

关系数据库系统

ORACLE

沈金发 郑南京 王令赤 编著

清华大学出版社

内 容 简 介

ORACLE 系统是一个使用方便、功能强、便于维护、便于推广的关系数据库系统。本书系统全面地介绍了 ORACLE 及其应用。本书既可以作为学习 ORACLE 系统的教材,也可作为 ORACLE 使用手册。本书的内容覆盖了目前最为流行的 ORACLE 5.1 和 ORACLE 6.0 版本。

(京)新登字 158 号

关 系 数 据 库 系 统 ORACLE

沈金发 郑甫京 王令赤 编著

☆

清华大学出版社出版

北京 清华园

清华大学印刷厂印刷

新华书店总店科技发行所发行

☆

开本:787×1092 1/16 印张:36 字数:923 千字

1992 年 9 月第 1 版 1992 年 6 月第 1 次印刷

印数:0001—8000

ISBN 7-302-01053 6/TP · 389

定价:20.00 元

前 言

ORACLE 数据库系统自 1979 年推出后,就受到了广泛注意。经过十多年的发展,ORACLE 系统已有大量的用户。近几年来,ORACLE 系统在国内也受到极大的重视,越来越多的用户了解、学习和使用 ORACLE 系统。

ORACLE 系统使用方便、功能强、便于维护、便于推广,尤其是它使用 SQL 语言作为用户的主要接口,给数据库应用系统的开发、维护和移植带来了很大的方便。ORACLE 不但成功地运行在大型机上,而且在小型机和微型机上得到广泛的应用。

本书系统地全面地介绍了 ORACLE 系统及其应用。本书既可以作为学习 ORACLE 的教材,也可以作为 ORACLE 的使用手册。

每一个 ORACLE 数据库系统都由两大部分组成。一部分是 ORACLE RDBMS,另一部分是各种 ORACLE 实用程序。而 ORACLE RDBMS 和各种实用程序的升级都是相互独立的。版本 6 ORACLE 系统与版本 5 ORACLE 系统的主要区别在于 ORACLE RDBMS 的版本不同,而其它 ORACLE 实用程序基本上没有升级,也就是说,两者的各种实用程序是基本相同的。对于一个了解版本 5 ORACLE 系统的用户来说,如果想学会使用版本 6 ORACLE 系统,只需了解版本 6 的 RDBMS 就可以了。

本书的内容覆盖了目前最为流行的 ORACLE 5.1 版本和最新的 ORACLE 6.0 版本。

本书附录给出了 ORACLE 的许多重要信息。考虑到 5.1 版本的 ORACLE 出错信息与 6.0 版本相同很多,为节省篇幅,附录中仅给出了 6.0 版本的 ORACLE 出错信息。

本书一至八章、十一至十三章由沈金发编写,九、十章及附录由郑南京编写,十四至二十一章由王令赤和郑南京编写。全书由沈金发统一修改。

由于时间仓促,水平有限,书中难免有缺点与错误,敬请广大读者指正。

作 者

1991 年 10 月

目 录

前言	I
----------	---

第一章 概述	1
--------------	---

1.1 关系数据库方法	1
1.2 ORACLE 简介	3
1.3 结构查询语言 SQL	4
1.4 SQL 命令缓冲区及其使用	6
1.5 ORACLE 系统的安装、启动和退出	9
1.6 SQL * PLUS 使用的键	12
1.7 帮助和错误处理	13

第二章 基本查询	14
----------------	----

2.1 简单查询	14
2.2 组函数	20
2.3 谓词 BETWEEN、IN 和 LIKE	20
2.4 排序显示	24
2.5 分组查询	26
2.6 并、交、差操作	28

第三章 表的基本操作	30
------------------	----

3.1 表的建立	30
3.2 数据的插入	32
3.3 数据的修改	34
3.4 数据的删除	35
3.5 更新的提交与撤消	36

第四章 连接查询	38
----------------	----

4.1 连接的基本方法	38
4.2 二元连接查询	39
4.3 多元连接查询	40
4.4 自身连接	42
4.5 外部连接	43

第五章 嵌套与树结构查询	45
--------------------	----

5.1 单层嵌套查询	45
5.2 多层嵌套查询	47
5.3 关键字 ANY 和 ALL, 谓词 EXISTS	49
5.4 相关嵌套查询	52
5.5 树结构查询	53

第六章 数据显示及使用	61
-------------------	----

6.1 数值显示及使用	61
6.2 字符值显示及使用	64
6.3 日期值显示及使用	67
6.4 空值显示及使用	72

第七章 视图的建立与使用	75
--------------------	----

7.1 视图的概念	75
-----------------	----

7.2 视图的基本操作	75
-------------------	----

7.3 多表视图	78
----------------	----

第八章 数据定义功能与授权控制	81
-----------------------	----

8.1 别名与重新命名	81
-------------------	----

8.2 表结构的修改	81
------------------	----

8.3 表与视图的复制和删除	82
----------------------	----

8.4 索引	83
--------------	----

8.5 使用权的授予	84
------------------	----

8.6 使用权的收回	85
------------------	----

第九章 表格设计 SQL * FORMS	87
----------------------------	----

9.1 SQL * FORMS 概述	87
--------------------------	----

9.2 SQL * FORMS 基本概念	89
----------------------------	----

9.3 定义一个简单的 FORM	93
------------------------	----

9.4 运行 FORM	96
-------------------	----

9.5 设计 FORM	107
-------------------	-----

9.6 建立和定义块	117
------------------	-----

9.7 建立和定义域	133
------------------	-----

9.8 屏幕画面的设计	146
-------------------	-----

9.9 触发器	151
---------------	-----

9.10 SQL * FORMS 的组成	183
----------------------------	-----

第十章 电子表格 SQL * CALC	188
---------------------------	-----

10.1 SQL * CALC 概述	188
--------------------------	-----

10.2 SQL * CALC 启动和屏幕画面	193
-------------------------------	-----

10.3 基本提示下的操作	195
---------------------	-----

10.4 制表命令	201
-----------------	-----

10.5 使用 ORACLE 表	217
------------------------	-----

第十一章 图形软件 SQL * Graph	233
-----------------------------	-----

11.1 简介	233
---------------	-----

11.2 绘制折线图	235
------------------	-----

11.3 绘制直方图	240
------------------	-----

11.4 绘制饼形图	243
------------------	-----

11.5 图形的完善	245
------------------	-----

第十二章 报表软件 SQL * Report	250
------------------------------	-----

12.1 概述	250
---------------	-----

12.2 报表正文格式化程序 RPF	253
--------------------------	-----

12.3 报表生成程序 RPT	263
-----------------------	-----

12.4 打印报表	277
-----------------	-----

第十三章 预编译接口	284
------------------	-----

13.1 PRO * C 程序的结构	284
--------------------------	-----

13.2 数据更新操作	289	第十九章 SQL 语句的优化	454
13.3 查询	291	19.1 查询优化	454
13.4 事务管理	294	19.2 优化 NOT 子句	455
13.5 出错检测	296	19.3 优化 OR 子句	455
13.6 动态语句	301	19.4 优化 ORDER BY 和 GROUP BY 子句	456
13.7 预编译	306	19.5 优化连接	457
第十四章 ORACLE 调用界面(OCI)	310	19.6 查询路径的等级划分	458
14.1 OCI 程序的基本结构	310	第二十章 数据库安全管理	459
14.2 各种 ORACLE 调用描述	319	20.1 三种特权:CONNECT、 RESOURCE 和 DBA	459
14.3 数据类型和数据转换	336	20.2 怎样进入 ORACLE 数据库	461
14.4 怎样编写 OCI 程序	339	20.3 ORACLE 用户除名	462
第十五章 怎样使用 SQL * NET	363	20.4 ORACLE 用户权限的授予与收回	463
15.1 引言	363	20.5 两个特殊的 ORACLE 术语	464
15.2 SQL * NET 概要介绍	364	20.6 授予和收回用户对数据库客体 进行存取的权利	465
15.3 SQL * NET 结构	365	20.7 审计程序(AUDIT)能做什么	466
15.4 使用 SQL * NET 进行数据库操作	368	20.8 表和视图的审计	467
15.5 SQL * NET 所支持的三种通信 规程介绍	371	20.9 系统的审计	469
第十六章 DBA 工具	377	20.10 审计信息在数据库中的描述	470
16.1 数据库管理员(DBA)	377	第二十一章 版本 6 ORACLE 关系 数据库系统介绍	474
16.2 启动和停止数据库(IOR 实用程序)	378	21.1 新工具:SQL * DBA 实用程序	474
16.3 ORACLE 显示系统实用程序(ODS)	392	21.2 数据字典	475
16.4 后映象日志(AIJ)	399	21.3 数据库文件结构	475
16.5 修改显示终端(CRT 实用程序)	404	21.4 数据库结构	476
16.6 ORACLE 数据加载器(ODL)	406	21.5 事务控制	477
16.7 卸出程序和装入程序 (EXPORT/IMPORT)	415	21.6 SQL 语言的变化	477
第十七章 ORACLE 数据库结构	422	21.7 其它变化	478
17.1 数据怎样存放在数据库中	422	21.8 版本 5 与版本 6 的对比	479
17.2 空间定义	424	附录 A SQL 命令清单	480
17.3 ORACLE 数据字典	428	附录 B SQL * PLUS 命令清单	486
17.4 ORACLE 数据类型	430	附录 C ORACLE 公司提供的终端定义	497
17.5 临时表	434	附录 D 各种 OCI 程序的 ORACLE 调用格式	498
17.6 簇	435	附录 E SQL * FORM 错误信息	500
17.7 索引	439	附录 F SQL * CALC 错误信息	505
第十八章 数据库的一致性和并行性	442	附录 G ORACLE 错误信息和编码	510
18.1 事务处理	442		
18.2 前映象文件	445		
18.3 ORACLE 锁	446		
18.4 死锁的检测及处理	452		
18.5 使用 ORACLE 锁	453		

第一章 概 述

1.1 关系数据库方法

关系数据库的雏形可见于 60 年代。自从 1970 年开始, E. F. Codd 先后定义了关系数据库的基本概念, 引进了规范化理论, 提出了数据子语言, 为关系数据库的全面开发奠定了坚实的基础。在此之后, 关系数据库得到飞速的发展, 无论在理论上, 还是在实践上都取得了巨大的进展。关系数据库结构简洁明了, 便于用户理解; 用户对数据库的操作采用非过程化的语言, 使用方便; 用户接口不涉及任何存储细节及其存取路径, 数据独立性高; 因此关系数据库极受人们的欢迎, 目前关系数据库已被广泛地应用。

一、关系与表

在关系数据库中, 数据库是以关系的方式组织数据。关系也称之为表。一个数据库可由若干个表组成, 用户对数据库的操作也就是对表的操作。每个表都有一个表名, 以便识别。表的每一列都有一个列名, 或称属性名。表由表的框架和表的内容组成(见图 1.1)。表的框架就是表的结构, 它由表名和列名组成。表的内容就是表内实际存放的数据。

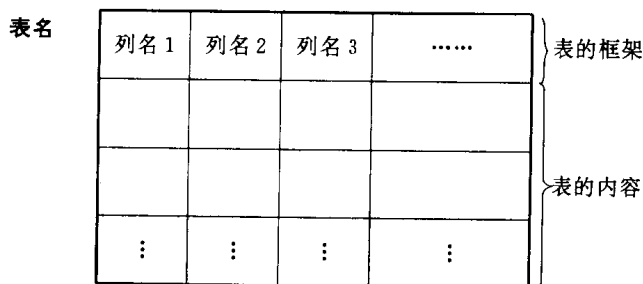


图 1.1

一个表可以存放数据。例如图 1.2 的学生和课程表, 它存放着学生和课程的数据, 表的每一行实际存放着一个学生或者一门课程的记录。表的一行也称为表的一个元组。

学生	学号	姓名	年龄	性别
	871024	王刚	21	男
	871721	张海	20	男
	880015	李华	20	女
	⋮	⋮	⋮	⋮

课程	课程号	课程名	学 时
	C-001	PASCAL	80
	C-005	编译原理	100
	C-007	操作系统	80

图 1.2

在数据库中,数据与数据是存在着联系的,这种联系也是数据库要处理的重要对象。在网状或层次数据库中,这种联系是从它们的结构中直接反映出来的。如图 1.3 所示,学生与课程之间存在着联系,这种联系表示哪些学生选修了哪些课程,哪些课程由哪些学生选修。

学生	学 号	姓 名	年 龄	性 别
课程	课程号	课程名	学 时	

图 1.3

在关系数据库中,联系也可以用表来表示。图 1.4 中的表,就表示了学生与课程之间的联系。从这个表中,我们很清楚地看到,哪些学生选修了哪些课,哪些课由哪些学生选修。

学 号	课 程 号
871024	C-001
871024	C-005
871721	C-005
871721	C-007
880015	C-001
⋮	⋮

图 1.4

二、候选码与主码

在表中一个能唯一标识行的属性或列称之为候选码。例如,“学生”表中,“学号”就是候选码,因为在表中,一旦确定了学号值,那么也就唯一确定了行。如学号为“871721”就确定第二行数据。在“课程”表中,“课程号”为候选码。有的表有多个候选码,有的表仅有一个候选码。当一个表有多个候选码时,可指定其中任何一个为主码;如一个表仅有一个候选码,则该候选码即为主码。

主码往往可以用来建立表与表之间的联系。由于主码可用来识别表中的数据,因此在使用中主码不要赋空值。

三、关系数据语言

在关系数据库中,提供给用户对数据库进行操作的语言,称之为关系数据语言。关系数据语言结构简单,是一种使用十分方便的用户接口。

关系数据语言可以分成两大类,即关系代数类和关系演算类:

关系数据语言	{	关系代数语言	{	元组关系演算语言
		关系演算语言		域关系演算语言

关系代数语言是由一组以关系作为运算对象的特定运算所组成的。用户通过这组运算对一个或多个关系不断地进行“分割”与“组合”而得到所需要的数据。关系代数的这组运算包括传统的集合运算,如并运算、差运算和交运算;还包括专门运算,如选择运算、投影运算、连接运算和除运算。

关系演算语言分成两种:一种是元组关系演算语言,另一种是域关系演算语言。ORACLE 系统使用的 SQL 语言主要是一种元组关系演算语言,但是它也引进了关系代数的一些运算功能。因此它在使用上就显得更灵活与方便。

1.2 ORACLE 简介

ORACLE 数据库系统是由 ORACLE 公司所研制与开发的一个关系数据库系统。ORACLE 数据库系统最早的版本是 1979 年推出的,那时它运行在 DEC PDP-11 计算机上。ORACLE 数据库经过十多年的发展,其功能与性能得到不断的完善,已成为功能最齐全的、最受欢迎的关系数据库系统。它不但成功地运行在大型机上,而且在小型机和微机上广泛使用。它还可以运行在多种操作系统上。这种对环境的广泛适应性,使 ORACLE 系统便于推广,更便于 ORACLE 应用系统的维护与移植,这也是它颇受欢迎的原因之一。目前广泛使用的 ORACLE 系统是 ORACLE5.1 版本和 ORACLE6.0 版本。

ORACLE 系统由 ORACLE 的核心、SQL * PLUS 接口、主语言接口以及各种系统实用程序组成。其构成如图 1.5 所示。

SQL * PLUS 允许用户使用 SQL 命令交互式访问数据库,也允许用户使用 SQL * PLUS 命令(有关 SQL * PLUS 命令将在 1.4 节简单介绍)与系统发生联系。

用户不但可以使用命令方式操作数据库,还可以使用程序方式使用数据库。ORACLE 系统提供了两种程序方式使用数据库的接口:一种是 OCI,在这种接口中,系统提供了一组调用 ORACLE 数据库的函数,用户可以象调用 C 语言的函数那样来调用 ORACLE 数据库。另一种接口是预编译接口,在这种接口中,允许用户在高级语言中(如 C 语言等),嵌入处理 ORACLE 数据库的 SQL 语句,这种程序是由高级语言与 SQL 语言共同组成的。

SQL * FORMS 是个快速开发应用软件的工具,利用它可不必编程,只需通过选择一些菜单和相应的功能键,就可完成应用的开发,通过已建立的屏幕格式能够方便地对数据库进行查询、插入、删除和修改操作。此外,SQL * PLUS 还提供了触发器机制,为保证数据的完整性、一致性和合法性提供了必要的检验手段。

SQL * Report 是专用于报表的软件,用户可以利用 SQL 的查询能力从数据库中取出数据,并将这些数据与附加的正文一起,按照事先设计的格式进行输出。

SQL * CALC 是类似于 Lotus 1-2-3 的电子表格,它不仅可以单独象 Lotus 1-2-3 那样使用,还可以使用 SQL 语言存取数据库中的数据,并与普通数据和公式一起进行拷贝、移动和保存。另外,SQL * CALC 与 Lotus 1-2-3 兼容,它可读取由 Lotus 1-2-3 生成的电子表格文件。

SQL * Graph 是绘制各种统计图形的软件,它需要与 SQL * PLUS 一起使用,可以由数据库中检索得到的数据绘制折线图、直方图和饼形图,并在屏幕上和绘图仪上输出。

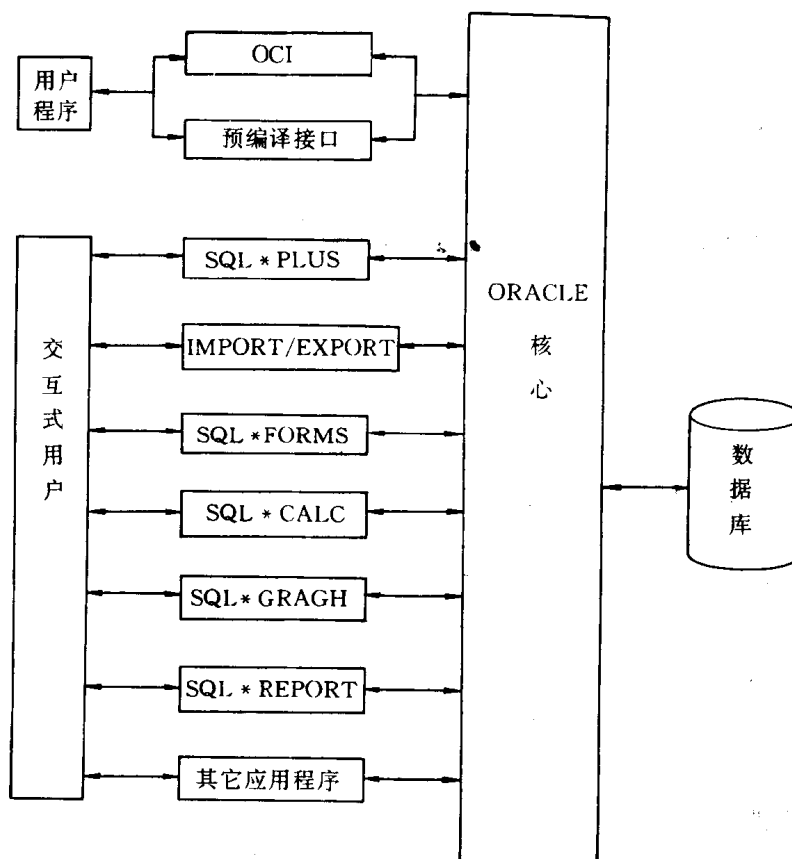


图 1.5

1.3 结构查询语言 SQL

SQL 语言的前身是一种称之为 SQUARE (Specifying Queries As Relational Expressions) 的数据语言, 早在 1973 年就被提出了。当时它仅作为关系数据库的交互式存取的查询语言。这种语言较多地采用了数学符号, 后经修改在 1974 年发展成 SEQUEL (Structured English Query Language)。SEQUEL 语言在结构上更类似英语, 这样更便于人们掌握与使用。1976 年又进一步对 SEQUEL 语言进行修改, 从而形成了 SQL (Structured Query Language) 语言。由于历史的原因, SQL 读成“sequel”, 也可读作“S. Q. L”。

SQL 语言从它一问世就受到人们广泛重视与欢迎。近几年来, SQL 语言已成为标准的数据库语言。现已有一百多个数据库管理产品支持 SQL 语言, 它已运行在从微机、小型机一直到大型机上。

SQL 语言有以下几个特点:

一、非过程化程度高

我们知道用户在使用计算机系统完成工作时, 是使用系统所提供的用户语言来表达或描述自己的处理要求的。例如常用的 FORTRAN、COBOL、C 等都是系统所提供的用户语

言。用户使用这些语言来编写程序,然后通过程序的执行来完成自己所要做的工作。如果系统提供的用户语言功能强,用户使用起来就方便得多,描述处理要求也容易。我们在使用 FORTRAN、COBOL 等语言编写程序对数据进行处理加工时,需要把对数据进行处理的每一过程都在程序中表达清楚,否则程序就不能正确地反映用户的要求与意图。我们把这样的语言称之为过程化语言。使用这种语言编写程序时要在程序中把用户要干什么、如何干都表达得很清楚。而非过程化语言的特点就是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,这由系统决定与完成。SQL 语言是一种非过程化程度相当高的语言,用户只需在程序中指出要做什么就可以了。

由于 SQL 语言的这种特点,使得 SQL 语言使用起来相当方便,程序的编写也相当简单。

二、用户性能好

衡量一个语言好坏的另一个标准是用户性能的好坏。所谓用户性能好是指一种语言在被一个新用户学习掌握时,用户不必花费太多的时间就能学会,并且很快就能熟练地掌握与使用。经过试验,SQL 语言是一种用户性能非常好的语言,它非常便于用户学习与掌握,这也是它受到广大用户喜爱的原因之一。

三、语言功能强

SQL 语言是一种关系数据语言。关系数据语言分为两大类,一类是关系代数语言,另一类是关系演算语言。这两类语言在结构上具有不同的特点,各自具有自己的长处。而 SQL 语言具有两类语言的特点。因此 SQL 语言是一种功能很强的语言。

四、提供有“视图”数据结构

SQL 语言可以对两种基本数据结构进行操作。一种称之为“表(table)”,也就是人们常说的“关系”。另一种就是“视图(view)”,视图可以是由某个表中满足一定条件的元组(行)组成,也可以是由某个表的某些属性(列)组成,还可以是若干个表经过一定运算的结果。总之,视图是由数据库中满足一定条件约束的数据所组成的。通常我们将表定义为基本关系,视图定义为虚关系。虚关系在数据库中不实际存放。在 SQL 语言中,用户可以对基本关系进行操作,也可以对视图进行操作。当对视图进行操作时,由系统转换成对基本关系的操作。

视图可以作为某个用户的专用数据部分,这样便于用户使用,提高了数据的独立性,有利于数据的安全保密。

五、两种使用方式

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。采用命令方式使用 SQL 语言时,用户通过交互式的方式,每输入一条命令,系统就执行该命令,并且显示执行的结果;SQL 语言还可以嵌入在象 COBOL、FORTRAN 和 C 等高级语言中,组成一个完整的程序。用户可根据自己不同的需要,灵活地选择相应的使用方式,以满足不同的要求。

六、提供数据控制功能

数据控制功能是数据库系统的重要部分。SQL 语言提供了事务控制，它能保证数据的共享以及并发使用而不产生干扰，也便于对数据库的恢复。SQL 语言另提供了授权控制，也称为存取控制，它是为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而采取的一种保护性措施。有关事务控制与授权控制将在有关章节中详细介绍。

1.4 SQL 命令缓冲区及其使用

一、SQL 和 SQL * PLUS 命令的输入

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数据进行操作，而 SQL * PLUS 命令主要用来控制查询结果显示形式、编辑 SQL 缓冲区命令以及设置一些选项，这些选项影响 SQL * PLUS 的性能。

当启动 SQL * PLUS，并出现 SQL>提示后，就可以输入 SQL 命令或 SQL * PLUS 命令。

SQL 命令可以分多行输入，所有 SQL 命令必须以分号(;)结束。在输入 SQL 命令过程中，如果按[RETURN]键，则表示从下一行继续输入，在下一行的开始处将自动出现一个行号。

例 显示 EMP 表，输入 SQL 命令的过程为：

1. 在 SQL>提示后输入第一行命令：

```
SQL>SELECT *
```

输入完第一行命令后，按[RETURN]键从下一行继续输入。

2. SQL * PLUS 将显示第二行的提示“2”，输入第二行命令：

```
2 FROM EMP;
```

分号(;)表示整个命令输入结束，按[RETURN]键。

3. SQL * PLUS 执行这条命令，在屏幕上显示执行结果。

4. 结果显示完后，SQL * PLUS 重新显示提示 SQL>。

多数 SQL 命令由多个子句构成，多个子句可以占一行。为了清楚起见，最好一个子句占一行。

SQL * PLUS 命令不能象 SQL 命令那样输入，它不必以分号结束。在输入 SQL * PLUS 命令过程中，按[RETURN]键表示输入完成。如果在一行末尾，光标会自动移到下一行，可以继续输入，直到完成输入按[RETURN]键为止。

二、SQL 缓冲区

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，即当输入新的 SQL 命令后，将删除原来保存的 SQL 命令。

用户可以利用 SQL 缓冲区在不再重新输入命令情况下，编辑和重新运行上一次执行的 SQL 命令。

在 SQL 缓冲区中不能保存 SQL * PLUS 命令。

表 1.1 列出了用于编辑和重新运行 SQL 缓冲区中命令的 SQL * PLUS 命令。

表 1.1

命 令	缩 写	意 义
APPEND text	A text	行尾增加 text
CHANGE	C/old/new	在当前行中将 old 改为 new
CHANGE	C/text/	从当前行中删除 text
CLEAR BUFFER	CL BUFF	删除所有行
DEL	(无)	删除当前行
INPUT	I	增加一行
INPUT	I text	增加一个由 text 组成的行
LIST	L	显示 SQL 缓冲区所有行
LIST n	Ln	显示一行
LIST m n	Lm n	显示多行(从 m 到 n)
RUN	R	运行当前 SQL 命令

下面分别介绍这些 SQL * PLUS 命令的使用。

1. 显示 SQL 命令

使用 LIST 命令可以显示当前 SQL 缓冲区中保存的 SQL 命令。如果 LIST 命令后面不带任何参数,那么将显示 SQL 命令的所有行;如果 LIST 命令后面带一个参数(LIST n),则显示一行,即第 n 行;如果 LIST 后面带有两个参数(LIST m n),那么显示多行,即从第 m 行到第 n 行。

例 显示 SQL 缓冲区的内容。

```
SQL>LIST
1  SELECT *
2  FROM EMP
3 * WHERE DEPTNO='D-03'
```

行号 3 后面的星号表示第 3 行为当前行。除 LIST 和 RUN 命令外,其它编辑命令都只对 SQL 缓冲区当前行操作。

另外,在 SELECT 命令尾部输入的分号没有显示,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,这样使得编辑更为方便。

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有一个参数的 LIST 命令选择当前行。

例 输入下面错误的命令:

```
SQL> SELECT EMNO, ENAME, SALARY
2  FROM EMP
3  WHERE SALARY>150;
SELECT EMNO, ENAME, SALARY
*
```

ERROR at line 1: ORA-0704: invalid column name

为了将 EMNO 改正为 EMPNO,先用 LIST 命令使第一行成为当前行,输入:

```
SQL>LIST 1
```

下面一行出现在屏幕上

```
1 * SELECT EMNO, ENAME, SALARY
```

然后用 CHANGE 命令进行改正,输入:

```
SQL>CHANGE/EMNO/EMPNO
```

改正后在屏幕上显示:

```
1 * SELECT EMPNO, ENAME, SALARY
```

这时可重新运行修改后的 SQL 命令。

3. 重新运行的 SQL 命令

当 SQL 缓冲区中 SQL 命令修改完后,就可以利用 RUN 命令重新运行。

例 运行当前 SQL 缓冲区中的命令。

```
SQL>RUN
```

4. 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,则在当前行的位置插入多个新行;如果输入 INPUT 命令的缩写形式:I text,那么在当前行的位置只插入一个新行。

例 假设当前 SQL 缓冲区的内容为:

```
SQL>LIST
```

```
1 SELECT *  
2 FROM EMP  
3 * ORDER BY SALARY DESC
```

如果在第五行之前输入一个新行,首先用 LIST 命令将第二行设置为当前行,然后输入 INPUT 命令。

```
SQL>LIST 2
```

```
2 * FROM EMP
```

```
SQL>INPUT
```

```
3i WHERE DEPTNO='D-03'  
4i
```

当输入 INPUT 命令之后,SQL * PLUS 出现一个行号,这个行号表明所要插入的行,这时可输入一个或多个新行。在连续按两次[RETURN]键之后,插入操作结束。其中一次按[RETURN]键表示结束一行输入,另一次按[RETURN]键表示不再输入新行。

输入完新行后,可用 RUN 命令重新运行 SQL 缓冲区中的 SQL 命令。

```
SQL>RUN
```

```
1 SELECT *  
2 FROM EMP  
3 WHERE DEPTNO='D-03'  
4 * ORDER BY SALARY DESC
```

EMPNO	ENAME	AGE	HIRE- DATE	SALARY	BONUS	TEL	DEPTNO
0594	LiJing	40	11-DEC-81	155	25	427364	D-03
0573	ZhaoYan.hua	35	21-OCT-86	105	20	273924	D-03
0854	LiuJian.jun	35	19-JUL-84	97	15	273921	D-03
0924	QianDong	25	23-AUG-89	80	15		D-03

5. 在一行后追加内容

使用 APPEND 命令可以在 SQL 缓冲区中当前行末尾追加内容。

例 假设当前 SQL 缓冲区的内容为:

```
SQL>LIST
```

```

1 SELECT EMPNO, ENAME
2 FROM EMP
3 * WHERE DEPTNO='D-03'

```

如果在第一行末尾追加两列 SALARY 和 BONUS,操作过程为:

```

SQL>LIST 1
1 * SELECT EMPNO, ENAME
SQL>APPEND, SALARY, BONUS
1 * SELECT EMPNO, ENAME, SALARY, BONUS
SQL>RUN
1 SELECT EMPNO, ENAME, SALARY, BONUS
2 FROM EMP
3 * WHERE DEPTNO='D-03'

```

EMPNO	ENAME	SALARY	BONUS
0854	LiuJian-jun	97	15
0573	ZhaoYan-hua	105	20
0594	LiJing	155	25
0924	QianDong	80	15

先用 LIST 命令将第一行设置为当前行,然后使用 APPEND 命令在当前行末尾追加内容,最后可用 RUN 命令重新运行 SQL 缓冲区中的 SQL 命令。

6. 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加 1。这样,在把第一行设置为当前行时,连续多次输入 DEL 后可删除 SQL 缓冲区中所有的行。

删除 SQL 缓冲区中所有行也可以使用 CLEAR BUFFER 命令。CLEAR BUFFER 命令的作用是清 SQL 缓冲区为空。

1.5 ORACLE 系统的安装、启动和退出

本节介绍 5.1 版 ORACLE RDBMS 及其工具软件在 PC-DOS 或 MS-DOS 下作为单用户运行的安装。关于在其它操作系统下的安装请参见有关资料。

一、ORACLE RDBMS V5.1 的环境要求

在 PC-DOS 或 MS-DOS 下正常运行 5.1 版 ORACLE RDBMS 需满足下列环境要求:

1. IBM-PC/AT,或百分之百兼容的微型计算机。
2. 640K 字节的使用内容。
3. 至少 896K 字节的扩充存储器。
4. 一个 5.25 英寸软盘驱动器。
5. 一个硬盘,安装单用户 ORACLE 的所有部分,至少需要 7.5 兆字节的空间。
6. 3.0 版本或更高版本 PC-DOS 或 MS-DOS。

二、ORACLE 系统安装

在微型计算机上安装 5.1 版 ORACLE 分为两个阶段:准备阶段和 ORACLE 软件安装阶段。准备阶段是为 ORACLE RDBMS 以及工具软件的安装作准备,确认 ANSI.SYS 文件以及建立 ORACLE 目录结构。目录结构如图 1.6 所示,其中:

BIN 为子目录下存放可执行程序 and 工具软件的批处理文件。

DBS 为子目录下存放 ORACLE 数据文件和参数文件。

CALC 为子目录下存放 SQL * CALC 使用的文件。

NET 为子目录下存放 SQL * NET 使用的文件。

PRO 为子目录下存放 ORACLE 程序设计语言接口文件。

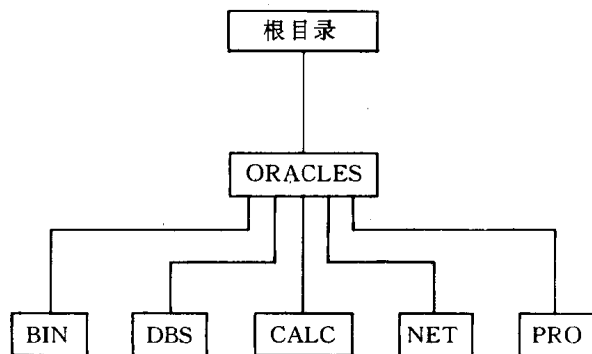


图 1.6 ORACLE 目录结构

ORACLE 软件安装阶段是安装 ORACLE RDBMS 和工具软件。工具软件根据需要可以进行有选择的安装,对那些暂时不使用的工具软件在使用时再进行安装。

下面详细介绍安装 ORACLE 的过程。

1. 准备阶段

第一步:备份 ORACLE 软件

使用操作系统提供的磁盘拷贝命令 DISKCOPY 将 ORACLE 全部软盘进行备份,将 ORACLE 源盘保存起来,使用拷贝得到的软盘进行 ORACLE 安装,这样可以避免由于不正确的安装对源盘软件的损坏。

第二步:设置当前驱动器

当前驱动器上将要安装 ORACLE。例如:若想将 ORACLE 安装在 C 驱动器上,那么需要把 C 驱动器作为当前驱动器,输入:

A>C :

C>

第三步:运行 ORAINST

将标有“ORACLE Install”的安装盘插入到 A 驱动器中,然后输入下面命令 运行 ORAINST。

C>A : ORAINST

第四步:确定 ORACLE 主目录

输入 ORACLE 主目录名。如果直接按回车键,则 ORACLE5 作为主目录名。

第五步:确定引导驱动器

ORAINST 要检测 AUTOEXEC. BAT 和 CONFIG. SYS 是否适合要求,因此需要知道哪个是操作系统引导盘。

第六步:修改 AUTOEXEC. BAT

AUTOEXEC. BAT 是个批处理文件,在每次启动操作系统 DOS 时,都将在根目录下查找这个文件,并执行保存在这个文件的命令。

根据系统的情况,对 AUTOEXEC. BAT 文件作如下修改:

(1) 在 AUTOEXEC. BAT 文件不存在情况下,将自动建立。

(2) 如果 AUTOEXEC. BAT 文件中没有 PATH 命令,则把下面的命令加入到文件中

(如果不在 C 驱动器上安装,应改为要安装的驱动器符)。

PATH C:\; C:\ORACLE5\BIN

(3) 如果 AUTOEXEC.BAT 文件中有 PATH 命令,则在此 PATH 命令加上 C:\ORACLE5\BIN。PATH 命令应保证包括含有 COMMAND.COM 的目录。

(4) 如果不在缺省目录 ORACLE5 中安装,则必须包含有 SET 命令。

SET CONFIG= $\langle$ ORACLE_HOME $\rangle$ CONFIG.ORA

$\langle$ ORACLE_HOME $\rangle$ 是用来安装 ORACLE 的驱动器和目录。

第七步:修改 CONFIG.SYS 文件

如果 CONFIG.SYS 文件不存在,则将自动建立 CONFIG.SYS 文件。如果 CONFIG.SYS 文件存在,则需要作下列修改:

(1) 若有 FILES 命令,但其值小于 40,则将它改为 FILES=40;若没有 FILES 命令,则将插入 FILES=40 命令。

(2) 若有 BUFFERS 命令,但其值小于 16,则将它改为 BUFFERS=16;若没有 BUFFERS 命令,则将插入 BUFFERS=16 命令。

(3) 若有 BREAK 命令,则使其成为 BREAK=ON;若没有 BREAK 命令,则将插入 BREAK=ON。

(4) 若没有引用 ANSI.SYS 的 DEVICE 命令,则插入 DEVICE=C:\ANSI.SYS 命令。

第八步:将 ANSI.SYS 文件存入根目录

在 DEVICE=C:\ANSI.SYS 命令插入到 CONFIG.SYS 文件中时,如果 ORAINST 找不到 ANSI.SYS 文件,则发出警告,要求把 ANSI.SYS 文件拷贝到引导盘的根目录下。

第九步:重新启动操作系统

在修改 AUTOEXEC.BAT 和 CONFIG.SYS 文件之后,要求重新启动操作系统,以便能够执行这两个文件中修改后的命令。

2. ORACLE 软件安装阶段

第一步:运行 ORAINST

经过第一阶段的操作,已经为安装 ORACLE 作好了准备,这时可以再次输入 ORAINST 命令开始进行安装。

C>ORAINST

第二步:选择安装什么类型的 ORACLE

ORAINST 将显示如下的菜单:

(P)rofessional Oracle

(N)etworkstation Oracle

按 P 键则安装单用户 ORACLE,按 N 键则安装网络工作站 ORACLE。

第三步:安装 ORACLE 工具

选择安装什么类型 ORACLE 后,还要选择是装入所有 ORACLE 工具,还是其中单个工具,然后进行安装,根据屏幕提示顺序在 A 驱动器中插入软盘。

第四步:结束 ORACLE 安装

每个工具软件安装完之后,都可按 Q 键结束安装过程,或者安装完所选择的工具之后,

自动结束安装过程。

3. ORACLE 安装完成后的工具软件安装

在第一次安装时可以不必将所有 ORACLE 工具软件都安装完,但至少应安装 ORACLE RDBMS 部分,对其它工具软件可以在以后需要的时候安装。

在安装剩余工具软件时,不必重新开始整个安装过程,可以按下述步骤进行:

第一步:启动 ORACLE

C>ORACLE

第二步:启动安装程序 ORAINST

C>ORAINST

以后各步与 ORACLE 安装阶段相同。

三、启动 ORACLE

如果是刚完成安装过程,ORACLE RDBMS 已经装入内存,就不必执行下面的命令;否则,需要执行下面的命令,启动 ORACLE RDBMS:

C> ORACLE

这个命令是把 ORACLE RDBMS 装入内存中,在命令执行后就可运行工具软件。例如:执行 SQL * PLUS 可输入下面的命令:

C> SQLPLUS

在用 QUIT 命令退出 SQL * PLUS 之后,返回到操作系统提示状态,这时,可继续运行其它工具软件,或者重新运行 SQL * PLUS。

四、退出 ORACLE

在使用 ORACLE 之后,建议用如下命令关闭数据库:

C> IOR S

一旦启动 ORACLE RDBMS,它就一直保存在内存中,要把 ORACLE RDBMS 从内存中清除,需输入命令:

C> REMORA ALL

在重新启动或关闭计算机之前,不必把 ORACLE RDBMS 从内存中清除

1.6 SQL * PLUS 使用的键

SQL * PLUS 为键盘上的一些键赋予了特殊的功能,用户按这些键可以完成一定的操作。表 1.2 列出了这些键。

由于不同的计算机键盘上这些键位置和名称有所不同,本书将根据这些键的功能分别为它们命名,在实际使用时通过询问数据库管理员 DBA 或查阅有关资料获取这些键与具体计算机键盘的对应关系。

表 1.2 中同时列出了这些键对应于 IBM-PC/AT 键盘上具体的键。