PAUSE コマンド, 6-29, 13-74  
LOGIN.SQL, 3-3  
PAUSE 変数, 5-14, 13-95, 13-111  
PL/SQL, 5-7  
SQL\*Plus での出力の書式設定, 13-144  
SQL\*Plus でのモード, 5-6  
SQL コマンド, 5-6  
実行, 13-61  
定義の表示, 5-14  
ブロック、PL/SQL, 5-7  
PLAN\_TABLE の作成, 9-2  
PLAN\_TABLE 表, 9-2  
PLUSTRACE ロール, 9-2  
PLUSTRACE ロールの作成, 9-3  
PNO 句, 13-122  
「Preferences」画面, 2-8  
PREFORMAT, 4-8  
PREFORMAT 句, 4-8  
PRINT 句, 13-34  
PRINT コマンド, 13-75  
Product User Profile 表, 10-13  
PRODUCT\_USER\_PROFILE 表, 10-2

PRODUCT\_USER\_PROFILE 表の作成, 10-2  
PROMPT 句, 6-27, 13-11  
PROMPT コマンド, 6-26, 13-76  
値の入力を求めるプロンプトのカスタマイズ, 6-28  
PUPBLD.SQL, 10-2

## Q

---

QUIT コマンド, 13-62  
「EXIT」を参照

## R

---

RAW  
DESCRIBE による列定義の表示, 13-52  
RECOVER 句, 13-132  
RECOVER コマンド, 13-77  
AUTOMATIC 句, 13-78  
CANCEL 句, 13-79, 13-81  
CONTINUE 句, 13-78  
DATABASE 句, 13-79  
FROM 句, 13-78  
LOGFILE 句, 13-78  
STANDBY DATABASE 句, 13-79  
STANDBY DATAFILE 句, 13-80  
STANDBY TABLESPACE 句, 13-80  
UNTIL CANCEL 句, 13-79  
UNTIL CONTROLFILE 句, 13-80  
UNTIL TIME 句, 13-79  
USING BACKUP CONTROL FILE 句, 13-79  
データベースのリカバリ, 11-5  
RECSEPCHAR 変数, 7-10, 13-95, 13-111  
RECSEP 変数, 7-10, 13-95, 13-111  
REDO ログ・ファイル  
ARCHIVE LOG コマンド, 13-13  
REFCURSOR 句  
VARIABLE コマンド, 13-143  
REFCURSOR バインド変数  
ストアド・ファンクション, 6-31  
RELEASE 句, 13-122  
REMARK コマンド, 6-9, 13-83  
RENAME コマンド  
使用禁止, 10-5  
REPFOOTER 句, 13-122  
REPFOOTER コマンド, 7-22, 13-84  
BOLD 句, 13-87  
CENTER 句, 13-87

COL 句, 13-87  
FORMAT 句, 13-87  
LEFT 句, 13-87  
OFF 句, 13-87  
RIGHT 句, 13-87  
SKIP 句, 13-87  
TAB 句, 13-87  
  現行の定義の非表示, 13-87  
  フッター要素の整列, 13-87  
  レポート・フッターのインデント, 13-87  
REPHEADER 句, 13-122  
REPHEADER コマンド, 7-22, 13-86  
  BOLD 句, 13-87  
  CENTER 句, 13-87  
  COL 句, 13-87  
  FORMAT 句, 13-87  
  LEFT 句, 13-87  
  OFF 句, 13-87  
  PAGE 句, 13-86  
  RIGHT 句, 13-87  
  SKIP 句, 13-87  
  TAB 句, 13-87  
  現行の定義の非表示, 13-87  
  ヘッダーのインデント, 13-87  
  ヘッダー要素の整列, 7-24, 13-87  
REPLACE 句  
  COPY コマンド, B-3, B-6  
  SAVE コマンド, 13-90  
RESTRICT, 4-9, 10-8, 13-131  
「Retype new password」フィールド, 4-18  
REVOKE コマンド, 10-2  
  使用禁止, 10-5  
RIGHT 句, 7-24, 13-87, 13-139  
ROLLBACK 句, 13-63  
  WHENEVER OSERROR, 13-148  
  WHENEVER SQLERROR, 13-150  
ROWID  
  DESCRIBE による列定義の表示, 13-52  
RUN コマンド, 5-7, 13-89  
  / (スラッシュ) コマンドとの類似, 13-89, 5-7  
  カレント PL/SQL ブロックの実行, 5-7  
  現行の SQL コマンドまたは PL/SQL ブロックの実  
  行, 5-7  
  最終行をカレント行にする, 6-5  
  使用禁止, 10-5

## S

---

「Save Script」ボタン, 2-6  
SAVE コマンド, 13-90  
  APPEND 句, 13-90  
  CREATE 句, 13-90  
  INPUT を使用したスクリプトの作成, 6-3  
  REPLACE 句, 13-90  
  使用禁止, 10-5  
  スクリプトへの格納, 13-90  
SCAN 変数, C-2, C-6  
SCREEN 句, 6-29, 13-28  
Secure Sockets Layer セキュリティ, 10-9  
SELECT コマンド  
  BREAK コマンド, 7-11, 13-19, 13-20  
  COLUMN コマンド, 13-30  
  COMPUTE コマンド, 7-11  
  COPY コマンド, B-3, B-6  
  DEFINE コマンド, 13-48  
  ORDER BY 句, 7-11  
  結果の書式設定, 6-31  
  使用禁止, 10-5  
  タイトルの変数に現在の日付を格納, 7-29  
SERVEROUTPUT 変数, 13-112  
SET AUTOTRACE, 9-2  
SET MARKUP  
  BODY 句, 4-6  
  ENTMAP 句, 4-6, 8-12  
  HEAD 句, 4-5  
  HTML, 4-5  
  PREFORMAT 句, 4-8  
  SPOOL 句, 4-5  
  「SPOOL コマンド」を参照  
  TABLE 句, 4-6  
  埋め込まれた CGI の例, 8-7  
  対話型 HTML の例, 8-3, 8-5  
SET ROLE コマンド  
  使用禁止, 10-5  
SET TRANSACTION コマンド  
  使用禁止, 10-5  
SET 句, 13-135  
SET コマンド, 3-3, 5-10, 13-92  
  APPINFO 変数, 9-10, 13-96  
  ARRAYSIZE 変数, 9-10, 13-93, 13-97, B-8  
  AUTOCOMMIT 変数, 5-11, 13-93, 13-98  
  AUTOPRINT 変数, 13-93, 13-98, 13-144  
  AUTORECOVERY 変数, 13-93, 13-98

AUTOTRACE 変数, 13-93, 13-99  
 BLOCKTERMINATOR 変数, 13-93, 13-99  
 BUFFER 変数, C-4  
 CLOSECURSOR 変数, C-2, C-5  
 CMDSEP 変数, 13-93, 13-100  
 COLSEP 変数, 7-33, 13-93, 13-101  
 COMPATIBILITY 変数, 13-93, 13-101  
 CONCAT 変数, 6-22, 13-93, 13-102  
 COPYCOMMIT 変数, 13-93, 13-102, B-8  
 COPYTYPECHECK 変数, 13-93, 13-102  
 DEFINE 句, 6-22  
 DEFINE 変数, 13-93, 13-102  
 DESCRIBE 変数, 13-93, 13-102  
 DOCUMENT 変数, C-2, C-5  
 ECHO 変数, 6-13, 13-94, 13-104  
 EDITFILE 変数, 13-94, 13-104  
 EMBEDDED 変数, 13-94, 13-104  
 ESCAPE 変数, 6-22, 13-94, 13-104  
 FEEDBACK 変数, 13-94, 13-105  
 FLAGGER 変数, 13-94, 13-105  
 FLUSH 変数, 13-94, 13-105  
 HEADING 変数, 13-106  
 HEADSEP 変数, 7-3, 13-94, 13-106  
 INSTANCE 変数, 13-94, 13-106  
 LINESIZE 変数, 7-24, 7-31, 13-94, 13-107  
 LOBOFFSET 変数, 13-94, 13-107  
 LOGSOURCE 変数, 13-94, 13-108  
 LONGCHUNKSIZE 変数, 13-94, 13-108  
 LONG 変数, 13-94, 13-108, B-8  
 MARKUP 句, 13-109  
 MAXDATA 変数, C-2, C-6  
 NEWPAGE 変数, 7-30, 13-94, 13-110  
 NULL 変数, 13-95, 13-110  
 NUMFORMAT 句, 3-3  
 NUMFORMAT 変数, 13-95, 13-111  
 NUMWIDTH 変数, 7-4, 13-33, 13-95, 13-111  
 PAGESIZE 句, 3-3  
 PAGESIZE 変数, 5-4, 7-30, 13-95, 13-111  
 PAUSE 句, 3-3  
 PAUSE 変数, 5-14, 13-95, 13-111  
 RECSEPCHAR 変数, 7-10, 13-95, 13-111  
 RECSEP 変数, 7-10, 13-95, 13-111  
 REFCURSOR 変数の書式設定に使用, 13-144  
 SCAN 変数, C-2, C-6  
 SERVEROUTPUT 変数, 13-112  
 SHIFTINOUT 変数, 13-95, 13-113  
 SPACE 変数, C-2, C-6  
 SQLBLANKLINES 変数, 13-114  
 SQLCASE 変数, 13-95, 13-115  
 SQLCONTINUE 変数, 13-95, 13-115  
 SQLNUMBER 変数, 13-95, 13-116  
 SQLPLUSCOMPATIBILITY 変数, 13-95, 13-116  
 SQLPREFIX 変数, 13-95, 13-116  
 SQLPROMPT 変数, 13-95, 13-116  
 SQLTERMINATOR 変数, 13-95, 13-117  
 SUFFIX 変数, 13-95, 13-117  
 TAB 変数, 13-96, 13-118  
 TERMOUT 変数, 13-96, 13-118  
 TIME 句, 3-3  
 TIME 変数, 13-96, 13-118  
 TIMING 変数, 13-96, 13-118  
 TRIMOUT 変数, 13-96, 13-118  
 TRIMSPOOL 変数, 13-96, 13-119  
 TRUNCATE 変数, C-2, C-7  
 UNDERLINE 変数, 13-96, 13-119  
 VERIFY 句, 6-18  
 VERIFY 変数, 6-22, 13-96, 13-119  
 WRAP 変数, 7-7, 13-96, 13-119  
 使用禁止, 10-5  
 置換変数への影響, 6-23  
 SET 変数, 5-10  
 「システム変数」を参照  
 SGA 句, 13-122  
 SHIFTINOUT 変数, 13-95, 13-113  
 SHOWMODE 変数, 13-95, 13-114  
 SHOW 句, 13-136  
 SHOW コマンド, 5-10, 13-120  
 ALL 句, 13-120  
 BTITLE 句, 13-121  
 ERRORS 句, 13-121  
 LABEL 変数, C-2, C-7  
 LNO 句, 13-121  
 PNO 句, 13-122  
 RELEASE 句, 13-122  
 REPFOOTER 句, 13-122  
 REPHEADER 句, 13-122  
 SPOOL 句, 13-122  
 SQLCODE 句, 13-122  
 TTITLE 句, 13-122  
 USER 句, 13-122  
 現行のページ・サイズの表示, 7-32  
 SHUTDOWN コマンド, 13-125  
 ABORT, 13-125  
 IMMEDIATE, 13-125

NORMAL, 13-125  
 TRANSACTIONAL LOCAL, 13-126  
 SKIP PAGE 句, 7-13, 13-21  
 SKIP 句  
   BREAK コマンド, 7-13, 13-20  
   REPHEADER および REPFOOTER コマンド, 13-87  
   TTITLE および BTITLE コマンド, 7-24, 13-139  
   下部タイトルの前に空白行を配置するための使用方法, 7-25  
 SPACE 変数, C-2, C-6  
 SPOOL 句, 4-5, 4-7, 13-122  
 SPOOL コマンド, 7-32, 13-127  
   HTML ファイル, 4-7  
   OFF 句, 7-32, 13-127  
   OUT 句, 7-33, 13-127  
   SET MARKUP とともに使用, 8-3  
   使用禁止, 10-5  
   スプールを OFF にする, 7-32, 13-127  
   ファイル名, 7-33, 13-127  
 SQL DML 文  
   レポート, 13-93, 13-99  
 SQL\*Plus  
   環境の設定, 3-2  
   起動, 4-2, 4-3  
   起動のショートカット, 4-3  
   グローバリゼーション・サポート, 12-1  
   グローバリゼーション・サポートの構成, 12-2  
   コマンド一覧, 13-2, 13-93  
   コマンド・プロンプト, 4-2  
   コマンドライン, 2-2  
   システム変数によるパフォーマンスへの影響, 9-10  
   実行計画, 9-3  
   終了, 4-13, 13-62  
   条件付き終了, 13-148, 13-150  
   制限, A-1  
   対象ユーザー, 1-5  
   チューニング, 9-2  
   データベース管理, 11-2  
   統計, 9-4  
   廃止されたコマンドのかわりに使用, C-2  
   バッチ・モードでのコマンドの実行, 6-16  
 SQL\*Plus のエラー・メッセージ, 14-2  
 SQL.PNO、レポート・タイトル内での参照, 7-26  
 SQL.SQLCODE  
   EXIT コマンドでの使用, 13-62  
 SQLBLANKLINES 変数, 13-95, 13-114  
 SQLCASE 変数, 13-95, 13-115  
 SQLCODE 句, 13-122  
   SHOW コマンド, 13-122  
 SQLCONTINUE 変数, 13-95, 13-115  
 SQLNUMBER 変数, 13-95, 13-116  
 SQLPLUS コマンド, 4-2  
   -? 句, 4-4  
   -MARKUP SPOOL 句, 4-5  
   -MARKUP オプション, 4-4  
   -MARKUP 句, 4-5  
   -SILENT オプション, 4-10, 8-11  
   -SILENT 句, 4-10  
   - 句, 4-4  
   /NOLOG 句, 4-11  
   @ (アットマーク), 4-3, 4-21, 6-14  
   BODY オプション, 4-6  
   ENTMAP オプション, 4-6  
   EXIT FAILURE, 6-15  
   HEAD オプション, 4-5  
   HTML オプション, 4-5  
   PREFORMAT オプション, 4-8  
   RESTRICT, 4-9, 10-8  
   SPOOL 句, 4-7  
   SYSDBA 句, 4-11  
   TABLE オプション, 4-6  
   構文, 4-3  
   構文の表示, 4-4  
   コマンド  
     SQLPLUS, 4-3  
     サービス名, 4-10, 4-21  
     失敗した接続, 6-15  
     スクリプトの実行, 6-14  
     接続識別子, 4-10  
     ユーザー名 / パスワード, 4-2, 4-10  
     リモート・データベースへの接続, 4-21  
 SQLPREFIX 変数, 13-95, 13-116  
 SQLPROMPT 変数, 13-95, 13-116  
 SQLTERMINATOR, 13-114  
 SQLTERMINATOR 変数, 13-68, 13-95, 13-117  
 SQL オプティマイザ, 9-3  
 SQL 句, 13-28  
 SQL バッファ, 5-2  
 SSL セキュリティ, 10-9  
 STANDBY DATAFILE 句, 13-80  
 STANDBY TABLESPACE 句, 13-80  
 STARTUP コマンド, 13-131  
   FORCE 句, 13-131  
   MOUNT 句, 13-132

NOMOUNT 句, 13-132  
OPEN 句, 13-132  
PFILE 句, 13-131  
RECOVER 句, 13-132  
RESTRICT 句, 13-131  
データベースの指定, 13-132  
START 句, 13-14, 13-136  
START コマンド, 6-13, 13-129  
  @@ (二重アットマーク) コマンドとの類似, 13-8,  
  13-130  
  @ (アットマーク) コマンドとの類似, 6-14, 13-6,  
  13-130  
  使用禁止, 10-5  
  スクリプト, 6-13, 13-129  
  スクリプトへのパラメータの受渡し, 6-25  
  引数, 6-25  
STD 関数, 7-16  
STOP 句, 13-14, 13-136  
STORE コマンド, 3-3, 13-135  
  SET 句, 13-135  
「Substitution Variables」フィールドの定義, 6-24  
SUFFIX 変数, 13-95, 13-117  
  @ (アットマーク) コマンドとの併用, 13-5  
  EDIT コマンドとの併用, 13-59  
  GET コマンドとの併用, 13-64  
  SAVE コマンドとの併用, 13-90  
  START コマンドとの併用, 13-129  
SUM 関数, 7-16  
SYSDATE, 7-29  
SYSDBA 句, 13-45  
SYSOPER 句, 4-11, 13-45  
「System Variables」画面, 2-14

## T

---

TABLESPACE 句, 13-79  
TABLE オプション, 4-6  
TABLE 句, 4-6  
TAB 句, 13-87, 13-139  
TAB 変数, 13-96, 13-118  
TERMOUT 変数, 13-96, 13-118  
  SPOOL コマンドでの使用, 13-127  
TIME 変数, 13-96, 13-118  
TIMING 句, 13-28  
TIMING コマンド, 9-7, 13-136  
  SHOW 句, 13-136  
  START 句, 13-136

STOP 句, 13-136  
  現行の領域の削除, 13-136  
  作成されたすべての領域の削除, 13-28  
TIMING 変数, 13-96, 13-118  
TNS 別名, 2-5, 2-11, 4-21  
TO 句, B-5  
TRIMOUT 変数, 13-96, 13-118  
TRIMSPOOL 変数, 13-96, 13-119  
TRUNCATED 句, 7-7, 13-35  
TRUNCATE コマンド  
  使用禁止, 10-5  
TRUNCATE 変数, C-2, C-7  
TTITLE 句, 13-122  
TTITLE コマンド, 7-22, 13-138  
  BOLD 句, 13-139  
  CENTER 句, 7-24, 13-139  
  COL 句, 7-25, 13-139  
  FORMAT 句, 7-27, 13-139  
  LEFT 句, 7-24, 13-139  
  OFF 句, 7-27, 13-138  
  ON 句, 7-28  
  RIGHT 句, 7-24, 13-139  
  SKIP 句, 7-24, 13-139  
  TAB 句, 13-139  
  旧形式, C-7  
  現行の定義の非表示, 7-27, 13-138  
  現行の定義の表示, 7-27, 13-139  
  現行の定義の表示と非表示の切替え, 7-28  
  タイトルのインデント, 7-25, 13-139  
  タイトル要素の整列, 7-24, 13-139  
  列値変数の参照, 7-28, 13-34

## U

---

UNDEFINE コマンド, 6-17, 13-141  
  DEFINE コマンド, 13-48  
UNDERLINE 変数, 13-96, 13-119  
Unicode, 12-2, 12-3  
UNTIL CANCEL 句, 13-79  
UNTIL CHANGE 句, 13-79  
UNTIL CONTROLFILE 句, 13-80  
UNTIL TIME 句, 13-79  
UPDATE コマンド、使用禁止, 10-5  
URL, 1-5  
  iSQL\*Plus の起動, 4-15  
  動的レポートの実行, 8-12

## Username

フィールド, 2-4, 2-11, 3-17, 4-18

USER 句, 13-122

USING BACKUP CONTROL FILE 句, 13-79

USING 句, B-3, B-6

UTF-8, 12-2, 12-3

## V

---

V\$SESSION 仮想表, 13-96

V\$SQLAREA 仮想表, 13-96

VARCHAR2

DESCRIBE による列定義の表示, 13-52

VARCHAR2 句

VARIABLE コマンド, 13-143

VARCHAR2 列

書式の変更, 7-6, 13-31

デフォルトの書式, 7-6

VARCHAR 列

書式の変更, 7-6

デフォルトの書式, 7-6, 13-31

VARIABLE コマンド, 13-142

CHAR 句, 13-142

CLOB 句, 13-143

NCHAR 句, 13-142

NCLOB 句, 13-143

NUMBER 句, 13-142

REFCURSOR 句, 13-143

VARCHAR2 句, 13-143

変数句, 13-142

VARIANCE 関数, 7-16

VERIFY 句, 6-18

VERIFY 変数, 6-22, 13-96, 13-119

## W

---

WARNING 句, 13-62

Web ブラウザ, 1-4, 2-3, 8-2

Web、レポートの出力, 8-2

WHENEVER OSERROR コマンド, 13-148

COMMIT 句, 13-148

CONTINUE 句, 13-148

EXIT 句, 13-148

NONE 句, 13-148

ROLLBACK 句, 13-148

WHENEVER SQLERROR コマンド, 6-16, 13-150

COMMIT 句, 13-150

CONTINUE 句, 13-150

EXIT 句, 13-150

NONE 句, 13-150

ROLLBACK 句, 13-150

Windows iSQL\*Plus Extension, 2-15

WORD\_WRAPPED 句, 7-7, 7-10, 13-35

「Work Screen」画面, 2-5

「Output」オプション, 2-9

WRAPPED 句, 7-7, 13-35

WRAP 変数, 7-7, 13-96, 13-119

## X

---

XMLType

DESCRIBE による列定義の表示, 13-52

値の挿入, 7-7

最大列幅の設定, 13-108

作成, 7-7

データの選択, 7-7

列の取得サイズの設定, 13-108

列の書式設定, 7-7

列幅, 7-6

レポート内での書式設定, 7-6

## あ

---

アクセスの拒否および付与, 10-2

値の入力を求めるプロンプト

ACCEPT の使用, 6-26

カスタマイズ, 6-28

置換変数, 6-18

パラメータを使用した回避, 6-25

アンパサンド (&)

パラメータ, 13-129

置換変数, 6-17

パラメータ, 6-25, 13-5

## い

---

異常終了モード, 13-125

インスタンス

起動, 13-131

停止, 13-125

インストール

デフォルト・ユーザー作成, 4-17

## う

---

ウィンドウ、出力、 2-9

## え

---

エスケープ文字、定義、 13-94, 13-104

エディタ、 5-2

エラー・メッセージ

    COPY コマンド、 14-43

    iSQL\*Plus、 14-33

    SQLPLUS、 14-2

エラー・メッセージ、解釈、 5-15

エラー、含まれている行をカレント行にする、 6-5

エンティティ、HTML、 8-12

## お

---

オブジェクト、定義の表示、 13-102

オンライン・ヘルプ、 4-12, 13-66

## か

---

カーソル変数、 13-144

改ページ、新しいページの開始、 7-30, 13-110

拡張子、 13-90, 13-95, 13-117, 13-135

カスケード・スタイルシート

    カスタマイズ、 3-12

    ユーザー・インタフェース、 3-12

各国語サポート、 12-2, 12-3

    「グローバル化・サポート」を参照

    フォント、 12-3

    文字コード、 12-3

画面

    「Change Password」、 4-18

    「Connection Identifier」フィールド、 2-5, 2-11

    DBA の「Login」画面、 2-10

    DBA の「Work Screen」画面、 2-12

    「Enter statements」フィールド、 2-6, 2-7

    「Expired Password」、 4-19

    「History」、 2-6

    「Interface Options」、 2-9

    「Login」、 2-4, 2-10

    「New password」フィールド、 4-18

    「Old password」フィールド、 4-18

    「Password」フィールド、 2-4, 2-11

    「Preferences」、 2-8

    「Retype new password」フィールド、 4-18

    「Script location」フィールド、 2-8

    「Substitution Variables」フィールドの定義、 6-24

    「System Variables」、 2-14

    「Username」フィールド、 2-4, 2-11, 3-17, 4-18

    「Work Screen」、 2-5

    権限リスト、 2-12

画面、消去、 6-29, 13-28

環境属性の保存、 13-135

カンマ、数値書式、 7-5

管理者

    Oracle HTTP Server、 10-12

## き

---

起動

    iSQL\*Plus、 4-13, 4-15

    Oracle HTTP Server、 3-14

    SQL\*Plus、 2-2, 4-2

    SQL\*Plus のショートカット、 4-3

起動時のデータベース名、 13-131

機能、 2-3

行

    COPY でのコミット後に数を設定、 13-102

    一度に取得する数の設定、 9-10, 13-93, 13-97

    終わりの空白の削除、 13-118

    カレント行の後への新しい行の追加、 6-7, 13-69

    カレント行の判断、 6-5

    カレント行の編集、 6-5

    計算の実行、 7-15, 13-39

    テキストの追加、 6-8, 13-12

    バッファからの削除、 6-9, 13-50

    バッファ内のすべての行の削除、 13-50

    バッファ内のすべての行の表示、 6-4, 13-71

    バッファの最後に追加、 13-69

    バッファの最初に追加、 13-69

    幅の変更、 7-31, 13-94, 13-107

行番号、SQL コマンド、 5-3

## <

---

空白行

    PL/SQL ブロック、 5-7

    SQL コマンド、 5-5

    SQL コマンドでの保存、 13-95, 13-114

クライアント層、 1-4

    構成、 3-5

グローバリゼーション・サポート, 12-1  
Oracle9i データベース, 12-2, 12-4

## け

---

### 権限

リスト, 2-12

## こ

---

### 構成

HTTP Server, 3-12  
HTTP Server 構成ファイルのテスト, 3-13  
httpd.conf, 3-13  
iSQL\*Plus, 3-5  
iSQL\*Plus Extension for Windows, 3-16  
iSQL\*Plus サーバー, 3-7  
iSQL\*Plus 用の Cookie, 3-5  
iSQL\*Plus 用の Java スクリプト, 3-5  
MIME タイプ, 3-5  
Oracle HTTP Server, 3-7  
Oracle Net, 3-16  
oracle\_apache.conf, 3-13  
Oracle9i, 3-16  
SQL\*Plus, 3-1  
クライアント層, 3-5  
グローバリゼーション・サポート, 12-1, 12-2,  
12-3  
中間層, 3-7  
データベース層, 3-16  
ファイルの関連付け, 3-5  
プロキシ・サーバー例外, 3-6

### 構文

COPY コマンド, B-5

### 構文規則

SQL\*Plus コマンド, 5-9  
SQL コマンド, 5-4

### コマンド

iSQL\*Plus での入力, 2-6  
iSQL\*Plus での未サポート, D-1

### SQL

構文に従う, 5-4  
実行, 5-7, 13-9, 13-89  
実行せずに入力, 5-5  
終了, 5-5  
終了および実行のために使用する文字設定,  
13-95, 13-117

入力および実行, 5-3  
バッファ内の現行のコマンドの表示, 6-5  
バッファ内の編集, 6-4  
ホスト・システム・エディタによる編集, 13-59  
保存, 13-90

### SQL\*Plus

SQL コマンドのエントリ中に入力, 13-116  
コマンド一覧, 13-2  
コマンド・プロンプトでの編集, 6-4  
終了, 5-10, 13-1  
次の行への継続, 5-9, 13-1  
入力および実行, 5-8  
廃止されたコマンドのかわりに使用, C-2  
略称, 5-8

### SQL、次の行への継続, 5-4

空白, 5-2  
再度使用可能, 10-4  
実行中の停止, 5-12  
実行に影響する変数, 5-10  
使用禁止, 10-4  
タイプ, 5-2  
タイミング統計の収集, 9-7, 13-136  
対話型の作成, 6-16  
タブ, 5-2  
バッファ内の現行のコマンドの表示, 13-71  
ホスト、SQL\*Plus からの実行, 5-12, 13-67

### コマンド・ファイル

@@ (二重アットマーク) コマンド, 13-7  
@ (アットマーク) コマンド, 13-5  
EDIT コマンド, 13-59  
GET コマンド, 13-64  
SAVE コマンド, 13-90  
SAVE での作成, 13-90  
START コマンド, 13-129  
Uniform Resource Locator, 13-5, 13-7, 13-129  
異常終了およびリターン・コードを返す終了,  
13-149, 13-151  
コメントの挿入, 13-83  
実行, 13-5, 13-129  
取得, 13-64  
登録, 9-10, 13-93, 13-96  
ネスト, 13-7  
バッチ・モードでの実行, 13-63  
パラメータの受渡し, 13-5, 13-129  
ホスト・システム・エディタによる編集, 13-59  
コマンド・ファイル拡張子, 13-95

## コマンド・プロンプト

SET SQLPROMPT, 13-95, 13-116

SQL\*Plus, 4-2

## コマンドライン

グローバル化・サポートの構成, 12-2

ユーザー・インタフェース, 2-2

## コメント

-- を使用した作成, 6-10

/\*...\*/ を使用した作成, 6-10

REMARK を使用した作成, 6-9, 13-83, C-2

スクリプトに挿入, 6-9, 13-83, C-2

## コロン (:)

バインド変数, 6-29

## コンテキスト・メニュー

iSQL\*Plus Extension for Windows, 2-15

## さ

---

### サーバー

iSQL\*Plus パラメータ, 2-14

### サービス名

CONNECT コマンド, 4-22

COPY コマンド, B-5, B-7, B-9

SQLPLUS コマンド, 4-21

### サイト・プロファイル

GLOGIN.SQL, 3-2

LOGIN.SQL, 3-2

「ユーザー・プロファイル」を参照

### サマリー行

同じブレイク列についての複数の出力, 7-20

計算および出力, 7-15, 13-39

異なる列についての同じタイプの計算, 7-20

総サマリーおよびサブサマリー (合計) の出力,  
7-19

レポートの終わりでの計算および出力, 7-19

### 3層モデル, 1-4

### サンプル表

アクセス, 1-7

解除, 1-7

削除, 1-8

作成, 1-8

サンプル表の削除, 1-8

サンプル表の作成, 1-8

サンプル表のロック解除, 1-7

## し

---

### システム管理値

タイトル内での書式設定, 7-27

タイトル内に表示, 7-26

タイトルの中に表示, 13-138

ヘッダーおよびフッターの中に表示, 13-86

### システム変数, 5-10, 13-92

iSQL\*Plus での設定, 2-14

SQL\*Plus のパフォーマンスへの影響, 9-10

格納およびリストア, 3-3

現行の設定の表示, 5-10, 13-120

新旧の値の表示, 13-95, 13-114

置換変数との併用, 6-22

### 実行

CREATE コマンド, 5-6

文, 5-3

### 実行計画, 9-3

### 実行統計

レポートへの挿入, 13-99

### 自動トレース・レポート, 9-2

シャープ記号 (#), 13-33

### 終了

iSQL\*Plus, 4-14

終了、条件付き, 13-148, 13-150

### 出力

REFCURSOR 変数, 13-144

SPOOL コマンド, 13-127

空白の書式設定, 13-118

バインド変数の自動出力, 13-98

表示中の一時停止, 5-14, 13-111

### 使用禁止

iSQL\*Plus, 3-15

PL/SQL コマンド, 10-5

SQL\*Plus コマンド, 10-4

SQL コマンド, 10-4

### 初期化パラメータ

表示, 13-121

書式モデル、数値, 7-4, 13-33

## す

---

### 数値書式

\$, 7-5

0, 7-5

9, 7-5

カンマ, 7-5

デフォルト設定, 13-95, 13-111  
スクリプト  
@ (アットマーク) コマンド, 6-13  
EDIT コマンド, 6-3  
SAVE コマンド, 6-3  
SQL\*Plus 起動時の実行, 4-11, 6-14  
SQLPLUS コマンド, 4-11, 6-14  
START コマンド, 6-13  
一連のスクリプトの順次実行, 6-14  
エンド・ユーザーによる入力, 6-16  
拡張子, 13-90, 13-117, 13-135  
「コマンド・ファイル」を参照  
コメントの挿入, 6-9  
システム・エディタでの作成, 6-2  
実行, 6-13  
ネスト, 6-14  
「場所」フィールド, 2-8  
バッチ・モードでの実行, 6-16  
パラメータの受渡し, 6-25  
複数の PL/SQL ブロックの挿入, 6-2  
複数の SQL コマンドの挿入, 6-2  
編集, 5-2  
ホスト・システム・エディタでの編集, 6-3  
リターン・コードを伴う強制終了, 6-16  
スクリプトの実行  
「実行」を参照  
ストアド・ファンクション, 6-31  
ストアド・プロシージャ  
作成, 5-6  
スラッシュ (/) コマンド, 13-9  
GET コマンドでロードされたファイル, 13-64

## せ

---

制限、SQL\*Plus, A-1  
セキュリティ  
HTTP, 10-9  
Oracle HTTP Server, 10-9  
Oracle Net, 10-9  
PRODUCT\_USER\_PROFILE 表, 10-2  
RESTRICT, 4-9, 10-8  
Secure Sockets LayerSSL, 10-9  
埋込み型の Web レポート, 8-7  
パスワードの変更, 13-73  
参照可能なパスワード, 4-10  
セッション  
識別, 1-5, 3-7, 4-13

ステートフルな動作, 3-7  
同時, 1-5  
接続識別子, 2-5, 2-11, 4-20, 4-21, 13-44  
CONNECT コマンド, 13-44  
COPY コマンド, B-2  
DESCRIBE コマンド, 13-52  
SQLPLUS コマンド, 4-10  
セミコロン (;)  
SQL\*Plus コマンド, 13-1  
PL/SQL ブロック, 5-7  
SQL\*Plus コマンド, 5-10  
SQL コマンド, 5-4, 5-5  
バッファに格納されない, 6-5

## そ

---

層間の通信, 1-4, 1-5

## た

---

タイトル  
インデント, 7-25, 13-139  
現行の定義の表示, 7-27, 13-23, 13-139  
現在の日付の表示, 7-29, 13-34, 13-37  
最終行から下部タイトルまでの間隔, 7-25  
システム管理値の書式設定, 7-27  
システム管理値の表示, 7-26, 13-138  
上部および下部の設定, 7-22, 13-23, 13-138, C-2  
上部タイトルからページの終わりまでの行の設定,  
13-95, 13-111  
定義の非表示, 7-27, 13-138  
定義の表示と非表示の切替え, 7-28  
ページ下部への表示, 7-22, 13-23, C-2  
ページ最上部から上部タイトルまでの行数の設定,  
7-30, 13-94, 13-110, C-2  
ページ上部への表示, 7-22, 13-138, C-2  
ページ番号の表示, 7-26, 13-140  
要素の書式設定, 13-139  
要素の整列, 7-24, 13-139  
列値の表示, 7-28, 13-34, 13-35  
レポートの始めまたは終わりに設定, 7-22  
タグ、HTML, 8-2

## ち

---

置換変数, 6-17, 6-23  
DEFINE コマンド, 6-20, 13-48

値の入力を求める不要なプロンプトの回避, 6-20  
アンパサンドおよび二重アンパサンド, 6-20  
解析, 9-10  
システム変数との併用, 6-22  
使用する位置および方法, 6-18  
制限, 6-22  
接頭辞, 13-93, 13-102, C-2  
直後に文字を追加, 6-20  
定義, 6-20  
未定義, 6-18  
連結文字, 13-93, 13-102  
中間層, 1-4  
構成, 3-7  
チューニング  
-idle-timeout, 9-13  
iSQL\*Plus のアクティブ統計, 9-11  
iSQLPlusHashTableSize, 9-13  
iSQLPlusNumberOfThreads, 9-13  
iSQLPlusTimeOutInterval, 9-13  
SET APPINFO OFF, 9-10  
SET ARRAYSIZE, 9-10  
SET DEFINE OFF, 9-10  
SET FLUSH OFF, 9-11  
SET SERVEROUTPUT OFF, 9-11  
SET TRIMOUT ON, 9-11  
SET TRIMSPOOL ON, 9-11  
SQL\*Plus, 9-2  
システム変数, 9-10

## て

---

データベース  
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-9  
SQL\*Plus を終了せずに切断, 4-17, 4-18, 13-58  
オープン, 11-3, 13-132  
起動, 11-2  
接続識別子, 13-44  
停止, 11-2, 11-3  
データのコピー, B-2, B-5  
デフォルト・データベースへの接続, 13-44  
マウント, 11-3, 13-132  
リカバリ, 11-5, 13-77  
リモート・データベースへの接続, 4-20, 13-44  
データベース・アクセス制限, 10-10  
データベース管理者, 11-2  
データベース層, 1-4

構成, 3-16  
データベースのオープン, 13-132  
データベースの変更、自動保存, 5-11, 13-93, 13-98  
データベースのマウント, 13-132  
データベース・ファイル  
リカバリ, 13-77  
テキスト, 4-6  
APPEND によるカレント行への追加, 6-8, 13-12  
CHANGE による旧から新への変更, 6-5, 13-25  
バッファからの消去, 6-4, 13-27  
テキスト・エディタ、ホスト・オペレーティング・シ  
ステム, 6-2, 13-59  
デフォルト・ユーザー  
インストール時に作成, 4-17

## と

---

問合せ  
COPY コマンド, B-3, B-6  
取得したレコード数の表示, 13-94, 13-105, 5-4  
問合せ結果  
画面に表示, 5-4  
ファイルへの格納, 7-33, 13-127  
プリンタへの出力, 7-33, 13-127  
問合せ実行パス  
レポートへの挿入, 13-99  
問合せの中断, 5-12  
問合せの停止, 5-12  
問合せの取消し, 5-12  
問合せのトレース, 9-7  
統計, 9-4  
iSQL\*Plus サーバー, 2-12  
iSQL\*Plus サーバーのアクティブ統計, 9-11  
iSQL\*Plus のアクティブ統計の解釈, 9-13  
タイミング統計の収集, 9-7  
動的レポート, 8-12

## な

---

長い SQL\*Plus コマンドの継続, 5-9, 13-1

## に

---

日本語, 12-2, 12-3  
入力  
[Return] のアクセプト, 6-29  
ユーザーからの値のアクセプト, 6-26, 13-10

入力領域, 2-6  
編集, 5-2  
認証  
htpasswd ユーティリティ, 10-12  
iSQL\*Plus, 10-10  
iSQL\*Plus サーバーの統計, 10-9  
iSQL\*Plus での DBA アクセス, 10-9, 10-11  
iSQL\*Plus ユーザーに対する有効化, 3-12  
接続エントリの追加, 10-12  
ユーザー名 / パスワード・エントリの追加, 3-12,  
10-11  
認証エントリの追加, 10-12

## は

---

廃止されたコマンド

BTITLE, C-3  
COLUMN コマンドの DEFAULT 句, C-3  
DOCUMENT, C-2, C-3  
NEWPAGE, C-2, C-4  
SET コマンドの BUFFER 変数, C-4  
SET コマンドの CLOSECURSOR 変数, C-2, C-5  
SET コマンドの DOCUMENT 変数, C-2, C-5  
SET コマンドの MAXDATA 変数, C-2, C-6  
SET コマンドの SCAN 変数, C-2, C-6  
SET コマンドの SPACE 変数, C-2, C-6  
SET コマンドの TRUNCATE 変数, C-2, C-7  
SHOW コマンドの LABEL 変数, C-2, C-7  
TTITLE コマンドの旧形式, C-7

ハイフン

長い SQL\*Plus コマンドの継続, 5-9, 13-1

バインド変数, 6-29

COPY コマンド, 13-144  
PL/SQL ブロック, 13-144  
SQL 文, 13-144  
作成, 13-142  
自動表示, 13-98, 13-144  
表示, 13-75

パスワード, 4-17

CONNECT コマンド, 4-17, 4-18, 4-21, 13-44  
COPY コマンド, B-5, B-7, B-9  
HTTP Server 認証, 2-10  
iSQL\*Plus での変更, 2-8, 4-18, 4-19  
PASSWORD コマンドで変更, 13-73  
SQLPLUS コマンド, 4-2, 4-10, 4-21  
参照可能な警告, 4-10

バックグラウンド・プロセス  
異常終了後の起動, 13-125

バッチ・モード, 6-16, 13-63

バッファ, 5-2

1 行の削除, 6-4, 13-50  
1 行の表示, 6-4, 13-71  
新しい行の挿入, 6-7  
ある範囲の行の削除, 6-4, 13-50  
ある範囲の行の表示, 6-4, 13-71  
カレント行の削除, 6-4, 13-50  
カレント行の表示, 6-4, 13-71  
行の削除, 13-50, 6-9  
行へのテキストの追加, 6-8, 13-12  
最終行の削除, 6-4, 13-50  
最終行の表示, 6-4, 13-71  
新規行の挿入, 13-69  
すべての行の削除, 13-27, 13-50  
すべての行の消去, 6-4  
すべての行の表示, 6-4, 13-71  
内容の実行, 5-7, 13-9, 13-89  
内容の表示, 6-5, 13-71  
内容の保存, 13-90  
ホスト・システム・エディタへのロード, 13-59

パフォーマンス

SQL 文, 9-2  
ダイアルアップ回線を超える, 13-118

パラメータ, 6-25, 13-5, 13-129

-idle-timeout, 9-13

iSQL\*Plus サーバー, 2-14

iSQLPlusHashTableSize, 9-13

iSQLPlusNumberOfThreads, 9-13

iSQLPlusTimeOutInterval, 9-13

## ひ

---

引数

START コマンド, 6-25

日付、タイトルの変数に現在の日付を格納, 7-29,  
13-34, 13-37

表

値のコピー, B-2, B-5, B-9  
コピー時に別のユーザーの表への参照, B-8  
コピー時の宛先の制御, B-3, B-6  
サンプルへのアクセス, 1-7  
列定義の表示, 5-13, 13-52

表領域、リカバリ, 13-77

ピリオド (.)

PL/SQL ブロックの終了, 13-93, 13-99, 5-7

## ふ

---

ファイル

関連付けの構成, 3-5

出力, 2-9

フラット, 7-32

ファイル拡張子, 3-3, 13-90, 13-95, 13-117, 13-135

ファイル名

@@ (二重アットマーク) コマンド, 13-7

@ (アットマーク) コマンド, 13-5

EDIT コマンド, 13-59

GET コマンド, 13-64

SAVE コマンド, 13-90

SPOOL コマンド, 7-33, 13-127

SQLPLUS コマンド, 4-11

フィールド

Connection Identifier, 2-5, 2-11

Enter statements, 2-6, 2-7

New password, 4-18

Old password, 4-18

Password, 2-4, 2-11

Privilege, 2-12

Retype new password, 4-18

Script location, 2-8

Username, 2-4, 2-11, 3-17, 4-18

置換変数の定義, 6-24

フォント

各国語サポート, 12-3

フッター

インデント, 13-87

現行の定義の表示, 13-84

システム管理値の表示, 13-86

定義の非表示, 13-87

ページ下部への表示, 13-84

要素の書式設定, 13-87

要素の整列, 13-87

レポートの終わりに設定, 7-22

負の無限大記号 (--), 13-33

ブラウザ、Web, 8-2

フラット・ファイル, 7-32

フラット・ファイルの作成, 7-32

ブレイク定義

現行の取消し, 7-15, 13-27

現行の表示, 7-15, 13-21

ブレイク列, 7-11, 13-18

値が変更されたときの空白の挿入, 7-13

値の重複の抑止, 7-12

複数の指定, 7-14

プロキシ・サーバー例外

Microsoft Internet Explorer, 3-6

Netscape Navigator, 3-6

構成, 3-6

ブロック、PL/SQL

終了のために使用する文字設定, 13-93, 13-99

タイミング統計, 13-118

ホスト・システム・エディタによる編集, 13-59

保存, 13-90

SQL コマンド, 5-6

SQL バッファからの実行, 5-7

SQL バッファに格納, 5-7

継続, 5-7

入力および実行, 5-7

バッファ内の現行のコマンドの表示, 6-5

バッファ内の編集, 6-4

ホスト・システム・エディタでの編集, 6-2

プロンプト

SET SQLPROMPT, 13-95, 13-116

文

実行, 5-3

文のトレース

データベース・リンクの使用, 9-7

問合せ実行パス用, 9-5

パフォーマンス統計用, 9-5

パラレル問合せオプション, 9-8

## へ

---

ページ

サイズの設定, 7-30

サイズを画面または用紙のサイズに合わせる, 7-30

デフォルトのサイズ, 7-30

長さの変更, 7-30, 13-95, 13-111

ページ番号、タイトルへの挿入, 7-14, 7-26

ヘッダー

インデント, 13-87

現行の定義の表示, 13-88

システム管理値の表示, 13-86

定義の非表示, 13-87

非表示, 7-24

ページ上部への表示, 13-86

要素の書式設定, 13-87

- 要素の整列, 7-24, 13-87
- 列ヘッダー, 13-106
- レポートの始めに設定, 7-22
- ヘルプ
  - iSQL\*Plus オンライン, 4-14
- ヘルプ、オンライン, 4-12, 13-66
- 編集
  - 外部エディタ, 5-2
  - スクリプト, 5-2
  - 入力領域, 5-2
- 変数
  - システム変数, 5-10
  - 置換, 6-23
  - 置換変数, 6-17
  - バインド変数, 6-29
  - ユーザー変数, 13-48

## ほ

---

- ホスト・オペレーティング・システム
  - SQL\*Plus からのコマンド実行, 5-12, 13-67
  - エディタ, 6-2, 13-59
  - ファイル、バッファへのロード, 13-64
- ボタン
  - Browse, 2-6, 2-15
  - Cancel, 2-8, 4-19, 6-24
  - Clear Screen, 2-6
  - Execute, 2-6
  - Load Script, 2-6
  - Login, 2-5, 2-12
  - Save Script, 2-6

## む

---

- 無限大記号 (~), 13-33

## め

---

- メッセージ、画面への送信, 6-26, 13-76
- メディア・リカバリ, 13-132

## も

---

- 文字コード
  - 各国語サポート, 12-3

## ゆ

---

- ユーザー・インタフェース, 2-3
  - コマンドライン, 2-2
- ユーザー・プロファイル, 3-2
  - GLOGIN.SQL, 3-2
  - LOGIN.SQL, 3-2
  - 「サイト・プロファイル」を参照
- ユーザー変数, 6-16
  - 1つの定義の表示, 6-17, 13-48
  - ACCEPT コマンド, 6-26, 13-10
  - 削除, 6-17, 13-141
  - すべての定義の表示, 6-17, 13-48
  - タイトルの中に表示, 13-138
  - 定義, 6-16, 13-48
  - ヘッダーおよびフッターの中に表示, 13-86
- ユーザー名, 4-17
  - CONNECT コマンド, 4-17, 4-18, 13-44
  - COPY コマンド, B-5, B-7, B-9
  - SQLPLUS コマンド, 4-2, 4-10, 4-21
  - インストール時に作成, 4-17
  - 異なるユーザー名での接続, 4-17, 4-18, 13-44
- ユーザー名 / パスワード
  - 認証エントリの追加, 10-12

## ら

---

- ラベル
  - COMPUTE コマンド, 7-16, 13-40

## り

---

- リカバリ
  - RECOVER コマンド, 13-77
  - リターン・コード、指定, 6-16, 13-63, 13-151
  - リモート・データベース
    - 接続識別子, 2-5, 2-11, 4-21

## れ

---

- 例
  - 埋め込まれた CGI レポート, 8-7
  - 対話型 HTML レポート, 8-3, 8-5
  - レコード・セパレータ、出力, 7-10, 13-95, 13-111
  - 列
    - 1つの列の表示属性の表示, 7-9, 13-30

- CHAR、VARCHAR、LONG および DATE の書式設定, 13-31
- MLSLABEL、RAW MLSLABEL、ROWLABEL の書式設定, 13-31
- NUMBER の書式設定, 7-4, 13-32
- 値がオーバーフローした場合の全列値の折返し表示, 7-7, 13-119
- 値がオーバーフローした場合の全列値の切捨て表示, 7-7, 13-119
- 値がオーバーフローした場合の列値の折返し表示, 7-7, 13-35
- 値がオーバーフローした場合の列値の切捨て表示, 7-7, 13-35
- 新しい行の開始, 13-34
- オーバーフローした値の後の行の出力, 7-10, 13-95, 13-111
- 下部タイトルへの値の表示, 7-29, 13-35
- コピー時の宛先の表での名前, B-3, B-6
- サマリー行の計算, 7-15, 13-39
- 出力を OFF または ON に設定, 7-28, 13-34
- 上部タイトルへの値の表示, 7-28, 13-34
- すべての単語を 1 列に折返し, 7-10
- すべての列の表示属性の表示, 7-9, 13-30
- すべての列の表示をデフォルトにリセット, 13-27
- 表間での値のコピー, B-2, B-5, B-9
- 表示属性のコピー, 7-8, 13-17, 13-34
- 表示属性の非表示, 7-9, 13-17, 13-35
- 表示属性の表示, 7-9, 13-17, 13-35
- 別名の割当て, 13-30
- 変数への値の格納, 7-28, 13-34
- 列の表示をデフォルトにリセット, 7-9, 13-30, C-2
- レポート内での書式設定, 7-2, 13-29
- 列セパレータ, 13-93, 13-101, C-2
- 列ヘッダー
  - CHAR および LONG 列で切り捨てられる場合, 7-6
  - DATE 列で切り捨てられる場合, 7-6
  - NUMBER 列で切り捨てられる場合, 7-4
  - 下線に使用する文字の変更, 13-96, 13-119
  - 切り捨てられる場合, 13-31
  - 整列, 13-34
  - 複数行への表示, 7-3, 13-33
  - 複数の単語に変更, 7-3, 13-33
  - 変更, 7-2, 13-33
  - レポートに出力しない, 13-94, 13-106
- レポート
  - SILENT モード, 8-11
  - URL からの実行, 8-12
  - Web 上, 8-2
  - 新しいページでの開始, 13-104
  - 埋め込まれた CGI の例, 8-7
  - 下部タイトルの作成, 7-22, 13-23, C-2
  - 間隔の設定およびサマリー行による明確化, 7-11
  - 自動トレース, 9-2
  - 上部タイトルの作成, 7-22, 13-138, C-2
  - タイトル, 13-138, C-2
  - 対話型 HTML の例, 8-3, 8-5
  - 動的作成, 8-12
  - 表示, 13-93, 13-99
  - フッターの作成, 13-84
  - ブレイク, 13-18
  - ヘッダーおよびフッターの作成, 7-22
  - ヘッダーの作成, 13-86
  - マスター / デティールの作成, 7-28, 13-34, 13-35
  - 列, 13-30
  - 列の書式設定, 7-4, 7-6, 13-29
  - 列ヘッダーの書式設定, 7-2, 13-29

## ろ

---

- ルール, 10-6
  - 再度使用可能, 10-7
  - 使用禁止, 10-7
- ログアウト, 4-14
- ログイン
  - Oracle, 4-21, 13-44
  - SQL\*Plus, 4-2
  - インストール時のデフォルト・ユーザー作成, 4-17
- ログオフ
  - Oracle, 4-17, 4-18, 13-58
  - SQL\*Plus, 4-13, 13-62
  - 条件付き, 13-148, 13-150

