

# SQL\*Plus

ユーザーズ・ガイドおよびリファレンス

リリース 9.0.1

2001 年 10 月

部品番号 : J04131-01

ORACLE®

---

SQL\*Plus ユーザーズ・ガイドおよびリファレンス, リリース 9.0.1

部品番号: J04131-01

原本名: SQL\*Plus User's Guide and Reference, Release 9.0.1

原本部品番号: A88827-02

原本著者: Simon Watt

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

Copyright © 1996, 2001, 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述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

はじめに .....	ix
対象読者 .....	x
構成 .....	x
関連マニュアル .....	xii
表記規則 .....	xiii

## 第 I 部 SQL\*Plus について

### 1 概要

SQL*Plus の概要 .....	1-2
基本概念 .....	1-2
SQL*Plus の対象ユーザー .....	1-3
このマニュアルの利用方法 .....	1-3
サンプル表 .....	1-3
SQL*Plus の実行に必要なもの .....	1-4
ハードウェアおよびソフトウェア .....	1-4
オペレーティング・システムに固有の情報 .....	1-5
ユーザー名およびパスワード .....	1-5
サンプル表へのアクセス .....	1-6

### 2 SQL\*Plus の基本

SQL*Plus の起動 .....	2-2
SQL*Plus の終了 .....	2-3
コマンドの入力と実行 .....	2-4

SQL コマンドの実行 .....	2-6
PL/SQL ブロックの実行 .....	2-9
SQL*Plus コマンドの実行 .....	2-11
コマンドの実行に影響する変数 .....	2-12
データベースへの変更の自動保存 .....	2-13
実行中のコマンドの停止 .....	2-14
実行したコマンドのタイミング統計の収集 .....	2-15
ホスト・オペレーティング・システムのコマンドの実行 .....	2-15
<b>役立つ情報</b> .....	2-16
表定義のリスト表示 .....	2-16
PL/SQL 定義のリスト表示 .....	2-17
表示の制御 .....	2-17
エラー・メッセージの解釈 .....	2-17

### 3 コマンド操作

<b>編集コマンド</b> .....	3-2
バッファ内容のリスト表示 .....	3-3
カレント行の編集 .....	3-3
新しい行の追加 .....	3-5
行へのテキストの追加 .....	3-6
行の削除 .....	3-7
システム・エディタを使用したコマンドの編集 .....	3-7
<b>後で使用するコマンドの保存</b> .....	3-8
コマンド・ファイルへのコマンドの格納 .....	3-8
コマンド・ファイルへのコメントの挿入 .....	3-12
コマンド・ファイルの取出し .....	3-16
コマンド・ファイルの実行 .....	3-17
コマンド・ファイルのネスト .....	3-18
コマンド・ファイルの変更 .....	3-19
リターン・コードを伴ったコマンド・ファイルの終了 .....	3-20
SQL*Plus 環境の設定 .....	3-20
SQL*Plus システム変数の格納および復元 .....	3-21
<b>対話型コマンドの作成</b> .....	3-22
ユーザー変数の定義 .....	3-22
置換変数の使用 .....	3-23

START コマンドを使用した値の指定 .....	3-29
ユーザーとの対話 .....	3-30
<b>バインド変数の使用方法</b> .....	<b>3-34</b>
バインド変数の作成 .....	3-34
バインド変数の参照 .....	3-34
バインド変数の表示 .....	3-35
REFCURSOR バインド変数の使用方法 .....	3-35
<b>文のトレース</b> .....	<b>3-39</b>
レポートの制御 .....	3-39
実行計画 .....	3-40
統計 .....	3-41
パラレル問合せおよび分散問合せのトレース .....	3-44

## 4 問合せ結果の書式の設定

<b>列の書式設定</b> .....	<b>4-2</b>
列ヘッダーの変更 .....	4-2
NUMBER 列の書式設定 .....	4-4
データ型の書式設定 .....	4-6
列の表示属性のコピー .....	4-8
列の表示属性のリスト表示およびリセット .....	4-9
列の表示属性の抑止および復元 .....	4-9
折り返した列の値の後に文字の行を出力する方法 .....	4-10
<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>4-11</b>
ブレイク列内の値の重複の抑止 .....	4-12
ブレイク列の値が変更されたときの空白の挿入 .....	4-13
各行の後に空白を挿入する方法 .....	4-14
複数の間隔設定技法の使用法 .....	4-14
ブレイク定義のリスト表示および取消し .....	4-16
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 .....	4-16
レポートの終わりでサマリー行の計算 .....	4-20
複数のサマリー値およびサマリー行の計算 .....	4-21
COMPUTE 定義のリスト表示および取消し .....	4-22
<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>4-23</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 .....	4-23
タイトルの中にページ番号およびその他のシステム管理値を表示する方法 .....	4-28

ページ・タイトル定義のリスト表示、抑止、復元 .....	4-29
タイトル内に列の値を表示する方法 .....	4-30
タイトル内に現在の日付を表示する方法 .....	4-31
ページ・サイズの設定 .....	4-32
<b>問合せ結果の格納および印刷</b> .....	4-34
ファイルへの結果の格納 .....	4-35
プリンタへの結果の出力 .....	4-35
<b>Web レポートの作成</b> .....	4-38
静的 Web レポートの作成 .....	4-38
CGI スクリプトを使用した動的 Web レポートの作成 .....	4-43
Web レポートでの SQL*Plus コマンドの表示の抑止 .....	4-49
HTML エンティティ .....	4-49

## 5 データベース管理

<b>概要</b> .....	5-2
<b>データベースの起動および停止の概要</b> .....	5-2
データベースの起動 .....	5-2
データベースの停止 .....	5-3
<b>REDO ログ・ファイル</b> .....	5-4
ARCHIVELOG モード .....	5-4
データベースのリカバリ .....	5-5

## 6 SQL\*Plus を使用したデータベースへの接続

<b>デフォルト・データベースへの接続</b> .....	6-2
<b>リモート・データベースへの接続</b> .....	6-3
SQL*Plus からのリモート・データベースへの接続 .....	6-3
SQL*Plus 起動時のリモート・データベースへの接続 .....	6-4
<b>データベース間でのデータのコピー</b> .....	6-4
COPY コマンド構文について .....	6-5
宛先の表の扱いを制御する方法 .....	6-6
COPY で表示されるメッセージの解釈 .....	6-8
別ユーザーの表の指定 .....	6-8
<b>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> .....	6-9

## 第 II 部 リファレンス

### 7 SQL\*Plus の起動およびヘルプの表示

SQLPLUS コマンドでの SQL*Plus の起動 .....	7-2
Option .....	7-2
Logon .....	7-9
Start .....	7-10
サイト・プロファイルの設定 .....	7-10
ユーザー・プロファイルの設定 .....	7-10
リターン・コードの受信 .....	7-11
SQL*Plus コマンドのヘルプ .....	7-12

### 8 コマンド・リファレンス

SQL*Plus コマンド一覧 .....	8-2
@ (アットマーク) .....	8-5
@@ (二重アットマーク) .....	8-7
/ (スラッシュ) .....	8-9
ACCEPT .....	8-10
APPEND .....	8-12
ARCHIVE LOG .....	8-13
ATTRIBUTE .....	8-16
BREAK .....	8-18
BTITLE .....	8-23
CHANGE .....	8-24
CLEAR .....	8-27
COLUMN .....	8-29
COMPUTE .....	8-39
CONNECT .....	8-44
COPY .....	8-46
DEFINE .....	8-49
DEL .....	8-51
DESCRIBE .....	8-53
DISCONNECT .....	8-59
EDIT .....	8-60
EXECUTE .....	8-62
EXIT .....	8-63

GET .....	8-65
HELP .....	8-66
HOST .....	8-67
INPUT .....	8-69
LIST .....	8-71
PASSWORD .....	8-73
PAUSE .....	8-74
PRINT .....	8-75
PROMPT .....	8-76
RECOVER .....	8-77
REMARK .....	8-83
REPFOOTER .....	8-84
REPHEADER .....	8-86
RUN .....	8-90
SAVE .....	8-91
SET .....	8-93
SHOW .....	8-118
SHUTDOWN .....	8-123
SPOOL .....	8-125
START .....	8-126
STARTUP .....	8-128
STORE .....	8-131
TIMING .....	8-132
TTITLE .....	8-134
UNDEFINE .....	8-137
VARIABLE .....	8-138
WHENEVER OSERROR .....	8-145
WHENEVER SQLERROR .....	8-147

## **A SQL\*Plus のエラー・メッセージ**

SQL*Plus のエラー・メッセージおよびコード .....	A-2
COPY コマンドのメッセージ .....	A-32

## **B リリース 9.0.1 での拡張機能**

SQL*Plus リリース 9.0.1 での拡張機能 .....	B-2
----------------------------------	-----



## C SQL\*Plus の制限

## D SQL コマンド・リスト

## E セキュリティ

PRODUCT_USER_PROFILE 表 .....	E-2
PUP 表の作成 .....	E-2
PUP 表の構造 .....	E-3
PUP 列の説明および使用方法 .....	E-3
PUP 表の管理 .....	E-4
SQL*Plus、SQL および PL/SQL コマンドの使用禁止 .....	E-4
ロールの作成および制御 .....	E-7
SET ROLE の使用禁止 .....	E-7
ユーザー・ロールの使用禁止 .....	E-8
SQLPLUS -RESTRICT でのコマンドの使用禁止 .....	E-9

## F 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 .....	F-2
BTITLE (旧形式) .....	F-3
COLUMN DEFAULT .....	F-3
DOCUMENT .....	F-3
NEWPAGE .....	F-4
SET BUFFER .....	F-4
SET CLOSECURSOR .....	F-5
SET DOCUMENT .....	F-5
SET MAXDATA .....	F-6
SET SCAN .....	F-6
SET SPACE .....	F-6
SET TRUNCATE .....	F-7
SHOW LABEL .....	F-7
TTITLE (旧形式) .....	F-7

## 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](#)
- [構成](#)
- [関連マニュアル](#)
- [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・エンド・ユーザー、テクニカル・エンド・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、取出しおよび実行
- 問合せ結果の書式設定、計算の実行、格納、印刷および Web への表示
- 表の列定義のリスト表示
- 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 データベース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。

# 構成

このマニュアルは、次のように構成されています。

## 第 1 部 SQL\*Plus について

SQL\*Plus ユーザーズ・ガイドおよびチュートリアル・コンテンツを含みます。

### 第 1 章「概要」

SQL\*Plus の概要、このマニュアルの使用法および SQL\*Plus を実行するための情報について説明します。

### 第 2 章「SQL\*Plus の基本」

SQL\*Plus の起動方法および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 3 章「コマンド操作」

コマンドの編集方法、コマンドを後で使用するために保存する方法、対話型コマンドの作成方法を、例を使用して説明します。

### 第 4 章「問合せ結果の書式の設定」

列を書式設定する方法、間隔を設定し、サマリー行を使用してレポートを見やすくする方法、ページのサイズおよびタイトルを定義する方法、問合せ結果の格納および出力方法、および問合せ結果を Web に出力する方法について例を使用して説明します。

## 第5章「データベース管理」

データベース管理者 (DBA) を対象としています。SQL\*Plus での基本データベース管理機能について説明します。

## 第6章「SQL\*Plus を使用したデータベースへの接続」

デフォルト・データベースおよびリモート・データベースに接続する方法、データベース間および同じデータベースでの2つの表の間でデータをコピーする方法について説明します。

## 第II部 リファレンス

SQL\*Plus コマンド・リファレンスおよび付録が含まれます。

## 第7章「SQL\*Plus の起動およびヘルプの表示」

オペレーティング・システムのプロンプトから SQL\*Plus にアクセスする方法、およびオンライン・ヘルプにアクセスする方法について説明します。

## 第8章「コマンド・リファレンス」

SQL\*Plus コマンドの概要およびそれぞれの SQL\*Plus コマンドの詳細をアルファベット順で説明します。

## 付録 A 「SQL\*Plus のエラー・メッセージ」

COPY コマンドのエラー・メッセージを含む SQL\*Plus が生成するメッセージについて説明します。原因およびエラーからリカバリするための適切な処置を説明します。

## 付録 B 「リリース 9.0.1 での拡張機能」

このリリースの新機能および拡張機能を説明します。

## 付録 C 「SQL\*Plus の制限」

SQL\*Plus の要素の最大値を示します。

## 付録 D 「SQL コマンド・リスト」

主要な SQL コマンドおよび句を示します。

## 付録 E 「セキュリティ」

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。

## 付録 F 「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。

## 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。

## 関連マニュアル

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』
- 『iSQL\*Plus ユーザーズ・ガイドおよびリファレンス』
- 『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 Heterogeneous Connectivity 管理者ガイド』
- 『Oracle9i 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・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書籍は、Oracle Technology Network (OTN) に接続すれば、無償でダウンロードできます。OTN を利用するには、オンライン登録をする必要があります。次の URL で登録できます。

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

すでに OTN のユーザー名およびパスワードを所有している場合は、OTN の Web サイトから無償でドキュメントを参照およびダウンロードすることが可能です。

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

ホワイト・ペーパー、サンプル・コード、FAQ およびその他の役立つ情報は、OTN の次の URL の SQL\*Plus セクションに定期的に掲載されます。

<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

### 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。
大文字の固定幅フォント	大文字の固定幅フォントは、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、RMAN 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。

規則	意味	例
小文字の固定幅 フォント	<p>小文字の固定幅フォントは、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。</p> <p><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。</p>	<p>sqlplus と入力して、SQL*Plus をオープンします。</p> <p>パスワードは、orapwd ファイルで指定します。</p> <p>/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。</p> <p>hr.departments 表には、department_id、department_name および location_id 列があります。</p> <p>QUERY_REWRITE_ENABLED 初期化パラメータを true に設定します。</p> <p>oe ユーザーとして接続します。</p> <p>JRepUtil クラスが次のメソッドを実装します。</p>
小文字の固定幅 イタリック体	<p>小文字の固定幅イタリック体は、プレースホルダまたは変数を示します。</p>	<p>managed_clause を指定できます。</p> <p>old_release.SQL を実行します。ここで、old_release とはアップグレード前にインストールしたりリースを示します。</p>

## コード例中の表記規則

コード例は、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。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す。



```
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。オプションのうちの1つを入力します。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。	<i>acctbal</i> NUMBER(11,2); <i>acct</i> 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、これらの用語は大文字/小文字が識別されないため、小文字でも入力できます。	SELECT <i>last_name</i> , <i>employee_id</i> 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、列名または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 <i>last_name</i> , <i>employee_id</i> FROM <i>employees</i> ;  sqlplus hr/hr  CREATE USER <i>mjones</i> IDENTIFIED BY <i>ty3MU9</i> ;



# 第 I 部

---

## SQL\*Plus について

この部では、SQL\*Plus の概要について説明します。SQL\*Plus の実行方法の概要および様々な例を使用した SQL\*Plus の使用方法を説明します。

この部では、次の章について説明します。

- 概要
- SQL\*Plus の基本
- コマンド操作
- 問合せ結果の書式の設定
- データベース管理
- SQL\*Plus を使用したデータベースへの接続



# 1

---

## 概要

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

- [SQL\\*Plus の概要](#)
- [このマニュアルの利用方法](#)
- [SQL\\*Plus の実行に必要なもの](#)

## SQL\*Plus の概要

SQL\*Plus プログラムは、SQL データベース言語およびその手続き型言語拡張機能である PL/SQL と一緒に使用できます。SQL データベース言語を使用すると、Oracle 内のデータの格納および取出しができます。PL/SQL を使用すると、手続き型ロジックを通して複数の SQL コマンドへリンクできます。

SQL\*Plus を使用すると、SQL コマンドおよび PL/SQL ブロックを実行できる他、多数の作業ができます。SQL\*Plus を通して実行可能な作業は、次のとおりです。

- SQL コマンドおよび PL/SQL ブロックの入力、編集、格納、取出しおよび実行
- 問合せ結果の書式設定、計算の実行、格納、印刷および Web への表示
- 表の列定義のリスト表示
- SQL 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
- 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

## 基本概念

次に、SQL\*Plus の基本概念の定義を説明します。

コマンド	ユーザーが SQL*Plus または Oracle に与える命令
ブロック	手続き型ロジックによって相互に関連付けられている PL/SQL コマンドのグループ
表	Oracle での記憶域の基本単位
問合せ	1 つ以上の表から情報を取り出す SQL SELECT コマンド
問合せ結果	問合せによって取り出したデータ
レポート	ユーザーが SQL*Plus コマンドによって書式を設定した問合せ結果

## 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、「Last 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 'Last Name'
```

同様に、EMPLOYEES という表の列定義をリスト表示するには、次のコマンドを入力します。

```
DESCRIBE EMPLOYEES
```

## このマニュアルの利用方法

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。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、このマニュアルと併用してください。

コマンドの入力方法を示す例では、このマニュアルの全編を通じて共通のコマンド構文および共通のサンプル表セットを使用しています。表については、次に説明します。

「はじめに」の「コード例中の表記規則」は、このマニュアルの第 7 章および第 8 章のコマンドを参照する場合に特に役立ちます。

## サンプル表

Oracle9i には、多くのサンプル・スキーマが含まれます。このマニュアルのチュートリアルおよび例は、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。

---

---

**注意：** 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 関数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)

## SQL\*Plus の実行に必要なもの

SQL\*Plus を実行するには、ハードウェア、ソフトウェア、オペレーティング・システム固有の情報、ユーザー名とパスワードおよび 1 つ以上の表へのアクセス権限が必要です。

### ハードウェアおよびソフトウェア

Oracle および SQL\*Plus は多数の異なる種類のコンピュータ上で実行できます。コンピュータのリソースを管理し、ハードウェアと SQL\*Plus などのプログラム間を調整するのは、オペレーティング・システムです。コンピュータの種類によって、使用されているオペレーティング・システムも様々です。ご使用のオペレーティング・システムの詳細は、コンピュータに添付されているドキュメントを参照してください。

SQL\*Plus を使用するには、あらかじめ Oracle および SQL\*Plus の両方をインストールしておく必要があります。

複数のユーザーがコンピュータを共有している組織では、Oracle の使用を管理するデータベース管理者 (DBA) が必要です。



DBA は、Oracle および SQL\*Plus をシステムにインストールする責任を負います。DBA になった場合、Oracle および SQL\*Plus のインストール方法については、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。

## オペレーティング・システムに固有の情報

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・システムの種類によって異なる点があります。そのような点については、SQL\*Plus がサポートしているホスト・コンピュータおよびオペレーティング・システム固有に出荷されている 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ください。

## ユーザー名およびパスワード

SQL\*Plus を起動する場合は、許可を受けた Oracle ユーザーであることを識別す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になります。Oracle9i のインストール中にデフォルト・ユーザーが作成され、メッセージが表示されます。作成されるデフォルト・ユーザーのユーザー名 / パスワードの組合せは、次のとおりです。

- SYS/CHANGE\_ON\_INSTALL
- SYSTEM/MANAGER
- HR/<YOUR\_SECRET\_PASSWORD>

デフォルト・パスワードは、できるかぎり早く変更する必要があります。パスワードの変更方法の詳細は、第 8 章の「PASSWORD」コマンドを参照してください。

デフォルト・ユーザーの詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

### マルチユーザー・システム

各ユーザーは、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を所有する必要があります。それらは、SQL\*Plus に使用するものと同じでも別でもかまいません。

### 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。

## サンプル表へのアクセス

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 SQL\*Plus にログインして、次のコマンドを入力します。



```
ALTER USER HR IDENTIFIED BY HR 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』を参照します。また、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 HR 表を作成できます。

```
$ORACLE_HOME/DEMO/SCHEMA/HUMAN_RESOURCES/HR_MAIN.SQL
```

サンプル表が不要になった場合は、Oracle の HR\_DROP.SQL というコマンド・ファイルを実行して、サンプル表を削除してください。これらのコマンド・ファイルの実行方法の詳細は、ご使用のオペレーティング・システム固有の Oracle インストレーション・ガイド、管理者リファレンスおよび『Oracle9i サンプル・スキーマ』を参照してください。

# 2

---

## SQL\*Plus の基本

この章では、SQL\*Plus の基本的な使用方法を説明します。この章では、次の項目について説明します。

- [SQL\\*Plus の起動](#)
- [SQL\\*Plus の終了](#)
- [コマンドの入力と実行](#)
- [役立つ情報](#)

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「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。

## SQL\*Plus 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 2-1 SQL\*Plus の起動

ここでは、SQL\*Plus の起動方法を示します。次の手順に従ってください。

1. コンピュータ上に SQL\*Plus がインストールされていることを確認します。
2. コンピュータの電源を（OFF の場合）ON にし、（必要な場合は）ホスト・オペレーティング・システムにログインします。すでにコンピュータを使用している場合は、ログオフやリセットの必要はありません。使用中のプログラムがあれば、終了します。

画面左側に 1 文字以上の文字が表示されます。これはオペレーティング・システムのコマンド・プロンプトで、オペレーティング・システムにコマンドを入力できる状態であることを示します。このマニュアルでは、オペレーティング・システムのプロンプトをドル（\$）記号で表記します。ご使用のコンピュータのオペレーティング・システムのプロンプトは、これと異なる場合があります。

3. SQLPLUS コマンドを入力し、[Return] を押します。これは SQL\*Plus を起動するオペレーティング・システム・コマンドです。

---

---

**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てください。

---

---



```
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 9.0.1.0.0 - Production on Thu June 14 16:29:01 2001  
(c) Copyright 1996, 2001 Oracle Corporation. All rights reserved.  
ユーザー名を入力してください:
```



4. ユーザー名を入力して [Return] を押します。「パスワードを入力してください」という旨の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。

ユーザー名と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 のリリースと、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 にコマンドを入力できる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 (  ) が表示される場合は、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 (  ) が表示される場合は、画面に表示されることを示します。

SQL\*Plus が起動しない場合、何が問題になっているかを解決するためのメッセージが表示されます。

## SQL\*Plus を起動するためのショートカット

SQL\*Plus を起動するとき、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、ユーザー名が HR でパスワードが HR の場合、



```
SQLPLUS HR/HR
```

と入力し、[Return] を押します。ホスト・オペレーティング・システムにログインするとき、自動的に SQL\*Plus にログインするような設定もできます。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。

## SQL\*Plus の終了

SQL\*Plus での作業が終了し、オペレーティング・システムに戻るときは、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 2-2 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 のリリース番号と、SQL\*Plus から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 2-1 の手順 3、4 および 5 に従って、再度 SQL\*Plus を起動してください。または、前述の「SQL\*Plus を起動するためのショートカット」で説明したショートカットを使用してログインしてください。

## コマンドの入力と実行

### コマンドの入力

コンピュータのカーソル（通常は、下線、長方形ブロックまたはスラッシュのポインタ）がコマンド・プロンプトの後に表示されます。このカーソルは、次に入力する文字が表示される画面上の位置を示しています。

SQL\*Plus に処理を指示するには、その処理に応じたコマンドをそのまま入力します。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。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。

---

---

**注意：** 文字間隔やインデントの例は、このマニュアルの全編にわたって示してあります。例にあるコマンドを入力する場合、空白はそのとおりに入れなくてもかまいませんが、入れた方が読みやすくなります。

---

---

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・プロンプトでは、次の 3 種類のコマンドを入力できます。

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

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

### 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、第 3 章の「コマンド・ファイルの実行」を参照してください。

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

---

---

**注意：** SQL\*Plus コマンドは、SQL バッファに格納されません。

---

---

## ヘルプの入手

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・システムに対応できる十分な大きさである。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・ファイルは、次のとおりです。

- HLPBLD.SQL。新規のヘルプ表を削除および作成する。
- HELPUS.SQL。ヘルプ表をヘルプ・データに移入する。
- HELPDROP.SQL。既存の SQL\*Plus ヘルプ表を削除する。

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/PASSWORD
```

PASSWORD には、SYSTEM ユーザーに対して定義したパスワード（デフォルトでは、MANAGER）を入力します。

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 コマンドに関するオンライン・ヘルプを表示するには、次の例で示すとおり、コマンド・プロンプトで HELP と入力し、その後にコマンドの名前を入力します。



```
HELP ACCEPT
```

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。ヘルプ・システムの詳細は、第7章の「SQL\*Plus コマンドのヘルプ」および第8章の「HELP」コマンドを参照してください。

## コマンドの実行

コマンドを入力して、SQL\*Plus にそのコマンドを実行するように指示すると、SQL\*Plus はそのコマンドを処理し、別のコマンドを入力できるようにコマンド・プロンプトを再表示します。

## 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 2-3 SQL コマンドの入力

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

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。入力が終わったら、[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 はコマンドを処理し、結果を次のように画面に表示します。



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 行が選択されました。

結果と取り出した行数が表示された後、再びコマンド・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します。詳細は、「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。取り出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、この章の後半にある「コマンドの実行に影響する変数」で説明します。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、単語を組み合わせる文にする方法を規定した構文規則があるように、SQL\*Plus にも、単語を組み合わせるコマンドにする方法を規定した構文規則があります。SQL\*Plus にコマンドを渡して実行させるためには、それらの規則に従う必要があります。

**1 つの SQL コマンドを別個の行に分割** SQL コマンドは、個々の単語を途中で改行するの  
なければ、任意の位置で別々の行に分割できます。したがって、例 2-3 で入力した問合せは、  
次のように 3 行に入力できます。



```
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  
なっています。たとえば、例 2-3 では、SELECT 句と FROM 句が別々の行に置かれていま  
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  
2-3 に示すように、コマンドの最終行の末尾にセミコロンを入力して [Return] を押します。  
SQL\*Plus は、コマンドを処理して SQL バッファに格納します (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  
ロンを入力して再び [Return] を押し、コマンドを実行します。

---

---

**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、第3章の「[コマンド・ファイル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1行を使用することによっても、コマンドの実行を SQL\*Plus に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、再び [Return] を押します。SQL\*Plus は、コマンドを実行してバッファに格納します（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の空白行は、コマンドの入力は終了したが、そのコマンドをまだ実行しないことを SQL\*Plus に伝えます。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。空白行の動作の詳細は、第8章の「[SET](#)」コマンドを参照してください。

---

---

再び [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。SQL\*Plus は、コマンドを実行せずに SQL バッファに格納します（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、この章の「[PL/SQL ブロックの実行](#)」を参照してください。PL/SQL サブプログラムを入力し終わったら、PL/SQL モードを終了するため、1 行にピリオド (.) のみを入力してください。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。

ストアド・プロシージャを作成するために CREATE を使用する場合、コンパイル・エラーがあるとメッセージが表示されます。それらのエラーを表示するには、SHOW ERRORS を使用します。次に例を示します。



```
SHOW ERRORS PROCEDURE ASSIGNVL
```

SHOW コマンドの詳細は、[第 8 章「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。次に例を示します。



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

EXECUTE コマンドの詳細は、[第 8 章「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・プロンプトから、RUN コマンドまたは / (スラッシュ) コマンドを入力することにより、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 コマンドでは、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 ブロックのリストが表示されます。/ (スラッシュ) コマンドは、SQL コマンドまたは PL/SQL ブロックをそのまま実行します。

## 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 モードに入るには、次のようにします。

- 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、PL/SQL サブプログラムは SQL コマンドと同じ方法で処理されますが、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。PL/SQL サブプログラムを終了および実行するには、新しい行にスラッシュ (/) のみを入力します。

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。それらの SQL コマンドは、他の SQL コマンドのようにセミコロン (;) で実行することはできません。

SQL\*Plus は、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を Oracle へ送信します。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;
```

.

/

サブプログラムを実行したとき、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する場合があります。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 コマンドと 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することができます。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扱いを受けます。個々の SQL\*Plus コマンドの詳細は、[第 8 章「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、多数の SQL\*Plus コマンドは、1 文字または数文字に省略できます。一部の SQL\*Plus コマンドの略称は、コマンドの説明とともに[第 3 章](#)、[第 4 章](#)および[第 6 章](#)に記載されています。すべての SQL\*Plus コマンドの略称については、[第 8 章「コマンド・リファレンス」](#)を参照してください。

### 例 2-4 SQL\*Plus コマンドの入力

この例では、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ために、SQL\*Plus コマンドを入力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。この行の入力が終わったら、[Return] を押します。SQL\*Plus は新しい書式を認識し、SQL\*Plus コマンド・プロンプトを再び表示して新しいコマンドを入力できる状態になります。

2. 最新の (例 2-3 からの) 問合せを再実行するため、RUN コマンドを入力します。



```
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 行が選択されました。

COLUMN コマンドによって、SALARY 列がドル記号 (\$) とカンマ (,) 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した。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 2-3 の問合せが再実行されました。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 コマンドを追加行に継続する方法** 長文の 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 (>) が個々の追加行のプロンプトとして表示されます。

次に例を示します。



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

SQL\*Plus では、ハイフンを継続文字とみなすため、SQL 文内でハイフンを入力しても無視されます。SQL\*Plus 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、その文を SQL 文とはみなしません。たとえば、次のように入力します。



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

次のエラーが返されます。



```
SELECT 200 100 FROM DUAL  
          *
```

1行でエラーが発生しました。  
ORA-00923: FROM キーワードが指定の位置にありません。

文を正しく解釈させるには、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。

## コマンドの実行に影響する変数

SQL\*Plus の SET コマンドを使用して、システム変数と呼ばれる多数の変数を指定できます。システム変数の設定は、SQL\*Plus のコマンド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の有無、ページ・サイズなど、SQL\*Plus 内の様々な条件を指定します。システム変数は、SET コマンド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、記載されている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（コンピュータ上に SQL\*Plus の LOGIN ファイルがある場合は、デフォルトの設定と異なる可能性があります）。

システム変数とそれらのデフォルト設定の詳細は、第 8 章の「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、第 3 章の「後で使用されるコマンドの保存」の「SQL\*Plus 環境の設定」および第 7 章の 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リスト表示するには、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 を使用してリスト表示できるその他の項目の詳細は、第 8 章の「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 コマンドである UPDATE、INSERT および DELETE は、独立して使用することも、PL/SQL ブロック内で使用することもできます。これらのコマンドで、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、あるいは自動コミット機能を使用するまでは、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か、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 2-5 自動コミット機能を ON にする方法

自動コミット機能を ON にするには、次のように入力します。



```
SET AUTOCOMMIT ON
```

または、次のように入力して自動コミット機能を ON にすることもできます。



```
SET AUTOCOMMIT IMMEDIATE
```

AUTOCOMMIT の設定を変更するまでは、データベースの変更を指定する SQL の各 DML コマンドからの変更を、SQL\*Plus が自動的にコミットします。SQL\*Plus は、1 回自動コミットするたびに次のメッセージを表示します。



```
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 は、SQL の DML コマンドを実行するたびにカウントし、10 回目の SQL の DML コマンドを実行するたびに変更をコミットします。

---

**注意：** この機能では、ブロックに入っている実際の SQL コマンドの数には関係なく、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び 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、次のように SHOW コマンドを入力します。



```
SHOW AUTOCOMMIT
```



```
AUTOCOMMIT OFF
```

詳細は、第 8 章の「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 ページ目を表示し終わり、残りのページを表示する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。

---

**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終わっ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 4 章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・システムを通じて停止できます。詳細は、ご使用のオペレーティング・システム固有のインストレーション・ガイドおよび管理者リファレンスを参照してください。

---



## 実行したコマンドの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、第 8 章の「TIMING」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、第 3 章の「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。

## 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。これは、既存のホスト・オペレーティング・システム・ファイルのリストを表示するときなどに便利です。

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 コマンドは、ホスト・コマンドの DIRECTORY \*.SQL を実行します。



```
HOST DIRECTORY *.SQL
```

ホスト・コマンドの実行が終わると、再び 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、引き続き起動する SQL\*Plus セッションに影響を与える可能性があります。

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

---

## 役立つ情報

SQL\*Plus の使用中に表の列定義のリストを表示したり、スクロール表示を開始または停止する必要のある場合があります。また、コマンドを誤って入力したときや、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、それらの状態に対して役立つ情報を説明します。

### 表定義のリスト表示

ある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 2-6 DESCRIBE コマンドの使用

EMP\_DETAILS\_VIEW というサンプル・ビューの列定義をリスト表示するには、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にアクセスするために使用します。SQL の SELECT コマンドを使用して、データベース内のこれらの情報や他の情報にアクセスすることもできます。詳細は、『Oracle9i SQL リファレンス』を参照してください。

## PL/SQL 定義のリスト表示

関数またはプロシージャの定義をリスト表示するには、SQL\*Plus の DESCRIBE コマンドを使用します。

### 例 2-7 DESCRIBE コマンドの使用

AFUNC という関数の定義をリスト表示するには、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を表示しているとき、画面を停止して内容を詳しく調べたい場合があります。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。

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させることもできます。詳細は、[第 8 章の「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 がコマンド内にエラーを検出した場合、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、[付録 A「SQL\\*Plus のエラー・メッセージ」](#)を参照してください。

### 例 2-8 エラー・メッセージの解釈

たとえば、次のように入力して、存在しない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訂正方法がわかります。詳しい説明が必要な場合は、問題の原因と訂正方法を判断するため、次のいずれかの手順を実行してください。

- 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、このマニュアルの付録 A「SQL\*Plus のエラー・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、このマニュアルの付録 A「SQL\*Plus のエラー・メッセージ」にある「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 8 章「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 に問い合せてください。

# 3

---

## コマンド操作

この章は、SQL\*Plus コマンド、SQL コマンドおよび PL/SQL ブロックの操作方法を学習する場合に役立ちます。この章では、次の項目について説明します。

- 編集コマンド
- 後で使用するコマンドの保存
- 対話型コマンドの作成
- バインド変数の使用方法
- 文のトレース

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](#)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。

## 編集コマンド

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・プロンプトに直接入力した 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。

バッファに現在格納されている SQL コマンドまたは PL/SQL ブロックを編集するには、多数の SQL\*Plus コマンドを使用します。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 3-1 に、コマンドを再入力せずにバッファ内のコマンドを検査または変更できる SQL\*Plus コマンドを示します。

**表 3-1 SQL\*Plus 編集コマンド**

コマンド	省略	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>	行末の <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 BUFF	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> で構成される 1 行を追加します。
LIST	L	SQL バッファに入っているすべての行をリスト表示します。
LIST <i>n</i>	L <i>n</i> または <i>n</i>	行 <i>n</i> をリスト表示します。
LIST *	L *	カレント行をリスト表示します。
LIST <i>n</i> *	L <i>n</i> *	行 <i>n</i> からカレント行までをリスト表示します。
LIST LAST	L LAST	最終行をリスト表示します。

表 3-1 SQL\*Plus 編集コマンド

コマンド	省略	用途
LIST <i>m n</i>	L <i>m n</i>	ある範囲 ( <i>m</i> ~ <i>n</i> ) の行をリスト表示します。
LIST * <i>n</i>	L * <i>n</i>	カレント行から行 <i>n</i> までをリスト表示します。

これらのコマンドは、コマンドの綴りを誤って入力した場合や、入力したコマンドを変更する場合に役立ちます。

## バッファ内容のリスト表示

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リスト表示した場合、カレント行にはアスタリスクが付きます。

### 例 3-1 バッファ内容のリスト表示

現行のコマンドをリスト表示するとします。LIST コマンドで、次のように指定します (例 2-3 の手順に従った後、SQL\*Plus を EXIT 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 コマンドの最後に入力したセミコロンはリストに表示されません。このセミコロンは、入力するとき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。次のいずれかの方法で、どの行がカレント行であるかを判断できます。

- 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、バッファ内のコマンドを実行するためにスラッシュ (/) コマンドを使用しても、カレント行は影響を受けません)。
- エラー・メッセージを受け取った場合は、その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 3-2 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 を故意に誤った綴りにして、次のようにコマンドを入力してください。



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

画面には次のメッセージが表示されます。



```
SELECT EMPLOYEE_ID, LAST_NAME, JO_ID, SALARY
*
1 行でエラーが発生しました。
```

ORA-00904: 列名が無効です。

エラー・メッセージには、問合せの 1 行目に無効な列名があ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訂正できます。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、誤りを訂正します。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 コマンドは、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します。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入力して [Return] を押します。

### 例 3-3 エラーの訂正

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



```
CHANGE /JO_ID/JOB_ID
```

訂正後の行が、次のように表示されます。



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



エラーの訂正が済んだので、RUN コマンドを使用してコマンドを再び実行できます。



RUN

SQL\*Plus は問合せおよびその結果を正しく表示します。



```
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 列は、例 2-4 で指定した書式を保持しています（例 2-4 を実行した後、SQL\*Plus を終了して再び起動した場合は、列が元の書式に戻されています）。

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、第 8 章の「CHANGE」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 3-4 行の追加

例 3-3 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
```

再び次のように新しい行の入力を求めるプロンプトが表示されます。



```
5
```

これ以上入力しないことを示すために [Return] を押し、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、変更する行をリスト表示します。
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入れられます。

### 例 3-5 行へのテキストの追加

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



```
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つ入力してください）。



```
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、削除する行をリスト表示します。
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、バッファの次の行がある場合は、その行がカレント行になります。

詳細は、[第8章](#)の「DEL」コマンドを参照してください。

## システム・エディタを使用したコマンドの編集

コンピュータのホスト・オペレーティング・システムには、1つ以上のテキスト・エディタがあります。ホスト・システムのファイルを作成し編集するには、それらのエディタを使用します。テキスト・エディタには、SQL\*Plus の編集コマンドと同じ一般的な機能がありますが、編集コマンドよりも使用しやすい場合もあります。



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。

EDIT

EDIT で、バッファの内容がシステムのデフォルト・テキスト・エディタにロードされます。その後、テキスト・エディタのコマンドを使用してテキストを編集できます。テキスト・エディタに編集後のテキストを保存して終了するよう指示した場合、テキストはバッファにロードされます。

バッファの内容をデフォルト以外のテキスト・エディタにロードするには、SQL\*Plus の DEFINE コマンドを使用して、エディタの名前を保持する変数 `_EDITOR` を定義します。たとえば、EDIT で使用するエディタを EDT として定義するには、次のコマンドを入力します。

```
DEFINE _EDITOR = EDT
```

EDIT で使用するエディタをユーザーまたはサイトのプロファイルの中で定義することもできます。詳細は、この章の「SQL\*Plus 環境の設定」、第 8 章の「DEFINE」コマンドおよび「EDIT」コマンドを参照してください。

## 後で使用するコマンドの保存

SQL\*Plus を使用して、1 つ以上のコマンドをコマンド・ファイルというファイルに格納できます。コマンド・ファイルを作成した後、そのファイルの取出し、編集および実行を行うことができます。コマンド・ファイルは、コマンド、特に複雑なコマンドや PL/SQL ブロックを今後も使用できるように保存しておくために使用します。

### コマンド・ファイルへのコマンドの格納

1 つ以上の SQL コマンド、PL/SQL ブロックおよび SQL\*Plus コマンドをコマンド・ファイルに格納できます。SQL\*Plus では、次のいずれかの方法でコマンド・ファイルを作成できます。

- コマンドを入力し、バッファの内容を保存します。
- INPUT を使用してコマンドを入力した後、バッファの内容を保存します。
- EDIT を使用し、ホスト・システムのテキスト・エディタによってファイルを最初から作成します。

SQL\*Plus コマンドはバッファ内に格納されないため、SQL\*Plus コマンドを保存するには最後の 2 つの方法を使用する必要があります。

## バッファの保存によるコマンド・ファイルの作成

現行の SQL コマンドまたは PL/SQL ブロックを後で使用するために保存するには、SAVE コマンドを入力します。このコマンドの後にファイル名を続けます。

```
SAVE file_name
```

SQL\*Plus は、SQL 問合せファイルであることを示すため、ファイル名に拡張子 SQL を追加します。別の拡張子を付けてコマンドまたはブロックを保存する場合は、ファイル名の終わりにピリオドを入力し、その後に拡張子を続けてください。

---

---

**注意：** SQL\*Plus で保存されたファイルの拡張子には、デフォルトで sql が付きます。拡張子を指定するには、SQL\*Plus の SET SUFFIX コマンドを使用します。

---

---

SQL\*Plus では、拡張子とファイル名をピリオドで区切ります。ご使用のオペレーティング・システムによっては、ファイル名と拡張子の区切りに別の文字または空白を使用する場合があります。

### 例 3-6 現行コマンドの保存

最初に、LIST と入力します。

```
LIST
```

次のように、バッファ内の現行のコマンドがリスト表示されます。

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

示されている問合せがバッファ内に存在しない場合は、この時点で問合せを再入力します。次に、SAVE コマンドとそれに続けてファイル名 EMPLINFO を入力します。

```
SAVE EMPLINFO
```

ファイル EMPLINFO.sql が作成されました。

次のように SQL\*Plus の HOST コマンドを入力し、それに続けてホスト・オペレーティング・システムでのファイルのリストに使用するコマンドを入力することによって、コマンド・ファイル EMPLINFO.SQL が存在するかどうかを確認できます。

```
HOST your_host's_file_listing_command
```

同じ方法で、バッファ内に現在格納されている PL/SQL ブロックを保存できます。



## INPUT および SAVE を使用したコマンド・ファイルの作成

コマンドの入力に INPUT を使用すると、SQL\*Plus コマンドを（1つ以上の SQL コマンドまたは PL/SQL ブロックとともに）バッファに入れることができます。それには、コマンド・プロンプトからコマンドを直接入力する場合と同様に、最初に SQL\*Plus コマンドを入力し、SQL コマンドまたは PL/SQL ブロックは最後に入力する必要があります。

コマンド・ファイルの中には、様々な問合せに使用する一連の SQL\*Plus コマンドを単独で格納することもできます。

### 例 3-7 INPUT および SAVE を使用したコマンドの保存

たとえば、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。

INPUT を使用して問合せを作成して保存するには、最初にバッファを消去する必要があります。



```
CLEAR BUFFER
```

次に、INPUT を使用してコマンドを入力します（コマンドの終わりにセミコロンを入力しないでください）。



```
INPUT
```

その後、スクリプトの各行を入力するプロンプトが表示されます。文の終わりにセミコロンを入力しないでください。入力すると、SQL\*Plus はスクリプトを正常に実行しません。SQL\*Plus は、バッファ内に SQL または PL/SQL 文のみが入っているとみなします。



```
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 列の書式モデルの 0（ゼロ）は、10 進値に対して最初の 0（ゼロ）を表示し、行の COMMISSION\_PCT の値が 0（ゼロ）の場合は、空白のかわりに 0（ゼロ）を表示することを示します。書式モデルおよび COLUMN コマンドの詳細は、[第 4 章「問合せ結果の書式の設定」](#) および『Oracle9i SQL リファレンス』を参照してください。

この時点で、次のように SAVE コマンドを使用し、拡張子 SQL が付いた SALES というファイルに問合せを格納してください。



```
SAVE SALES
```



ファイル SALES.SQL が作成されました。

問合せの終わりにセミコロンを入力しないでください。セミコロンを挿入した場合、SQL\*Plus はバッファの内容を実行しようとします。SQL\*Plus は、バッファ内に SQL コマンドのみが入っているとみなすため、バッファ内の SQL\*Plus コマンドはエラーとなります。コマンド・ファイルの実行方法については、この章の後半で説明します。

複数の SQL コマンドを入力するには、すべての SQL コマンドを、セミコロンを付けずに入力します。その後、APPEND を使用して最後のコマンドを除くすべてのコマンドにセミコロンを追加します。(SAVE は、ファイルの終わりに自動的にスラッシュを追加します。このスラッシュは、コマンド・ファイルが実行されたときに最後のコマンドを実行するよう SQL\*Plus に指示します。)

複数の PL/SQL ブロックを入力するには、ブロックとブロックの間の行にピリオドまたはスラッシュを挿入せずに、次々にブロックを入力します。その後、最後のブロックを除く各ブロックについて、そのブロックの最終行をリスト表示してカレント行にし、次の書式の INPUT を使用しスラッシュのみの 1 行を挿入します。



```
INPUT /
```

## システム・エディタを使用したコマンド・ファイルの作成

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コマンド・ファイルを作成するには、次のように EDIT を入力し、その後にファイル名を入力します。



```
EDIT SALES
```

SAVE コマンドと同様、EDIT は、ファイル名に拡張子 SQL を追加します。ただし、ユーザーがファイル名の終わりにピリオドと別の拡張子を入力した場合は除きます。テキスト・エディタを使用してコマンド・ファイルを保存する場合、そのコマンド・ファイルは、同じファイルに戻されます。

各 SQL コマンドの終わりにはセミコロンを、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 1 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)。

EDIT を使用してコマンド・ファイルを作成する場合、ファイルの終わりに SQL\*Plus コマンドを挿入することもできます。ただし、これは SAVE コマンドを使用してコマンド・ファイルを作成するときにはできません。SAVE コマンドを使用すると、ファイルの終わりにスラッシュが追加されるためです。このスラッシュがあると、SQL\*Plus はコマンド・ファイルを 2 回実行します。1 回は最後の SQL コマンドの終わりでセミコロン (または最後の PL/SQL ブロックの後にスラッシュ) を検出したとき、もう 1 回はファイルの終わりでスラッシュを検出したときです。

## コマンド・ファイルへのコメントの挿入

コマンド・ファイル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
- 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
- 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

コマンド・ファイル内のコメントの使用の詳細は、この章の「[コメント挿入時の注意](#)」を参照してください。

### REMARK コマンドの使用方法

REMARK コマンドは、コマンド・ファイル内の1行に指定し、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。1つの SQL コマンドで、行と行の間に REMARK コマンドを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、コマンド・ファイル内の独立した行に入力するか、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。入力しないと、SQL\*Plus バッファ内の任意のコマンドを実行するときに、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、SQL\*Plus はそのコメントをバッファ内に格納しません。

## -- 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。終わりのデリミタはないので、コメントを複数の行にわたって入力することはできません。

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、コメントのみを 1 行にして入力します。



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

SQL\*Plus コマンドで「--」を使用してコメントを挿入できるのは、その行全体がコメントである場合のみです。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扱われ、実行されません。



```
-- SET LONG 777
```

## コメント挿入時の注意

通常、SQL\*Plus は、コメントとして識別される入力を解析または実行しません。

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をスキャンし、SQL、PL/SQL または SQL\*Plus のいずれのコマンドであるか判断します。コメントの挿入箇所によっては、SQL\*Plus がコマンドを正しく認識できず、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. 文の先頭から数個目のキーワードには、コメントを挿入しないでください。次に例を示します。



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

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。

例に示す位置にコメントを挿入すると、SQL\*Plus は、このコマンドが PL/SQL のコマンドであることを認識できません。SQL\*Plus は、コメントの最初にあるスラッシュ (/) を文の終了記号と解釈して、ブロックをサーバーに送信します。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 /
```

プロシージャが作成されました。

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
```

次のエラーが発生します。



```
SELECT 'Y' FROM DUAL; -- TESTING
                        *
1行でエラーが発生しました。
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ず、コメントとして認識されません。

3. コメント行の終わりに、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;
```

次のエラーが戻ります。



```
-- COMMENT
*
2行でエラーが発生しました。
ORA-00923: FROM キーワードが指定の位置にありません。
```

SQL\*Plus は、セミコロンを文の終了記号として解釈し、処理のために不完全な SQL コマンドをサーバーに送信し、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行が選択されました。

SQL\*Plus は、アンパサンド (&) の後のテキストを置換変数として解釈し、変数の値を求めるプロンプトを表示します。SET DEFINE OFF を使用して、置換文字をスキャンしないようにすることができます。

置換変数および終了文字の詳細は、第 8 章の「SET」コマンドの DEFINE、SQLTERMINATOR および SQLBLANKLINES を参照してください。

## コマンド・ファイルの取出し

コマンド・ファイルの内容をバッファ内に入れる場合は、そのコマンドが格納されているファイルからコマンドを取り出す必要があります。SQL\*Plus コマンドの GET を使用し、コマンド・ファイルを取り出します。

バッファ内にある問合せを SAVE コマンドでファイルに保存するのと同様に、GET コマンドを使用してファイル内の問合せをバッファに取り出すことができます。

```
GET file_name
```

一部のオペレーティング・システムでは、SQL\*Plus によってファイル名にピリオドと拡張子 SQL が追加されます。ただし、ユーザーがファイル名の終わりにピリオドと別の拡張子を入力した場合は除きます。ファイルの接尾辞の設定の詳細は、[第8章「コマンド・リファレンス」](#)の SET SUFFIX を参照してください

### 例 3-8 コマンド・ファイルの取出し

たとえば、後のセッションで SALES ファイルを取り出す必要があるとします。ファイルは、GET コマンドを入力することによって取り出すことができます。ファイル SALES を取り出すには、次のように入力します。

```
GET SALES
```

SQL\*Plus は、ファイル SALES.SQL の内容を SQL バッファの中に取り出し、画面にリスト表示します。

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

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。SQL\*Plus コマンド、複数の文またはブロックはロードされますが、「/」または RUN を使用して実行する場合は、エラーが発生します。



## コマンド・ファイルの実行

START コマンドはコマンド・ファイルを取り出し、そこに含まれるコマンドを実行します。SQL コマンド、PL/SQL ブロックおよび SQL\*Plus コマンドを含むコマンド・ファイルを実行するには、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 3-9 コマンド・ファイルの実行

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

```
START SALES
```

SQL\*Plus はファイル SALES 内のコマンドを実行し、コマンドの結果を画面に表示します。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 がコマンドを入力す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コマンド・ファイル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、コマンド・ファイルを実行することもできます。

```
@SALES
```

指定されたコマンド・ファイル内のコマンドを、@ 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、コマンド・ファイル内の最後の 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 起動時のコマンド・ファイルの実行

SQL\*Plus を起動するときにコマンド・ファイルを実行するには、次の 4 つのオプションのいずれかを使用します。

- SQLPLUS コマンドと、それに続けてユーザー名、スラッシュ、パスワード、空白、@ およびコマンド・ファイル名を指定します。



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

SQL\*Plus が起動し、コマンド・ファイルが実行されます。

- SQLPLUS コマンドと、それに続けてユーザー名、空白、@ およびコマンド・ファイル名を指定します。



```
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、コマンド・ファイルが実行されます。

- 実行するコマンド・ファイルの 1 行目に、ユーザー名を挿入します。SQLPLUS コマンドと、それに続けて @ およびファイル名を指定します。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、コマンド・ファイルが実行されます。
- 実行するコマンド・ファイルの 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 コマンドと、それに続けて @ およびファイル名を指定します。SQL\*Plus が起動し、コマンド・ファイルが実行されます。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。

## コマンド・ファイルのネスト

一連のコマンド・ファイル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コマンド・ファイルを作成します。それぞれの START コマンドの後には、コマンド・ファイルの名前を続けます。次に、START コマンドが入ったコマンド・ファイルを実行します。たとえば、SALESRPT というコマンド・ファイルに次の 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、@@ コマンドが便利な場合があります。詳細は、[第 8 章の「@@ \(二重アットマーク\)」](#) コマンドを参照してください。

---

## コマンド・ファイルの変更

次の2つの方法で、既存のコマンド・ファイルを変更できます。

- EDIT コマンドの使用
- GET、SQL\*Plus 編集コマンドおよび SAVE の使用

既存のコマンド・ファイルを 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。

GET、SQL\*Plus 編集コマンドおよび SAVE を使用して既存のファイルを編集するには、最初に GET でファイルを取り出し、次に SQL\*Plus 編集コマンドでファイルを編集し、最後に SAVE コマンドでファイルを保存します。

既存のコマンド・ファイルの内容をバッファ内のコマンドまたはブロックで置き換える場合は、SAVE コマンドを使用し、その後にファイル名および REPLACE という単語を指定する必要があります。

次に例を示します。



```
GET MYREPORT
```



```
1* SELECT * FROM EMP
```



```
CHANGE/EMP/EMP_DETAILS_VIEW
```



```
1* SELECT * FROM EMP_DETAILS_VIEW
```



```
SAVE MYREPORT REPLACE
```



ファイル MYREPORT が書き込まれました。

既存のコマンド・ファイルの終わりにバッファの内容を追加する場合は、次のように SAVE コマンドを使用し、その後にファイル名および APPEND という単語を続けて入力します。



```
SAVE file_name APPEND
```

## リターン・コードを伴ったコマンド・ファイルの終了

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コマンド・ファイルが SQL エラーを生成した場合、そのコマンド・ファイルは異常終了し、リターン・コードを付けて終了できます。それには、SQL\*Plus コマンドの `WHENEVER SQLERROR` を使用します。詳細は、第 8 章の「[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、第 8 章の「[WHENEVER OSERROR](#)」コマンドを参照してください。

## SQL\*Plus 環境の設定

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。それには、拡張子が `SQL` の、`LOGIN` というホスト・オペレーティング・システム・ファイル（ユーザー・プロファイル）を使用します。こ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。

このファイルには、任意の SQL コマンド、PL/SQL ブロックまたは SQL\*Plus コマンドを追加できます。ユーザーが SQL\*Plus を起動すると、そのユーザーの `LOGIN` ファイルが（最初にユーザーのローカル・ディレクトリ、次にシステム依存のパスから）自動的に検索され、そこにあるコマンドが実行されます。（サイト・プロファイル、たとえば `LOGIN.SQL` の前に実行する `GLOGIN.SQL` を使用することもできます。サイト・プロファイルおよびユーザー・プロファイルの関係については、「[サイト・プロファイルの設定](#)」を参照してください。）

## LOGIN ファイルの変更

`LOGIN` ファイルは、他のコマンド・ファイルと同様に、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。

<code>SET LINESIZE</code>	数値を後に続けて、問合せ結果のページ幅の文字数を設定します。
<code>SET NUMFORMAT</code>	数値書式（\$99,999 など）を後に続けて、問合せ結果で数値を表示するためのデフォルト書式を設定します。
<code>SET PAGESIZE</code>	数値を後に続けて、1 ページごとの行数を設定します。
<code>SET PAUSE</code>	<code>ON</code> を後に続けると、SQL*Plus は出力の各ページの冒頭で SQL*Plus を一時停止させることができます（ <code>[Return]</code> を押すと、スクロールを続行します）。テキストを後に続けると、SQL*Plus が一時停止するたびに、そのテキストが表示されるように設定できます（ <code>PAUSE</code> を <code>ON</code> に設定しておく必要もあります）。
<code>SET TIME</code>	<code>ON</code> を後に続けて、それぞれのコマンド・プロンプトの前に現在の時刻を表示します。



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、第 8 章の「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復元

現行の SQL\*Plus システム (SET) 変数を、STORE コマンドでホスト・オペレーティング・システム・ファイル (コマンド・ファイル) に格納できます。変数を変更した場合、このコマンド・ファイル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値を格納するには、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てください。または、SET SUFFIX コマンドを使用し、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。

### システム変数の復元

格納済のシステム変数を復元するには、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、コマンド・ファイルを実行することもできます。

#### 例 3-10 SQL\*Plus システム変数の格納および復元

SQL\*Plus システム変数の現行の値を新規コマンド・ファイル plusenv.sql に格納するには、次のように入力します。

```
STORE SET plusenv
```

ファイル plusenv が作成されました。



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コマンド・ファイルから復元できます。



```
START plusenv  
SHOW PAGESIZE
```



```
PAGESIZE 24
```

## 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・ユーザーが入力できるコマンド・ファイルを設定できます。

- ユーザー変数の定義
- 置換変数の使用
- START コマンドを使用した値の指定
- 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、単一のコマンド・ファイルの中で繰り返し使用できます。ユーザー変数を定義して、タイトルの中で使用したり、(長い文字列を短い名前の変数の値として定義することによって) キーストロークを保存することもできます。

### 例 3-11 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リスト表示するには、コマンド・プロンプトに DEFINE を単独で入力します。DEFINE を使用して明示的に定義したユーザー変数は、CHAR の値のみになることに注意してください（ユーザーが変数に割り当てた値は、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（例 3-7 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。そのためには、コマンドを実行するたびに異なる CHAR 値を編集して WHERE 句に入れ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 は、コマンド内で置換変数を検出した場合、そのコマンドに置換変数自体ではなく置換変数の値が入っているものとしてコマンドを実行します。

たとえば、変数 SORTCOL の値が JOB\_ID で、変数 MYTABLE の値が EMP\_DETAILS\_VIEW の場合、SQL\*Plus は、

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

というコマンドを次のコマンドとして実行します。

```
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 は、コマンド内で未定義の置換変数を検出した場合、値の入力を求めるプロンプトを表示します。

このプロンプトには、任意の文字列を入力できます。空白および句読点を含む文字列でもかまいません。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合、そこに引用符が挿入されていないければ、プロンプトが表示されたときに引用符を挿入する必要があります。

SQL\*Plus は、キーボードからの応答を読み込みます。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。端末が使用不可能な場合（たとえば、バッチ・モードでコマンド・ファイルを実行した場合）、SQL\*Plus はリダイレクトされたファイルを使用します。

プロンプトで値を入力すると、SQL\*Plus は、置換変数が入っている行を 2 回表示します。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ため、置換変数に渡す値と同じ名前の置換変数を作成しないでください。置換変数に指定する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 3-12 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）を計算するために使用する STATS というコマンド・ファイルを、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
```



ファイル STATS が作成されました。

この時点で、コマンド・ファイル 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

```

SQL\*Plus は、次の出力を表示します。



```

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 行が選択されました。

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

数値列の最小値、合計および平均値を含めるために、ファイル `STATS` を拡張するとします。**例 3-12** では、値の入力を求めるプロンプトが `GROUP_COL` について 2 回、`NUMBER_COL` について 1 回表示され、それぞれの `GROUP_COL` または `NUMBER_COL` には前に 1 つのアンパサンドが付いています。さらに 3 つの関数を、それぞれの前にアンパサンドを 1 つ付けてコマンド・ファイルに追加した場合には、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` は、前に 2 つのアンパサンドが付いた置換変数を自動的に `DEFINE` し、アンパサンドが前に 1 つのみ付いた置換変数は `DEFINE` しません。ユーザーが変数を `DEFINE` した場合、`SQL*Plus` は (`&` 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を置換します。`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。

### 例 3-13 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コマンド・ファイル `STATS` を拡張した後にそのファイルを実行するには、まず、次のように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` を取り出し、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

```

ファイル STATS2 が作成されました。

最後に、コマンド・ファイル 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

```

SQL\*Plus は、次の出力を表示します。



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 行が選択されました。

NUMBER\_COL および GROUP\_COL の値の入力を求めるプロンプトが 1 回のみ表示されました。現行セッションで STATS2 を再び実行する予定だった場合は、TABLE の入力を求めるプロンプトは表示されますが（名前に 1 つのアンパサンドが付いており、この変数は DEFINE されていないため）、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（名前に二重アンパサンドが付いており、これらの変数は DEFINE されているため）。

先へ進む前に、次のようにしてシステム変数 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、他にも置換が意味をなさないコマンド、たとえば SQL\*Plus コメント (REMARK、「/\*...\*/」または「--」) など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 は、& または && で始まるテキストを、他のテキスト文字列と同じように単なる文字列として扱います。

### 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	置換文字の前に使用できるエスケープ文字を定義します。SQL*Plus は、エスケープ文字を検出すると、置換文字を変数置換の要求ではなく通常の文字として扱います。デフォルトのエスケープ文字はバックslash (\) です。
SET VERIFY ON	置換の前後にコマンド・ファイル各行を表示します。
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。

これらのシステム変数の詳細は、第 8 章の「SET」コマンドを参照してください。



## START コマンドを使用した値の指定

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コマンド・ファイル内のパラメータに値を渡すことによってバイパスできます。

そのためには、コマンド・ファイル内で置換変数のかわりにアンパサンド (&) とそれに続けて数値を指定します。このコマンド・ファイルを実行するたびに、START はファイル内の各 &1 を START *filenames* の後の最初の値 (引数) に置換し、各 &2 を 2 番目の値に置換するというように、指定した数字の順序で置換します。

たとえば、MYFILE というコマンド・ファイル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、コマンド・ファイル 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 コマンドとともに引数を使用した場合、SQL\*Plus はコマンド・ファイル内のそれぞれのパラメータを適切な引数の値で DEFINE します。

### 例 3-14 START を使用してパラメータを渡す方法

表示する職種をパラメータで指定する新しいコマンド・ファイルを 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
ファイル ONEJOB が作成されました。
```

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 は、次のように、パラメータをその値に置換する前と後に、そのパラメータが含まれている SQL コマンドの行を表示し、その後結果を表示します。



```
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つのコマンド・ファイルの中でいくつでも使用できます。また、1つのコマンド・ファイルの中で1つ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。コマンドをコマンド・ファイルに格納し、START または @ を使用して実行してください。

---

---

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、画面にメッセージを送信したり、[Return] キーを押すのみなどのユーザーからの入力を受信できます。PROMPT および ACCEPT を使用して、SQL\*Plus が置換変数用に自動生成する値のプロンプトをカスタマイズすることもできます。

### 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・ユーザーにメッセージを送信したり、エンド・ユーザーからの入力値を受け入れることができます。PROMPT は、指定され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のユーザー変数に格納します。値の入力を求めるプロンプトを複数の行にわたって表示する場合は、ACCEPT と組み合わせて PROMPT を使用します。

**例 3-15 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、まず次のようにバッファを消去します。



```
CLEAR BUFFER
```

次に、コマンド・ファイルを設定し、PROMPT1 という名前でのこのファイルを保存します。



```
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
```



ファイル PROMPT1.sql が作成されました。

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、[第 4 章の「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、コマンド・ファイルを実行し、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、置換変数と組み合わせて PROMPT および ACCEPT を使用します。

### 例 3-16 置換変数と組み合わせて PROMPT および ACCEPT を使用する方法

例 3-15 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あるコマンド・ファイルに PROMPT および ACCEPT を挿入します。次のコマンドを使用して、最初にバッファを消去します。



```
CLEAR BUFFER
```

このような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、このコマンド・ファイルを実行します。SQL\*Plus は、PROMPT および ACCEPT で指定されたテキストを使用して、次のように ENUMBER の値の入力を求めるプロンプトを表示します。



```
START PROMPT2
```

次のように、Employee ID の入力を求めるプロンプトが表示されます。



```
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 を指定したので、SQL\*Plus は数値以外の値を受け入れません。

「Employee ID.」というプロンプトに、数字のかわりに文字を入力してみてください。SQL\*Plus は、エラー・メッセージを表示し、正しい数字の再入力を求めるプロンプトを再表示します。



```
START PROMPT2
```

SQL\*Plus が Employee ID の入力を求めるプロンプトを表示すると、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 [Return] の入力のアクセプト

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で [Return] を入力するようにする場合は、SQL\*Plus コマンドの PAUSE を使用します。たとえば、コマンド・ファイルに次の行を挿入することもできます。



```
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、コマンド・ファイル内の適切な場所に、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、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じように使用でき、SQL\*Plus からアクセスすることもできます。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利用できます。

バインド変数は SQL\*Plus から認識できるので、バインド変数の値を SQL\*Plus で表示したり、SQL\*Plus で実行する PL/SQL サブプログラム中でバインド変数を参照することができます。

## バインド変数の作成

バインド変数は、SQL\*Plus の中で VARIABLE コマンドを使用して作成されます。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、第 8 章の「**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リスト表示するには、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。次に例を示します。



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



PL/SQL プロシージャが正常に完了しました。

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。バインド変数を表示する方法の詳細は、第8章「コマンド・リファレンス」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 バインド変数も、ストアド・プロシージャ内で PL/SQL カーソル変数を参照するために使用できます。これによって、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 のリリース 7.3 以上を使用する必要があります。

---

### 例 3-17 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 プロシージャが正常に完了しました。

この時点で、SELECT 文の結果を SQL\*Plus で PRINT 文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 ブロックを再実行します。

### 例 3-18 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、プロシージャへパラメータとして渡されます。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;
```



/  
パッケージが作成されました。

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;
```



/  
プロシージャが作成されました。



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 プロシージャが正常に完了しました。

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 プロシージャが正常に完了しました。



```
PRINT pcv
```



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

**例 3-19 ストアド・ファンクションでの REF CURSOR 変数の使用方法**

次のようにして、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;
```



/  
ファンクションが作成されました。

このファンクションを実行します。



```
VARIABLE rc REF CURSOR
EXECUTE :rc := EmpInfo_fn
```



PL/SQL プロシージャが正常に完了しました。

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 プロシージャが正常に完了しました。



```
PRINT rc
```



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

## 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。

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』および第 8 章の「SET」コマンドの AUTOTRACE 変数を参照してください。

### 例 3-20 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/HR
@$ORACLE_HOME/RDBMS/ADMIN/UTLXPLAN.SQL
```



表が作成されました。

**例 3-21 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;
ロールが削除されました。
create role plustrace;
ロールが作成されました。
.
.
.
grant plustrace to dba with admin option;
権限付与が成功しました。
```

**例 3-22 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;
権限付与が成功しました。
```

**実行計画**

「Execution Plan」は、SQL オプティマイザの問合せ実行パスを示します。両方の表が、フル・テーブル・スキャンによってアクセスされ、ソートされ、その後マージされます。

「Execution Plan」のそれぞれの行には、順次行番号が付きます。SQL\*Plus は、親操作の行番号も表示します。

「Execution Plan」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	それぞれの実行ステップの行番号を表示します。
PARENT_ID_PLUS_EXP	それぞれのステップとその親の関係を表示します。この列は、大きなレポートに便利です。
PLAN_PLUS_EXP	レポートのそれぞれの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。

「Execution Plan」の出力は、EXPLAIN PLAN コマンドを使用して生成されます。EXPLAIN PLAN の出力の解釈の詳細は、『Oracle9i データベース・パフォーマンス・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。

統計およびその解釈方法の詳細は、『Oracle9i データベース・パフォーマンス・ガイドおよびリファレンス』を参照してください。

### 例 3-23 パフォーマンス統計および問合せ実行パス用の文のトレース

SQL バッファに次の文が含まれているとします。



```
SELECT E.LAST_NAME, E.SALARY, J.JOB_TITLE  
FROM EMPLOYEES E, JOBS J  
WHERE E.JOB_ID=J.JOB_ID AND E.SALARY>12000
```

この文を実行すると、次のように文のトレースが自動的に行われます。



```
SET AUTOTRACE ON
/
```



```
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行が選択されました。

#### 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
```

---

---

**注意：** 出力は、接続先のサーバーのバージョン、またはサーバーの構成によって異なる場合があります。

---

---

**例 3-24 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行が選択されました。

**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、大規模な問合せをチューニングしていて、問合せレポートを表示しない場合に便利です。

**例 3-25 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「Execution Plan」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。

**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の中で文をトレースする場合、「Execution Plan」には、コストベースのオプティマイザの行数見積り（カーディナリティ）が表示されます。一般に、各ノードでのコスト、カーディナリティおよびバイト数は、累積結果を表しています。たとえば、結合ノードのコストは結合操作を実行するためのコストのみを考慮したのではなく、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・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	それぞれの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の問合せのテキストを表示します。

列の書式は、COLUMN コマンドで変更できます。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。

---

**注意：** このレポートの 2 番目のセクションを表示するには、Oracle7 リリース 7.3 以上を使用する必要があります。

---



## 例 3-26 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);
```



表が作成されました。



```
CREATE TABLE D2_T2 (UNIQUE1 NUMBER) PARALLEL -
(degree 6);
```



表が作成されました。



```
CREATE UNIQUE INDEX D2_I_UNIQUE1 ON D2_T1(UNIQUE1);
```



索引が作成されました。



```
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
```

「Execution Plan」の行0は、コストベースのオプ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。

# 4

---

## 問合せ結果の書式の設定

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。この章では、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
- Web レポートの作成

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「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、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、第 8 章の「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 は、問合せ結果を表示する場合、列または式の名前をデフォルトの列ヘッダーとして使用します。しかし、列名は多くの場合、短くて暗号のようであり、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

次に示す書式で COLUMN コマンドの HEADING 句を使用すると、さらに便利な列ヘッダーを定義できます。

```
COLUMN column_name HEADING column_heading
```

#### 例 4-1 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 コマンドを入力するときに、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することができます。詳細は、第 8 章の「SET」コマンドを参照してください。)

#### 例 4-2 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

それぞれの列のヘッダーに下線を付けるために使用する文字を変更するには、使用する文字を SET コマンドの UNDERLINE 変数に設定します。

**例 4-3 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、引用符で囲んでください。そうしないと、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せん。

---

**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できます。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。

**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 を使用しなかった場合、列の幅は、必ず 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、許容される最大文字数ま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ことによって、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、書式を設定する列を識別し、使用するモデルを示すことができます。

```
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たり、負の値の右側（左側でなく）にマイナス記号を表示したり、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 4-4 を参照）。書式モデルのすべてのリストおよび詳細は、第 8 章の「COLUMN」コマンドを参照してください。

### 例 4-4 NUMBER 列の書式設定

SALARY の表示にドル記号やカンマを使用し、0（ゼロ）値に空白ではなく数字の 0（ゼロ）を表示するには、次のコマンドを入力します。

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

この時点で、現行の間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
Cambault	\$11,000	.3
Zlotkey	\$10,500	.2

ドル記号などの他の書式を使用しているときに、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）を使用してください。

---

**注意：** 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できます。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

### 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。

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 パラメータから得られます。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、第 8 章の「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することによって変更できます。

COLUMN コマンドの中で、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。LONG、CLOB または NCLOB のいずれかの列に幅を指定した場合は、LONGCHUNKSIZE、または指定した幅のどちらか小さい方が列幅として使用されます。詳細は、第 8 章の「COLUMN」コマンドを参照してください。



**例 4-5 文字列の書式設定**

LAST\_NAME 列の幅を 4 文字に設定し、現行の問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
```

---

**注意：** 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。

---

SET コマンドの WRAP 変数を ON (デフォルト値) に設定した場合、従業員名は例 4-5 に示すように、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。ある列についての `WRAP` の設定は、`COLUMN` コマンドの `WRAPPED`、`WORD_WRAPPED` および `TRUNCATED` 句を指定することによって上書きできます。これらの句の詳細は、第 8 章の「`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
```

## 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` 句は、前に定義した列の表示属性を新しい列にコピーするよう `SQL*Plus` に指示します。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 4-6 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 NAME	MONTHLY SALARY	BONUS
Russell	\$14,000	\$0
Partners	\$13,500	\$0
Errazuriz	\$12,000	\$0
Cambrault	\$11,000	\$0
Zlotkey	\$10,500	\$0

## 列の表示属性のリスト表示およびリセット

ある列の現行の表示属性をリスト表示するには、次に示すように COLUMN コマンドと、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リスト表示するには、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
```

すべての列の属性をリセットするには、CLEAR コマンドの COLUMNS 句を使用します。

### 例 4-7 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、次のコマンドを入力します。



```
CLEAR COLUMNS
```



```
columns は消去されました。
```

## 列の表示属性の抑止および復元

特定の列に指定した表示属性を抑止および復元できます。列の表示属性を抑止するには、次の形式で COLUMN コマンドを入力します。

```
COLUMN column_name OFF
```

OFF 句は、列にデフォルトの表示属性を使用するよう SQL\*Plus に指示しますが、COLUMN コマンドで定義された属性は削除しません。COLUMN コマンドで定義した属性を復元するには、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に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、SET コマンドの RECSEP および RECSEPCHAR 変数を使用します。

RECSEP は、文字の行を出力するタイミングを決定します。RECSEP を EACH に設定すると、各行の後に文字の行が出力され、WRAPPED に設定すると、折返し行の後に出力され、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。

列の値が追加行に折り返される場合は、すべての単語を追加行に折り返すこともできます。そのためには、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 4-8 折り返した列の値の後に文字の行を出力する方法

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



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

最後に、次の問合せを入力します。



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

この時点で、次のように JOB\_TITLE 列の幅を制限し、必要な場合はすべての単語を追加行に折り返すよう SQL\*Plus に指示します。



```
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行が選択されました。

RECSEP を EACH に設定した場合、SQL\*Plus は各行の後に（前述の例では各部門の後に）文字の行を出力します。

先へ進む前に、次のように RECSEP を OFF に設定して、レコード・セパレータの出力を抑制します。



```
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、出力の中で一緒に表示されます。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）を追加することによって、出力をユーザーにとってより実用的なものにできます。

BREAK コマンドの中で指定する列は、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ことによって、出力に意味のあるレコードのサブセットを作成することができます。その後、同じ BREAK コマンドの中でサブセットに書式設定を追加でき、COMPUTE コマンドの中でブレイク列を指定することによって（合計、平均などを含んだ）サマリー行を追加できます。

たとえば、BREAK コマンドや COMPUTE コマンドを使用せずに、次の問合せを実行したとします。



```
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
80	Partners	13500
90	King	24000
90	Kochhar	17000
90	De Haan	17000

6 行が選択されました。

このレポートをさらに使用しやすくするには、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 値の重複を抑制し、部門が変わるときに空白行を入力したり改ページできます。BREAK を COMPUTE と組み合わせて使用し、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）を含んだサマリー行を計算して出力することもできます。

## ブレイク列内の値の重複の抑止

BREAK コマンドは、デフォルトでは指定された列または式の値の重複を抑制します。したがって、ORDER BY 句で指定した列の値の重複を抑制するには、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 句を使用しなかった場合は、列の値を変更するたびにブレイクが発生します。

---

---

**例 4-9 ブレー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を表示しないようにするには、次のコマンドを入力します。



```
BREAK ON DEPARTMENT_ID;
```

次の問合せ（バッファに格納されている現行の問合せ）の場合



```
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行が選択されました。

**ブレーク列の値が変更されたときの空白の挿入**

ブレーク列の中で値が変更されるたびに、空白行を挿入するか、または改ページすることができます。 $n$  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 4-10 ブレーク列の値が変更されたときの空白の挿入**

部門と部門の間に 1 行の空白行を挿入するには、次のコマンドを入力します。



```
BREAK ON DEPARTMENT_ID SKIP 1
```

この時点で、次のように問合せを再実行します。



/



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

6行が選択されました。

## 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することもできます。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` を指定しなければ、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方法

たとえば、`ORDER BY` 句に複数の列があり、それぞれの列の値が変更されたときに空白を挿入するとします。`BREAK` コマンドは、入力するたびに前の `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、または順序付きの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、単一の `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 4-11 間隔設定技法の統合

次のコマンドを使用して、最初にバッファを消去します。



```
CLEAR BUFFER
```

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ため、ページ番号付きの TTITLE を作成するには、次のように入力します。



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

次のように新規問合せを実行し、結果を表示します。



```

                                     Page: 1
DEPARTMENT_ID JOB_ID      LAST_NAME      SALARY
-----
                20 MK_MAN   Hartstein      13000

                                     Page: 2
DEPARTMENT_ID JOB_ID      LAST_NAME      SALARY
-----
                80 SA_MAN   Russell        14000
                Partners    13500

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

6 行が選択されました。

## ブレイク定義のリスト表示および取消し

上部タイトルの定義を変更しないで、その表示を抑止するには、継続する前に次のように入力します。



```
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リスト表示できます。



```
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
```

すべてのコマンド・ファイルの先頭に **CLEAR BREAKS** コマンドを指定することによって、あるファイル内で実行する問合せに、前回の **BREAK** コマンドが影響しないようにすることができます。

## 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ついて様々な計算を実行できます。この計算は、**SQL\*Plus** の **COMPUTE** コマンドの関数で実行します。次の形式で **BREAK** コマンドおよび **COMPUTE** コマンドを同時に指定してください。

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

**BREAK** コマンドには、複数のブレイク列と、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。**BREAK** コマンドと **COMPUTE** コマンドを同時に使用して、**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  
... 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8章「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 4-1 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 コマンドの中で指定した関数は、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、順序付きの列の値が変更されたときに、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かった場合は、省略されていない関数キーワードが出力されます。

計算ラベルを抑止するには、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、第 8 章の「COMPUTE」コマンドを参照してください。

#### 例 4-12 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、次のように現行の BREAK 定義をリスト表示します。

```
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
                Partners                13500

*****
sum                      27500

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

*****
sum                      58000
```

6行が選択されました。

部門 10 および 20 の給与の合計を計算し、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 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行が選択されました。

レポートの終わりで給与を計算するには、次のようにします。



```

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行が選択されました。

---

**注意：** SALARY 列の書式は、個々の SALARY 値のみでなく、SALARY の合計を表示する際にも影響します。NUMBER 列の書式を設定する際に、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 4-13 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、新規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 コマンドと、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 4-14 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 4-13 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 は指定されたそれぞれの列について値を計算して出力します。

#### 例 4-15 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、最初に次のような BREAK コマンドと 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  
Raphaely 11000  
*****  
avg 4150  
sum 24900
```

6行が選択されました。

## COMPUTE 定義のリスト表示および取消し

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 定義をリスト表示できます。



```
COMPUTE
```

#### 例 4-16 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 は消去されました。  
CLEAR COMPUTES  
computes は消去されました。
```



すべてのコマンド・ファイルの先頭に CLEAR BREAK コマンドおよび CLEAR COMPUTES コマンドを指定することによって、前に入力した BREAK コマンドおよび COMPUTE コマンドが、ある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、各ページの下に表示するタイトルも設定できます。TTITLE コマンドは上部タイトルを定義し、BTITLE コマンドは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 コマンドはレポート・ヘッダーを定義、REPFOOTER コマンドはレポート・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 8 章](#)の「TTITLE」コマンドおよび「REPHEADER」コマンドを参照してください。

#### 例 4-17 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行が選択されました。

#### 例 4-18 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 行が選択されました。

レポート・ヘッダーの定義を変更せずにその出力を抑止するには、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、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、行サイズを減らして、タイトルをデータの近くに中央揃えに配置するのも良い方法です。

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 4-19 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行が選択されました。

LEFT、RIGHT および CENTER 句は、後続の値を行の始め、終わりおよび中央に配置します。SKIP 句は、1 行以上、下へ移動するよう SQL\*Plus に指示します。

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。前述の例では、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 コマンドの始めに SKIP  $n$  句を使用します。たとえば、前述の例では、下部タイトルの前で 1 行のみスキップさせるには、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 4-20 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 行が選択されました。

## 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 コマンドの中でそれらの変数を参照します。



```
TTITLE CENTER LINE1 SKIP 1 CENTER LINE2 SKIP 1 -  
CENTER LINE3
```

## タイトルの中にページ番号およびその他の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の中に表示するには、たとえば次のようにシステム値の名前をタイトル要素として入力します。

```
TTITLE LEFT system-maintained_value_name
```

タイトルの中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。タイトルの中に表示できるシステム管理値のリストについては、[第8章「コマンド・リファレンス」](#)の「TTITLE」コマンドを参照してください。

### 例 4-21 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。



```
/
```



```
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行が選択されました。

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 4-22 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行が選択されました。

**ページ・タイトル定義のリスト表示、抑止、復元**

ページ・タイトルの定義をリスト表示するには、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
```

タイトル定義を抑止するには、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、SQL\*Plus はレポートにタイトルを表示しなくなりますが、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、次のように入力すれば復元できます。



```
TTITLE ON
BTITLE ON
```

## タイトル内に列の値を表示する方法

マスター / ディテール・レポートを作成し、各ページの一番上には変化するマスター列の値を、その下にはマスター列の値に対する詳細な問合せ結果を表示することができます。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 コマンドの中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 句と BREAK コマンドの中にマスター列を挿入する必要があります。

### 例 4-23 マスター / ディテール・レポートの作成

2 人のマネージャの従業員番号を別々のページの一番上にそれぞれ表示し、そのマネージャの部下をマネージャの従業員番号と同じページに表示するレポートを作成します。最初に、次のように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の中に表示されるので、ディテールの列としては出力されません。前述の NOPRINT 句は、MANAGER\_ID 列を出力しないよう SQL\*Plus に指示します。

次に、ページ・タイトルにラベルおよび値を挿入し、適正な BREAK コマンドを入力して、最後の例の下部タイトルが表示されるのを抑止します。



```
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行が選択されました。

ページの一番下に列の値を出力する場合は、**COLUMN** コマンドを次の形式で使用します。

```
COLUMN column_name OLD_VALUE variable_name
```

**SQL\*Plus** は、マスター列の新しい値を検出した後、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を出力します。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。非定型のレポートなら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した方が便利です。そのためには、現在の日付を保持する変数を作成します。

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。



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

**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、タイトルの中で **\_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、第3章の「**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・プロンプトから入力することもできます。詳細は、第8章の「**COLUMN**」コマンドを参照してください。

## ページ・サイズの設定

一般に、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 は長すぎて1ページに収めきれないレポートを複数の連続したページに表示し、各ページに独自のタイトルおよび列ヘッダーを付けます。SQL\*Plus が各ページに表示するデータの量は、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の一番下までの1ページ当り行数:24
- 1行当りの文字数:80

これらの設定は、ご使用のコンピュータ画面のサイズ、または印刷の場合は用紙サイズ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。たとえば、印刷するページは一般に66行の長さなので、レポートの印刷時にページの長さを変更してもかまいません。

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 は新しいページを開始するために0（ゼロ）行のみスキップし、改ページ文字を表示して印刷します。ほとんどのタイプのコンピュータ画面では、改ページ文字は画面を消去してカーソルを1行目の先頭に移動します。レポートを印刷しているときに改ページ文字が検出されると、その検出位置が現在のページの最後でなくても、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、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。

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減らすこともできます。また、幅の広い用紙に出力するために行サイズを増やすこともできます。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 4-24 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し（またはプリンタを改ページ）、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           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 行が選択されました。

この時点で、次のように 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リスト表示するには、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

問合せ結果を、印刷する前にワード・プロセッサで編集したり、書状、メモまたはその他の文書に挿入する場合は、問合せ結果をファイルに格納します。

問合せ結果を画面に表示したまま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かった場合、**SPOOL** はそのファイルを出力ファイルとして認識するため、ファイル名にデフォルトのファイル拡張子を追加します。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。

次のように **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** は情報をファイルにスプールし続けます。

```
SPOOL OFF
```

### フラット・ファイルの作成

異なるソフトウェア・プロダクト間でデータを移動するときには、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** ヘッダーを移行するには、**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 HEADING OFF
SET MARKUP HTML OFF SPOOL OFF
```

これらのコマンドを入力した後、前述のような 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、列に線を付けるときに便利です。詳細は、[第 8 章](#)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

問合せ結果を画面に表示したまま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 コマンドを入力した後に画面に表示されていたすべての情報が、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、前述のようにして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
```

SQL\*Plus はスプールを停止し、スプールされたファイルの内容をご使用のホスト・コンピュータの標準（デフォルト）プリンタにコピーします。SPOOL OUT では、スプール・ファイルは印刷後に削除されません。

#### 例 4-25 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、EMPRPT という名前のコマンド・ファイルを作成します。

最初に EDIT を使用して、ご使用のホスト・オペレーティング・システムのテキスト・エディタでコマンド・ファイルを作成します。(INPUT および SAVE を使用しないでください。これらを使用すると、SQL\*Plus はファイルの終わりにスラッシュを追加し、コマンド・ファイルを 2 回実行します。1 回はセミコロンの結果として実行され、1 回はスラッシュによって実行されます。)



```
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、コマンド・ファイル EMPRPT 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
                Partners                $13,500
*****
sum                                $27,500

                90 King
                Kochhar                $17,000
                De Haan                $17,000
*****
sum                                $58,000

sum                                $98,500
                                -----
                                COMPANY CONFIDENTIAL

```

6行が選択されました。

## Web レポートの作成

SQL\*Plus を使用すると、完全なスタンドアロン 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、スタンドアロン Web ページを生成できます。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。

取り出されたデータは、デフォルトでは HTML 形式で記述されますが、オプションで HTML の <PRE> タグに直接出力できるため、SQL\*Plus で表示される形式と同じように、Web ブラウザで正確に表示することができます。これらのコマンドの詳細は、第 7 章の「SQLPLUS コマンドでの SQL\*Plus の起動」の SQLPLUS -MARKUP コマンドおよび第 8 章の「SET」の SET MARKUP コマンドを参照してください。

プログラム・スクリプトに SQL\*Plus を埋め込む場合に、SQLPLUS -MARKUP HTML ON は有効です。起動時、コマンドを実行する前に、<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 とあわせて使用すると、SQL\*Plus プロンプトおよびバナー情報の表示の抑止、および一部のコマンドの使用制限に効果的です。

SET MARKUP HTML ON SPOOL ON は、その後にスプールされたそれぞれのファイルに対する完全な HTML ページを生成します。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 タグがクローズされます。

SET MARKUP HTML ON SPOOL OFF を使用すると、既存の Web ページの埋込みに適した 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。

## 静的 Web レポートの作成

SET MARKUP コマンドを SQL\*Plus セッションで対話的に使用し、スプール・ファイルに HTML で書き出します。出力結果は Web ブラウザで表示できます。

SET MARKUP 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、出力ファイルを作成する、またはファイルへの書出しを開始するわけではありません。SET MARKUP HTML ON SPOOL ON によって、スプール・ファイルのオープンおよびクローズが、<HTML> および </HTML> タグを生成するトリガーとなります。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、SQL\*Plus の SPOOL コマンドを使用します。

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 を指定します。そうすると、出力ファイルの種類を簡単に識別でき、また、Web ブラウザが 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と、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クローズするには、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・ファイルがクローズされ、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 4-26 対話型セッションでのスタンドアロン Web 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 セッションをオープンして、次のように入力します。



```
SET MARKUP HTML ON SPOOL ON PREFORMAT OFF ENIMAP ON -
HEAD '<TITLE>Department Report</TITLE>' -
<STYLE type="text/css"> -
<!-- BODY {background: #FFFFFFC6} --> -
</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 コマンドは、SQL 問合せ出力に対する新しいヘッダー名を作成します。最初のコマンドは、DEPARTMENT\_NAME 列にマップするエンティティを OFF にし、出力データのこの列に HTML のハイパーリンクを正しく作成します。



```
COLUMN DEPARTMENT_NAME HEADING 'DEPARTMENT' ENIMAP OFF
COLUMN DEPARTMENT_NAME HEADING 'DEPARTMENT'
COLUMN CITY HEADING 'CITY'
```

SET MARKUP HTML SPOOL ON を使用すると、SQL\*Plus が HTML をスプール・ファイルに書き出します。次の SPOOL コマンドは、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 コマンドの後の入力または標準出力への表示は、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;
```

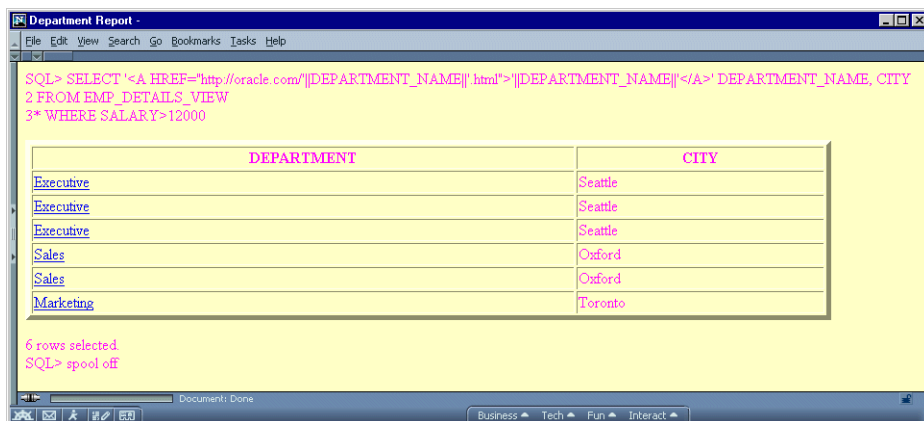
SPOOL OFF コマンドを入力します。



```
SPOOL OFF
```

ファイルをクローズする前に、スプール・ファイル *report.html* 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* です。これは、Web ブラウザへロードされるスタンドアロン Web ページです。Web ブラウザで *report.html* をオープンすると、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 文は抑止されていません。スクリプトの起動時に、`-SILENT` オプションの `SET ECHO OFF` を使用すると、抑止できます。

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 の特殊文字が、対応するエンティティ、&quot;、&amp;、&lt; および &gt; に置き換えられません。
- 文字列と変数要素を連結してハイパーリンクを作成するには、SELECT 文に引用符および連結文字を使用します。

Web ブラウザまたはテキスト・エディタで *report.html* ソースを表示すると、次のように、Department 列の表セルに完全な形式のハイパーリンクが設定されていることがわかります。

```
<html>
<head>
<TITLE>Department Report</TITLE> <STYLE type="text/css"> <!-- BODY
{background: #FFFFFF6} --> </STYLE>
<meta name="generator" content="SQL*Plus 9.0.1">
</head>
<body TEXT="#FF00ff">
SQL&gt; SELECT '&lt;A
  HREF=&quot;http://oracle.com/'||DEPARTMENT_NAME||'.html&quot;&gt;';' ||DEPARTMENT_N
  AME||'&lt;/A&gt;'; DEPARTMENT_NAME, CITY
&lt;br>
  2 FROM EMP_DETAILS_VIEW
&lt;br>
  3* WHERE SALARY&gt;12000
&lt;br>
&lt;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>
&lt;p>
```

6行が選択されました。<br>

```
SQL> spool off  
<br>  
</body>  
</html>
```

#### 例 4-27 SQLPLUS コマンドを使用したスタンドアロン Web レポートの作成

オペレーティング・システムのプロンプトで次のコマンドを入力します。

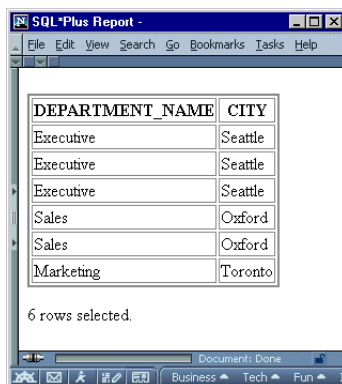
```
SQLPLUS -S -M "HTML ON TABLE 'BORDER=2'" HR/HR@Ora9i @depart.sql>depart.html
```

*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>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 スクリプトからの Web 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* をオープンします。



DEPARTMENT_NAME	CITY
Executive	Seattle
Executive	Seattle
Executive	Seattle
Sales	Oxford
Sales	Oxford
Marketing	Toronto

6 rows selected.

この例の SQLPLUS コマンドには、ネストした引用符による 3 つのレイヤーが含まれています。レイヤーには、内側から順に次のものがあります。

- "2" は、BORDER に対する HTML 属性値を引用符で囲んだものです。
- BORDER="2" は、TABLE オプションに対するテキスト引数を引用符で囲んだものです。
- "HTML ON TABLE 'BORDER="2"'" は、-MARKUP オプションに対する引数を引用符で囲んだものです。

引用符のネスト方法は、オペレーティング・システムやプログラム・スクリプト言語によって異なります。

## CGI スクリプトを使用した動的 Web レポートの作成

SET MARKUP コマンドを対話的に使用するかわりに、SQLPLUS-MARKUP コマンドを使用すると、インターネットで使用可能なモードで SQL\*Plus セッションを開始できます。これによって、Common Gateway Interface (CGI) またはオペレーティング・システム・コマンド・ファイル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 のコマンド・ファイル (バッチ・ファイル)、Java、Java スクリプト、Perl のファイルなど、Web サーバーがサポートす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、HTML の Web 出力を生成できます。そのため、Web ベースのレポートを動的に作成できます。

### 例 4-28 CGI スクリプトからの Web 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。

- CGI スクリプトをコールする 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 ページでは、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プロンプトが表示されます。

---

---

**注意：** ログイン情報をスクリプト・ファイルに埋め込んだり、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。

---

---



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

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

<!-- Change the URL here. On Windows NT you may need to use
http://host.domain/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://host.domain/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, 2001. 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 8.1.7 (or later) is
#   installed on your webserver, and the webserver 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/901"; # 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-Fa-f0-9]{2})/pack("c",hex($1))/ge;
  $val =~ s/%([A-Fa-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;
```

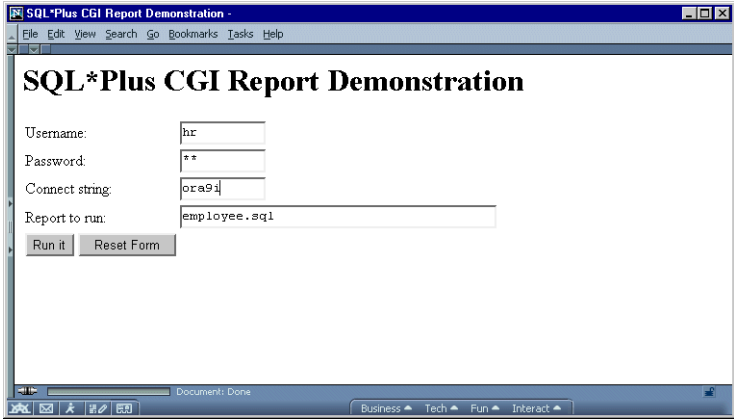
### CGI を使用した SQL スクリプトの例

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* をオープンします。

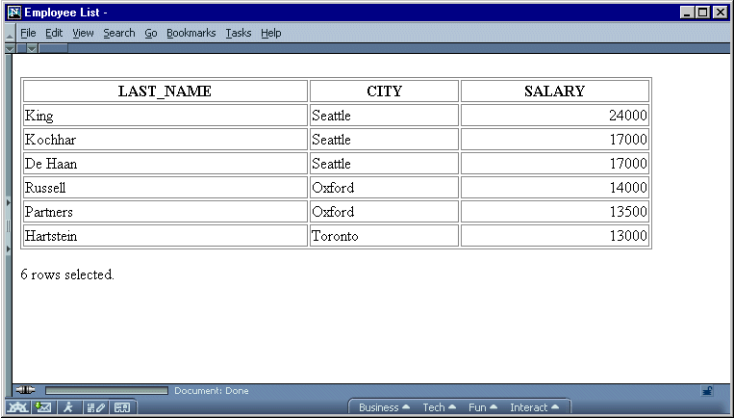


The screenshot shows a web browser window titled "SQL\*Plus CGI Report Demonstration". The page contains a form with the following fields and values:

- Username: hr
- Password: \*\*
- Connect string: ora9i
- Report to run: employee.sql

Below the form are two buttons: "Run it" and "Reset Form".

「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。



The screenshot shows a web browser window titled "Employee List". The page displays a table with the following data:

LAST_NAME	CITY	SALARY
King	Seattle	24000
Kochhar	Seattle	17000
De Haan	Seattle	17000
Russell	Oxford	14000
Partners	Oxford	13500
Hartstein	Toronto	13000

Below the table, it says "6 rows selected."

## Web レポートでの SQL\*Plus コマンドの表示の抑止

SILENT モードを使用して SQL\*Plus セッションを起動することをお勧めします。これによって、Web ブラウザには SQL 問合せの結果のみが表示されます。

CGI スクリプトまたはオペレーティング・システム・コマンド・ファイルを使用して、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便利です。これによって、SQL\*Plus コマンドおよび SQL\*Plus パナーの表示を抑止できます。HTML 出力には、SQL 問合せの結果データのみが表示されません。

## 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 4-2 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;	アンパサンド

Web ブラウザには「>」が表示されますが、エンコードされた実際の 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あった場合、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 を使用して、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。



---

## 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。この章では、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、[付録 E「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には、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、データベースとすでに起動されているインスタンスが対応付けられます。

### 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 8 章](#)を参照してください。

### 例 5-1 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 5-2 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 5-3 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 内のデータベースおよびリカバリ・データがすべてデータ・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、インスタンスからデータベースが切り離され、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 メモリーが回収され、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 8 章「コマンド・リファレンス」を参照してください。

#### 例 5-4 データベースの停止

通常、オープンおよびマウントされた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.
```

## 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、データベースが壊れた場合に、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変更がすべて永久に保存されているので、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 8 章「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・ファイルの詳細をリスト表示するには、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8章「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 を使用したデータベースへの接続

この章では、SQL\*Plus を使用してデータベースに接続する方法を説明します。この章では、次の項目について説明します。

- デフォルト・データベースへの接続
- リモート・データベースへの接続
- 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](#)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。

## デフォルト・データベースへの接続

あるデータベース内のデータにアクセスするには、最初にそのデータベースに接続する必要があります。通常、SQL\*Plus を起動すると、起動時に入力したユーザー名およびパスワードでデフォルトの Oracle データベースに接続します。いったんログインすると、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。インストール中に作成されるデフォルト・ユーザーのリストについては、第 1 章の「ユーザー名およびパスワード」を参照してください。

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
```

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は、どのデータベースにも接続していない状態になります。

データベースへの接続の際にアカウントの有効期限が切れたユーザーとしてログイ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、そのユーザーとしての接続は、アカウントのロックが DBA によって解除されるまで許可されません。

SQL\*Plus のコマンド・プロンプトで SQL\*Plus の DISCONNECT コマンドを入力することによって、SQL\*Plus を終了せずに現在 Oracle に接続しているユーザー名を切断できます。

デフォルト・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。

## リモート・データベースへの接続

多数の大きなシステムでは、複数のコンピュータ上で Oracle を実行します。それらのコンピュータは、多くの場合、ネットワーク上で接続されているので、異なるコンピュータ上のプログラム間でデータを効率よく高速に交換できます。ネットワーク化されたコンピュータは、互いに物理的に近い位置に設置されている場合と、遠く離れた位置に設置され、通信リンクで接続されている場合があります。

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・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。接続するデータベースに Oracle Net があり、両方のデータベースが互換性のあるネットワーク・ドライバを備えている場合は、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 つの方法のいずれかで接続できます。

- SQL\*Plus から、CONNECT コマンドを使用する
- SQL\*Plus を起動するときに、SQLPLUS コマンドを使用する

## SQL\*Plus からのリモート・データベースへの接続

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、CONNECT コマンドに、次のいずれかの形式で Oracle Net のデータベース指定を含めます（入力するユーザー名およびパスワードは、接続するデータベースに対して有効である必要があります）。

- CONNECT HR@connect\_identifier
- CONNECT HR/HR@connect\_identifier

SQL\*Plus は必要に応じてパスワードの入力を求めるプロンプトを表示し、指定されたデータベースにユーザーを接続します。

他のデータベース接続と同様に、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イ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、そのユーザーとしての接続は、アカウントのロックが DBA によって解除されるまで許可されません。

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。

サービス名を指定するために入力する正確な文字列は、コンピュータが使用する Oracle Net プロトコルによって異なります。詳細は、第 8 章の「CONNECT」および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、または DBA に問い合せてください。

## SQL\*Plus 起動時のリモート・データベースへの接続

SQL\*Plus を起動したときにリモート・データベースに接続するには、次のいずれかの形式で SQLPLUS コマンドに Oracle Net サービス名を挿入します。

- SQLPLUS HR@connect\_identifier
- SQLPLUS HR/HR@connect\_identifier

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 は必要に応じてユーザー名およびパスワードの入力を求めるプロンプトを表示し、SQL\*Plus を起動して、指定されたデータベースにユーザーを接続します。これは、ユーザーが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するデータベースです。

他のデータベース接続と同様に、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イ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、そのユーザーとしての接続は、アカウントのロックが DBA によって解除されるまで許可されません。

この場合も、この方法で接続した後、リモート・データベース内の表を直接操作できます。

---

---

**注意：** 接続識別子の @ 記号を、コマンド・ファイルの実行に使用する @ コマンドと混同しないでください。

---

---

## 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/HR@BOSTONDB -
TO TODD/FOX@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。両方の句を挿入すれば、スクリプトを読みやすくする利点があります。

COPY コマンドの動作は、宛先の表がすでに存在するかどうかおよびユーザーが入力したアクション句（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、元の表から 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/HR@BOSTONDB -  
REPLACE EMPCOPY2 -  
USING SELECT LAST_NAME, SALARY -  
FROM EMP_DETAILS_VIEW -  
WHERE DEPARTMENT_ID = 30
```

長い COPY コマンドは、コマンド・プロンプトから直接入力するよりも、コマンド・ファイル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 6-1 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/HR@BOSTONDB -  
CREATE EMPCOPY -  
USING SELECT * FROM HR
```



```
Array fetch/bind size is 15. (arraysize is 15)  
Will commit when done. (copycommit is 0)  
Maximum long size is 80. (long is 80)
```

その後、SQL\*Plusは表EMPLOYEE\_COPYを作成し、行をコピーします。



表SALESMANが作成されました。

```
5行がHR@BOSTONDBから選択されました。  
5行がSALESMANに挿入されました。  
5行がDEFAULT HOST接続のSALESMANにコミットされました。
```

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HR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コマンドは長さが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 コマンドの変数について、それらの設定を変更する方法も含め、詳細は、第 8 章の「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リスト表示された後、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を示されます。その後、選択、挿入およびコミットが実行された行数がリスト表示されます。

## 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/HR@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/HR@MYDATABASE -  
INSERT EMPLOYEE_COPY2 -  
USING SELECT * FROM EMPLOYEE_COPY
```

リモート・データベース上にある表と表の間でコピーするには、次のように FROM 句および TO 句の両方に同じユーザー名、パスワードおよびサービス名を挿入します。

```
COPY FROM HR/HR@BOSTONDB -  
TO HR/HR@BOSTONDB -  
INSERT EMPLOYEE_COPY2 -  
USING SELECT * FROM EMPLOYEE_COPY
```



# 第 II 部

---

## リファレンス

この部では、SQL\*Plus の起動方法の概要を説明します。それぞれの SQL\*Plus コマンドの説明を含むコマンド・リファレンスについても説明します。

この部では、次の章について説明します。

- [SQL\\*Plus の起動およびヘルプの表示](#)
- [コマンド・リファレンス](#)



---

## SQL\*Plus の起動およびヘルプの表示

この章では、オペレーティング・システムのプロンプトから SQL\*Plus にアクセスする方法を説明します。この章では、次の項目について説明します。

- [SQLPLUS コマンドでの SQL\\*Plus の起動](#)
- [SQL\\*Plus コマンドのヘルプ](#)

## SQLPLUS コマンドでの SQL\*Plus の起動

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] ]
```

各項目の意味は、次のとおりです。

**Option** 構文は次のとおりです。

```
-H[ELP] | -V[ERSION]
| [ [-M[ARKUP] markup_option] [-R[ESTRICT] {1|2|3}] [-S[ILENT]] ]
markup_option の構文は次のとおりです。
```

```
HTML [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。

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

**Start** 構文は次のとおりです。

```
@{uri|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 は、コマンド・ファイルの 1 行目に有効なログインが含まれているものとみなします。*start* も *logon* も指定しなかった場合、SQL\*Plus はログイン情報の入力を求めるプロンプトを表示します。

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



## MARKUP オプション

### -M [MARKUP]

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 タグがクローズされます。

-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便利です。

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。

このリリースでは、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、デフォルトは HTML 表への出力です。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、第 8 章の「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。オプションの ON および OFF は、HTML 出力を生成するかどうかを指定します。デフォルトは OFF です。

MARKUP HTML ON を使用すると、MARKUP オプションでの指定に従って 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の SET MARKUP HTML コマンドで設定したオプション・セットに従って 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 で HTML の出力が可能になり、その後 SQL\*Plus コマンドの SPOOL *filename* で、スプール・ファイルへの出力の書き出しを開始できます。詳細は、後述の SPOOL {ON|OFF} および第 8 章の「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、このような自由なテキストの入力について、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、このような自由なテキストの入力について、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、このような自由なテキストの入力について、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、カスタマイズした出力に書き出すことができます。

<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。出力で使用するエンティティの詳細は、第 8 章の「COLUMN」コマンドを参照してください。

---

---

**SPOOL {ON|OFF}**

SPOOL ON または SPOOL OFF では、SQL\*Plus の SPOOL *filename* コマンドで作成された各ファイルの開始および終了時に、HTML の開始タグである <HTML> タグや <BODY> タグ、および終了タグである </BODY> タグおよび </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してください。

SET MARKUP HTML SPOOL ON オプションでは、<HTML> タグのスパール・ファイルへの書き込みを指定します。SQLPLUS の SPOOL *filename* コマンドを実行するまで、スパール・ファイルは作成されず、SET MARKUP HTML SPOOL ON オプションでヘッダーおよびフッター・タグをスパール・ファイルに書き込むこともできません。

---

---

SPOOL *filename* コマンドを実行した場合、SQL\*Plus では、複数の HTML タグがスパール・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、次の終了タグが書き込まれ、ファイルがクローズされます。

```
</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、SQL\*Plus が <PRE> タグまたは HTML 表に出力を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、出力を HTML 表に送る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、Web 表用のレポートとは、次のように異なる意味があります。

- 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、ブラウザが空白文字として解釈する場合があります。
- 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 は新規に HTML 表を開始し、タイトルの後に出力行を表示します。ご使用のブラウザで、各列のデータの幅によって、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 [RESTRICT] {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ことと同様です。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。

-RESTRICT オプションが指定されていない場合、PUP 表で使用禁止に指定されていなければ、すべてのコマンドが使用可能です。

-RESTRICT を指定すると、LOGIN.SQL は読み込めません。GLOGIN.SQL は読み込めますが、制限付きのコマンドが使用されていると、正常に実行されません。

表 7-1 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 7-1 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・メッセージを抑止します。コマンド・プロンプト、コマンドのエコー、SQL\*Plus の起動時に通常表示されるバナーも含まれます。username または password を省略すると、SQL\*Plus はプロンプトを表示せずに入力を待ちます。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・システムのコマンド・ファイルを使用して、Web 用のレポートを作成する場合に便利なモードです。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抑止され、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。*username* および *password* を省略すると、SQL\*Plus は入力を求めるプロンプトを表示します。

*password* のみを省略すると、*password* の入力を求めるプロンプトが表示されます。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。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・ログインを表します。デフォルト・ログインを使用する場合、*connect\_identifier* を指定できません。デフォルト・ログインでは、SQL\*Plus はユーザー名 OPS\$name を使用してユーザーをログインさせようとします。*name* は、ご使用のオペレーティング・システムのユーザー名です。接頭辞「OPS\$」は、テキストの他の文字列にも設定されることがあります。たとえば、INIT.ORA パラメータ・ファイルの設定を LOGONname または USERIDname に変更することがあります。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、/ および /NOLOG でも特権付きの接続を使用できます。

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/MANAGER 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 コマンド・ファイルから表示するには、/NOLOG を使用します。コマンド・ファイルの 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

```
@{uri/file_name[.ext]} [arg ...]
```

コマンド・ファイルの名前および実行する引数を指定します。コマンド・ファイルは、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。*uri* は、このリリースでは、Windows プラットフォームのみでサポートされています。

SQL\*Plus 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、引数をコマンド・ファイルに渡します。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、第 8 章の「START」コマンドを参照してください。

## サイト・プロファイルの設定

SQL\*Plus はサイト・プロファイル（データベース管理者が作成した SQL\*Plus コマンド・ファイル）をサポートします。このファイルは、通常、拡張子 SQL が付いている GLOGIN という名前になっています。ユーザーが SQL\*Plus を起動して、SQL\*Plus が Oracle 接続を確立するたびに、SQL\*Plus はこのコマンド・ファイルを実行します。DBA は、サイト・プロファイルを使用して、特定サイトのすべてのユーザー用の SQL\*Plus 環境のデフォルトを設定します。ユーザーは、直接サイト・プロファイルにアクセスできません。サイト・プロファイルのデフォルト名およびデフォルト位置は、ご使用のシステムによって異なります。サイト・プロファイルの詳細は、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。

## ユーザー・プロファイルの設定

SQL\*Plus 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します。SQL\*Plus は、現行のディレクトリ内で名前が LOGIN で拡張子が SQL のファイルを検索します。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。



## リターン・コードの受信

無効なユーザー名かパスワードを指定したため、または、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、SQL\*Plus は、EXIT FAILURE コマンドと同じエラー状態を戻します。詳細は、第 8 章の「EXIT」コマンドを参照してください。

### 例 7-1 SQL\*Plus の起動

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



```
SQLPLUS HR/HR
```

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には (POLICY は、有効な Oracle Net データベース接続識別子とします)、次のように入力します。



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

ユーザー名 HR およびパスワード HR で SQL\*Plus を起動し、名前が STARTUP で拡張子が SQL のコマンド・ファイルを実行するには、次のように入力します。



```
SQLPLUS HR/HR @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、出力をファイルに獲得して、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/HR
```

セッション中、EDIT および HOST コマンドのアクセスを受け付けないように SQL\*Plus を起動するには、次のように入力します。



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

### 例 7-2 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 | [ [-M <o>] [-R <n>] [-S] ]
      <logon> ::= <username>[/<password>] [@<connect_identifier>] | / |/NOLOG
      <start> ::= @<uri>|<filename>[.<ext>] [<parameter> ...]
      -H displays the SQL*Plus version banner and usage syntax
      -V displays the SQL*Plus version banner
      -M <o> uses HTML markup options <o>
      -R <n> uses restricted mode <n>
      -S uses silent mode
```

## SQL\*Plus コマンドのヘルプ

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。次に例を示します。

```
HELP ACCEPT
```

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

```
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、[第 8 章](#)の「**HELP**」コマンドを参照してください。

---

## コマンド・リファレンス

この章では、SQL\*Plus のコマンドをアルファベット順に説明します。この章は、参照用に利用してください。それぞれのコマンドについて、次の内容を説明します。

<b>構文</b>	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、「はじめに」の「コード例中の表記規則」を参照してください。
<b>項</b>	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。
<b>使用方法</b>	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。
<b>例</b>	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。

長文の 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 (>) が個々の追加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
@	8-5	指定したコマンド・ファイルの SQL*Plus 文を実行する。コマンド・ファイルは、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る。
@@	8-7	コマンド・ファイルを実行する。@ (アットマーク) コマンドと同じ。コマンドをコールしたコマンド・ファイルと同じパス内で、指定のコマンド・ファイルを検索するため、ネストしたコマンド・ファイルの実行に有効。
/	8-9	SQL コマンドまたは PL/SQL ブロックを実行する。
ACCEPT	8-10	1 行の入力を読み取り、それを所定のユーザー変数に格納する。
APPEND	8-12	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する。
ARCHIVE LOG	8-13	オンライン REDO ログ・ファイルの自動アーカイブを開始または終了する、指定した REDO ログ・ファイルを手動で (明示的に) アーカイブする、または REDO ログ・ファイルの情報を表示する。
ATTRIBUTE	8-16	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リスト表示する。
BREAK	8-18	レポート内のどこでどのように書式設定を変更するかを指定する、または現行のブレイク定義をリスト表示する。
BTITLE	8-23	各レポート・ページの下部に指定のタイトルを配置し書式設定する、または現行の BTITLE 定義をリスト表示する。
CHANGE	8-24	バッファ内のカレント行のテキストを変更する。
CLEAR	8-27	指定したオプション (たとえば BREAKS や COLUMNS) などの現在の値または設定を、リセットまたは消去する。
COLUMN	8-29	特定の列の表示特性を指定するか、1 つまたはすべての列の現在の表示特性をリスト表示する。
COMPUTE	8-39	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出力するか、またはすべての COMPUTE 定義をリスト表示する。
CONNECT	8-44	指定したユーザー名で Oracle に接続する。
COPY	8-46	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する。
DEFINE	8-49	ユーザー変数を指定し、それに CHAR 値を割り当てるか、1 つまたはすべての変数の値および変数型をリスト表示する。

コマンド	ページ	説明
DEL	8-51	バッファの1つ以上の行を削除する。
DESCRIBE	8-53	指定した表、ビューまたはシノニムの列定義をリスト表示する、あるいは指定したファンクションまたはプロシージャの仕様をリスト表示する。
DISCONNECT	8-59	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 からログオフするが、SQL*Plus は終了しない。
EDIT	8-60	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・エディタを起動する。
EXECUTE	8-62	1つの PL/SQL 文を実行する。
EXIT	8-63	SQL*Plus を終了し、オペレーティング・システムに制御を戻す。
GET	8-65	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する。
HELP	8-66	SQL*Plus ヘルプ・システムにアクセスする。
HOST	8-67	SQL*Plus を終了せずに、ホスト・オペレーティング・システム・コマンドを実行する。
INPUT	8-69	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する。
LIST	8-71	バッファ内の1つ以上の行をリスト表示する。
PASSWORD	8-73	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する。
PAUSE	8-74	指定したテキストを表示し、その後ユーザーが [Return] を押すまで待つ。
PRINT	8-75	バインド変数の現在の値を表示する。
PROMPT	8-76	指定したメッセージをユーザーの画面に送る。
QUIT	8-63	SQL*Plus を終了し、オペレーティング・システムに制御を戻す。EXIT と同じ機能。
RECOVER	8-77	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メディア・リカバリを実行する。
REMARK	8-83	コマンド・ファイルの中でコメントを開始する。
REPFOOTER	8-84	各レポートの下部に指定のレポート・フッターを配置し書式設定するか、または現行の REPFOOTER 定義をリスト表示する。
REPHEADER	8-86	各レポートの上部に指定のレポート・ヘッダーを配置し書式設定するか、または現行の REPHEADER 定義をリスト表示する。

コマンド	ページ	説明
RUN	8-90	現在、バッファに格納されている SQL コマンドまたは PL/SQL ブロックをリスト表示し実行する。
SAVE	8-91	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（コマンド・ファイル）に保存する。
SET	8-93	現行セッション用に SQL*Plus 環境を変更するためのシステム変数を設定する。
SHOW	8-118	SQL*Plus システム変数の値または現行の SQL*Plus 環境を表示する。
SHUTDOWN	8-123	現在実行中の Oracle インスタンスを停止する。
SPOOL	8-125	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る。
START	8-126	指定したコマンド・ファイルの SQL*Plus 文を実行する。コマンド・ファイルは、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る。
STARTUP	8-128	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する。
STORE	8-131	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（コマンド・ファイル）に保存する。
TIMING	8-132	経過時間に関するタイミング・データを記録するか、現行のタイマーのタイトルとタイミング・データをリスト表示するか、アクティブ・タイマーの数をリスト表示する。
TTITLE	8-134	各レポート・ページの上部に指定のタイトルを配置し書式設定する、または現行の TTITLE 定義をリスト表示する。
UNDEFINE	8-137	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する。
VARIABLE	8-138	PL/SQL で参照できるバインド変数を宣言する。
WHENEVER OSERROR	8-145	オペレーティング・システム・コマンドがエラーを生成した場合に、SQL*Plus を終了する。
WHENEVER SQLEERROR	8-147	SQL コマンドまたは PL/SQL ブロックがエラーを生成した場合、SQL*Plus を終了する。

## @ (アットマーク)

### 構文

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

指定したコマンド・ファイルの SQL\*Plus 文を実行します。コマンド・ファイルは、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。uri は、このリリースでは、Windows プラットフォームのみで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。

#### uri

指定した Web サーバーで実行するスクリプトの Uniform Resource Identifier (URI) 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]

実行するコマンド・ファイルを指定します。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。

@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。該当の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。

#### arg...

コマンド・ファイル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、該当の値をコマンド・ファイルのパラメータ (&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コマンド・ファイルを再度実行する場合には、新しい引数を入力するか、現行の値を使用する場合は引数を省略することができます。

パラメータの使用方法の詳細は、第 3 章の「対話型コマンドの作成」の「START コマンドを使用した値の指定」を参照してください。

## 使用方法

コマンド・ファイルには、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、どのコマンドでも挿入できます。

コマンド・ファイル内で 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と（付録 E 「セキュリティ」の「SQL\*Plus、SQL および PL/SQL コマンドの使用禁止」を参照）、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、この章の「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 であるコマンド・ファイルを実行するには、次のように入力します。



```
@PRINTRPT
```

名前が WKRPT で拡張子が QRY であるコマンド・ファイルを実行するには、次のように入力します。



```
@WKRP.T.QRY
```

次のようにして、URI で指定して YEAREND という名前のスクリプトを実行し、値を通常どおりの方法で YEAREND で参照する変数に渡せます。



```
@HTTP://HOST.DOMAIN/YEAREND.SQL VAL1 VAL2  
@FTP://HOST.DOMAIN/YEAREND.SQL VAL1 VAL2  
@GOPHER://HOST.DOMAIN/YEAREND.SQL VAL1 VAL2
```

SQL レポートを作成するために構成された Web サーバーでは、次のように入力して、SQL\*Plus に動的スクリプトを実行するように要求します。



```
@HTTP://HOST.DOMAIN/SCRIPTSERVER?ENDOFYEAR VAL1 VAL2
```



## @@ (二重アットマーク)

### 構文

`@@file_name[.ext]`

コマンド・ファイルを実行します。このコマンドは、@ (アットマーク) コマンドと同じです。このコマンドがコールされたコマンド・ファイルと同じパスまたは `uri` 内で、指定のコマンド・ファイルを検索する追加の機能があるため、ネストしたコマンド・ファイルの実行に有効です。`uri` は、このリリースでは、Windows プラットフォームのみで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。

`file_name[.ext]`

実行するネストしたコマンド・ファイルを指定します。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。

あるコマンド・ファイルの中で `@@file_name.ext` を入力すると、SQL\*Plus は、そのコマンド・ファイル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から、またはこのコマンドがコールされたコマンド・ファイルと同じ `uri` から `file_name.ext` を実行します。該当のファイルが見つからない場合、SQL\*Plus は、システム依存パスを検索します。オペレーティング・システムによっては、このパス検索がサポートされていない場合もあります。オペレーティング・システム固有の情報については、ご使用のオペレーティング・システム固有の 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

コマンド・ファイルには、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、どのコマンドでも挿入できます。

コマンド・ファイル内で 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と、同時に @@ コマンドも使用禁止になります。詳細は、この章の後半にある「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。これを回避するには、SQLTERMINATOR をもう 1 つ追加します。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

PRINTRPT という次のようなコマンド・ファイルがあるとします。



```
SELECT DEPARTMENT_ID, CITY FROM EMP_DETAILS_VIEW WHERE SALARY>12000;  
@EMPRPT  
@@ WKRPT
```

PRINTRPT を開始し、それが @ コマンドに到達すると、現行の作業ディレクトリ内で EMPRPT という名前のコマンド・ファイルが検索され、実行されます。PRINTRPT が @@ コマンドに到達すると、このコマンドは、PRINTRPT と同じパスの中で WKRPT というコマンド・ファイルを検索し、実行します。

同じコマンド・ファイル PRINTRPT が Web サーバーにあり、START HTTP://HOST.DOMAIN/PRINTRPT で実行したとします。@ コマンドに到達すると、現行のローカル作業ディレクトリ内で EMPRPT という名前のコマンド・ファイルが検索され、実行されます。PRINTRPT が @@ コマンドに到達すると、このコマンドは、PRINTRPT と同じ uri の中で WKRPT というコマンド・ファイルを検索し、実行します。

## / (スラッシュ)

### 構文

/

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、カレント行の番号をエラーがある行の番号に変更します。

### 例

次の SQL 問合せを入力します。



```
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

### 構文

```
ACC[EPT] variable [NUM[BER] |CHAR|DATE] [FOR[MAT] format] [DEF[AULT] default]  
[PROMPT text|NOPR[OMPT]] [HIDE]
```

1 行の入力を読み取り、それを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、変数は、SQL\*Plus によって作成されます。

#### 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「COLUMN」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**

ユーザーからの変数の値を受け入れる前に、画面に *text* が表示されます。

**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、その表示を抑止するには、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: '
```

「Enter employee lastname:」というプロンプトを表示し、`lastname` という名前の **CHAR** 変数に応答を入れるには、次のように入力します。



```
ACCEPT lastname CHAR FORMAT 'A20' -
PROMPT 'Enter employee lastname: '
```

# APPEND

## 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わらせませす（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

1 つの空白と CITY という列名をバッファの最初の行に追加するには、その行をカレント行にするために、次のようにその行をリスト表示します。



```
1
```



```
1* SELECT DEPARTMENT_ID,
```

次に、APPEND を入力します。



```
APPEND CITY
```

```
1
```



```
1* SELECT DEPARTMENT_ID, CITY
```

APPEND と CITY の間に空白が 2 つあることに注意してください。最初の空白は APPEND と追加する文字を区切るためのものです。2 番目の空白が、追加される最初の文字になります。

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 番目のセミコロンは APPEND コマンドの終了記号と解釈します。

## 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・ファイルの手動での（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。

ログ順序は、ログ・ライターが他の REDO ログ・ファイル・グループに書き込みを開始するたびに、毎回増加します。ログ順序は、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、「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 コマンドの使用の詳細は、『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リスト表示します。

## 項

指定した属性についてのみ現行の表示特性をリスト表示するには、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リスト表示するには、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 または \$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 または \$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、属性の特性が使用禁止になります。ON を指定すると、特性が復元されます。

## 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_element   構文は次のとおりです。
                  {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]]
```

レポート内で次のような書式設定をどこでどのように変更するかを指定します。

- 特定の列について重複する値の表示の抑止
- 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)

また、現行の BREAK 定義もリスト表示します。

## 項

項または句の詳細は、次の説明を参照してください。

### 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) の中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* によって、*column* の中で重複する値の出力が抑止され、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* の重複値の出力を抑止し、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 が行うレポート内の位置にマークが設定されます。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。

**[SKI [P]] PAGE**

ブレイクが発生した行を出力する前に、1 ページ分として定義されている行数のみスキップします。1 ページの行数は、SET コマンドの PAGESIZE 句を使用して設定できます。PAGESIZE は、SQL\*Plus が 1 ページとみなす行数を変更するのみです。したがって、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・ブレイクが行われません。

**NODUP [LICATES]**

あるブレイク列の値が前の行の列値と同じ場合は、その値ではなく空白が出力されます。

**DUP [LICATES]**

選択した各行のブレイク列の値が出力されます。

現行のブレイク定義をリスト表示するには、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、再び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	Fleur	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 行が選択されました。



# BTITLE

## 構文

```
BTI [TLE] [printspec [text|variable] ...] [ON|OFF]
```

各レポート・ページの下部に指定のタイトルを配置し書式設定する、または現行の BTITLE 定義をリスト表示する。

旧形式の BTITLE の詳細は、[付録 F「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 定義をリスト表示するには、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'
```

750 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

### 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*

「/」または「!」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* CHANGE/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は、CLEAR BUFFER の働きは CLEAR SQL と同じです（付録 F「[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、該当の列の 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。

## 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 F「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

ブレークを消去するには、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]
ENIMAP {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_WRAPPED] |TRU[NCATED]
```

特定の列について次のような表示属性を指定します。

- 列ヘッダーのテキスト
- 列ヘッダーの位置
- NUMBER データの書式
- 列データの折返し

さらに、1 つまたはすべての列の現行の表示属性をリスト表示します。

## 項

指定した列または式についてのみ現行の表示属性をリスト表示するには、*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リスト表示するには、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せん。

**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、該当の列の 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、第7章の「[MARKUP オプション](#)」およびこの章の後半の「[SET](#)」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に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 または \$9.999 などのテキスト定数である必要があります。

**文字列** CHAR、NCHAR、VARCHAR2 (VARCHAR) および NVARCHAR2 (NCHAR VARYING) 列のデフォルトの幅は、データベース内の列の幅です。SQL\*Plus は、これらのデータ型を左揃えで書式設定します。値が列幅に収まらない場合、SQL\*Plus は、SET WRAP の設定に応じて、文字列を折り返すか切り捨てます。

LONG 列または CLOB 列、NCLOB 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。LONG、CLOB または NCLOB のいずれかの列に幅を指定した場合は、LONGCHUNKSIZE、または指定した幅のどちらか小さい方が列幅として使用されます。

**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 8-1 に示す要素の 1 つを指定します。

**表 8-1 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する。先行 0 (ゼロ) には空白が表示される。0 (ゼロ) の値には 0 (ゼロ) が表示される。
0	0999 9990	先行 0 (ゼロ) またはこの桁の 0 (ゼロ) の値は、0 と表示される。
\$	\$9999	値の前にドル記号が付けられる。
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る。
MI	9999MI	負の値の後に「-」が表示される。正の値の場合は、後続空白が表示される。
S	S9999	正の値の場合は「+」、負の値の場合は、「-」が、この桁に戻される。
PR	9999PR	負の値は <山カッコ> で囲まれて表示される。正の値の場合は、先行空白および後続空白が表示される。
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る。
G	9G999	この桁には、桁グループ・セパレータが表示される。
C	C999	この桁には、ISO 通貨記号が表示される。
L	L999	この桁には、各国通貨記号が表示される。
, (カンマ)	9,999	この桁にはカンマが表示される。
. (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) が表示される。
V	999V99	値に 10 <sup>n</sup> を掛ける。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る (書式には確実に 4 つの“E”を含める必要がある)。

表 8-1 数値書式

要素	例	説明
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る。値は 1 ～ 3999 の範囲内の整数。
DATE	DATE	値が MM/DD/YY 書式の日付として表示される。これは、ユリウス暦日付を表す NUMBER 列を書式設定するために使用される。

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 FORMAT 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 / ディテール・レポートに使用すると便利です。マスター / ディテール・レポートを作成するには、ORDER BY 句にもこの列を挿入しておく必要があります。

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ることなく列の属性が使用禁止にされます。ON を指定すると、属性が復元されます。

**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 が使用可能になっている場合、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 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。



```
COLUMN SALARY FORMAT $9,999,990.99
```

長い式を含む列に別名 NET を割り当て、結果をドル書式で表示し、NULL 値には <NULL> を表示するには、次のように入力します。



```
COLUMN SALARY+COMMISSION_PCT+BONUS-EXPENSES-INS-TAX ALIAS NET
COLUMN NET FORMAT $9,999,999.99 NULL '<NULL>'
```

この例では、この列指定を 2 つのコマンドに分けています。最初のコマンドは別名 NET を定義し、2 番目のコマンドは NET を使用して書式を定義しています。

また、最初のコマンドでは、SELECT 文で入力したとおり、正確に式を入力する必要があります。そうしないと、SQL\*Plus は、COLUMN コマンドを適正な列に対応付けることができません。

REMARKS という名前の列の中の長い値を折り返すには、次のように入力します。



```
COLUMN REMARKS FORMAT A20 WRAP
```



CUSTOMER	DATE	QUANTITY	REMARKS
123	25-AUG-2001	144	This order must be shipped by air freight to ORD

WRAP を WORD\_WRAP で置換した場合、REMARKS は次のようになります。

CUSTOMER	DATE	QUANTITY	REMARKS
123	25-AUG-2001	144	This order must be shipped by air freight to ORD

TRUNCATE を指定した場合、REMARKS は次のようになります。

```
CUSTOMER  DATE          QUANTITY REMARKS
-----
123       25-AUG-2001      144 This order must be s
```

現行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、今まで使用していた 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、第4章の「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きます。



```

                                     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          80
                                     -----
                                     $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行が選択されました。

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';
```



セッションが変更されました。

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** 定義をリスト表示します。(サマリーの作成方法の詳細は、[第 4 章の「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 8-2](#) 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 8-2 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、省略されていない関数キーワードです。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。ラベルを抑止するには、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、ラベルを抑止できます。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 定義をリスト表示するには、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。

## 例

AC\_MGR および SA\_MAN 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 行が選択されました。

レポート上の 12,000 を超える給与の合計を計算するには、次のように入力します。



```
COMPUTE SUM OF SALARY ON REPORT
BREAK ON REPORT
COLUMN DUMMY HEADING ''
SELECT ' ' DUMMY, SALARY, EMPLOYEE_ID
FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000
ORDER BY SALARY;
```



```

          SALARY EMPLOYEE_ID
-----
          13000          201
          13500          146
          14000          145
          17000          101
          17000          102
          24000          100
-----
sum          98500
```

6 行が選択されました。

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 行が選択されました。

## 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。

### 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、デフォルトのログインを使用してユーザーをログインします (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せん。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については、この章の「PASSWORD」コマンドを参照してください。

#### **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せてください。SQL\*Plus は、データベース指定については、プロンプトを表示せず、その指定がないとき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・ログインを表します。デフォルト・ログインを使用する場合、*connect\_identifier* を指定できません。デフォルト・ログインでは、SQL\*Plus はユーザー名 *OPSS\$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・ログインである / (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イ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、「CREATE」コマンドおよび「ALTER USER」コマンドを参照してください。また『Oracle9i SQL リファレンス』の「CREATE PROFILE」コマンドも参照してください。

## 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/HR@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/HR AS SYSDBA
```

現行ノードのインスタンスにデフォルト・ユーザーの権限で接続するには、次のように入力します。



```
CONNECT / AS SYSDBA
```

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

## 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
```

COPY コマンドでは、Oracle8 以上で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。

## 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。FROM 句を挿入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 句を挿入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。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 変数 LONG 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 変数 COPYCOMMIT 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 コマンドの 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、その表およびその内容を置換します。WESTEMPLOYEES 中の列は、ソース表 EMPLOYEES 中の列と同じ名前になります。



```
COPY FROM HR/HR@HQ TO JOHN/CHROME@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、コピーの際に 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/HR@ORACLE01 -  
CREATE SALESMEN (EMPLOYEE_ID, SA_MAN) -  
USING SELECT EMPLOYEE_ID, LAST_NAME FROM EMPLOYEES -  
WHERE JOB_ID='SA_MAN';
```

# DEFINE

## 構文

```
DEF [INE] [variable] | [variable = text]
```

ユーザー変数を指定し、それに CHAR 値を割り当てるか、1 つまたはすべての変数の値および変数型をリスト表示します。

## 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（またはリスト表示するか）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* の値と型をリスト表示するには、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リスト表示するには、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・クエリーまたはコマンド・ファイルを実行するたびに、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` の定義をリスト表示するには、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

## 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 $n$ が削除されます。
<code>n m</code>	行 $n \sim m$ が削除されます。
<code>n *</code>	行 $n$ からカレント行までが削除されます。
<code>n LAST</code>	行 $n$ 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 $n$ 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リスト表示するか、あるいは指定したファンクションまたはプロシージャの仕様をリスト表示します。

## 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*

記述する表、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記述には、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 値が許されるかどうか (NULL か NOT NULL か)。
- 列のデータ型 (NUMBER、CHAR、VARCHAR2 (VARCHAR)、LONG、DATE、RAW、LONGRAW または ROWID)
- 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 コマンドでは、SET DESCRIBE コマンドの深さレベルの設定に、オブジェクトを繰り返し記述できます。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。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記述には、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
```



```
PROCEDURE customer_lookup
Argument Name      Type      In/Out  Default?
-----
CUST_ID            NUMBER    IN
CUST_NAME          VARCHAR2  OUT
```



プロシージャ `aproc` および `bproc` を含むパッケージ `APACK` を作成し、この定義を表示するには、次のように入力します。



```
CREATE PACKAGE apack AS
PROCEDURE aproc(P1 CHAR, P2 NUMBER);
PROCEDURE bproc(P1 CHAR, P2 NUMBER);
END apack;
```



```
/
パッケージが作成されました。
```



```
DESCRIBE apack
```



```
PROCEDURE APROC
Argument Name                Type                In/Out Default?
-----
P1                            CHAR                IN
P2                            NUMBER              IN
PROCEDURE BPROC
Argument Name                Type                In/Out Default?
-----
P1                            CHAR                IN
P2                            NUMBER              IN
```

`STREET` 属性および `CITY` 属性を含むオブジェクト型 `ADDRESS` を作成し、この定義を表示するには、次のように入力します。



```
CREATE TYPE ADDRESS AS OBJECT
( STREET VARCHAR2(20),
  CITY   VARCHAR2(20)
);
```



```
/
型が作成されました。
```



```
DESCRIBE address
```



```
Name                Null?  Type
-----
STREET              VARCHA(20)
CITY                VARCHA(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)
);
/
```



型が作成されました。



```
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;
/
```



型が作成されました。



```
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;
/
```



型が作成されました。



```
DESCRIBE addr_varray
```



```
addr_varray VARRAY(10) OF ADDRESS
Name                                         Null?      Type
-----
STREET                                       VARRAY2 (20)
CITY                                         VARRAY2 (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
);
/
```



表が作成されました。



```
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

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 LINESUM 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)

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。

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が、SQL\*Plus を終了しません。

## 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、コマンド・ファイルの内部で 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、EXIT または QUIT を使用します。

## 例

次に示すように、コマンド・ファイルを CONNECT コマンドで開始し、DISCONNECT コマンドで終了することができます。



```
CONNECT HR/HR
SELECT LAST_NAME, DEPARTMENT_NAME FROM EMP_DETAILS_VIEW;
DISCONNECT
SET INSTANCE FIN2
CONNECT HR2/HR2
```

# EDIT

## 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（通常はコマンド・ファイル）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、指定の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「**DEFINE**」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・エディタの通常の方法に従って編集セッションを終了します。エディタを終了するとき、SQL\*Plus はファイル名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
```

単一の 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 プロシージャが正常に完了しました。

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]
```

SQL\*Plus を終了し、オペレーティング・システムに制御を戻します。

## 項

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) を表します。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。したがって、SQL\*Plus コマンド・ファイルをバッチ・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。

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での 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 に制限されます。

EXIT オプションに構文エラーがあった場合または非数値変数を使用した場合は、SQL\*Plus によって EXIT FAILURE COMMIT が実行されます。

条件付きの終了については、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
```

リターン・コードの位置は、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・システムがプログラムからデータを取り出す方法については、DBA に問い合せてください。SQL.SQLCODE の詳細は、この章の「[TTITLE](#)」コマンドを参照してください。

# GET

## 構文

```
GET 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\_name* [.ext]**

ロードするファイル（通常はコマンド・ファイル）を指定します。

**LIS[T]**

ロード後、ファイルの内容がリスト表示されます。これがデフォルトです。

**NOL[IST]**

リスト表示が抑止されます。

## 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 ヘルプ・システムにアクセスします。項目をリスト表示する場合は、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

## 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。

---

---

**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抑止できます。HOST コマンドの抑止の詳細は、[付録 E「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。

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 コマンドが実行される前に SQLTERMINATOR (デフォルトではセミコロン) を削除します。これを回避するには、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

## 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 つ以上の行をリスト表示します。

## 項

項または句の詳細は、次の説明を参照してください。

<i>n</i>	行 <i>n</i> がリスト表示されます。
<i>n m</i>	行 <i>n</i> から行 <i>m</i> がリスト表示されます。
<i>n *</i>	行 <i>n</i> からカレント行までがリスト表示されます。
<i>n LAST</i>	行 <i>n</i> から最終行までがリスト表示されます。
<i>*</i>	カレント行がリスト表示されます。
<i>* n</i>	カレント行から行 <i>n</i> までがリスト表示されます。
<i>* LAST</i>	カレント行から最終行までがリスト表示されます。
<i>LAST</i>	最終行がリスト表示されます。

すべての行をリスト表示するには、句を指定しないで **LIST** のみを入力します。リスト表示された最終行またはリスト表示された 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

バッファの内容のリスト表示するには、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リスト表示されます。

```
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）から最終行までをリスト表示するには、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

## 構文

```
PASSW[ORD] [username]
```

入力デバイスにパスワードをエコー表示させないで、パスワードを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* はデフォルトで現在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

HR/HR でログインしていて、パスワードを HRHR に変更する場合は、次のようにします。



```
PASSWORD
Changing password for HR
Old password: HR
New password: HRHR
Retype new password: HRHR
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
```

---

**注意：** 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

### 構文

PAU[SE] [*text*]

空白行を 1 行と、それに続いてテキストを含む行を 1 行表示し、ユーザーが [Return] を押すまで待つか、空白行を 2 行表示しユーザーの応答を待ちます。

### 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、コマンド・ファイル内に次の 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 ...]
```

バインド変数の現在の値を表示します。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 4 章](#)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**」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 プロシージャが正常に完了しました。
```



```
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ります。

## 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

このコマンドは、ユーザーに情報を与えるために、コマンド・ファイル内で使用できます。

## 例

次の例は、ASKFORDEPT というコマンド・ファイル内で ACCEPT とともに PROMPT を使用していることを示しています。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す。



```
PROMPT
PROMPT Please enter a valid department
PROMPT For example:  10, 20, 30, 40
ACCEPT NEWDEPT NUMBER PROMPT 'Department ID?> '
SELECT DEPARTMENT_NAME FROM EMP_DETAILS_VIEW
WHERE DEPARTMENT_ID = &NEWDEPT
```

START または@を使用して、このファイルを実行するとします。



```
@ASKFORDEPT
```



```
Please enter a valid department
For example:  10, 20, 30, 40
Department ID?>
```

プロンプト Department ID?> が表されたら、部門番号を入力できます。デフォルトでは、SQL\*Plus は置換の前および後で& NEWDEPT を含む行をリスト表示します。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。

## 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、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。対話方式のリカバリ・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**

キャンセル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・ログをただちに適用します。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・ログはただちに適用されます。

**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 clause)**

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、セッションに制御をただちに戻します。

#### **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 ログがただちに適用されます。

#### **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]
```

コマンド・ファイルの中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。

コマンド・ファイルに、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、[第 3 章の「コマンド・ファイルへのコメントの挿入」](#)を参照してください。

### 例

次のコマンド・ファイルには、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 [OOTER] [PAGE] [printspec [text|variable] ...] | [ON|OFF]
```

各レポートの下部に指定のレポート・フッターを配置し書式設定するか、または現行の REPFOOTER 定義をリスト表示します。

### 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 定義をリスト表示するには、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時と同じ順序で表示し、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・フッターは抑止されます。

---

### 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行が選択されました。

レポート・フッターの定義を変更しないで、レポート・フッターを抑止するには、次のように入力します。



REPFOOTER OFF

REPHEADER 表示を停止するには、次のように OFF に設定します。



REPHEADER OFF

## REPHEADER

### 構文

```
REP[EADER] [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のレポート・ヘッダーを配置し書式設定するか、または現行のREPHEADER 定義をリスト表示します。

### 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

指定のレポート・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.RELEASE は現行の Oracle リリース番号です。SQL.CODE は現行のエラー・コードです。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**

定義に影響を与えないで、レポート・ヘッダーまたはレポート・フッターをオフ（表示を抑制）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（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、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、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の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。

指定のデータ項目の前に適切な FORMAT 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 定義をリスト表示するには、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時と同じ順序で表示し、それぞれの定数または変数を、先行する *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 行が選択されました。

レポート・ヘッダーの定義を変更せずにその出力を抑止するには、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、リスト表示し実行します。

## 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。

## 例

SQL バッファに次の問合せが含まれているとします。



```
SELECT DEPARTMENT_ID
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000
```

RUN を使用して問合せを行うには、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 行が選択されました。

# SAVE

## 構文

```
SAVE[E] 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・ファイル（コマンド・ファイル）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\_name* [.ext]**

バッファ内容を保存するコマンド・ファイルを指定します。

### **CREATE**

指定された名前で、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]**

既存のファイル内容が置換されます。ファイルが存在しない場合、REPLACE はファイルを作成します。

### **APP [END]**

指定したファイルの終わりに、バッファの内容が追加されます。

## 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。

指定するファイル名が *file* という語の場合は、名前を一重引用符で囲む必要があります。

**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

## 構文

SET *system\_variable* *value*

*system\_variable* および *value* は、次のいずれかの句を表します。

```

APPI [INFO] {ON|OFF|text}
ARRAY [SIZE] {15|n}
AUTO [COMMIT] {ON|OFF|IMM[EDIATE] |n}
AUTOP [RINT] {ON|OFF}
AUTORECOVERY [ON|OFF]
AUTOT [RACE] {ON|OFF|TRACE [ONLY]} [EXP [LAIN]] [STAT [ISTICS]]
BLO [CKTERMINATOR] {.|c}
CMDS [EP] {;|c|ON|OFF}
COLSEP {_|text}
COM [PATIBILITY] {v7|v8|NATIVE}
CON [CAT] {.|c|ON|OFF}
COPYC [OMMIT] {0|n}
COPYTYPECHECK {ON|OFF}
DEF [INE] {&|c|ON|OFF}
DESCRIBE [DEPTH {1|n|ALL}] [LINENUM {ON|OFF}] [INDENT {ON|OFF}]
ECHO {ON|OFF}
EDITF [ILE] file_name [.ext]
EMB [EDED] {ON|OFF}
ESC [APE] {\|c|ON|OFF}
FEED [BACK] {6|n|ON|OFF}
FLAGGER {OFF|ENTRY |INTERMED [IATE] |FULL}
FLU [SH] {ON|OFF}
HEA [DING] {ON|OFF}
HEADS [EP] {||c|ON|OFF}
INSTANCE [instance_path|LOCAL]
LIN [ESIZE] {80|n}
LOBOF [FSET] {n|1}
LOGSOURCE [pathname]
LONG {80|n}
LONGC [HUNKSIZE] {80|n}
MARK [UP] HTML [ON|OFF] [HEAD text] [BODY text] [TABLE text] [ENIMAP
{ON|OFF}] [SPOOL {ON|OFF}] [PRE [FORMAT] {ON|OFF}]
NEWP [AGE] {1|n|NONE}
NULL text
NUMF [ORMAT] format
NUM [WIDTH] {10|n}
PAGES [IZE] {24|n}
PAU [SE] {ON|OFF|text}
RECSEP {WR [APPED] |EA [CH] |OFF}

```

```
RECSEPCHAR {_|c}
SERVEROUT [PUT] {ON|OFF} [SIZE n] [FOR [MAT]
{WRA [PPED] |WOR [D_WAPPED] |TRU [NCATED] }]
SERVEROUT [PUT] {ON|OFF} [SIZE n] [FOR [MAT]
{WRA [PPED] |WOR [D_WAPPED] |TRU [NCATED] }]
SHIFT [INOUT] {VIS [IBLE] |INV [ISIBLE] }
SHOW [MODE] {ON|OFF}
SQLBL [ANKLINES] {ON|OFF}
SQLC [ASE] {MIX [ED] |LO [WER] |UP [PER] }
SQLCO [NTINUE] {> |text}
SQLN [UMBER] {ON|OFF}
SQLPLUSCOMPAT [IBILITY] {x.y[.z] }
SQLPRE [FIX] {#|c}
SQLP [ROMPT] {SQL>|text}
SQLT [ERMINATOR] {;|c|ON|OFF}
SUF [FIX] {SQL|text}
TAB {ON|OFF}
TERM [OUT] {ON|OFF}
TI [ME] {ON|OFF}
TIMI [NG] {ON|OFF}
TRIM [OUT] {ON|OFF}
TRIMS [POOL] {ON|OFF}
UND [ERLINE] {-|c|ON|OFF}
VER [IFY] {ON|OFF}
WRA [P] {ON|OFF}
```

次に、現行のセッションに対する SQL\*Plus 環境設定を変更するシステム変数の例を示します。

- NUMBER 型データの表示幅
- HTML 形式でのチューニング
- 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



## 項

項、句またはシステム変数の詳細は、次の説明を参照してください。

### APPI [INFO] {ON|OFF|text}

DBMS\_APPLICATION\_INFO パッケージを介してコマンド・ファイルが自動登録されるように設定します。この設定によって、各コマンド・ファイル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コマンド・ファイルが登録されます。OFF を指定すると、コマンド・ファイル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、コマンド・ファイル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、コマンド・ファイルの深さレベルです。x は、コマンド・ファイル名を切り捨てる場合は「<」、そうでない場合は空白です。filename はコマンド・ファイル名で、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（この名前の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・ガイドおよびリファレンス』を参照してください。

### ARRAY [SIZE] {15|n}

SQL\*Plus が一度にデータベースからフェッチす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。

**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、自動コミットが抑止されるので、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す。

---

---

**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。

**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、予期できる名前ですべての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。

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選択できます。データベース・リカバリの詳細は、「RECOVER」コマンドを参照してください。

**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) のレポートが表示されます。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すが、問合せデータは、存在しても出力されません。EXPLAIN を指定すると、EXPLAIN PLAN が実行されて、問合せ実行パスが表示されます。STATISTICS を指定すると、SQL 文統計が表示されます。EXPLAIN PLAN の詳細は、『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・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せん。

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、第 3 章の「文のトレース」を参照してください。

**BLO[CKTERMINATOR] {\_ | c}**

c には、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。

**CMDS [EP] {; | c | ON | OFF}**

c には、1 行に入力した複数の 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 (;) に設定されます。

**COLSEP** {\_ | *text*}

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。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す。詳細は、この章の「[SET RECSEP](#)」を参照してください。

**COM[PATIBILITY]** {V7 | V8 | **NATIVE**}

現行の接続先である Oracle 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。データベースによって設定を判断させるには、COMPATIBILITY を NATIVE に設定します (Oracle9i に接続する場合、デフォルトで NATIVE に設定されます)。COMPATIBILITY を、接続先の Oracle 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。

---

**注意：** Oracle9i に接続している場合でも、COMPATIBILITY に V7 または V8 を設定することができます。これによって、Oracle7 SQL、Oracle8 または Oracle8i SQL を Oracle9i に対して実行できます。

---

**CON[CAT]** {. | c | **ON** | **OFF**}

通常なら、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、置換変数の指定を終了させ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。

**COPYC[OMMIT]** {0 | n}

COPY コマンドで、データベースに対する変更をコミットするときの基準となる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。

**COPYTYPECHECK** {**ON** | **OFF**}

COPY コマンドを使用して表にデータを挿入または追加する間は、データ型の比較が抑止されます。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。

**DEF[INE] {&|c|ON|OFF}**

cには、接頭辞の置換変数に使用される文字を設定します。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、付録 F「廃止された SQL\*Plus コマンド」の「SET SCAN」コマンドを参照してください。

**DESCRIBE [DEPTH {1|n|ALL}] [LINENUM {ON|OFF}] [INDENT {ON|OFF}]**

オブジェクトを繰り返し説明するための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。

オブジェクト説明の詳細は、この章の「DESCRIBE」コマンドを参照してください。

**ECHO {ON|OFF}**

START コマンドの実行時に、コマンド・ファイル内の各コマンドの実行時にそのコマンドをリスト表示するかどうかを制御します。ON を指定すると、コマンドがリスト表示されます。OFF を指定すると、リスト表示が抑止されます。

**EDIT[ILE] file\_name[.ext]**

EDIT コマンド用のデフォルトのファイル名を設定します。EDIT コマンドの詳細は、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、このコマンドの 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。

**EMB[EDDED] {ON|OFF}**

ページのどこから各レポートが始まるかを制御します。OFF を指定すると、各レポートは新しいページの一番上から始まります。ON を指定すると、新しいページのどこからでもレポートを開始できます。直前のレポートが終了した直後に次のレポートを出力するには、EMBEDDED を ON に設定します。

**ESC [APE] { \ | c | ON | OFF }**

エスケープ文字として入力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。

**FEED [BACK] { 6 | n | ON | OFF }**

問合せによって n 個以上のレコードを選択する場合に、問合せから戻されたレコードの数を表示します。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。

**FLAGGER { OFF | ENTRY | INTERMED [IATE] | FULL }**

SQL 文が ANSI/ISO SQL92 規格に準拠していることを確認するためのチェックが実行されます。規格外の構造体が見つかったと、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 コマンドと同等に機能します。

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。

**FLU [SH] { ON | OFF }**

出力をユーザーの表示デバイスに、いつ送るかを制御します。OFF を指定すると、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、コマンド・ファイルを非対話形式で実行する場合 (コマンド・ファイルの実行が完了するまで出力またはプロンプト、また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。

**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、列ヘッダーの出力が抑止されます。

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、列ヘッダーの出力を抑止するのみです。

**HEADS [EP] {|c|ON|OFF}**

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。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。

**INSTANCE [*instance\_path*|LOCAL]**

指定したインスタンス・パスを、ご使用のセッション用にデフォルト・インスタンスに変更します。SET INSTANCE コマンドを使用して、データベースへは接続しません。デフォルト・インスタンスは、インスタンスが何も指定されていないときにコマンドに対して使用されます。

SET INSTANCE を最初に使用する前の一部のコマンドは、デフォルト・インスタンスと通信します。

ご使用のオペレーティング・システムでデフォルト値のインスタンスをリセットするには、*instance\_path* を指定しないで SET INSTANCE を入力するか、または SET INSTANCE LOCAL を入力します。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。

**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。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。

**LOBOF[FSET] {*n*|1}**

CLOB および NCLOB データを取り出して表示する開始位置を設定します。

**LOGSOURCE [*pathname*]**

リカバリ中に取り出されたアーカイブ・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されます。

**LONG {80|*n*}**

LONG、CLOB および NCLOB 値を表示する場合、および LONG 値をコピーする場合の最大幅を（バイト単位で）設定します。*n* の最大値は、2GB です。

**LONGC[HUNKSIZE] {80|*n*}**

SQL\*Plus が LONG、CLOB または NCLOB 値を取り出すときの増分値のサイズを（バイト単位で）設定します。

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

HTML マークアップ・テキストを出力します。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 出力の書き込みを開始するには、SET MARKUP HTML ON SPOOL ON および SQL\*Plus の SPOOL コマンドを使用する必要があります。SET MARKUP のオプションと動作は、SQLPLUS-MARKUP と同じです。

詳細は、第 7 章の「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、8-113 ページの「MARKUP」および第 4 章の「Web レポートの作成」を参照してください。

**NEWP[AGE] {1|*n*|NONE}**

各ページの最上部から上部タイトルまでの間に入れる空白行の数を設定します。値 0 を指定すると、(最初のページも含めて) 各ページの開始位置に改ページ文字が設定され、ほとんどの端末で画面がクリアされます。NEWPAGE を NONE に設定した場合、SQL\*Plus はレポートのページ間で空白行または改ページ文字を出力しません。



**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。指定の列の 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。

**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。

**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。

**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抑止できます。

**PAU [SE] { ON | OFF | *text* }**

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。

**RECSEPCHAR { \_ | *c* }**

レコード・セパレータを表示または出力します。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。

RECSEPCHAR には、レコードセパレータ文字を定義します。デフォルトは、1 個の空白です。

**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。

**SERVEROUT [PUT] {ON|OFF} [SIZE n] [FOR [MAT]  
{WRA [PPED] | WOR [D\_WAPPED] | TRU [NCATED] }]**

SQL\*Plus のストアド・プロシージャまたは PL/SQL ブロックの出力 (DBMS\_OUTPUT.PUT\_LINE) を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 の出力が抑止されます。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 1,000,000 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。

**SHIFT [INOUT] {VIS [IBLE] | INV [ISIBLE] }**

シフト文字を表示する端末に適した位置合せを実現します。SET SHIF TINOUT 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。

**SHOW [MODE] {ON|OFF}**

SET を使用して、SQL\*Plus システム変数の設定を変更したときに、新旧の設定値を SQL\*Plus でリスト表示するかどうかを制御します。ON を指定すると、設定値がリスト表示されます。OFF を指定すると、リスト表示が抑止されます。SHOWMODE ON の動作は、すでに廃止された SHOWMODE BOTH と同じです。

**SQLBL[ANKLINES] {ON|OFF}**

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。

**SQLC[ASE] {MIX[ED] | LO[WER] | UP[PER]}**

実行の直前に、SQL コマンドおよび PL/SQL ブロックの大文字や小文字を変換します。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 パッファの内容自体は変更されません。

**SQLCO[NTINUE] {≥ | text}**

1 つの SQL\*Plus コマンドをハイフン (-) を使用して追加行に続けた後で、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。

**SQLN[UMBER] {ON|OFF}**

SQL コマンドまたは PL/SQL ブロックの 2 行目以降の行のためのプロンプトを設定します。ON を設定すると、プロンプトは行番号に設定されます。OFF を設定すると、プロンプトは SQLPROMPT の値に設定されます。

**SQLPLUSCOMPAT[IBILITY] {x.y[.z]}**

VARIABLE の動作または出力を x.y[.z] で指定し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 または 9.0.1 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 9.0.1 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。

**SQLPRE [FIX] {#|c}**

SQL\*Plus の接頭文字を設定します。SQL コマンドまたは PL/SQL ブロックを入力している途中で、別の行に、SQL\*Plus 接頭文字を付けて SQL\*Plus コマンドを入力できます。SQL\*Plus は、入力中の SQL コマンドまたは PL/SQL ブロックに影響を与えずに、別の行に入力したコマンドを即時実行します。接頭文字は、英数字以外の文字にする必要があります。

**SQLP [ROMPT] {SQL>|text}**

SQL\*Plus コマンド・プロンプトを設定します。

**SQLT [ERMINATOR] {\_ |c|ON|OFF}**

c には、SQL コマンドを終了し、実行するために使用する文字を設定します。英数字または空白は使用できません。OFF は、SQL\*Plus でコマンド終了記号が認識されないことを表します。SQL コマンドを終了するときは、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。

**SUF [FIX] {SQL|text}**

コマンド・ファイル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。

**TAB {ON|OFF}**

SQL\*Plus が空白をどのように端末に出力するかを指定します。OFF を指定すると、空白は空白として出力されます。ON を指定すると、TAB 文字が使用されます。TAB の設定間隔は 8 文字です。TAB のデフォルト値は、システムによって異なります。

**TERM [OUT] {ON|OFF}**

コマンド・ファイルから実行するコマンドによって生成される出力の表示を制御します。OFF を指定すると、表示が抑止されるので、コマンド・ファイル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。

**TI [ME] {ON|OFF}**

現在時刻の表示を制御します。ON を指定すると、各コマンド・プロンプトの前に現在時刻が表示されます。OFF を指定すると、時刻表示が抑止されます。

**TIMI [NG] {ON|OFF}**

計時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、その計時統計が表示されます。OFF を指定すると、各コマンドの計時統計が抑止されます。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。複数のコマンドの計時情報については、この章の「TIMING」コマンドを参照してください。

**TRIM [OUT] {ON|OFF}**

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。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。

**TRIMS [POOL] {ON|OFF}**

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ます。OFF を指定すると、SQL\*Plus で後続空白を挿入できます。TRIMSPool ON は、端末出力には影響を与えません。

**UND [ERLINE] {-|c|ON|OFF}**

c には、SQL\*Plus レポートの列ヘッダーに下線を付けるために使用する文字を設定します。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。

**VER [IFY] {ON|OFF}**

SQL\*Plus が置換変数を値に置き換える前および後で、SQL 文または PL/SQL コマンドのテキストを SQL\*Plus でリスト表示するかどうかを制御します。ON を指定すると、テキストがリスト表示されます。OFF を指定すると、リスト表示が抑止されます。

**WRA [P] {ON|OFF}**

SELECT によって選択された行が長すぎてカレント行の幅に収まらない場合に、その行の表示を SQL\*Plus で切り捨てるかどうかを制御します。OFF を指定すると、選択された行が切り捨てられます。ON を指定すると、選択され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。

## 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リスト表示できます。

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。

## 例

次に、選択済の SET コマンド変数の使用例を示します。

### APPINFO

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 プロシージャが正常に完了しました。
```



```
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
```

**AUTORECOVERY**

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
```

**CMDSEP**

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
Fleur                     $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 行が選択されました。

**COLSEP**

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

**COMPATIBILITY**

Oracle7 SQL 構文で作成されたコマンド・ファイル 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 用に作成されたコマンド・ファイル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
```

または、コマンド・ファイルの最初にコマンド SET COMPATIBILITY V7 を追加し、その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。

**DESCRIBE**

2 レベルの深さでビュー EMP\_DETAILS\_VIEW を記述して、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を記述します。



```
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)

## ESCAPE

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
```

## HEADING

レポート内での列ヘッダーの表示を抑止するには、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
```

## INSTANCE

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。

## LOBOFFSET

CLOB 列のデータを取り出すための開始位置を 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。

## LOGSOURCE

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
```

## LONG

LONG 値を表示およびコピーするためにフェッチする最大文字数を 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 番目の文字になるまで切捨てを行いません。LONG のデフォルトは、80 文字です。

## LONGCHUNKSIZE

SQL\*Plus が LONG 値を取り出すときに使用する増分値のサイズを 100 文字に設定するには、次のように入力します。



```
SET LONGCHUNKSIZE 100
```

LONG データは、値全体が取り出されるまで、または SET LONG の値に到達するまで（いずれか早い方）、100 文字の増分値で取り出されます。

## MARKUP

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。

---

入力コマンドを使用して、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。



```
INPUT
SET MARKUP HTML ON SPOOL ON HEAD '<TITLE>Employee List</title> -
STYLE TYPE="TEXT/CSS"><!--BODY {background: fffc6} --></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。次のように、スクリプトを保存し、START を使用して実行します。



```
SAVE employee.sql
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
John	Russell	14000
Karen	Partners	13500
Michael	Hartstein	13000

6 rows selected.

## SERVEROUTPUT

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;
```



```
/
タスクが完了しました。
PL/SQL プロシージャが正常に完了しました。
```

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.');
```



/  
トリガーが作成されました。



```
INSERT INTO SERVER_TAB VALUES ('TEXT');
```



タスクが完了しました。  
1行が作成されました。

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?');
```



/  
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?');
```



/  
If there is nothing  
shall we continue wi

**SHIFTINOUT**

これらをサポートする端末にシフト文字が表示されるようにするには、次のように入力します。



```
SET SHIFTINOUT VISIBLE
SELECT LAST_NAME, JOB_ID FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000;
```



```
LAST_NAME      JOB_ID
-----
: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。

**SQLBLANKLINES**

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
```

**SQLCONTINUE**

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。

### SQLPROMPT

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。

SID 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 X NEW_VALUE Y
SELECT RTRIM(INSTANCE, CHR(0)) X FROM V$THREAD;
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;
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。

### SUFFIX

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。

# 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
SPOO [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。

## 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リスト表示されます。

### **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、『Oracle8i 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 [OFTER]**

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。

ユーザーは "SYS" です。

**例**

現行の LINESIZE をリスト表示するには、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
```



```
Errors for PROCEDURE HR.PROC1:
LINE/COL ERROR
```



```
-----
3/1      PLS-00049: バインド変数 'P1' が正しくありません。
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。Oracle7 サーバーの Oracle インスタンスに対して SHUTDOWN は使用できません。

## 項

項または句の詳細は、次の説明を参照してください。

### 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。

SYSOPER または SYSDBA でデータベースに接続する必要があります。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

## 構文

SPO [OL] [*file\_name* [.ext] | OFF | OUT]

問合せの結果をファイルに格納するか、または必要に応じてそのファイルをプリンタに送ります。

## 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**

スプールの停止します。

### **OUT**

スプールの停止して、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、いくつか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リスト表示するには、句を指定しないで SPOOL のみを入力します。

## 使用方法

コマンド・ファイル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、DLARY という名前のファイルに表示出力を記録するには、次のように入力します。



SPOOL DIARY

スプールの停止し、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] {uri|file_name[.ext] } [arg...]
```

指定したコマンド・ファイルの **SQL\*Plus** 文を実行します。コマンド・ファイルは、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。uri は、このリリースでは、**Windows** プラットフォームのみで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。

### uri

指定した **Web** サーバーで実行するスクリプトの **URI** を指定します。**SQL\*Plus** では、**HTTP**、**FTP** および **Gopher** プロトコルをサポートします。

### file\_name[.ext]

実行するコマンド・ファイルを指定します。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、指定のファイル名および拡張子を持つファイルを検索します。該当の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 ...

コマンド・ファイル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、該当の値をコマンド・ファイルのパラメータ（&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 が置換されません。



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コマンド・ファイルを 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の詳細は、第 3 章の「対話型コマンドの作成」の「START コマンドを使用した値の指定」を参照してください。

## 使用方法

@ (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。

コマンド・ファイル内で EXIT または QUIT コマンドを使用すると、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;
```

このコマンド・ファイルを実行するには、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://HOST.DOMAIN/PROMOTE.SQL ST_MAN 7000
START FTP://HOST.DOMAIN/PROMOTE.SQL ST_MAN 7000
START GOPHER://HOST.DOMAIN/PROMOTE.SQL ST_MAN 7000
```

HOST.DOMAIN を、スクリプトがある Web サーバーの host.domain 名で置き換える必要があります。

どちらの場合も、SQL\*Plus は、次のコマンドを実行します。

```
SELECT LAST_NAME, LAST_NAME
FROM EMP_DETAILS_VIEW
WHERE JOB_ID='ST_MAN' AND SALARY>7000;
```

SQL\*Plus に問合せ結果を出力します。

# STARTUP

## 構文

```
STARTUP [FORCE] [RESTRICT] [PFILE=filename] [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
```

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。STARTUP を使用して Oracle7 インスタンスを起動できません。

## 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。デフォルトで使用されるファイルは、プラットフォームによって異なります。たとえば、UNIX の場合は \$ORACLE\_HOME/dbs/init\$ORACLE\_SID.ora、Windows の場合は %ORACLE\_HOME%\%database%\initORCL.ora がデフォルト・ファイルです。

### 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。

**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
```

現行のインスタンスを停止するには、すぐに、マウントまたはオープンしないで再起動し、データベース管理者のみが接続できるようにします。次のように入力します。



```
STARTUP FORCE RESTRICT NOMOUNT
```

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
```

特定のデータベースを停止し、ただちに再起動およびオープンし、データベース管理者のみがアクセスできるようにし、パラメータ・ファイル MYINIT.ORA を使用するには、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
```



アイドル・インスタンスに接続しました。



```
STARTUP MOUNT
```



ORACLE インスタンスが起動しました。

```
Total System Global Area      7629732 bytes
Fixed Size                      60324 bytes
Variable Size                   6627328 bytes
Database Buffers                409600 bytes
Redo Buffers                    532480 bytes
```

# STORE

## 構文

```
STORE SET file_name [.ext] [CRE[ATE] | REP[LACE] | APP[END]]
```

現行の SQL\*Plus 環境の属性を、ホスト・オペレーティング・システム・ファイル（コマンド・ファイル）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コマンド・ファイルが作成されます。

STORE コマンド句（CREATE、REPLACE または APPEND）と同じ名前でファイルを保存するには、名前を一重引用符で囲むか、またはファイル拡張子を指定する必要があります。

## 例

現行の SQL\*Plus システム変数を、名前が DEFAULTENV でデフォルトのコマンド・ファイル拡張子が付いたファイルに格納するには、次のように入力します。



```
STORE SET DEFAULTENV
```

現行の SQL\*Plus システム変数を、名前が DEFAULTENV で拡張子が OLD の既存ファイルに追加するには、次のように入力します。



```
STORE SET DEFAULTENV.OLD APPEND
```

## TIMING

### 構文

```
TIMI [NG] [START text|SHOW|STOP]
```

経過時間に関する計時データを記録するか、現行のタイマーの名前およびタイミング・データをリスト表示するか、またはアクティブ・タイマーの数をリスト表示します。

### 項

項または句の詳細は、次の説明を参照してください。

#### 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計時データがリスト表示されます。

#### STOP

現行のタイマー名および計時データがリスト表示されてから、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・タイマーの数をリスト表示するには、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リスト表示するには、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リスト表示して、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  
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のタイトルを配置し書式設定するか、または現行の TTITLE 定義をリスト表示します。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、[付録 F](#) 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、タイトルをオフに (表示を抑止) します。



**ON**

タイトルをオンに（表示を復元）します。上部タイトルを定義すると、SQL\*Plus は自動的に TTITLE を ON に設定します。

**COL *n***

カレント行の列 *n* までインデントします（列 *n* を過ぎて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、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、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の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。

指定のデータ項目の前に適切な FORMAT 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 定義をリスト表示するには、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。有効な *printspec* 句 (LEFT、SKIP、COL など) がコマンド名の直後に続いていれば、SQL\*Plus は新しい書式で 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](#)」コマンドの「[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、定数および変数を指定時と同じ順序で表示し、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、その表示を抑止するには、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[ITABLE] [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、[第 3 章の「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リスト表示されます。VARIABLE の後に変数名のみを指定すると、その変数がリスト表示されます。

### 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。文字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、NLS\_LENGTH\_SEMANTICS 環境変数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 Oracle9i より前のサーバーに接続している場合、または SQLPLUSCOMPATIBILITY システム変数が 9.0.1 未満に設定し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。文字セマンティクスを持つ VARCHAR2 変数  $n$  の最大値は、選択したキャラクタ・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、NLS\_LENGTH\_SEMANTICS 環境変数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 Oracle9i より前のサーバーに接続している場合、または SQLPLUSCOMPATIBILITY システム変数が 9.0.1 未満に設定し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](#)」の「[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る値は、バインド変数 :id に入ります。そのバインド変数は、**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することができます。

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 プロシージャが正常に完了しました。

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行が選択されました。

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
```

定義済変数をリスト表示するには、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つの変数をリスト表示します。



```
VARIABLE txt
```



```
variable txt
datatype CHAR(20)
```

次の例では、個人別に給与をリスト表示し、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 プロシージャが正常に完了しました。



```
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 列にデータをリスト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 プロシージャが正常に完了しました。

clob\_col 列から 200 文字を出力するには、次のように入力します。



```
SET LINESIZE 70
SET LONG 200
PRINT T
```



```
T
```

```
-----
Remember to run the Departmental Salary Bill report each month. This r
eport contains confidential information.
```

出力位置を 21 番目の文字に設定するには、次のように入力します。



```
SET LOBOFFSET 21
PRINT T
```



```
T
```

```
-----
Departmental Salary Bill report each month. This report contains confi
dential information.
```

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 I/O エラーなど）が発生した場合に、SQL\*Plus を終了します。

## 項

項または句の詳細は、次の説明を参照してください。

**EXIT** [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かった場合、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

次のコマンド・ファイル中のコマンドは、出力ファイルへの書込み時に障害が発生すると、SQL\*Plus が終了し、保留中の変更があればそれがすべてコミットされます。



```
WHENEVER OSERROR EXIT  
START no_such_file
```



O/S メッセージ: No such file or directory  
Oracle との接続が切断されました。

# 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 ブロックがエラーとなった場合、SQL\*Plus を終了します。

## 項

項または句の詳細は、次の説明を参照してください。

**EXIT** [**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**

次のコマンド・ファイル中のコマンドは、SQL UPDATE コマンドが失敗すると、SQL\*Plus が終了し、SQL エラー・コードが戻ります。



```
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
SELECT COLUMN_DOES_NOT_EXIST FROM DUAL;
```



```
select column_does_not_exist from dual
*
```

1 行でエラーが発生しました。  
ORA-00904: 列名が無効です。

Oracle との接続が切断されました。

次の SQL コマンド・エラーの場合は、SQL\*Plus が終了し、変数 MY\_ERROR\_VAR の値が戻ります。



```
DEFINE MY_ERROR_VAR = 99
WHENEVER SQLERROR EXIT my_error_var
UPDATE non_existed_table set col1 = col1 + 1;
```



```
UPDATE NON_EXISTED_TABLE set col1 = col1 + 1
*
```

1 行でエラーが発生しました。  
ORA-00942: 表またはビューが存在しません。

Oracle との接続が切断されました。

次の例では、WHENEVER SQLERROR コマンドが SQL\*Plus コマンドには影響しないで、SQL コマンドおよび PL/SQL ブロックに影響することを示します。



```
WHENEVER SQLERROR EXIT SQL.SQLCODE
COLUMN LAST_NAME HEADING "EMPLOYEE NAME"
```



```
Unknown COLUMN option "HEADING"
```



```
SHOW non_existed_option
```



```
Unknown SHOW option "NON_EXIST_OPTION"
```



```
GET non_existed_file.sql
```



```
Unable to open "NON_EXIST_FILE.SQL"
```

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;
```

\*

2 行でエラーが発生しました。

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

Oracle との接続が切断されました。





---

---

## SQL\*Plus のエラー・メッセージ

この付録では、SQL\*Plus が生成するエラー・メッセージについて説明します。Oracle が生成するエラー・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。

ここでは、次の項目について説明します。

- [SQL\\*Plus のエラー・メッセージおよびコード](#)
- [COPY コマンドのメッセージ](#)

## SQL\*Plus のエラー・メッセージおよびコード

**SP2-0002 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、[第 8 章「コマンド・リファレンス」](#)の「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、[第 8 章「コマンド・リファレンス」](#)の「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、[第 8 章「コマンド・リファレンス」](#)の「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り当てられません。**

**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、第 8 章「コマンド・リファレンス」の「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 - 無効なモードです (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します。

**SP2-00030 他の行を入れる空間がありません。**

**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、第 8 章「コマンド・リファレンス」を確認してください。

**SP2-00039 "command\_name" への代入中にコマンド行がオーバーフローしました。**

**原因:** コマンド行の最大長を超えています。

**処置:** 置換文字列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:** 使用するコマンドの正しいオプションについては、第 8 章「コマンド・リファレンス」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:** 使用するコマンドの正しいオプションについては、第 8 章「コマンド・リファレンス」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。

SP2-00054 definition\_name 定義を割り当てる空間がありません。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空間がなくなりました。古い定義（ある場合）が残されました。

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、[第8章「コマンド・リファレンス」](#)の「[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空間がありません。

**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、[第 8 章「コマンド・リファレンス」](#)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ていません。

**処置:** 列名を指定し、再実行してください。

**SP2-00096 INTO 変数 *variable\_name* を割り当てる空間がありません。**

**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、ファイルがオープンできない。

**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、ファイルがオープンできない。

**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、[第 8 章「コマンド・リファレンス」](#)の「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 (空間がありません)。**

**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) を割り当てられません。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、[第 8 章「コマンド・リファレンス」](#) の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、ファイルを開くことができない。

**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 がアクセス不能です。**

**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:** データベース管理者に連絡して、ヘルプ・システムをインストールしてください。

**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:** 使用するコマンドの正しいオプションについては、第 8 章「コマンド・リファレンス」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:** 第 8 章「コマンド・リファレンス」の「ACCEPT」コマンドを参照して、構文を修正してください。

**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 文字以内でなければなりません。**

**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:** 置換変数が使用されており、SQL\*Plus がその変数に対する値を検出できませんでした。

**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* 用の空間がありません (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:** 正しい構文については、[第 8 章「コマンド・リファレンス」](#)を確認してください。

**SP2-00245 印刷用の一時記憶域を割り当てられません。**

**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* は印刷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：使用するコマンドの正しいオプションについては、第8章「コマンド・リファレンス」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：使用するコマンドの正しいオプションについては、第8章「コマンド・リファレンス」の構文を確認してください。

SP2-00262 CCBDEF ポインタ・アレイを割り当てる空間がありません。

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空間がありません。

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 *parameter\_number* が範囲外です (範囲: *lower\_range* ~ *upper\_range*)。

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、制限内の値を入力してください。

**SP2-00268** *option\_name* オプションが有効数値ではありません。

**原因:** パラメータに数値（整数）以外の値が入力されました。

**処置:** 有効な数値（整数）を入力してください。

**SP2-00271** *variable\_name* はバッファ変数ではありません。

**原因:** 指定された変数はバッファとして定義されていません。

**処置:** バッファ変数名が正しいことを確認して、再試行してください。

**SP2-00272** *character\_name* 文字に英数字または空白は使用できません。

**原因:** SET コマンドの指定する文字には、英数字または空白は使用できません。

**処置:** 使用するコマンドの正しいオプションについては、[第 8 章「コマンド・リファレンス」](#)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:** 使用するコマンドの正しいオプションについては、[第 8 章「コマンド・リファレンス」](#)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:** 使用するコマンドの正しいオプションについては、[第 8 章「コマンド・リファレンス」](#)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 コマンド手順が *number\_of\_nested\_procedures* の深さ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。使用するコマンドの正しいオプションについては、[第 8 章「コマンド・リファレンス」](#)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:** 使用するコマンドの正しいオプションについては、[第 8 章「コマンド・リファレンス」](#)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空間がありません。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。

**SP2-00332 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、第 8 章「コマンド・リファレンス」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、第 8 章「コマンド・リファレンス」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*>] [<*login*>] [<*start*>]]**

<*option*> の使用法は、-H | -V | [[-M <*o*>] [-R <*n*>] [-S]] です。

"-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。

"-M <*o*>" は HTML マークアップ・オプションを使用します。<*o*>

"-V" は SQL\*Plus バージョン・バナーを表示します。

"-S" はサイレント・モードを使用します。

"-R <*n*>" は制限モードを使用します。<*n*>

<*login*> ::= <*username*>[/<*password*>][@<*connect\_string*>] | | /NOLOG

<*start*> ::= @<*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、第 7 章「SQL\*Plus の起動およびヘルプの表示」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:** 使用するコマンドの正しいオプションについては、第 8 章「コマンド・リファレンス」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: エントリを修正して、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。

SP2-00480 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>: hr/hr@d:chicago-mktg などのデータベース文字列

<opt>: APPEND または CREATE、INSERT、REPLACE キーワードのうちの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。使用するコマンドの正しいオプションについては、第8章「コマンド・リファレンス」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、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「COPY」コマンドの構文を確認してください。

**SP2-00498 カッコ付きの列のリストか USING キーワードが欠落しています。**

**原因:** カッコ付きのリストが列リスト内で欠落しているか、または USING キーワードが COPY コマンド内で欠落しています。

**処置:** 正しいオプションについては、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「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 コマンドでエラーが発生しました。コマンドで列の最大数を超えました。

**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 または ATTRIBUTE コマンドのいずれかで使用した列名が NULL です。コマンドに無効な列名が指定されました。

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が欠落しているコマンドが検出されました。

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: 問合せ文の前に USING キーワードを指定してください。

**SP2-00519 FROM 文字列の Oracle Net @ データベース仕様が欠落しています。**

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: データベースを再指定してください。FROM 句を省略して、ソースのデフォルトを SQL\*Plus が接続しているデータベースにしてください。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 @ データベース仕様が欠落しています。**

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: データベースを再指定してください。TO 句を省略して、ソースのデフォルトを SQL\*Plus が接続しているデータベースにしてください。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。

処置: 正しいオプションについては、[第 8 章「コマンド・リファレンス」](#)の「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。

**SP2-00545 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: 使用するコマンドの正しいオプションについては、[第 8 章「コマンド・リファレンス」](#)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* が範囲外です (範囲: *lower\_value* ~ *upper\_value*)。

原因: 指定された SET オプションが範囲外に達しました。

処置: オプションの制限を確認して、操作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: VARIABLE コマンドの使用方法のメッセージです。

処置: 正しい使用方法については、第 8 章「コマンド・リファレンス」の「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: PRINT コマンドの使用方法のメッセージです。

処置: 正しい使用方法については、第 8 章「コマンド・リファレンス」の「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: SHOW ERRORS コマンドの使用方法のメッセージです。

処置: 正しいオプションについては、第 8 章「コマンド・リファレンス」の「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: 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。

**SP2-00556** 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: EXECUTE コマンドの使用方法のメッセージです。

処置: 正しい使用方法については、第 8 章「コマンド・リファレンス」の「EXECUTE」コマンドの構文を確認してください。

**SP2-00560 使用方法: DESCRIBE [ スキーマ. ] オブジェクト [ .subobject | @db\_link ] [ 列 ]**

**原因:** DESCRIBE コマンドの使用方法のメッセージです。

**処置:** 正しい使用方法については、第 8 章「コマンド・リファレンス」の「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 コマンドで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の列の記述が不正です。

**処置:** DESCRIBE コマンドの列の指定を削除して、再試行してください。

**SP2-00568 バインド変数が宣言されていません。**

**原因:** バインド変数が宣言されていません。

**処置:** 処置は必要ありません。

**SP2-00570 使用方法: SET SERVEROUTPUT { ON | OFF } [ SIZE n ] [ FOR[MAT] { WRA[PPED] | WOR[D\_W]RAPPED } | 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。

**SP2-00577 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:** パッケージは、スタンドアロンで記述できません。プロシージャなどのサブオブジェクトとともに提供される必要があります。

**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を記述してください。

**SP2-00582 使用方法: [EXIT | QUIT] [SUCCESS | FAILURE | WARNING | n | 変数 | :バインド変数] [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、第 8 章「[コマンド・リファレンス](#)」の「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、第 8 章「[コマンド・リファレンス](#)」の「COMPUTE」コマンドの構文を確認してください。

**SP2-00591 必要な動的領域 (*number\_of\_bytes*bytes) を割り当てられません。**

**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、第 8 章「[コマンド・リファレンス](#)」の「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 コマンドの使用方法です。

**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] 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、第 8 章「コマンド・リファレンス」の「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。

**SP2-00599 使用方法: SET EDITF[ILE] ファイル名 [*.ext*]**

**原因:** SET EDITFILE コマンドの後に必要なファイル名が欠落していました。

**処置:** 正しいオプションについては、第 8 章「コマンド・リファレンス」の「SET EDITFILE」コマンドの構文を確認してください。

**SP2-00603 STORE コマンドが不正です。**

**STORE [SET] *filename.ext* [CRE[ATE]|REP[LACE]|APP[END]]**

**原因:** 無効な STORE オプションが指定されました。有効なコマンド句は、CREATE、REPLACE または APPEND です。

**処置:** 正しいオプションについては、第 8 章「コマンド・リファレンス」の「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取り専用になっています。

**処置:** ディスク領域が十分であること、およびディレクトリにファイルを作成できることを確認してください。



**SP2-00607 ファイル "file\_name" をクローズできません。**

**原因:** STORE コマンドは、指定されたファイルをクローズできません。別のリソースがそのファイルをロックしている可能性があります。

**処置:** クローズする前にファイルがロックされていないことを確認してください。

**SP2-00608 オブジェクト "object\_name" はリモート・オブジェクトです。詳しい説明はできません。**

**原因:** リモート・オブジェクトは DESCRIBE できません。

**処置:** 処置は必要ありません。

**SP2-00609 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、第 8 章「コマンド・リファレンス」の「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:** AUTOTRACE コマンドが、実行する適切な権限を持っていない現行のユーザーによって発行されました。

**処置:** AUTOTRACE を実行する権限およびオブジェクトを持っていることを確認してください。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。

**SP2-00614 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 セッション識別子なし。PLUSTRACE ロールが使用可能かチェックしてください。**

**原因:** セッション識別子を検出できませんでした。

**処置:** PLUSTRACE ロールが使用可能であることを確認してください。

**SP2-00619 接続中にエラーが発生しました。**

**原因:** AUTOTRACE がデータベース・インスタンスへの 2 番目の接続を確立しようとして、エラーが発生しました。

**処置:** データベースのアクティブなセッション数の制限を超えていないことを確認してください。

**SP2-00620 接続切断中にエラーが発生しました。**

**原因:** AUTOTRACE がデータベース・インスタンスから切断しようとして、エラーが発生しました。

**処置:** データベースが使用可能であることを確認してください。

**SP2-00621 統計表示収集中にエラー (ORA-error\_number) が発生しました。**

**原因:** AUTOTRACE の使用中、統計表示収集で 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し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 のインストール・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" の印刷中にエラーが発生しました。**

**原因:** 指定された変数の印刷中にエラーが発生しました。

**処置:** 指定され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ず、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:** Oracle インスタンスへ接続して、操作を再試行してください。データベースに接続して、PASSWORD コマンドを再発行してください。

**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。

**SP2-00645 オペレーティング・システムのエラーが発生しました。EDIT コマンドを完了できません。**

**原因:** EDIT コマンド使用時に、オペレーティング・システム・エラーが発生しました。

**処置:** ファイルが正常に作成されたことを確認し、書込み中のデバイスがまだ使用可能であることを確認してください。

**SP2-00650 新しいパスワードが合致しません。**

**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、その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 のみがシフト対応のキャラクタ・セットに影響します。**

**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 は起動できません。

**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:** 名前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:** 名前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 Bind 変数の長さは *variable\_length\_units\_of\_variable* を超過できません。

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 データ "*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: 有効な日付または日付書式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 コマンドの使用方法です。

処置: 正しいオプションについては、第 8 章「コマンド・リファレンス」の「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] [*login*] [AS [SYSDBA | SYSOPER]]

ただし *logon ::= username[/password][@connect\_string] | |*

原因: SQL\*Plus の CONNECT コマンドの使用方法です。

処置: 正しい使用方法については、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「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、第 8 章「コマンド・リファレンス」の「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:** 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 コマンドの発行中に、インスタンスへの接続で問題が発生します。

**処置:** インスタンスから切断して、コマンドを再発行してください。

**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:** 構文を確認して、再入力してください。

**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:** 使用するコマンドの正しいオプションについては、第 8 章「コマンド・リファレンス」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：コマンド行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 コマンドの使用法のメッセージです。  
処置：正しいオプションについては、第 8 章「コマンド・リファレンス」の「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わせてください。
- SP2-00745 使用方法：SET SQLPLUSCOMPAT[IBILITY] バージョン . リリース .[update]  
原因：SET SQLPLUSCOMPAT[IBILITY] コマンドに無効なオプションが使用されました。  
処置：正しいオプションについては、第 8 章「コマンド・リファレンス」の「SET SQLPLUSCOMPAT[IBILITY]」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。

## COPY コマンドのメッセージ

**CPY0002 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。

**CPY0003 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。

**CPY0004 ソース表および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。

**CPY0005 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。

**CYP0006 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：**選択した列の行数がコピー先の表の行数と一致していることを確認した上で、再度 COPY コマンドを指定してください。

**CPY0007 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：**選択した列の行数がコピー先の表の行数と一致していることを確認した上で、再度 COPY コマンドを指定してください。



**CPY0008 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していることを確認した上で、再度 COPY コマンドを指定してください。

**CPY0009 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していることを確認した上で、再度 COPY コマンドを指定してください。



---

## リリース 9.0.1 での拡張機能

ここでは、SQL\*Plus リリース 9.0.1 での拡張機能について説明します。

- SQLPLUS コマンドライン・スイッチ
- iSQL\*Plus
- START、@、@@
- SET APPINFO
- SET SQLPLUSCOMPATIBILITY
- HR サンプル・スキーマ

## SQL\*Plus リリース 9.0.1 での拡張機能

### 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[ERSION]
-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。iSQL\*Plus の詳細は、『iSQL\*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください。

### START、@、@@

URI から SQL を読み込み、実行する新しいコマンドが追加されました。URI 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。

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

「uri」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://host.domain/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

SET コマンド、SET SQLPLUSCOMPAT[IBILITY] {x.y[.z]} が新規に追加されました。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・スキーマの新規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 の制限

表 C-1 に、SQL\*Plus のそれぞれの要素の制限または最大値を示します。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 C-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
ネストしたコマンド・ファイルの最大数	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

---



# D

---

---

## SQL コマンド・リスト

表 D-1 に、主要な SQL コマンドを示します。これらのコマンドの詳細は、『Oracle9i SQL リファレンス』を参照してください。

---

**表 D-1 SQL コマンド・リスト**

**主要な SQL コマンドおよび句**

---

ALTER	LOCK TABLE
ANALYZE	NOAUDIT
AUDIT	RENAME
COMMENT	REVOKE
CREATE	SAVEPOINT
DROP	SET ROLE
EXPLAIN	SET TRANSACTION
GRANT	TRUNCATE
INSERT	UPDATE

---

---

## セキュリティ

この付録では、データベース表および SQL\*Plus コマンドへのアクセスを制御するため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

## 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 および 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 がこのセキュリティを実行します。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。

ユーザーが SQL\*Plus にログインする場合、SQL\*Plus は、PUP 表から制限を読み込み、セッションが終了するまでその制限を保持します。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 AS SYSDBA または AS 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 は PUP 表を読み込みません。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 という名前のコマンド・ファイルを、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 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旨の警告が表示されます。

---

---

## PUP 表の構造

PUP 表は、次の列を含んでいます。

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

PUP 表の各列の説明および使用方法については、次のリストを参照してください。

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、GET 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、この付録の「SQL*Plus、SQL および PL/SQL コマンドの使用禁止」を参照してください。ルールを使用禁止にする方法の詳細は、この付録の「SQLPLUS -RESTRICT でのコマンドの使用禁止」を参照してください。</p>
Scope	SQL*Plus では、この列は無視されます。この列には、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。

Numeric_Value	SQL*Plus では、この列は無視されます。この列には、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、この付録の「 <a href="#">SQLPLUS-RESTRICT</a> でのコマンドの使用禁止」を参照してください。
Date_Value	SQL*Plus では、この列は無視されます。この列には、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	SQL*Plus では、この列は無視されます。この列には、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。コマンド・ファイル PUPBLD.SQL を実行すると、PUBLIC に PUP 表に対する SELECT 権限が付与されます。

## 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 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 を入れた行を挿入します。

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	HOST			DISABLED	
SQL*Plus	%	INSERT			DISABLED	
SQL*Plus	%	UPDATE			DISABLED	
SQL*Plus	%	DELETE			DISABLED	

コマンドを再び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	PASSWORD
EDIT	QUIT
EXECUTE	RUN
EXIT	SAVE
GET	SET (後述の注意を参照)
HOST (または、ご使用のオペレーティング・システムの HOST の別名 (VMS では \$、UNIX では ! など))	SPOOL
	START

---

---

**注意：** SQL\*Plus の SET コマンドを使用禁止にすると、SQL の 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	LOCK
ANALYZE	NOAUDIT
AUDIT	RENAME
CONNECT	REVOKE
CREATE	SELECT
DELETE	SET ROLE
DROP	SET TRANSACTION
GRANT	TRUNCATE
INSERT	UPDATE

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 E-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. コマンドを使用して、SYSTEM としてのログインします。



```
SQLPLUS SYSTEM/MANAGER
```

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/HR で接続して、SELECT してみてください。



```
CONNECT HR/HR;  
SELECT * FROM EMP_DETAILS_VIEW;
```

このコマンドによって、次のエラー・メッセージが表示されます。



```
SP2-00544: 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 で再接続して、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する場合、ユーザーは、そのパスワードを知らないとロールにアクセスできません。

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。このようにすると、SQL\*Plus ユーザーは、SQL\*Plus を起動したとき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、この付録の「[SET ROLE の使用禁止](#)」を参照してください。

ロールを再び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

RESTRICT オプションの詳細は、7-8 ページの「[SQLPLUS -R\[ESTRICT\] {1 | 2 | 3}](#)」コマンドを参照してください。



---

---

## 廃止された SQL\*Plus コマンド

この付録では、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、これまでと同じように 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	代替のコマンド	代替コマンドの説明
BTITLE (旧形式)	BTITLE (8-23 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 定義をリスト表示します。
COLUMN DEFAULT	COLUMN CLEAR (8-30 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 (8-83 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 (8-102 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数を設定します。
SET BUFFER	EDIT (8-60 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 (8-99 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 (8-98 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 (8-107 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 (8-134 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 定義をリスト表示します。

## 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

コマンド・ファイル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

コマンド・ファイルにコメントを挿入する現行のメソッドの詳細は、[第3章の「後で使用するコマンドの保存」](#)の「[コマンド・ファイルへのコメントの挿入](#)」および[第8章「コマンド・リファレンス」](#)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、[第8章の「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および抑止します。

### 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の表示が抑止されます。

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が抑止され、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、現行の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[Nuncate] {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 [TL]E 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、第 8 章の「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、第 8 章の「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表示され、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'
```



---

---

# 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 \(archived redo log\)](#)」も参照。

## 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 4GB 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 上の文書のほとんど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。

### **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 までの長さに行える。

## 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 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 RDBMS (リレーショナル・データベース管理システム)。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 以前の用語。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 \(redo log\)](#)」も参照。

## **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 \(SQL statement\)](#)」も参照。

## **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ない。

### **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](#)」も参照。

### **SYSOPER**

DBA が STARTUP、SHUTDOWN、ARCHIVE LOG、RECOVER などの操作を実行できる権限。「[SYSDBA](#)」も参照。

### **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。

## **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、再び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」も参照。

## 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)」も参照。

## 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)」も参照。

## 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の種類。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)」も参照。

## 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)」も参照。

## 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 (computed column)」も参照。

### 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)

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。コマンド・ファイルは、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 @@ コマンドでコマンド・ファイルを実行できる。

### 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 \(prompt\)](#)」も参照。

**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 文を取り出して処理する。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ー列を使用して、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。

**サマリー行 (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ー行を作成する。

**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および制御情報など）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)」も参照。

## 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、問合せてそれらの列から取り出す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 句に事前定義済み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・オブジェクトの集まり)。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。置換変数をコマンド・ファイル内で使用して、コマンド・ファイル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コマンド・ファイル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 および LONG RAW 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（ある場合）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、データベースに一連のデータ・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発生した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、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 コマンド・ファイル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り出す SQL SELECT 文。問合せは、データを変更しないで、データを取り出すのみの読取り専用操作。多くの場合、問合せは、DML 文とみなされる。

### 問合せ結果 (query results)

問合せによって取り出したデータ。

### 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、所有者のみが認識している。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る。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る。

### **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します。

### **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)。コマンド・ファイル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 コマンド内でコマンド・ファイル名に続くデータ項目。引数は、コマンド・ファイル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 \(byte\)](#)」も参照。

## 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。表はエントリおよび関係を表し、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ます。

## 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、その問合せの間にのみ有効になる。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 \(alias\)](#)」も参照。

## ファイル (file)

1 つの単位として扱われるデータの集まり。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 を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求める。「[コマンドライン \(command line\)](#)」も参照。

### **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 \(SQL statement\)](#)」も参照。

### **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 \(column heading\)](#)」も参照。

### **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ります。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ります。

### **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 \(account\)](#)」も参照。

### **ユリウス暦日付 (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。ユリウス暦日付を使用すると、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 以前の用語。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)**

取り出す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 \(remote database\)](#)」も参照。

### **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。ログイ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・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・バッファがいっぱいになると、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。

# 索引

## 記号

# シャープ記号, 8-33  
\$ 数値書式, 4-5  
&&, 3-26  
& (アンパサンド)  
置換変数, 3-23  
\* (アスタリスク)  
DEL コマンド, 3-2, 8-51  
LIST コマンド, 3-2, 8-71  
-- (コメント・デリミタ), 3-13  
~ 負の無限大記号, 8-33  
-MARKUP, 4-38, 7-3  
SPOOL 句, 7-4  
SQLPLUS コマンド句, 7-4  
-SILENT オプション, 4-49, 7-8  
- (ハイフン)  
句, 7-2  
長い SQL\*Plus コマンドの継続, 2-12, 8-1  
. (ピリオド), 2-10  
/\*...\*/ (コメント・デリミタ), 3-12  
/NOLOG オプション, 7-10  
/ (スラッシュ) コマンド  
RUN との類似, 2-9, 8-9, 8-90  
カレント PL/SQL ブロックの実行, 2-10  
現行の SQL コマンドの実行, 2-9  
コマンド・プロンプトでの入力, 2-9, 8-9  
使用方法, 8-9  
デフォルト・ログイン, 7-9, 8-44  
バッファ行番号プロンプトでの入力, 2-8, 8-9  
:BindVariable 句  
EXIT コマンド, 8-63  
: (コロン)  
バインド変数, 3-34  
; (セミコロン), 2-6

@@ (二重アットマーク) コマンド, 3-21, 8-7  
START との類似, 8-7, 8-127  
コマンド・ファイル, 8-7  
@ (アットマーク)  
CONNECT コマンド, 6-3, 8-44  
COPY コマンド, 6-5, 8-46  
SQLPLUS コマンド, 3-18, 6-4, 7-2  
START との類似, 3-17, 8-6, 8-127  
コマンド, 3-17, 3-21, 8-5  
コマンド引数, 8-5  
コマンド・ファイル, 3-17, 8-5  
コマンド・ファイルへのパラメータの受渡し, 8-5  
[Cancel] キー, 2-14  
\_EDITOR、EDIT コマンド, 3-8, 8-60  
~ 無限大記号, 8-33

## 数字

0、数値書式, 4-5  
3 層, B-2  
3 層アーキテクチャ, B-2  
9、数値書式, 4-5

## A

ACCEPT コマンド, 3-30, 8-10  
CHAR 句, 8-10  
DATE 句, 8-10  
DEFAULT 句, 8-11  
DEFINE コマンド, 8-49  
FORMAT 句, 8-10  
HIDE 句, 8-11  
NOPROMPT 句, 8-11  
NUMBER 句, 3-33  
PROMPT 句, 3-31, 8-11

値の入力を求めるプロンプトのカスタマイズ, 3-32  
ALIAS 句, 8-30  
  ATTRIBUTE コマンド, 8-16  
ALL 句, 8-118  
ALTER コマンド  
  使用禁止, E-5  
ANALYZE コマンド  
  使用禁止, E-5  
APPEND 句  
  COPY コマンド, 6-7, 8-47  
  SAVE コマンド, 3-19, 8-91  
APPEND コマンド, 3-2, 3-6, 8-12  
APPINFO 句, 8-95  
ARCHIVE LOG  
  コマンド, 5-4, 8-13  
  モード, 5-4  
ARCH バックグラウンド・プロセス, 8-14  
ARRAYSIZE 変数, 8-95  
  COPY コマンドとの関連, 6-8, 8-48  
ATTRIBUTE コマンド, 8-16  
  ALIAS 句, 8-16  
  CLEAR COLUMN コマンド, 8-17  
  CLEAR 句, 8-16  
  FORMAT 句, 8-17  
  LIKE 句, 8-17  
  OFF 句, 8-17  
  ON 句, 8-17  
  すべての属性の表示特性のリスト表示, 8-16  
  属性の表示特性のリスト表示, 8-16  
  表示特性の制御, 8-17  
  複数入力, 8-17  
  列の消去, 8-27, 8-30  
  列の表示属性の復元, 8-17  
  列の表示属性の抑止, 8-17  
AUDIT コマンド  
  使用禁止, E-5  
AUTOCOMMIT 変数, 2-13, 8-96  
AUTOMATIC 句, 8-78  
AUTOPRINT 変数, 8-96  
AUTORECOVERY 変数, 8-96  
AUTOTRACE 変数, 3-39, 8-97  
AVG 関数, 4-17

## B

---

BEGIN コマンド, 2-9  
  使用禁止, E-6

BLOCKTERMINATOR, 8-97, 8-105, 8-106  
BODY オプション, 7-4  
BODY 句, 7-4  
BOLD 句, 8-87, 8-135  
BREAKS 句, 4-16, 8-27  
BREAK コマンド, 4-11, 8-18  
  BREAKS の消去, 4-16  
  COMPUTE との併用, 4-16, 8-18, 8-20, 8-41  
  DUPLICATES 句, 8-21  
  ON column 句, 4-12, 8-18  
  ON expr 句, 8-19  
  ON REPORT 句, 4-20, 8-20  
  ON ROW 句, 4-14, 8-20  
  REFCURSOR 変数の書式設定に使用, 8-140  
  SET COLSEP との併用, 8-98  
  SKIP PAGE 句, 4-13, 4-14, 8-21  
  SKIP 句, 4-14, 8-20  
  SQL の ORDER BY 句, 4-11, 4-12, 4-14, 8-19  
  値の重複の抑止, 4-12  
  各行の後に空白を挿入, 4-14  
  現行のブレイク定義のリスト表示, 4-16, 8-21  
  総およびサブサマリーの出力, 4-20  
  タイトルの変数に現在の日付を格納, 4-31  
  タイトルへの列値の表示, 4-30  
  定義の取消し, 8-27  
  複数のブレイク列の指定, 4-14, 8-18  
  ブレイク列が変更されたときの空白の挿入, 4-13  
  レポートの終わりでのサマリー行の出力, 4-20  
BTITLE 句, 8-119  
BTITLE コマンド, 4-23, 8-23  
  BOLD 句, 8-135  
  CENTER 句, 8-135  
  COL 句, 8-135  
  FORMAT 句, 8-135  
  LEFT 句, 8-135  
  OFF 句, 8-134  
  RIGHT 句, 8-135  
  SKIP 句, 8-135  
  TAB 句, 8-135  
  TTITLE コマンド, 8-23  
  下部タイトルの前に空白行を出力, 4-26  
  旧形式, F-3  
  現行の定義の抑止, 8-134  
  タイトルのインデント, 8-135  
  タイトル要素の整列, 8-135  
  列値変数の参照, 8-35  
BUFFER 句, 3-2, 3-10, 8-27

BUFFER 変数, F-4

## C

---

CANCEL 句, 8-79, 8-81

CENTER 句, 4-26, 8-87, 8-135

CHANGE コマンド, 3-2, 3-3, 8-24

CHAR 句, 8-10

VARIABLE コマンド, 8-138

CHAR 列

デフォルトの書式, 4-6, 8-31, 4-6, 8-31

CLEAR 句, 4-9, 8-30

ATTRIBUTE コマンド, 8-16

CLEAR コマンド, 8-27

BREAKS 句, 4-16, 8-27

BUFFER 句, 3-2, 3-10, 8-27

COLUMNS 句, 4-9, 8-27

COMPUTES 句, 8-27

SCREEN 句, 3-33, 8-28

SQL 句, 8-28

TIMING 句, 8-28

CLOB 句

VARIABLE コマンド, 8-139

CLOB 列

最大幅の設定, 8-102

デフォルトの書式, 4-6, 8-31

取出し位置の設定, 8-102

取出しサイズの設定, 8-102

CLOSECURSOR 変数, F-2, F-5

CMDSEP 変数, 8-97

COLSEP 変数, 8-98

COLUMNS 句, 4-9, 8-27

COLUMN コマンド, 4-2, 8-29

ALIAS 句, 8-30

BREAK コマンド, 8-20

CLEAR 句, 4-9, 8-30

DEFAULT 句, F-3

DEFINE コマンド, 8-49

ENTMAP 句, 8-30

FOLD\_AFTER 句, 8-30, 8-31

FOLD\_BEFORE 句, 8-31

FORMAT 句, 4-5, 4-6, 8-31

HEADING 句, 4-2, 8-33

HEADSEP 文字, 8-33

JUSTIFY 句, 8-34

LIKE 句, 4-8, 8-34

NEW\_VALUE 句, 4-30, 4-31, 8-34

NEWLINE 句, 8-34

NOPRINT 句, 4-30, 8-34

NULL 句, 8-35

NUMBER 列の書式設定, 4-4, 8-32

OFF 句, 4-9, 8-35

OLD\_VALUE 句, 4-31, 8-35

ON 句, 4-9, 8-35

PRINT 句, 8-34

REFCURSOR 変数の書式設定に使用, 8-140

TRUNCATED 句, 4-8, 8-35

WORD\_WRAPPED 句, 4-8, 4-10, 8-35

WRAPPED 句, 4-8, 8-35

下部タイトルへの列値の表示, 4-31, 8-35

上部タイトルへの列値の表示, 4-30, 8-34

タイトルの変数に現在の日付を格納, 4-31, 8-37

表示をデフォルトにリセット, 4-9, 8-30, F-2

複数入力, 8-36

列の書式設定, 4-6

列の表示属性の復元, 4-9, 8-35

列の表示属性の抑止, 4-9, 8-35

列の表示属性のリスト表示, 4-9, 8-29

COL 句, 4-27, 8-87, 8-135

COMMIT 句, 8-63

WHENEVER OSERROR, 8-145

WHENEVER SQLERROR, 8-147

COMMIT コマンド, 2-13

COMPATIBILITY 変数, 8-98

COMPUTES 句, 8-27

COMPUTE コマンド, 4-11, 8-39

AVG 関数, 4-17

COUNT 関数, 4-17

LABEL 句, 4-17, 4-20, 8-40

MAXIMUM 関数, 4-17

MINIMUM 関数, 4-17

NUMBER 関数, 4-17

OF 句, 4-16

OF での SELECT 式の参照, 8-40

ON column 句, 4-16, 8-40

ON expr 句, 8-40

ON REPORT 句, 4-20, 8-40

ON ROW 句, 8-40

ON での SELECT 式の参照, 8-40

REFCURSOR 変数の書式設定に使用, 8-140

STD 関数, 4-17

SUM 関数, 4-17

VARIANCE 関数, 4-17

同じ列についての複数のサマリーの出力, 4-21

異なる列についてのサマリーの計算, 4-21  
最大 LABEL 長, 8-40  
すべての定義のリスト表示, 4-22, 8-40  
総およびサブサマリーの出力, 4-20  
定義の取消し, 4-22, 8-27  
ブレイクでのサマリー行の出力, 4-16  
レポートの終わりでのサマリー行の出力, 4-20  
CONCAT 変数, 3-28, 8-98  
CONNECT コマンド, 6-2, 6-3, 8-44  
  @ (アットマーク), 6-3, 8-44  
  SYSDBA 句, 7-9, 8-45  
  SYSOPER 句, 7-9, 8-45  
  サービス名, 6-3  
  使用禁止, E-5  
  接続識別子, 6-3  
  パスワードの変更, 8-44, 8-45, 8-73  
  ユーザー名 / パスワード, 6-2, 6-3, 6-4, 8-44  
CONTINUE 句  
  WHENEVER OSERROR, 8-145  
  WHENEVER SQLERROR, 8-147  
COPYCOMMIT 変数, 8-98  
  COPY コマンドとの関連, 6-8, 8-48  
COPYTYPECHECK 変数, 8-98  
COPY コマンド, 6-4, 8-46  
  @ (アットマーク), 6-5, 8-46  
  1 つのデータベース上にある表の間でのデータのク  
  ピー, 6-9  
  APPEND 句, 6-7, 8-47  
  ARRAYSIZE 変数, 6-8, 8-48  
  COPYCOMMIT 変数, 6-8, 8-48  
  CREATE 句, 6-6, 8-47  
  FROM 句, 6-5, 8-46  
  INSERT 句, 6-6, 8-47  
  LONG 変数, 6-8, 8-48  
  REPLACE 句, 6-6, 8-47  
  SELECT によるソース表の命名, 6-6, 8-47  
  TO 句, 6-5, 8-46  
  USING 句, 6-6, 8-47  
  アクションの決定, 6-5  
  宛先の表, 6-5, 8-47  
  宛先の列の指定, 6-5, 8-47  
  コピーするデータの指定, 6-6, 8-47  
  コミットが行われる場合, 8-48  
  サービス名, 6-5, 6-7, 6-9  
  サンプル・コマンド, 6-5, 6-6  
  使用禁止, E-5  
  ソースの行および列の決定, 6-6, 8-47

データベース間のデータのコピー, 6-4  
問合せ, 6-6, 8-47  
必須の接続識別子, 8-47  
表へのデータの挿入, 6-6, 6-7, 8-47  
表内のデータの置換, 6-6, 8-47  
表の作成, 6-6, 8-47  
別のユーザーの表への参照, 6-8  
メッセージの解釈, 6-8  
ユーザー名 / パスワード, 6-5, 6-7, 6-9, 8-46  
COUNT 関数, 4-17  
CREATE 句  
  COPY コマンド, 6-6, 8-47  
CREATE コマンド  
  PL/SQL の入力, 2-8  
  使用禁止, E-5

## D

---

DATABASE 句, 8-79  
DATAFILE 句, 8-80  
DATE 句, 8-10  
DATE 列  
  デフォルトの書式, 4-6, 8-31, 8-38, 4-6  
DB2, 8-98  
DBA, 5-2  
DBA 権限, 8-128  
DBA モード, 8-128  
DBMS\_APPLICATION\_INFO パッケージ, 8-95  
DECLARE コマンド  
  使用禁止, E-6  
DECLARE コマンド (PL/SQL), 2-9  
DEFAULT 句, 8-11  
DEFINE コマンド, 3-22, 8-49  
  CHAR 値, 8-49  
  SET DEFINE ON|OFF, 8-99  
  UNDEFINE コマンド, 3-23, 8-137  
  置換変数, 3-26, 8-49  
  ホスト・システム・エディタ, 3-8, 8-50  
DEFINE 変数, 3-28, 8-99  
DELETE コマンド  
  使用禁止, E-5  
DEL コマンド, 3-2, 3-7, 8-51  
  アスタリスクの使用, 3-2, 8-51  
DESCRIBE コマンド (SQL\*Plus), 2-16, 8-53  
  connect\_identifier, 8-53  
  リスト表示できる PL/SQL プロパティ, 8-54  
  リスト表示できる表プロパティ, 8-53



DISABLED キーワード、使用禁止コマンド、E-4  
DISCONNECT コマンド、6-2, 8-59  
DOCUMENT コマンド  
  新しいバージョンの REMARK、F-4  
  廃止されたコマンド  
    DOCUMENT、F-2, F-3  
DOCUMENT 変数、F-2, F-5  
DROP コマンド  
  使用禁止、E-5  
DUPLICATES 句、8-21

## E

---

ECHO 変数、3-17, 8-99  
EDITFILE 変数、8-99  
EDIT コマンド、3-8, 8-60  
  \_EDITOR の定義、3-8, 8-60  
  コマンド・ファイルの作成、3-11  
  コマンド・ファイルの変更、3-19, 8-60  
  使用禁止、E-5  
  デフォルトのファイル名の設定、8-99  
EMBEDDED 変数、8-99  
EMP 表、1-4  
ENTMAP、7-5  
ENTMAP 句、4-49, 7-5, 8-30  
ESCAPE 変数、3-28, 8-100  
EXECUTE コマンド、8-62  
  使用禁止、E-5  
EXIT 句  
  WHENEVER OSERROR、8-145  
  WHENEVER SQLERROR、8-147  
EXIT コマンド、2-3, 8-63  
  :BindVariable 句、8-63  
  COMMIT 句、8-63  
  FAILURE 句、8-63  
  ROLLBACK 句、8-63  
  SET MARKUP とともに使用、4-39  
  WARNING 句、8-63  
  コマンド・ファイル内、8-127  
  使用禁止、E-5

## F

---

FAILURE 句、8-63  
FEEDBACK 変数、8-100  
FLAGGER 変数、8-100  
FLUSH 変数、8-100

FOLD\_AFTER 句、8-31  
FOLD\_BEFORE 句、8-31  
FORCE 句、8-128  
FORMAT 句、8-10, 8-31  
  ATTRIBUTE コマンド、8-17  
  COLUMN コマンド、4-5, 4-6  
  REPHEADER および REPFOOTER コマンド、8-87  
  TTITLE および BTITLE コマンド、4-28, 8-135  
FROM 句、8-78  
FROM 句 (SQL\*Plus)、6-5, 8-46

## G

---

GET コマンド、3-16, 8-65  
  LIST 句、8-65  
  NOLIST 句、8-65  
  コマンド・ファイルの取出し、3-16, 8-65  
  コマンド・ファイルの変更、3-19  
  使用禁止、E-5  
GLOGIN.SQL、3-20, 3-41, 3-44, 7-10  
  「LOGIN.SQL」を参照  
GRANT コマンド、E-2  
  使用禁止、E-5

## H

---

HEADING 句、4-2, 8-33  
HEADING 変数、8-101  
HEADSEP 変数、8-101  
  COLUMN コマンドでの使用、4-3, 8-34  
HEAD オプション、7-4  
HEAD 句、7-4  
HELP コマンド、8-66  
HIDE 句、8-11  
HOST コマンド、2-15, 8-67  
  コマンド・ファイル名のリスト表示、3-9  
  使用禁止、E-5  
HTML、4-38  
  エンティティ、4-49  
  オプション、7-4  
  句、7-4  
  タグ (tag)、4-38  
  ファイルへのスプール、7-6

## I

---

IMMEDIATE モード、8-123

INIT.ORA ファイル  
パラメータ・ファイル, 8-128  
INPUT コマンド, 3-2, 3-5, 8-69  
SAVE を使用したコマンド・ファイルの作成, 3-10  
複数行の入力, 8-69  
INSERT 句, 6-6, 8-47  
INSERT コマンド  
使用禁止, E-5  
INSTANCE 変数, 8-101  
iSQL\*Plus, B-2

## J

---

JUSTIFY 句, 8-34

## L

---

LABEL 変数  
SHOW コマンド, F-2, F-7  
LEFT 句, 4-26, 8-87, 8-135  
LIKE 句, 4-8, 8-17, 8-34  
LINESIZE 変数, 4-25, 4-32, 8-102  
LIST 句, 8-13, 8-65  
LIST コマンド, 3-2, 8-71  
アスタリスクの使用, 3-2, 8-71  
カレント行の判断, 3-3, 8-71  
最終行をカレント行にする, 3-3, 8-71  
LNO 句, 8-119  
LOBOFFSET 変数, 8-102  
LOCK TABLE コマンド  
使用禁止, E-5  
LOG\_ARCHIVE\_DEST パラメータ, 8-13  
LOGFILE 句, 8-78  
LOGIN.SQL, 3-20, 7-10  
「GLOGIN.SQL」を参照  
SET コマンドの挿入, 3-20  
タイトルの変数に現在の日付を格納, 4-31  
入力するサンプル・コマンド, 3-20  
LONGCHUNKSIZE 変数, 4-6, 8-31, 8-102  
LONG 変数, 8-102  
COPY コマンドへの影響, 6-8, 8-48  
LONG 列  
最大幅の設定, 8-102  
デフォルトの書式, 4-6, 8-31  
取出しサイズの設定, 8-102

## M

---

MARKUP, 4-38, 7-3, 8-102  
BODY 句, 7-4  
ENTMAP 句, 7-5  
HEAD 句, 7-4  
PREFORMAT 句, 7-7  
SPOOL 句, 7-4  
TABLE 句, 7-5  
例, 8-113  
MAXDATA 変数, F-2, F-6  
MAXIMUM 関数, 4-17  
MINIMUM 関数, 4-17  
MOUNT 句, 8-128

## N

---

NCHAR 句  
VARIABLE コマンド, 8-138  
NCHAR 列  
デフォルトの書式, 4-6, 8-31, 4-6, 8-31  
NCLOB 句  
VARIABLE コマンド, 8-139  
NCLOB 列  
最大幅の設定, 8-102  
デフォルトの書式, 4-6, 8-31  
取出し位置の設定, 8-102  
取出しサイズの設定, 8-102  
NEW\_VALUE 句, 4-30, 8-34  
タイトルの変数に現在の日付を格納, 4-31, 8-34  
NEWLINE 句, 8-34  
NEWPAGE コマンド, F-2, F-4  
NEWPAGE 変数, 4-32, 8-102  
NEXT 句, 8-14  
NLS\_DATE\_FORMAT, 8-10, 8-38  
NOAUDIT コマンド  
使用禁止, E-5  
NOLIST 句, 8-65  
NOLOG, 7-10  
NOMOUNT 句, 8-129  
NONE 句  
WHENEVER OSERROR, 8-145  
WHENEVER SQLERROR, 8-147  
NOPRINT 句, 4-17, 4-30, 8-34  
NOPROMPT 句, 8-11  
NORMAL モード, 8-123  
NULL 句, 8-35

NULL 値  
表示するテキストの設定, 8-35, 8-103  
NULL 変数, 8-103  
NUMBER 関数, 4-17  
NUMBER 句, 3-33, 8-10  
VARIABLE コマンド, 8-138  
NUMBER 列  
デフォルトの書式, 8-32  
書式の変更, 4-4  
デフォルトの書式, 4-4, 8-33  
NUMFORMAT 句  
LOGIN.SQL, 3-20  
NUMFORMAT 変数, 8-103  
NUMWIDTH 変数, 8-103  
NUMBER 列の書式への影響, 4-4, 8-33  
NVARCHAR2 列  
デフォルトの書式, 4-6, 8-31, 4-6, 8-31

## O

---

OFF 句, 8-35  
ATTRIBUTE コマンド, 8-17  
COLUMN コマンド, 4-9, 8-35  
REPFOOTER コマンド, 8-87  
REPHEADER コマンド, 8-87  
SPOOL コマンド, 4-34, 8-125  
TTITLE および BTITLE コマンド, 4-29, 8-134  
OF 句, 4-16  
OLD\_VALUE 句, 4-31, 8-35  
ON column 句  
BREAK コマンド, 4-12, 8-18  
COMPUTE コマンド, 4-16, 8-40  
ON expr 句  
BREAK コマンド, 8-19  
COMPUTE コマンド, 8-40  
ON REPORT 句  
BREAK コマンド, 4-20, 8-20  
COMPUTE コマンド, 4-20, 8-40  
ON ROW 句  
BREAK コマンド, 4-14, 8-20  
COMPUTE コマンド, 8-40  
ON 句  
ATTRIBUTE コマンド, 8-17  
COLUMN コマンド, 4-9, 8-35  
TTITLE および BTITLE コマンド, 4-29  
OPEN 句, 8-129

Oracle Net  
接続識別子, 8-44  
プロトコル, 6-3  
ORDER BY 句  
出力で値を一緒に表示, 4-11  
タイトルへの列値の表示, 4-30  
OUT 句, 4-35, 8-125

## P

---

PAGESIZE 句  
LOGIN.SQL, 3-20  
PAGESIZE 変数, 2-7, 4-32, 8-103  
PAGE 句, 8-86  
PARAMETERS 句, 8-119  
PASSWORD コマンド, 8-44, 8-73  
使用禁止, E-5  
PAUSE コマンド, 3-33, 8-74  
LOGIN.SQL, 3-20  
PAUSE 変数, 2-17, 8-103  
PL/SQL, 1-2, 2-9  
SQL\*Plus での出力を書式設定, 8-140  
SQL\*Plus でのモード, 2-8  
SQL コマンド, 2-8  
実行, 8-62  
定義のリスト表示, 2-17  
ブロック、PL/SQL, 2-9  
PLAN\_TABLE 表, 3-39  
PLUSTRACE ロール, 3-39  
PNO 句, 8-120  
PREFORMAT, 7-7  
PREFORMAT 句, 7-7  
PRINT 句, 8-34  
PRINT コマンド, 8-75  
PRODUCT\_USER\_PROFILE 表, E-2  
PRODUCT\_USER\_PROFILE 表の作成, E-2  
PROMPT 句, 3-31, 8-11  
PROMPT コマンド, 3-30, 8-76  
値の入力を求めるプロンプトのカスタマイズ, 3-32  
PUPBLD.SQL, E-2

## Q

---

QUIT コマンド, 8-63  
「EXIT」を参照

## R

---

### RECOVER

句, 8-129

### RECOVERY コマンド

データベースのリカバリ, 5-5

### RECOVER コマンド, 8-77

AUTOMATIC 句, 8-78

CANCEL 句, 8-79, 8-81

CONTINUE 句, 8-78

DATABASE 句, 8-79

FROM 句, 8-78

LOGFILE 句, 8-78

STANDBY DATABASE 句, 8-79

STANDBY DATAFILE 句, 8-80

STANDBY TABLESPACE 句, 8-80

UNTIL CANCEL 句, 8-79

UNTIL CONTROLFILE 句, 8-80

UNTIL TIME 句, 8-79

USING BACKUP CONTROL FILE 句, 8-79

### RECSEPCHAR 変数, 4-10, 8-103

### RECSEP 変数, 4-10, 8-103

### REDO ログ・ファイル

ARCHIVE LOG コマンド, 8-13

### REFCURSOR 句

VARIABLE コマンド, 8-139

### REFCURSOR バインド変数, 3-35

ストアド・ファンクション, 3-35

### RELEASE 句, 8-120

### REMARK コマンド, 3-12, 8-83

### RENAME コマンド

使用禁止, E-5

### REPFOOTER 句, 8-120

### REPFOOTER コマンド, 4-23, 8-84

BOLD 句, 8-87

CENTER 句, 8-87

COL 句, 8-87

FORMAT 句, 8-87

LEFT 句, 8-87

OFF 句, 8-87

RIGHT 句, 8-87

SKIP 句, 8-87

TAB 句, 8-87

現行の定義の抑止, 8-87

フッター要素の整列, 8-87

レポート・フッターをインデント, 8-87

### REPHEADER 句, 8-120

### REPHEADER コマンド, 4-23, 8-86

BOLD 句, 8-87

CENTER 句, 8-87

COL 句, 8-87

FORMAT 句, 8-87

LEFT 句, 8-87

OFF 句, 8-87

PAGE 句, 8-86

RIGHT 句, 8-87

SKIP 句, 8-87

TAB 句, 8-87

現行の定義の抑止, 8-87

ヘッダー要素の整列, 4-25, 8-87

ヘッダーをインデント, 8-87

### REPLACE 句

COPY コマンド, 6-6, 8-47

SAVE コマンド, 3-19, 8-91

### RESTRICT, 7-8, 8-128, E-9

### REVOKE コマンド, E-2

使用禁止, E-5

### RIGHT 句, 4-26, 8-87, 8-135

### ROLLBACK 句, 8-63

WHENEVER OSERROR, 8-145

WHENEVER SQLEERROR, 8-147

### RUN コマンド, 2-9, 8-90

/ (スラッシュ) コマンドとの類似, 2-9, 8-90

カレント PL/SQL ブロックの実行, 2-10

現行の SQL コマンドまたは PL/SQL ブロックの実行, 2-9

最終行をカレント行にする, 3-3

使用禁止, E-5

## S

---

### SAVE コマンド, 3-9, 8-91

APPEND 句, 3-19, 8-91

CREATE 句, 8-91

INPUT を使用したコマンド・ファイルの作成, 3-10

REPLACE 句, 3-19, 8-91

コマンド・ファイルへのコマンドの格納, 3-9, 8-91

コマンド・ファイルの変更, 3-19

使用禁止, E-5

### SCAN 変数, F-2, F-6

### SCREEN 句, 3-33, 8-28

### SELECT コマンド

BREAK コマンド, 4-11, 8-19, 8-20

COLUMN コマンド, 8-30

COMPUTE コマンド, 4-11  
 COPY コマンド, 6-6, 8-47  
 DEFINE コマンド, 8-49  
 ORDER BY 句, 4-11  
 使用禁止, E-5  
 タイトルの変数に現在の日付を格納, 4-31  
 SELECT 文  
   結果の書式設定, 3-35  
 SERVEROUTPUT 変数, 8-104  
 SET AUTOTRACE, 3-39  
 SET MARKUP  
   BODY 句, 7-4  
   ENTMAP 句, 4-49, 7-5  
   HEAD 句, 7-4  
   HTML, 7-4  
   PREFORMAT 句, 7-7  
   SPOOL 句, 7-4  
   TABLE 句, 7-5  
   埋め込まれた CGI の例, 4-43  
   対話型 HTML の例, 4-39, 4-42  
   「SPOOL コマンド」を参照  
 SET ROLE コマンド  
   使用禁止, E-5  
 SET TRANSACTION コマンド  
   使用禁止, E-5  
 SET 句, 8-131  
 SET コマンド, 2-12, 3-21, 8-93  
   APPINFO 変数, 8-95  
   ARRAYSIZE 変数, 6-8, 8-95  
   AUTOCOMMIT 変数, 2-13, 8-96  
   AUTOPRINT 変数, 8-96, 8-140  
   AUTORECOVERY 変数, 8-96  
   AUTOTRACE 変数, 8-97  
   BLOCKTERMINATOR 変数, 8-97  
   BUFFER 変数, F-4  
   CLOSECURSOR 変数, F-2, F-5  
   CMDSEP 変数, 8-97  
   COLSEP 変数, 4-35, 8-98  
   COMPATIBILITY 変数, 8-98  
   CONCAT 変数, 3-28, 8-98  
   COPYCOMMIT 変数, 6-8, 8-98  
   COPYTYPECHECK 変数, 8-98  
   DEFINE 句, 3-28  
   DEFINE 変数, 8-99  
   DESCRIBE 変数, 8-99  
   DOCUMENT 変数, F-2, F-5  
   ECHO 変数, 3-17, 8-99  
   EDITFILE 変数, 8-99  
   EMBEDDED 変数, 8-99  
   ESCAPE 変数, 3-28, 8-100  
   FEEDBACK 変数, 8-100  
   FLAGGER 変数, 8-100  
   FLUSH 変数, 8-100  
   HEADING 変数, 8-101  
   HEADSEP 変数, 4-3, 8-101  
   INSTANCE 変数, 8-101  
   LINESIZE 変数, 4-25, 4-32, 8-102  
   LOBOFFSET 変数, 8-102  
   LOGSOURCE 変数, 8-102  
   LONGCHUNKSIZE 変数, 8-102  
   LONG 変数, 6-8, 8-102  
   MARKUP 句, 8-102  
   MAXDATA 変数, F-2, F-6  
   NEWPAGE 変数, 4-32, 8-102  
   NULL 変数, 8-103  
   NUMFORMAT 句, 3-20  
   NUMFORMAT 変数, 8-103  
   NUMWIDTH 変数, 4-4, 8-33, 8-103  
   PAGESIZE 句, 3-20  
   PAGESIZE 変数, 2-7, 4-32, 8-103  
   PAUSE 句, 3-20  
   PAUSE 変数, 2-17, 8-103  
   RECSEPCHAR 変数, 4-10, 8-103  
   RECSEP 変数, 4-10, 8-103  
   REFCURSOR 変数の書式設定に使用, 8-140  
   SCAN 変数, F-2, F-6  
   SERVEROUTPUT 変数, 8-104  
   SHIFTINOUT 変数, 8-104  
   SPACE 変数, F-2, F-6  
   SQLBLANKLINES 変数, 8-105  
   SQLCASE 変数, 8-105  
   SQLCONTINUE 変数, 8-105  
   SQLNUMBER 変数, 8-105  
   SQLPLUSCOMPATIBILITY 変数, 8-105, B-3  
   SQLPREFIX 変数, 8-106  
   SQLPROMPT 変数, 8-106  
   SQLTERMINATOR 変数, 8-106  
   SUFFIX 変数, 8-106  
   TAB 変数, 8-106  
   TERMOUT 変数, 4-31, 8-106  
   TIME 句, 3-20  
   TIME 変数, 8-106  
   TIMING 変数, 8-107  
   TRIMOUT 変数, 8-107

- TRIMSPOOL 変数, 8-107
- TRUNCATE 変数, F-2, F-7
- UNDERLINE 変数, 4-3, 8-107
- VERIFY 句, 3-24
- VERIFY 変数, 3-28, 8-107
- WRAP 変数, 4-7, 8-107
- 使用禁止, E-5
- SET コマンド変数
  - システム変数, 2-12
- SGA 句, 8-120
- SHIFTINOUT 変数, 8-104
- SHOWMODE 変数, 8-104
- SHOW 句, 8-132
- SHOW コマンド, 2-13, 8-118
  - ALL 句, 8-118
  - BTITLE 句, 8-119
  - ERRORS 句, 8-119
  - LABEL 変数, F-2, F-7
  - LNO 句, 8-119
  - PNO 句, 8-120
  - RELEASE 句, 8-120
  - REPFOOTER 句, 8-120
  - REPHEADER 句, 8-120
  - SPOOL 句, 8-120
  - SQLCODE 句, 8-120
  - TTITLE 句, 8-120
  - USER 句, 8-120
  - 現在のページ・サイズのリスト表示, 4-34
- SHUTDOWN コマンド, 8-123
  - ABORT, 8-123
  - IMMEDIATE, 8-123
  - NORMAL, 8-123
- SKIP PAGE 句, 4-13, 4-14, 8-21
- SKIP 句
  - BREAK コマンド, 4-13, 4-14, 8-20
  - REPHEADER および REPFOOTER コマンド, 8-87
  - TTITLE および BTITLE コマンド, 4-26, 8-135
  - 下部タイトルの前に空白行を配置するための使用方法, 4-26
- SPACE 変数, F-2, F-6
- SPOOL 句, 7-4, 7-6, 8-120
- SPOOL コマンド, 4-34, 8-125
  - HTML ファイル, 7-6
  - OFF 句, 4-34, 8-125
  - OUT 句, 4-35, 8-125
  - SET MARKUP とともに使用, 4-39
  - 使用禁止, E-5
  - スプールを OFF にする, 4-34, 8-125
  - ファイル名, 4-35, 8-125
- SQL DML 文
  - レポート, 8-97
- SQL\*Plus
  - LOGIN.SQL, 3-20
  - 概要, 1-2
  - 環境の設定, 3-20
  - 起動, 2-2, 7-2
  - 起動のショートカット, 2-3
  - 基本概念, 1-2
  - コマンド一覧, 8-2
  - コマンド・プロンプト, 2-3
  - 実行に必要なもの, 1-4
  - 終了, 2-3, 8-63
  - 条件付き終了, 8-145, 8-147
  - 制限, C-1
  - 対象ユーザー, 1-3
  - データベース管理, 5-2
  - 廃止されたコマンドのかわりに使用, F-2
  - バッチ・モードでのコマンドの実行, 3-20, 8-64
- SQL\*Plus の起動, 2-2
  - ショートカット, 2-3
- SQL.PNO、レポート・タイトル内での参照, 4-28
- SQL.SQLCODE
  - EXIT コマンドでの使用, 8-63
- SQLBLANKLINES 変数, 8-105
- SQLCASE 変数, 8-105
- SQLCODE 句, 8-120
  - SHOW コマンド, 8-120
- SQLCONTINUE 変数, 8-105
- SQLNUMBER 変数, 8-105, B-3
- SQLPLUS コマンド, 2-2, 7-2
  - ? 句, 7-2
  - MARKUP SPOOL 句, 7-4
  - MARKUP オプション, 7-3
  - MARKUP 句, 7-4
  - SILENT オプション, 4-49, 7-8
  - SILENT 句, 7-8
  - 句, 7-2
  - /NOLOG 句, 7-10
  - @ (アットマーク), 3-18, 6-4, 7-2
  - BODY オプション, 7-4
  - ENTMAP オプション, 7-5
  - EXIT FAILURE, 7-11
  - HEAD オプション, 7-4
  - HTML オプション, 7-4

PREFORMAT オプション, 7-7  
RESTRICT, 7-8, E-9  
SPOOL 句, 7-6  
SYSDBA 句, 7-9  
TABLE オプション, 7-5  
構文, 7-2  
コマンド・ファイルの実行, 3-18  
サービス名, 6-4, 7-9  
失敗した接続, 7-11  
接続識別子, 7-9  
表示構文, 7-2  
ユーザー名 / パスワード, 2-2, 7-9  
リモート・データベースへの接続, 6-4  
SQLPREFIX 変数, 8-106  
SQLPROMPT 変数, 8-106  
SQLTERMINATOR, 8-105  
SQLTERMINATOR 変数, 8-67, 8-106  
SQL 句, 8-28  
SQL コマンド、主要なリスト, D-1  
SQL データベース言語, 1-2  
SQL バッファ, 2-4  
STANDBY DATAFILE 句, 8-80  
STANDBY TABLESPACE 句, 8-80  
STARTUP コマンド, 8-128  
FORCE 句, 8-128  
MOUNT 句, 8-128  
NOMOUNT 句, 8-129  
OPEN 句, 8-129  
PFILE 句, 8-128  
RECOVER 句, 8-129  
RESTRICT 句, 8-128  
データベースの指定, 8-128  
START 句, 8-14, 8-132  
START コマンド, 3-17, 8-126  
@@ (二重アットマーク) コマンドとの類似, 8-7, 8-127  
@ (アットマーク) コマンドとの類似, 3-17, 8-6, 8-127  
コマンド・ファイル, 3-17, 8-126  
コマンド・ファイルへのパラメータの受渡し, 3-29, 8-126  
使用禁止, E-5  
引数, 3-29, 8-126  
STD 関数, 4-17  
STOP 句, 8-14, 8-132  
STORE コマンド, 3-21, 8-131  
SET 句, 8-131

SUFFIX 変数, 8-106  
@(アットマーク) コマンドとの併用, 8-5  
EDIT コマンドとの併用, 8-60  
GET コマンドとの併用, 8-65  
SAVE コマンドとの併用, 8-91  
START コマンドとの併用, 8-126  
SUM 関数, 4-17  
SYSDATE, 4-31  
SYSDBA 句, 8-45  
SYSOPER 句, 7-9, 8-45

## T

---

TABLESPACE 句, 8-79  
TABLE オプション, 7-5  
TABLE 句, 7-5  
TAB 句, 8-87, 8-135  
TAB 変数, 8-106  
TERMOUT 変数, 8-106  
SPOOL コマンドでの使用, 8-125  
タイトルの変数に現在の日付を格納, 4-31  
TIME 句  
LOGIN.SQL, 3-20  
TIME 変数, 8-106  
TIMING 句, 8-28  
TIMING コマンド, 2-15, 8-132  
SHOW 句, 8-132  
START 句, 8-132  
STOP 句, 8-132  
現行の領域を削除, 8-132  
作成されたすべての領域の削除, 8-28  
TIMING 変数, 8-107  
TO 句, 6-5, 8-46  
TRIMOUT 変数, 8-107  
TRIMPOOL 変数, 8-107  
TRUNCATED 句, 4-8, 8-35  
TRUNCATE コマンド  
使用禁止, E-5  
TRUNCATE 変数, F-2, F-7  
TTITLE 句, 8-120  
TTITLE コマンド, 4-23, 8-134  
BOLD 句, 8-135  
CENTER 句, 4-26, 8-135  
COL 句, 4-27, 8-135  
FORMAT 句, 4-28, 8-135  
LEFT 句, 4-26, 8-135  
OFF 句, 4-29, 8-134

ON 句, 4-29  
RIGHT 句, 4-26, 8-135  
SKIP 句, 4-26, 8-135  
TAB 句, 8-135  
旧形式, F-7  
現行の定義の復元, 4-29  
現行の定義の抑止, 4-29, 8-134  
現行の定義のリスト表示, 4-29, 8-136  
タイトルのインデント, 4-27, 8-135  
タイトル要素の整列, 4-25, 8-135  
列値変数の参照, 4-30, 8-34

## U

---

UNDEFINE コマンド, 3-23, 8-137  
  DEFINE コマンド, 8-49  
UNDERLINE 変数, 4-3, 8-107  
UNTIL CANCEL 句, 8-79  
UNTIL CHANGE 句, 8-79  
UNTIL CONTROLFILE 句, 8-80  
UNTIL TIME 句, 8-79  
UPDATE コマンド  
  使用禁止, E-5  
USER 句, 8-120  
USING BACKUP CONTROL FILE 句, 8-79  
USING 句, 6-6, 8-47

## V

---

V\$SESSION 仮想表, 8-95  
V\$SQLAREA 仮想表, 8-95  
VARCHAR2 句  
  VARIABLE コマンド, 8-139  
VARCHAR2 列  
  デフォルトの書式, 4-6, 8-31, 4-6  
VARCHAR 列  
  デフォルトの書式, 4-6, 8-31  
VARIABLE コマンド, 8-138  
  CHAR 句, 8-138  
  CLOB 句, 8-139  
  NCHAR 句, 8-138  
  NCLOB 句, 8-139  
  NUMBER 句, 8-138  
  REFCURSOR 句, 8-139  
  VARCHAR2 句, 8-139  
  変数句, 8-138  
VARIANCE 関数, 4-17

VERIFY 句, 3-24  
VERIFY 変数, 3-28, 8-107

## W

---

WARNING 句, 8-63  
Web  
  レポートの出力, 4-38  
Web ブラウザ, 4-38  
WHENEVER OSERROR コマンド, 8-145  
  COMMIT 句, 8-145  
  CONTINUE 句, 8-145  
  EXIT 句, 8-145  
  NONE 句, 8-145  
  ROLLBACK 句, 8-145  
WHENEVER SQLERROR コマンド, 3-20, 8-147  
  COMMIT 句, 8-147  
  CONTINUE 句, 8-147  
  EXIT 句, 8-147  
  NONE 句, 8-147  
  ROLLBACK 句, 8-147  
WORD\_WRAPPED 句, 4-8, 4-10, 8-35  
WRAPPED 句, 4-8, 8-35  
WRAP 変数, 4-7, 8-107

## あ

---

アクセス、拒否および付与, E-2  
値の入力を求めるプロンプト  
  ACCEPT を使用した, 3-30  
  カスタマイズ, 3-32  
  置換変数を使用した, 3-24  
  パラメータによるバイパス, 3-29  
アンパサンド (&)  
  置換変数, 3-23  
  パラメータ, 3-29, 8-5, 8-126

## い

---

異常終了モード, 8-123  
インスタンス  
  起動, 8-128  
  停止, 8-123  
インストール  
  デフォルト・ユーザー作成, 1-5



## え

---

エスケープ文字、定義、8-100  
エラー・メッセージ、解釈、2-17  
エラー、含まれている行をカレント行にする、3-3  
エンティティ、HTML、4-49

## お

---

オブジェクト、説明、8-99  
オンライン・ヘルプ、2-5, 7-12, 8-66

## か

---

カーソル変数、8-140  
改ページ、新しいページを始める、4-32, 8-102  
拡張機能リスト、リリース 8.1.6, B-2  
拡張子、8-91, 8-106, 8-131  
画面、消去、3-33, 8-28  
環境属性の保存、8-131  
カンマ、数値書式、4-5

## き

---

起動時のデータベース名、8-128  
行  
COPY でのコミット後に数を設定、8-98  
一度に取り出す数の設定、8-95  
終わりの空白を削除、8-107  
カレント行の後への新規行の追加、3-5, 8-69  
カレント行の判断、3-3  
カレント行の編集、3-3  
計算の実行、4-16, 8-39  
テキストの追加、3-6, 8-12  
バッファからの削除、3-7, 8-51  
バッファ内のすべての行の削除、8-51  
バッファ内のすべての行のリスト表示、3-2, 8-71  
バッファの最後に追加、8-69  
バッファの最初に追加、8-69  
幅の変更、4-32, 8-102  
行番号、SQL コマンド、2-6

## く

---

空白行  
PL/SQL ブロック、2-10  
SQL コマンド、2-8

SQL コマンドでの保存、8-105  
空白行、表示、8-74

## こ

---

構文  
COPY コマンド、6-5  
構文規則  
SQL\*Plus コマンド、2-12  
SQL コマンド、2-7  
コマンド、1-2  
SQL  
構文に従う、2-7  
実行、2-9, 8-9, 8-90  
実行せずに入力する、2-8  
終了、2-7  
終了および実行のために使用する文字設定、8-106  
主要なリスト、D-1  
次の行への継続、2-7  
入力および実行、2-6  
バッファ内の現行のコマンドのリスト表示、3-3  
バッファ内の編集、3-2  
ホスト・システム・エディタによる編集、3-7, 8-60  
保存、3-9, 8-91  
SQL\*Plus  
SQL コマンドのエントリ中に入力、8-106  
コマンド一覧、8-2  
コマンド・プロンプトでの編集、3-2  
終了、2-12, 8-1  
次の行への継続、2-12, 8-1  
入力および実行、2-11  
廃止されたコマンドのかわりに使用、F-2  
略称、2-11  
空白、2-4  
コマンド・ファイルへの格納、3-8  
実行中の停止、2-14  
実行に影響する変数、2-12  
種類、2-4  
使用禁止、E-4  
タイミング統計の収集、2-15, 8-132  
対話式での記述、3-22  
タブ、2-4  
バッファ内の現行のコマンドのリスト表示、8-71  
再び使用可能にする、E-4  
ホスト、SQL\*Plus からの実行、2-15, 8-67

コマンド・ファイル, 3-8  
  @@ (二重アットマーク) コマンド, 8-7  
  @ (アットマーク) コマンド, 3-17, 8-5  
  EDIT コマンド, 3-19, 8-60  
  GET および SAVE を使用した編集, 3-19  
  GET コマンド, 3-16, 8-65  
  HOST コマンドによる名前のリスト表示, 3-9  
  INPUT および SAVE での作成, 3-10  
  SAVE コマンド, 3-9, 3-10, 8-91  
  SAVE での作成, 3-9, 8-91  
  SQL\*Plus の起動時の実行, 3-18, 7-10  
  SQLPLUS コマンド, 3-18, 7-10  
  START コマンド, 3-17, 8-126  
  Uniform Resource Identifier, 8-5, 8-7, 8-126  
  異常終了およびリターン・コードを返す終了,  
    3-20, 8-146, 8-148  
  一連のコマンド・ファイルの順次実行, 3-18  
  エンド・ユーザーによる入力, 3-22  
  コメントの挿入, 3-12, 8-83  
  システム・エディタでの作成, 3-11  
  実行, 3-17, 8-5, 8-126  
  登録, 8-95  
  取出し, 3-16, 8-65  
  ネスト, 3-18, 8-7  
  バッチ・モードでの実行, 3-20, 8-64  
  バッファ内容の保存, 3-9, 8-91  
  パラメータの受渡し, 3-29, 8-5, 8-126  
  複数の PL/SQL ブロックの挿入, 3-11  
  複数の SQL コマンドの挿入, 3-11  
  ホスト・システム・エディタによる編集, 3-19,  
    8-60  
コマンド・ファイル拡張子, 8-91, 8-106, 8-131  
コマンド・プロンプト  
  SET SQLPROMPT, 8-106  
  SQL\*Plus, 2-3  
  ホスト・オペレーティング・システム, 2-2  
コメント  
  -- を使用した作成, 3-13  
  /\*...\*/ を使用した作成, 3-12  
  REMARK を使用した作成, 3-12, 8-83, F-2  
  コマンド・ファイルへの挿入, 3-12, 8-83, F-2  
コロロン (:)  
  バインド変数, 3-34

## さ

---

サービス名  
  CONNECT コマンド, 6-3  
  COPY コマンド, 6-5, 6-7, 6-9  
  SQLPLUS コマンド, 6-4  
サイト・プロファイル  
  GLOGIN.SQL, 7-10  
  LOGIN.SQL, 7-10  
  「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, 4-21  
  計算および出力, 4-16, 8-39  
  異なる列についての同じタイプの計算, 4-21  
  総およびサブサマリー (合計) の出力, 4-20  
  レポートの終わりでの計算および出力, 4-20  
サンプル表  
  アクセス, 1-6

## し

---

システム管理値  
  タイトル内での書式設定, 4-29  
  タイトルの中に表示, 4-28, 8-134  
  ヘッダーおよびフッターの中に表示, 8-86  
システム変数, 2-12, 8-108  
  格納および復元, 3-21  
  現行の設定のリスト表示, 2-13, 8-118  
  現在の設定の変更, 8-93  
  新旧の値のリスト表示, 8-104  
  置換変数との併用, 3-28  
実行  
  CREATE コマンド, 2-8  
実行統計  
  レポートへの挿入, 8-97  
シャープ記号 (#), 8-33  
終了、条件付き, 8-145, 8-147  
出力  
  REFCURSOR 変数, 8-140  
  SPOOL コマンド, 8-125  
  空白の書式設定, 8-106  
  問合せ結果, 1-2  
  バインド変数の自動出力, 8-96  
  表示中の一時停止, 2-17, 8-103  
使用禁止  
  PL/SQL コマンド, E-6  
  SQL\*Plus コマンド, E-4

SQL コマンド, E-4  
初期化パラメータ  
表示, 8-119  
書式モデル、数値, 4-5, 8-33

---

## す

数値書式  
\$, 4-5  
0, 4-5  
9, 4-5  
カンマ, 4-5  
デフォルト設定, 8-103  
ストアド・ファンクション, 3-35  
ストアド・プロシージャ  
作成, 2-8  
スラッシュ (/) コマンド, 8-9  
GET コマンドによりロードされたファイルとの併  
用, 8-65

---

## せ

制限、SQL\*Plus, C-1  
セキュリティ  
PRODUCT\_USER\_PROFILE 表, E-2  
RESTRICT, 7-8, E-9  
埋込み型の Web レポート, 4-44  
パスワードの変更, 8-73  
接続識別子, 6-3, 8-44  
CONNECT コマンド, 8-44  
COPY コマンド, 8-46  
DESCRIBE コマンド, 8-53  
SQLPLUS コマンド, 7-9  
セミコロン (;)  
PL/SQL ブロック, 2-10  
SQL\*Plus コマンド, 2-12, 8-1  
SQL コマンド, 2-6, 2-7  
コマンド・ファイル入力時には不要, 3-11  
バッファに格納されない, 3-3

---

## そ

属性  
表示特性, 8-16

---

## た

タイトル  
インデント, 4-27, 8-135  
現在の定義のリスト表示, 4-29, 8-23, 8-136  
現在の日付の表示, 4-31, 8-34, 8-37  
最終行から下部タイトルまでの間隔, 4-26  
システム管理値の書式設定, 4-29  
システム管理値の表示, 4-28, 8-134  
上部および下部の設定, 4-23, 8-23, 8-134, F-2  
上部タイトルからページの終わりまでの行の設定,  
8-103  
定義の復元, 4-29  
定義の抑止, 4-29, 8-134  
ページ下部への表示, 4-23, 8-23, F-2  
ページ最上部から上部タイトルまでの行数の設定,  
4-32, 8-102, F-2  
ページ上部への表示, 4-23, 8-134, F-2  
ページ番号の表示, 4-28, 8-136  
要素の書式設定, 8-135  
要素の整列, 4-25, 8-135  
列値の表示, 4-30, 8-34, 8-35  
レポートの始めまたは終わりに設定, 4-23  
タグ、HTML, 4-38

---

## ち

置換変数, 3-23  
DEFINE コマンド, 3-26, 8-49  
値の入力を求める不要なプロンプトの回避, 3-26  
システム変数との併用, 3-28  
使用するおよび方法, 3-24  
制限, 3-28  
接頭辞, 8-99, F-2  
単一および二重のアンパサンド, 3-26  
直後に文字を追加, 3-26  
定義, 3-26  
未定義, 3-24  
連結文字, 8-98  
チューニング, 3-39

---

## 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ク  
ピー, 6-9  
SQL\*Plus を終了せずに切断する, 6-2, 8-59

オープン, 5-3, 8-129  
～間のデータのコピー, 6-4, 8-46  
起動, 5-2  
接続識別子, 8-44  
停止, 5-2, 5-3  
デフォルト・データベースへの接続, 6-2, 8-44  
マウント, 5-3, 8-128  
リカバリ, 5-5, 8-77  
リモート・データベースへの接続, 6-3, 8-44  
データベース管理者, 1-4, 5-2  
データベースのオープン, 8-129  
データベースの変更、自動保存, 2-13, 8-96  
データベースのマウント, 8-128  
データベース・ファイル  
リカバリ, 8-77  
テキスト, 7-4, 7-5  
APPEND によるカレント行への追加, 3-6, 8-12  
CHANGE による旧から新への変更, 3-3, 8-25  
バッファからの消去, 3-2, 8-27  
テキスト・エディタ、ホスト・オペレーティング・システム, 3-7, 8-60  
デフォルト・ログイン  
インストール時に作成, 1-5

## と

---

問合せ, 1-2  
COPY コマンド, 6-6, 8-47  
取り出したレコード数の表示, 2-6, 8-100  
問合せ結果, 1-2  
画面に表示, 2-6  
ファイルへの格納, 4-35, 8-125  
プリンタへの出力, 4-35, 8-125  
問合せ実行パス  
レポートへの挿入, 8-97  
問合せの中断, 2-14  
問合せの停止, 2-14  
問合せの取消し, 2-14  
統計, 3-41  
動的レポート, B-2

## な

---

長い SQL\*Plus コマンドの継続, 2-12, 8-1

## に

---

入力

[Return] のアクセプト, 3-33, 8-74  
ユーザーからの値のアクセプト, 3-30, 8-10

## は

---

廃止されたコマンド

BTITLE, F-3  
COLUMN コマンドの DEFAULT 句, F-3  
NEWPAGE, F-2, F-4  
SET コマンドの BUFFER 変数, F-4  
SET コマンドの CLOSECURSOR 変数, F-2, F-5  
SET コマンドの DOCUMENT 変数, F-2, F-5  
SET コマンドの MAXDATA 変数, F-2, F-6  
SET コマンドの SCAN 変数, F-2, F-6  
SET コマンドの SPACE 変数, F-2, F-6  
SET コマンドの TRUNCATE 変数, F-2, F-7  
SHOW コマンドの LABEL 変数, F-2, F-7  
TTITLE コマンドの旧形式, F-7

ハイフン

長い SQL\*Plus コマンドの継続, 2-12, 8-1

バインド変数, 3-34

COPY コマンド, 8-140  
PL/SQL ブロック, 8-140  
REFCURSOR, 3-35  
SQL 文, 8-140  
作成, 8-138  
自動表示, 8-96, 8-140  
表示, 8-75

パスワード, 1-5

CONNECT コマンド, 6-2, 6-3, 8-44  
COPY コマンド, 6-5, 6-7, 6-9  
PASSWORD コマンドで変更, 8-73  
SQLPLUS コマンド, 2-3, 6-4, 7-9

バックグラウンド・プロセス

異常終了後の起動, 8-123

バッチ・モード, 3-20, 8-64

バッファ, 2-4

1 行の削除, 3-2, 8-51  
1 行のリスト表示, 3-2, 8-71  
ある範囲の行の削除, 3-2, 8-51  
ある範囲の行のリスト表示, 3-3, 8-71  
カレント行の削除, 3-2, 8-51  
カレント行のリスト表示, 3-2, 8-71  
行の削除, 3-7, 8-51

行へのテキストの追加, 3-6, 8-12  
最終行の削除, 3-2, 8-51  
最終行のリスト表示, 3-2, 8-71  
新規行の挿入, 3-5, 8-69  
すべての行の削除, 3-2, 8-27, 8-51  
すべての行のリスト表示, 3-2, 8-71  
内容の実行, 2-9, 8-9, 8-90  
内容の保存, 3-9, 8-91  
内容のリスト表示, 3-3, 8-71  
ホスト・システム・エディタへのロード, 3-8, 8-60  
パフォーマンス  
SQL 文, 3-39  
パフォーマンス、ダイアルアップ行, 8-107  
パラメータ, 3-29, 8-5, 8-126  
パラメータ・ファイル (INIT.ORA ファイル)  
代替を指定, 8-128

## ひ

---

### 引数

START コマンド, 3-29, 8-126  
日付、タイトルの変数に現在の日付を格納, 4-31,  
8-34, 8-37

### 表

値のコピー, 6-4, 6-9, 8-46  
コピー時に別のユーザーの表を参照, 6-8  
コピー時の宛先の制御, 6-6, 8-47  
サンプルへのアクセス, 1-6  
列定義のリスト表示, 2-16, 8-53

### 表領域

リカバリ, 8-77

### ピリオド (.)

PL/SQL ブロックの終了, 2-10, 8-97

## ふ

---

### ファイル

コマンド・ファイル, 3-8  
フラット, 4-34  
ファイル拡張子, 3-21, 8-91, 8-106, 8-131  
ファイル名  
@@ (二重アットマーク) コマンド, 8-7  
@ (アットマーク) コマンド, 8-5  
EDIT コマンド, 8-60  
GET コマンド, 8-65  
SAVE コマンド, 3-9, 8-91  
SPOOL コマンド, 4-35, 8-125

SQLPLUS コマンド, 7-10  
START コマンド, 8-126  
フッター  
インデント, 8-87  
現行の定義のリスト表示, 8-84  
システム管理値の表示, 8-86  
定義の抑止, 8-87  
ページ下部への表示, 8-84  
要素の書式設定, 8-87  
要素の整列, 8-87  
レポートの終わりに設定, 4-23  
負の無限大記号 (--), 8-33  
ブラウザ、Web, 4-38  
フラット・ファイル, 4-34  
フラット・ファイルの作成, 4-34  
ブレイク定義  
現行の取消し, 4-16, 8-27  
現行のリスト表示, 4-16, 8-21  
ブレイク列, 4-11, 8-18  
値が変更されたときの空白の挿入, 4-13  
値の重複の抑止, 4-12  
複数の指定, 4-14  
ブロック、PL/SQL, 1-2  
SQL コマンド, 2-8  
SQL バッファからの実行, 2-9  
SQL バッファに格納, 2-9  
計時統計, 8-107  
継続, 2-10  
コマンド・ファイルへの格納, 3-8  
終了のために使用する文字設定, 8-97  
入力および実行, 2-10  
バッファ内の現行のコマンドのリスト表示, 3-3  
バッファ内の編集, 3-2  
ホスト・システム・エディタによる編集, 3-7, 8-60  
保存, 3-9, 8-91  
プロンプト  
SET SQLPROMPT, 8-106  
文のトレース, 3-39  
データベース・リンクの使用, 3-44  
問合せ実行パス用, 3-41  
パフォーマンス統計用, 3-41  
パラレル問合せオプション, 3-45

## へ

---

### ページ

サイズの設定, 4-32

サイズを画面または用紙のサイズに合わせる, 4-32  
デフォルトのサイズ, 4-32  
長さの変更, 4-32, 8-103  
ページ番号、タイトルへの挿入, 4-15, 4-28  
ヘッダー  
インデント, 8-87  
現行の定義のリスト表示, 8-88  
システム管理値の表示, 8-86  
定義の抑止, 8-87  
ページ上部への表示, 8-86  
要素の書式設定, 8-87  
要素の整列, 4-25, 8-87  
抑止, 4-25  
列ヘッダー, 8-101  
レポートの始めに設定, 4-23  
ヘルプ、オンライン, 2-5, 7-12, 8-66  
変数  
システム変数, 2-12  
置換変数, 3-23  
バインド変数, 3-34  
ユーザー変数, 8-49

## 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, 2-15, 8-67  
エディタ, 3-7, 8-60  
コマンド・プロンプト, 2-2  
ファイル、バッファへのロード, 8-65

## む

---

無限大記号 (∞), 8-33

## め

---

メッセージ、画面への送信, 3-30, 8-76  
メディア・リカバリ, 8-129

## ゆ

---

ユーザー・プロファイル, 3-20  
GLOGIN.SQL, 7-10  
LOGIN.SQL, 7-10  
「サイト・プロファイル」を参照  
ユーザー変数, 3-22  
1つの定義のリスト表示, 3-23, 8-49

ACCEPT コマンド, 3-30, 8-10  
削除, 3-23, 8-137  
すべての定義のリスト表示, 3-23, 8-49  
タイトルの中に表示, 8-134  
定義, 3-22, 8-49  
ヘッダーおよびフッターの中に表示, 8-86  
ユーザー名, 1-5  
CONNECT コマンド, 6-2, 6-3, 8-44  
COPY コマンド, 6-5, 6-7, 6-9  
SQLPLUS コマンド, 2-2, 6-4, 7-9  
インストール時に作成, 1-5  
異なるユーザー名での接続, 6-2, 8-44

## ら

---

ラベル  
COMPUTE コマンド, 4-17, 8-40

## り

---

リカバリ  
RECOVER コマンド, 8-77  
リターン・コード、指定, 3-20, 8-64, 8-148

## れ

---

例  
埋め込まれた CGI レポート, 4-43  
対話型 HTML レポート, 4-39, 4-42  
レコード・セパレータ、出力, 4-10, 8-103  
列  
1つの列の表示属性のリスト表示, 4-9, 8-29  
CHAR、VARCHAR、LONG および DATE の書式  
設定, 8-31  
MLSLABEL、RAW MLSLABEL、ROWLABEL の書  
式設定, 8-31  
NUMBER の書式設定, 4-4, 8-32  
値がオーバーフローした場合の全列値の折返し表  
示, 4-7, 8-107  
値がオーバーフローした場合の全列値の切捨て表  
示, 4-7, 8-107  
値がオーバーフローした場合の列値の折返し表示,  
4-8, 8-35  
値がオーバーフローした場合の列値の切捨て表示,  
4-8, 8-35  
新しい行の開始, 8-34  
オーバーフローする値の後の行の出力, 4-10, 8-103

下部タイトルへの値の表示, 4-31, 8-35  
コピー時の宛先の表での名前, 6-5, 8-47  
サマリー行の計算, 4-16, 8-39  
出力を OFF または ON に設定, 4-30, 8-34  
上部タイトルへの値の表示, 4-30, 8-34  
すべての単語を 1 列に折返し, 4-10  
すべての列の表示属性のリスト表示, 4-9, 8-29  
すべての列の表示をデフォルトにリセット, 4-9,  
8-27  
表間での値のコピー, 6-4, 6-9, 8-46  
表示属性のコピー, 4-8, 8-17, 8-34  
表示属性の復元, 4-9, 8-17, 8-35  
表示属性の抑止, 4-9, 8-17, 8-35  
別名の割当て, 8-30  
変数への値の格納, 4-30, 8-34  
列の表示をデフォルトにリセット, 4-9, 8-30, F-2  
レポート内での書式設定, 4-2, 8-29  
列セパレータ, 8-98, F-2  
列ヘッダー  
CHAR および LONG 列で切り捨てられる場合, 4-6  
DATE 列で切り捨てられる場合, 4-6  
NUMBER 列で切り捨てられる場合, 4-4  
下線に使用する文字の変更, 4-3, 8-107  
切り捨てられる場合, 8-31  
整列, 8-34  
複数行への表示, 4-3, 8-33  
複数のワードに変更, 4-3, 8-33  
変更, 4-2, 8-33  
レポートへの出力の抑止, 8-101  
レポート, 1-2  
SILENT モード, 4-49  
Web 上, 4-38  
新しいページでの開始, 8-99  
埋め込まれた CGI の例, 4-43  
下部タイトルの作成, 4-23, 8-23, F-2  
間隔の設定およびサマリー行による明確化, 4-11  
上部タイトルの作成, 4-23, 8-134, F-2  
タイトル, 8-134, F-2  
対話型 HTML の例, 4-39, 4-42  
動的, B-2  
表示, 8-97  
フッターの作成, 8-84  
ブレイク, 8-18  
ヘッダーおよびフッターの作成, 4-23  
ヘッダーの作成, 8-86  
マスター / デティールの作成, 4-30, 8-34, 8-35  
列, 8-30

列の書式設定, 4-4, 4-6, 8-29  
列ヘッダーの書式設定, 4-2, 8-29

## ろ

---

ロール, E-7  
使用禁止, E-8  
再び使用可能にする, E-8  
ログイン  
Oracle, 6-2, 6-3, 8-44  
SQL\*Plus, 2-3  
インストール時のデフォルト・ユーザー作成, 1-5  
ログオフ  
Oracle, 6-2, 8-59  
SQL\*Plus, 2-3, 8-63  
条件付き, 8-145, 8-147

