

ORACLE

● ORACLE 数据库丛书

ORACLE 7

使用技巧

● 北方交通大学自动化系统研究所 编著

● 科学出版社

ORACLE 数据库丛书

ORACLE 7 使用技巧

北方交通大学
自动化系统研究所 编著

科学出版社

1995

(京) 新登字 092 号

内 容 简 介

本书集中介绍了 ORACLE 体系结构及工具的使用, 主要包括: 基本数据库查询语言 SQL * Plus 及其在 ORACLE 中的高级应用; 过程化 SQL 语言 PL/SQL 及其在各种工具中的使用技巧; 最新表格生成工具 SQL * Forms 3.0; 最新菜单生成工具 SQL * Menu 5.0; 与 C 语言的接口工具 Pro * C; DBA, 其中详细阐述 ORACLE 7 体系结构、进程管理、内存管理、用户权限管理及 ORACLE 7 最新的多线索机制等。

本书适合 ORACLE 各级用户及数据库系统开发、管理人员参阅。

ORACLE 数据库丛书 ORACLE 7 使用技巧

北方交通大学 编著
自动化系统研究所

责任编辑: 留霞 刘晓融

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

香河县第二印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1995 年 10 月第 一 版 开本: 787×1092 1/16

1995 年 10 月第 一 次印刷 印张: 44

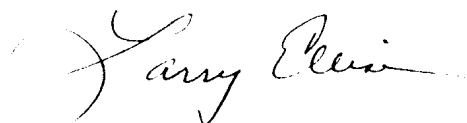
印数: 1-5 000 字数: 1026 000

ISBN 7-03-004782-6/TP·453

定价: 58.50 元

我很高兴地看到这样优秀的 Oracle 图书
在中国出版。

拉瑞·埃里森

A handwritten signature in cursive script, reading "Larry Ellison".

Oracle 公司总裁，首席执行官

1995 年 3 月 21 日

序

在信息管理高度发展的今天, Oracle 以其远见卓识和开拓进取而著称于世。自 1979 年世界上第一个以 SQL 语言为基础的关系型数据库问世, 到 1992 年全球第一个协同服务器 ORACLE 7 及 1995 年 ORACLE 2000 系列的发表, Oracle 公司始终坚持投入巨额资金于信息管理现代化技术及产品的研究与开发, 在数据库技术领域创造了若干个第一, 成为行业的带头人。

Oracle 中国公司坚守: 用户的成功即是我们的成功。几年来, Oracle 中国公司以其卓越的技术及全面的服务, 赢得了国内广大用户的信任。用户几乎遍及中国所有省市的每个行业, 国家大多数重点项目采用了 Oracle 的技术与产品。

Oracle 在不断开展各项业务联盟的同时, 还积极与国内近 20 所著名的院校开展联合教育计划。这项计划的实施已在普及先进的数据库技术和培养计算机应用人才方面, 取得了可喜的成绩, 并得到社会各界的充分肯定。

值北方交通大学自动化系统研究所 (ASI) -ORACLE 联合大学教师编写的《ORACLE 7 使用技巧》及《ORACLE 7 技术手册》出版之际, 谨为此两书作序。这两本书汲取了 ORACLE 数据库原版资料的精华, 对数据库理论和 ORACLE 内核分别进行了阐述与剖析, 同时配以丰富的实例。这两本书面向 ORACLE 各层次用户, 对了解 ORACLE 体系结构及内核, 对使用 ORACLE 各种工具进行工程开发及作为 DBA 进行数据库管理均是理想的、不可多得的工具书。希望这两本书能使更多的朋友认识 ORACLE, 了解 ORACLE, 使用 ORACLE, 使 ORACLE 在社会主义现代化建设中与您共创美好的未来!

冯星君

Oracle 中国公司总经理

前 言

关系型数据库管理系统 (RDBMS) 和 SQL 语言的产生, 给数据处理带来了一场革命。RDBMS 技术和产品为计算机应用设计人员和软件开发人员提供了高效率的软件生产工具, 极大地促进了信息产业的发展。

作为关系型数据库的先驱和基于标准 SQL 数据库语言的产品, ORACLE 数据库自 1979 年推出后, 受到社会广泛的注意。十多年来, ORACLE 数据库融汇了先进的技术, 并极有预见性地领导着全球数据库技术的发展。它从 1979 年的第 2 版开始, 经历了可移植的第 3 版, 可靠的第 4 版, 客户/服务器和协同服务器的第 5 版, 高可靠 OLTP 的第 6 版, 以至发展到今天在性能、系统可靠性、分布式处理能力等方面都有极大优化改善的 7.0 版——ORACLE 7。ORACLE 7 主要提供了以下新功能:

1. 体系结构发生了变化

ORACLE 7.0 版本引入了共享 SQL 和多线索服务器体系结构。这两个特性结合在一起, 减少了 ORACLE 的资源占用, 并增加了 ORACLE 的能力, 使之在低档平台上用较少的资源就可以支持更多的用户, 而在高档平台上可以支持成百甚至上千个用户。

2. 增加了新的分布式数据库能力

提供了对用户应用完全透明的二阶段提交机制, 以防备系统和网络故障; 包含数据操纵语句的事务可以在单个事务中执行于多处地点; 支持透明异步复制 (快照)。

3. 提供基于角色 (Role) 分工的安全保密管理

通过组访问控制 (Group Access Controls) 以及授予角色一些新的“系统特权”等手段, 使得数据库安全保密更易于管理。

4. 其他

ORACLE 7 在数据库管理功能、数据完整性检查、数据库查询性能、数据安全性等方面有明显的改善。

针对不同的应用界面, ORACLE 分别提供了不同的开发工具, 以辅助程序的生成。强有力的开发工具, 已成为 ORACLE 技术的另一标志。用 ORACLE 工具开发出来的应用程序, 具有良好的移植性, 可方便地在不同硬件平台、网络环境、图形界面中移植。ORACLE 数据库具有完整的多媒体支持能力, 为声音、图形、图像、动画等类型数据提供了灵活多样的表现方式。

北方交通大学自动化系统研究所是北方交通大学下属的一所高技术研究机构, 先后使用 ORACLE 5.1 B, ORACLE 6.0 和 ORACLE 7 在铁路、石油、外贸、经贸等行业开发了大量的 MIS 系统, 积累了丰富的应用开发经验。1994 年成立的北方交通大学-ORACLE 联合大学, 已培养了数百名 ORACLE 高级专业人才, 并于今年初成立了国内第一家 ORACLE 硕士培训中心。同时, 作为 ORACLE 公司的技术支持中心, 北方交通大学自动化系统研究所承担着 ORACLE 公司的许多技术支持工作。针对目前国内 ORACLE 7 中文技术资料缺乏, 特别是在应用技术方面的专著几乎为空白的现状, 我所在 ORACLE 公司的支持下, 组织了所内有经验的教师和技术开发人员, 参考 ORACLE 7 的最新资料及发展动向, 并结合多年的实际应用开

发经验，编写成《ORACLE 7 使用技巧》和《ORACLE 7 技术手册》两书。

《ORACLE 7 使用技巧》主要介绍 ORACLE 7 的应用工具、DBA 及选件。全书分十章，详细介绍了基本数据库查询语言 (SQL * Plus)、过程化 SQL 语言 (PL/SQL)、最新表格生成工具 (SQL * Forms 3.0)、最新菜单生成工具 (SQL * Menu5.0)、最新报表生成工具 (SQL * Report Writer V1.1) 及 C 语言接口工具 (PRO * C)。所有的讨论都是从基本概念、设计方法入手，重点介绍了使用技巧和高级应用，并配有大量示例。此外，书中还采用大量篇幅在数据库系统管理 (DBA) 部分，详细阐述了 ORACLE 7 体系结构、进程管理、内存权限管理、备份与恢复、审计功能及 ORACLE 7 最新的多线索机制、共享 SQL 及性能调整技术。在分布式选件部分，介绍了网络环境下 ORACLE 的应用，即如何构造 Client-Server 结构及分布式数据库环境。在实用工具部分，介绍了 Export, Import, SQL * Loader 工具使用方法及最新 ORACLE * TERMINAL 工具的功能及使用技巧。在过程化选件部分，介绍了最新 ORACLE 7 的 Procedure, Package, Database Trigger 功能及应用方法。

《ORACLE 7 技术手册》主要内容为语言参考、开发工具使用参考和错误信息列表，分别详细描述了 SQL 命令、SQL 语言数据类型和函数、PL/SQL 语法、SQL * Forms 封装过程和封装函数、SQL * Forms 系统变量、SQL * Menu 封装过程、宏等内容。对所有的命令或函数的基本用途、使用条件、语法图、关键字和参数以及相关条目等给予了详细描述；并对大多数命令或函数，配备了相应的使用示例。附录 A 至附录 C 是按错误号排列的 ORACLE 主要的错误信息，包括 SQL * Forms, DBA 和 LCC 错误信息，以及 ORA 错误信息。使用时可根据所得到的错误代码查找其原因和相应的解决措施。附录 D 描述了所有数据字典视图的内容及其结构，可以在使用数据字典时作为参考手册查询。附录 E 描述了所有 ORACLE 在启动例程时所需的参数，可用于调整例程配置时作为参考手册查询。

上述两书由樊兰非、吴琛主编，由樊兰非、陈京审校。参加编写的人员及分工如下：

《ORACLE 7 使用技巧》：

- | | |
|-----|-------|
| 第一章 | 霍宇红 |
| 第二章 | 陈京 |
| 第三章 | 樊兰非 |
| 第四章 | 许向东 |
| 第五章 | 孙玮 |
| 第六章 | 李红辉 |
| 第七章 | 吴琛 |
| 第八章 | 徐悦、陈京 |
| 第九章 | 时新德 |
| 第十章 | 李淑忠 |

《ORACLE 7 技术手册》：

- | | |
|--------|---------|
| 第一章 | 张晓鹏 |
| 第二章 | 张晓鹏、孙宏声 |
| 第三章 | 孙宏声 |
| 第四、五章 | 樊兰非 |
| 第六、七章 | 许向东 |
| 附录 A、B | 郭巍 |
| 附录 C | 冯凤娟 |
| 附录 D | 买学军 |
| 附录 E | 张戈 |

参加这两本书审校的人员还有吴伟、刘晓锐等。张秋、娄群为两书的编辑做了大量工作。在两书的编写、出版过程中，得到了 ORACLE 公司的大力帮助，以及科学出版社编辑们的紧密配合，在此谨致深切谢意。

由于 ORACLE 7 是一个很新的技术，奢望用较少篇幅来全面介绍和刻画有关特征、技术内容，确实是一件非常困难的事。尽管如此，我所十几名年轻的作者仍以顽强的精神，从大量的原版参考文献中精选了有关内容，并在此基础上融合了自己在工程设计实践中的经验、体会和应用技巧，经过近一年的努力完成了书稿的编写。令人欣慰的是，ORACLE 公司总裁拉

瑞·埃里森先生今年3月首次访华期间，专程来我所参观 ORACLE 联合大学，与本书作者们进行了亲切座谈，并欣然为本书命笔题词。这无疑对我们是极大的鼓舞。

在本书出版发行之际，我们祈愿读者能借助本书，快速、高效、深入地参与 ORACLE 数据库的管理、使用、维护和开发研究工作。

本书内容繁多，涉及面广，书中难免有不妥和谬误之处，恳请读者批评指正。

刘峰

北方交通大学

自动化系统研究所副所长

1995年9月18日

目 录

序

前言

第一章 SQL * PLUS	(1)
1.1 简介	(1)
1.2 数据操纵语言	(7)
1.3 数据定义语言	(22)
1.4 SQL * PLUS 的环境命令	(41)
1.5 SQL * PLUS 中使用的函数	(57)
1.6 高级查询	(68)
1.7 实体的授权和权限管理	(86)
1.8 技巧及高级应用	(94)
第二章 PL/SQL	(104)
2.1 PL/SQL 简介	(104)
2.2 设计一个简单的 PL/SQL 块	(108)
2.3 数据操作	(117)
2.4 流程控制	(124)
2.5 游标的控制	(129)
2.6 例外处理	(140)
2.7 PL/SQL 子程序	(147)
2.8 包	(153)
第三章 SQL * Forms (V3.0)	(160)
3.1 SQL * Forms (V3.0) 简介	(160)
3.2 运行 SQL * Forms 应用程序	(161)
3.3 设计 SQL * Forms 应用程序	(173)
3.4 form 的定义	(193)
3.5 块的定义	(195)
3.6 域的定义	(208)
3.7 触发子的定义	(220)
3.8 页的定义	(236)
3.9 应用程序设计举例	(243)
3.10 应用程序调试和文档	(252)
第四章 SQL * Menu (V5.0)	(255)
4.1 SQL * Menu (V5.0) 简介	(255)
4.2 设计 SQL * Menu 基本步骤	(270)
第五章 Pro * C	(288)
5.1 ORACLE 预编译器	(288)
5.2 Pro * C 的基本结构	(290)
5.3 运行 Pro * C 预编译器	(315)

5.4	应用程序举例	(354)
第六章	SQL * ReportWriter (V1.1)	(376)
6.1	SQL * ReportWriter 简介	(376)
6.2	SQL * ReportWriter 的使用	(377)
6.3	创建报表	(382)
6.4	技巧及高级应用	(393)
附录	本章中使用的报表和视图	(400)
第七章	ORACLE 7 数据库管理	(403)
7.1	概述	(403)
7.2	建立数据库	(405)
7.3	数据库的启动和关闭	(410)
7.4	ORACLE 体系结构及进程管理	(415)
7.5	ORACLE 存储管理	(436)
7.6	ORACLE 用户及权限管理	(460)
7.7	数据库的备份与恢复	(475)
7.8	ORACLE 审计	(492)
7.9	ORACLE 7 数据库性能调整	(499)
第八章	实用程序	(507)
8.1	卸出	(507)
8.2	装入	(517)
8.3	SQL * Loader	(530)
8.4	ORACLE * TERMINAL	(564)
第九章	分布式处理与分布式数据库	(575)
9.1	概述	(575)
9.2	SQL * Net 产品	(581)
9.3	配置 SQL * Net	(588)
9.4	SQL * Net 配置工具	(597)
9.5	配置文件应用	(607)
9.6	使用 SQL * Net	(612)
9.7	网络配置实例	(621)
9.8	分布式系统中表复制策略	(643)
9.9	分布式系统中的安全性	(648)
第十章	过程化选件	(656)
10.1	概述	(656)
10.2	存储过程	(657)
10.3	存储函数	(668)
10.4	包	(669)
10.5	数据库触发子	(675)

第一章 SQL * PLUS

本章介绍关系数据库的 SQL 语言及其在 ORACLE 中的应用技巧。

1.1 简介

SQL * PLUS 是 ORACLE 的交互查询工具，它允许用户使用 SQL 命令交互式地访问数据库，也允许用户使用 SQL * PLUS 命令与系统发生联系。

1.1.1 结构查询语言 SQL

SQL 语言是一种使用方便灵活的语言，因为它是一种非过程化程度相当高的语言，使用这种语言编程，用户只需在程序中指出要干什么，至于如何干，用户不必在程序中指出，而由系统来决定完成。

SQL 语言可以通过两种方式使用，一种是命令方式，另一种是程序方式。采用命令方式时，用户通过交互来执行该命令。SQL 语言还可以嵌入 C，COBOL 和 FORTRAN 等高级语言中，组成一个完整的程序。用户可以根据不同需要选择不同的使用方式，以满足不同的要求。

SQL 语言还提供了事务控制，以保证数据共享以及并发使用时不产生干扰，也便于对数据库的恢复。此外，SQL 语言还提供了授权控制，以保证数据的安全与保密，防止非法用户对数据库的使用与破坏。

正因为 SQL 语言的这些特点，使它受到人们的广泛重视与欢迎。近几年来，SQL 语言已成为标准的数据库语言，现在已有一百多个数据库管理产品支持 SQL 语言。

1.1.2 启动 SQL * PLUS

启动 SQL * PLUS 的步骤如下：

(1) 将 ORACLE 软件安装到机器上。

(2) 启动机器进行操作系统登录，成功后用户将在屏幕上看到操作系统的提示符，说明进入操作系统已经成功。

(3) 启动 ORACLE 数据库系统，将 ORACLE RDBMS 装入内存中，命令执行后，就可以运行工具软件了。

(4) 在操作系统提示符下执行 SQL * PLUS：

```
$ SQLPLUS [RETURN]
```

SQL * PLUS 将显示它的版本号、日期和版权信息，并提示你输入用户名。因为 ORACLE 保护对它所有数据的访问，所以与它连接通常需要一个用户标识 (user-name) 和口令 (password)。

Enter user-name:

输入用户名后按 [RETURN]，SQL * PLUS 提示你输入口令：

Enter password:

如果口令输入正确, SQL * PLUS 将宣布你已经与 ORACLE 连接上了, 然后显示:

SQL>

说明现在你已经进入 SQL * PLUS 中, 并等待你的指示。

如果输入的用户名或口令不正确, 在向 ORACLE 三次注册失败后, 屏幕上将出现终止服务信息, 并退出 SQL * PLUS。

在进入 SQL * PLUS 时, 也可以在命令行上直接输入用户名和口令, 中间以 “/” 分隔, 比如, 你的用户名是 SCOTT, 口令是 TIGER, 那么你输入下列命令:

\$ SQLPLUS SCOTT/TIGER

就可以进入 SQL * PLUS 中。

当你想停止工作并离开 SQL * PLUS, 则可以输入 EXIT 或 QUIT 命令。

SQL> EXIT

1.1.3 预备知识

1. 基表

ORACLE 是一种关系型数据库系统, 它以关系的方式组织数据, 这里所说的关系, 我们称之为基表。一个数据库可以由若干个表组成, 用户对数据库的操作也就是对表的操作。

ORACLE 用基表的形式存储信息, 如表 1.1 所示。

表 1.1 EMP 表

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SLESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30

每个基表都有一个表名, 以便识别。每个基表都包含一个或一个以上的列, 每列包含一类信息, 并都有一个列名, 以描述保存在那一列中的信息的种类。表名和列名构成了基表的框架, 也就是基表的分类信息。具体信息一行一行存放, 表的每一行表示一组独立的数据, 它由具有不同列值的各列组成。

2. 基表之间的关系

在数据库中, 数据与数据之间是存在着联系的, 我们可以将某个基表中的信息与其他基表中的信息建立关系, 例如, 在前面的 EMP 表中, 每个雇员均有部门编号 (DEPTNO), 它对应于 DEPT 表中的部门编号 (DEPTNO), 见表 1.2。

表 1.2 DEPT 表

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

这样可便于信息的独立组织，因为我们可以 EMP 表中存放雇员信息，而无需在其中存放 DEPT 表中有关部门的信息。当然，我们还可以在多个基表之间建立联系。

3. 数据字典

数据字典是 ORACLE 关系数据库管理系统 (RDBMS) 最重要的成分之一，是数据库系统的宝贵资源。数据库管理员 (DBA) 使用它来监视 ORACLE RDBMS 的使用，并协助用户完成他们的工作。数据字典是由 ORACLE 自动建立并更新的一组基表和视图，在这里记录了用户的姓名、基表和视图的定义、用户权限以及数据存储等方面的信息。

数据字典中包括下列信息的描述：

- (1) ORACLE 所有用户的标识及用户的存取权限。
- (2) 每个用户所拥有的数据库实体 (表、空间定义、视图、索引、聚簇和同义词) 的定义。
- (3) 为数据库实体分配的空间情况。

我们可以通过查询语句查看数据字典中的有关内容。例如：用户需要查看当前用户创建的所有基表、视图和同义词清单，可以使用下列命令：

```
SQL> SELECT * FROM CAT;
```

CAT 表中存放的就是有关用户创建的基表、视图和同义词情况的描述。

数据字典由几个基础表及派生出来的视图组成。基础表是由 ORACLE RDBMS 自动更新，用户基本上不访问基础表，ORACLE RDBMS 不断地更新数据字典的基础表，而视图总是引用基础表中的最新数据。

一般来说，用户不能直接更新 ORACLE 的数据字典，因为用户操作不当、删除或修改了某些数据字典的内容，就会造成数据库的破坏。

1.1.4 SQL 和 SQL * PLUS 命令

SQL * PLUS 可接收两种类型的命令：SQL 命令和 SQL * PLUS 命令。SQL 命令主要用来对数据库进行操作，而 SQL * PLUS 命令主要用来设置查询结果的显示格式，设置一些环境选项。

1. SQL 命令

SQL 命令主要用于 ORACLE 数据库中创建、存储、改变、检索和维护信息。当输入 SQL 命令时，SQL * PLUS 将其保存在内部缓冲区中，这个内部缓冲区称为 SQL 缓冲区。SQL 缓冲区只保存一条 SQL 命令，当输入新的 SQL 命令时，原来保存的 SQL 命令就被删除。用户可以利用 SQL 缓冲区在不重新输入命令的情况下，编辑、运行上一次执行的 SQL 命令。

SQL 命令可以分一行或多行输入，如果在一行输入，则是

```
SQL> SELECT EMPNO, ENAME, JOB, SAL FROM EMP WHERE SAL<2500;
```

在 SQL 命令输入过程中，如果按 RETURN 键，则表示命令分多行输入，在下一行的开始处将自动出现一个行号。例如：

```
SQL>SELECT EMPNO, ENAME, JOB, SAL
2   FROMEMP
3   WHERE SAL<2500;
```

SQL 命令完成，可以用分号 (;)、斜线 (/) 或空行来结束命令的输入。输入分号并按 RE-

TURN 键，则结束 SQL 命令的输入，并运行 SQL 缓冲区中的命令。在单独的一行上键入斜线 (/) 并按 RETURN 键，SQL * PLUS 将运行 SQL 命令。如果输入一个空行，即在命令的最后一行后按 RETURN 键，SQL * PLUS 显示一个行号，此时再按 RETURN 键，SQL * PLUS 结束 SQL 命令的输入，回到 SQL * PLUS 的提示符状态。

2. SQL * PLUS 命令

SQL * PLUS 命令用于设置查询结果的输出格式，形成复杂的报表，编辑 SQL 命令，设置系统变量并可提供帮助信息。

SQL * PLUS 的很多命令可以简写成一个或几个字母，SQL * PLUS 命令的输入与 SQL 命令不同，例如，我们想改变 EMP 表的 SAL 列的显示格式，可以键入

```
SQL> COLUMN SAL FORMAT $99,999 HEADING SALARY
```

COLUMN 命令为 SAL 列的数据显示设置了“\$”和逗号“,”，并给出了新的列标题，

通过运行前面的 SQL 命令可以清楚地看到 SQL * PLUS 命令的结果。

在输入 SQL * PLUS 命令过程中，按 RETURN 键表示输入完成，而不必输入分号 (;)。

如果命令输入占满一行，光标会自动移到下一行，可以继续输入，直到按 RETURN 键结束输入。

1.1.5 获取帮助

1. 随机帮助

在实际使用 SQL * PLUS 时，有时需要能有一种快捷的方法，获得有关命令的语法、含义的说明。SQL * PLUS 提供了 HELP 命令，为使用者提供随机帮助信息。如果要获取所有 SQL 和 SQL * PLUS 命令的信息，只需输入

```
SQL> HELP
```

如果希望获取某一个命令的信息，可以在 HELP 之后输入该命令的名字，例如：

```
SQL> HELP ACCEPT
```

2. DESCRIBE 命令

当我们需要对基表进行操作的时候，往往需要获取关于基表的结构信息，可以使用 DESCRIBE 命令查看基表的结构：

```
SQL> DESCRIBE DEPT
```

命令执行结果如下：

NAME	NULL?	TYPE
DEPTNO	NOT NULL	NUMBER (2)
DNAME		CHAR (14)
LOC		CHAR (13)

基表的结构信息包括：

NAME 列名

NULL 标识该列是否允许为空值，如缺省则表示允许为空值

TYPE 说明该列的数据类型，可以是字符型：CHAR (w)，VARCHAR2 (w)；数

字型: NUMBER (w, d); 日期型: DATE; 原始数据类型: RAW 以及 LONG 型等等

其中 w 表示列的宽度, d 表示小数点之后的位数

还可以使用 DESCRIBE 命令获取有关函数、过程以及包的描述, 例如: 要求显示有关函数 AFUNC 的信息:

```
SQL> DESCRIBE afunc
```

结果是

```
function afunc (F1 CHAR, F2 NUMBER) return NUMBER;
```

1.1.6 编辑 SQL * PLUS 命令的使用

1. 命令行编辑命令

我们可以使用 SQL * PLUS 的一些行编辑命令编辑 SQL 缓冲区中的 SQL 命令, 表 1.3 列出了编辑功能的 SQL * PLUS 命令。

表 1.3

命 令	缩 写	意 义
APPEND text	A text	text 行尾增加
CHANGE/old/new	C /old/new	在当前行中将 old 改为 new
CHANGE/text	C /text	从当前行中删除 text
CLEAR BUFFER	CL BUFF	删除 SQL 缓冲区的所有行
DEL	(无)	删除当前行
INPUT	I	增加一行或多行
INPUT text	I text	增加一个由 text 组成的行
LIST	L	显示 SQL 缓冲区的所有行
LIST n	L n 或 n	显示一行
LIST *	L *	显示当前行
LIST LAST	LAST	显示最后一行
LIST m n	L m n	显示多行 (从 m 到 n)

(1) 显示 SQL 缓冲区的命令

使用 LIST 命令可以显示当前 SQL 缓冲区的内容。如果 LIST 命令后面不带任何参数, 将显示 SQL 缓冲区中的所有行; 如果 LIST 后面带一个参数 (如 LIST n), 则显示第 n 行的内容; 如果 LIST 后面带有两个参数 (LIST m n), 那将显示多行, 即从第 m 行到 n 行。例如:

```
SQL>LIST
```

```
1  SELECT EMPNO, ENAME, JOB, SAL
```

```
2 * FROM EMP WHERE SQL<2500;
```

行号 2 后面的星号表示第 2 行为当前行。除 LIST 命令外, 其他编辑命令都只对 SQL 缓冲区的当前行操作。在 SELECT 命令尾部输入的分号 (;) 没有显示, 因为 SQL * PLUS 并不把它保存在 SQL 缓冲区中, 所以在缓冲区末尾增加新行时, 不必从最后一行去掉分号, 使编辑更为方便。

(2) 修改 SQL 命令

使用 CHANGE 命令可以从当前行中删除一部分内容, 或者用新的内容替换原先的内容。

在使用 CHANGE 命令之前必须先用带一个参数的 LIST 命令选择当前行。例如，下列命令将 DEPTNO 错输成 DPTNO：

```
SQL> SELECT DPTNO, ENAME, SAL
      2 FROM EMP
      3 WHERE DEPTNO=10;
```

系统给出出错信息：

```
SELECT DPTNO, ENAME, SAL
*
ERROR at Line 1:
ORA-0904: invalid column name
```

使用 CHANGE 命令进行修改：

```
SQL> CHANGE/DPTNO/DEPTNO
```

改正后屏幕上显示：

```
1 * SELECT DEPTNO, ENAME, SAL
```

这时可用 RUN 命令重新运行修改后的 SQL 命令。

(3) 增加新行

使用 INPUT 命令可以在 SQL 缓冲区增加一个或多个新行。如果只输入 INPUT，则在当前行之后插入多个新行；如果输入 INPUT text，则在当前行之后插入由 text 组成的新行。

例如，输入 INPUT 命令：

```
SQL> INPUT
      4
```

这时在 INPUT 命令之后出现一个行号，这个行号表明所要插入的行，此时可输入一个或多个新行。输入一行内容后按 RETURN 键，SQL * PLUS 给出一个新行号，如屏幕所示：

```
      4 ORDER BY SAL
      5
```

再按 RETURN 键，表示插入结束，可用 RUN 命令重新运行来检验最终的查询结果。

(4) 在一行后追加内容

使用 APPEND 命令可以在 SQL 缓冲区中的当前行末尾追加内容。首先用 LIST 命令将需要追加内容的行置为当前行，然后使用 APPEND text 命令在当前行末尾追加内容。例如，要在第四行末尾追加 DESC 选项，先将第四行置为当前行：

```
SQL> LIST 4
      4 * ORDER BY SAL
```

然后输入下列命令追加 DESC：

```
SQL> APPEND DESC
      4 * ORDER BY SAL DESC
```

最后用 RUN 命令重新运行。

(5) 删除 SQL 缓冲区中的行

使用 DEL 命令可以删除 SQL 缓冲区中的当前行，当前行以下各行的行号顺序减 1，如果把第一行设为当前行，连续多次输入 DEL 后，可以删除 SQL 缓冲区中的所有行。删除 SQL 缓冲区中所有行也可以使用 CLEAR BUFFER 命令，它的作用是将 SQL 缓冲区清空。

2. EDIT 命令

EDIT 命令用于调用一个操作系统提供的文本编辑器。编辑 SQL 缓冲区里的内容或操作

系统下的文本文件，并将要编辑的文件名传送给该编辑器。如果文件扩展名（例如 .ext）被省略，则系统自动追加 .SQL 的扩展名，如果文件名和扩展名均省略，则编辑器被调用，并将 SQL * PLUS 产生的文件名传给它，这个文件中包含着当前缓冲区的内容。如果当前缓冲区为空，且 EDIT 被调用时没有给出文件名，则 EDIT 命令失效。

```
SQL> EDIT
```

也可以是

```
SQL> EDIT 文件名
```

可以使用 DEFINE 命令设置系统变量 _EDITOR 来改变文本编辑器的类型，一般在 LOGIN.SQL 文件中完成它的设置，当 SQL * PLUS 被调用时将读出这些信息。例如，在 UNIX 平台上，定义 Vi 作为 SQL * PLUS 调用的外部编辑器：

```
SQL> DEFINE _EDITOR=Vi
```

3. SAVE 命令

如果要把 SQL 命令存储到磁盘上，或者想使用自己的编辑程序，用户可以通过 SAVE 命令将内容写到一个文件上。

```
SQL> SAVE DEPTINFO
```

这时 SQL * PLUS 响应：

```
Create file DEPTINFO
```

文件名后缀缺省为 .SQL。这时用户需要保存的内容已经存入了 DEPTINFO.SQL 文件中，可以对文件进行编辑、运行。

如果想将修改过的内容保存到原有的命令文件中，则在 SAVE 命令后增加 REPLACE 选项：

```
SQL> SAVE DEPTINFO REPLACE
```

如果想将 SQL 缓冲区中的内容追加到已经存在的命令文件末尾，则在 SAVE 命令后增加 APPEND 选项：

```
SQL> SAVE file-name APPEND
```

4. GET 命令

GET 命令把磁盘上的命令文件内容调入当前缓冲区，并显示在屏幕上，格式是

```
SQL> GET file-name
```

文件后缀缺省为 .SQL，文件被调入缓冲区后，可以对它进行编辑、运行。

5. 运行命令文件

START 命令将指定文件调入缓冲区并执行文件内容：

```
SQL> START file-name
```

文件后缀缺省为 .SQL。也可以使用 “@” 命令运行文件：

```
SQL> @file-name
```

1.2 数据操纵语言

数据操纵语言 (DML: Data Manipulation Language) 给用户或应用程序提供访问数据库