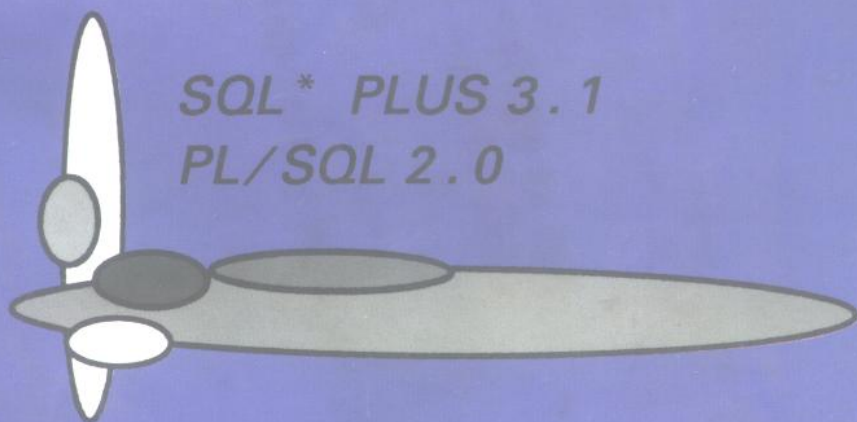


ORACLE 7 应用开发工具系列教程之一

SQL * PLUS 和 PL/SQL 程序设计

刘金亭 苏红宇 编



电子工业出版社

SQL * PLUS 和 PL/SQL 程序设计

重
印
社

SQL * Plus 和 PL/SQL 程序设计

刘金亭 苏红宇 编

电子工业出版社

内 容 简 介

1992年 Oracle 公司和清华大学联合创办 Oracle——清华联合大学,旨在培训更多的数据库人员,促进和提高数据库应用及管理领域的水平。ORACLE7 应用开发工具系列教程是为配合 Oracle 大学而编写的。考虑到读者多为 Oracle 用户,在编写过程中既注重实践环节和易读易学,又注重内容的系统性,对每个应用开发工具和产品都作了比较全面的介绍。因此本系列教程不仅是 Oracle 大学的培训教材,也是数据库系统开发人员和最终用户的参考资料。

《ORACLE7 应用开发工具系列教程之一:SQL * Plus 和 PL/SQL 程序设计》全面地介绍了 ORACLE7 支持的 SQL * Plus 3.1 和过程化查询语言 PL/SQL2.0。全书由两篇组成:第一篇 SQL * Plus 3.1,主要介绍 SQL * Plus 基础、操作界面、数据库查询、数据库维护和格式化查询结果等。第二篇 过程化查询语言 PL/SQL,主要介绍 PL/SQL 的基本概念、基础、控制语句、ORACLE 支持、例外处理、过程、包和应用环境等。

SQL * Plus 和 PL/SQL 程序设计

刘金亭 苏红宇 编

责任编辑:宋玉升

*

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

三河市双峰印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:9.125 字数:220 千字

1996 年 1 月第一版 1996 年 1 月第一次印刷

印数:4000 册 定价:14.00 元

ISBN7-5053-3323-2/TP·1260

前 言

作为技术支持和服务的延伸,在 1992 年 Oracle 公司和清华大学 联合创办了 Oracle-清华联合大学,旨在为我国培训更多的数据库人员,促进和提高我国在数据库应用及管理领域的水平。Oracle 公司为联合大学提供有最新版本的 ORACLE7 分布式关系数据库管理系统、应用开发工具(SQL 工具系列和 CDE 协同开发环境等)以及系统的技术资料,清华大学为联合大学提供有经验丰富的师资和良好的上机实习环境。

本系列教程是为配合 Oracle 大学教学而编写的,具有较强的系统性,内容包括 ORACLE7 所支持的最新应用开发工具(如 SQL * Plus, SQL * Forms, SQL * ReportWriter, SQL * Menu, CDE 等)和相关产品(PL/SQL, PRO * C, OCI 等)。考虑到读者多为 Oracle 用户,在编写过程中既注重实践环节和易读易学,又注重内容的系统性,对每个应用开发工具和产品都作了比较全面介绍。因此本系列教程不仅是 Oracle 大学的培训教材,也是数据库系统开发人员和最终用户的参考资料。

《ORACLE7 应用开发工具系列教程之一:SQL * Plus 和 PL/SQL 程序设计》全面地介绍了 ORACLE7 支持的 SQL * Plus 3.1 和过程化查询语言 PL/SQL2.0。全书由两篇组成:第一篇 SQL * Plus 3.1,主要介绍 SQL * Plus 基础、操作界面、数据库查询、数据库维护和格式化查询结果等。第二篇 过程化查询语言 PL/SQL,主要介绍 PL/SQL 的基本概念、基础、控制语句、ORACLE 支持、例外处理、过程、包和应用环境等。

本书第一篇和附录 A 由苏红宇编写,第二篇和附录 B 由刘金亭编写,全书由刘金亭统稿。本书编写过程中得到了 ORACLE-中国公司和清华大学计算中心的大力支持,同时得到了中国软件行业协会和电子工业出版社的支持和帮助,吴克忠教授、胡昆山高工和宋玉升编审审阅了全稿,并提出许多宝贵意见,在此表示感谢。由于时间仓促,书中错误难免,请读者指正。

编者

1995 年 8 月

目 录

第一篇 SQL * Plus 3.1

第一章 SQL * Plus 概述	(1)
§ 1.1 数据库的基本概念	(1)
1.1.1 什么是数据库和数据库管理系统	(1)
1.1.2 什么是表和视图	(1)
1.1.3 什么是数据字典	(3)
§ 1.2 SQL 语言概述	(3)
1.2.1 什么是 SQL 语言	(3)
1.2.2 SQL 语言简介	(3)
§ 1.3 SQL * Plus 和 SQL、PL/SQL	(5)
参考文献	(6)
第二章 SQL * Plus 界面	(7)
§ 2.1 登录和退出 SQL * Plus	(7)
2.1.1 登录 SQL * Plus	(7)
2.1.2 退出 SQL * Plus	(7)
§ 2.2 命令的编辑与运行	(7)
2.2.1 编辑和运行 SQL 命令	(8)
2.2.2 编辑和运行 PL/SQL 块	(12)
2.2.3 编辑和运行 SQL * Plus 命令	(13)
2.2.4 建立和编辑批处理命令文件	(14)
§ 2.3 获得帮助信息	(14)
2.3.1 SQL * Plus 命令帮助	(14)
2.3.2 列出一个表的列定义	(14)
2.3.3 解释出错信息	(15)
第三章 数据库查询	(16)
§ 3.1 普通查询	(16)
3.1.1 SELECT 子句和 FROM 子句	(16)
3.1.2 WHERE 子句	(20)
3.1.3 ORDER BY 子句	(24)
3.1.4 HAVING 和 GROUP BY 子句	(28)
§ 3.2 高级查询	(28)
3.2.1 连接查询	(28)
3.2.2 子查询	(30)
3.2.3 设置操作符	(31)
第四章 数据库操作	(34)
§ 4.1 插入命令—INSERT	(34)
§ 4.2 修改命令—UPDATE	(36)

§ 4.3	删除数据—DELETE	(38)
§ 4.4	事务控制命令	(39)
4.4.1	提交命令(COMMIT)	(39)
4.4.2	回滚命令(ROLLBACK)	(40)
第五章	表和视图	(41)
§ 5.1	表	(41)
5.1.1	表的建立	(41)
5.1.2	表的修改	(42)
5.1.3	表的删除	(43)
§ 5.2	视图	(43)
5.2.1	视图的建立	(43)
5.2.2	视图的删除	(45)
§ 5.3	表和视图的拷贝	(45)
第六章	函数	(47)
§ 6.1	简介	(47)
§ 6.2	单行函数	(47)
6.2.1	字符函数	(47)
6.2.2	日期函数	(49)
6.2.3	数值函数	(50)
6.2.4	转换函数	(50)
6.2.5	其它函数	(52)
§ 6.3	分组函数	(53)
6.3.1	分组函数	(53)
6.3.2	SELECT 命令中的 GROUP BY 子句和 HAVING 子句	(54)
第七章	格式化查询结果	(56)
§ 7.1	简介	(56)
§ 7.2	格式化列	(56)
7.2.1	改变列标题	(56)
7.2.2	格式化数值列	(58)
7.2.3	格式化 CHAR、VARCHAR2、LONG 和 DATE 列	(59)
7.2.4	复制列的显示属性	(60)
7.2.5	列出以及重新设置列的显示属性	(60)
7.2.6	抑制和恢复列的显示属性	(61)
7.2.7	在 WRAP 列值后面打印分隔行	(62)
§ 7.3	使用空行和统计行	(62)
7.3.1	在中断列压缩重复值	(63)
7.3.2	在中断列值改变时插入空行	(64)
7.3.3	列出和删除当前的中断定义	(65)
7.3.4	当中断值发生变化时计算统计值	(65)
7.3.5	在报表的结束处计算统计行	(66)
7.3.6	计算多个统计值和统计行	(67)
7.3.7	列出和删除 COMPUTE 定义	(68)

§ 7.4	定义页标题和页面大小	(68)
7.4.1	设置顶部标题和底部标题	(68)
7.4.2	在标题中显示页码和其它系统维护值	(70)
7.4.3	列出、禁止和恢复页标题的定义	(71)
7.4.4	在标题中显示列值	(72)
7.4.5	在页标题中显示当前日期	(73)
7.4.6	设置页的大小	(73)
§ 7.5	存储和打印查询结果	(74)
7.5.1	将查询结果存入文件	(74)
7.5.2	打印查询结果	(74)

第二篇 PL/SQL 2.0 程序设计

第一章	PL/SQL 概述	(75)
§ 1.1	PL/SQL 的特点	(75)
§ 1.2	PL/SQL 的体系结构	(77)
	参考资料	(77)
第二章	PL/SQL 基础	(79)
§ 2.1	字符集和合法单元	(79)
2.1.1	标识符	(79)
2.1.2	定界符	(80)
2.1.3	字符	(80)
2.1.4	注释	(81)
§ 2.2	数据类型	(81)
2.2.1	BINARY-INTERGER 类型	(82)
2.2.2	BOOLEAN 类型	(82)
2.2.3	支持的 SQL 类型	(82)
2.2.4	RECORD 类型和 TABLE 类型	(82)
§ 2.3	数据类型转换	(83)
§ 2.4	变量和常量	(84)
2.4.1	变量说明和作用域	(84)
2.4.2	RECORD 类型变量	(84)
2.4.3	PL/SQL 表 TABLE 类型变量	(86)
§ 2.5	表达式	(86)
§ 2.6	PL/SQL 程序结构	(87)
第三章	控制语句	(89)
§ 3.1	分支语句	(89)
3.1.1	IF-THEN 语句	(89)
3.1.2	IF-THEN-ELSE 语句	(89)
3.1.3	IF-THEN-ELSIF 语句	(90)
§ 3.2	循环语句	(91)
3.2.1	LOOP 循环和 EXIT 语句	(91)
3.2.2	WHILE 循环	(92)

3.2.3	FOR 循环	(93)
§ 3.3	GOTO 语句和 NULL 语句	(94)
第四章	与 ORACLE 交互	(95)
§ 4.1	SQL 支持和远程访问	(95)
§ 4.2	光标操作	(97)
4.2.1	显式光标	(97)
4.2.2	隐式光标	(100)
4.2.3	封装光标	(101)
4.2.4	光标 FOR 循环	(101)
§ 4.3	事务处理	(102)
§ 4.4	数据库触发器	(102)
第五章	例外处理	(105)
§ 5.1	例外的概念	(105)
§ 5.2	预定义例外	(106)
§ 5.3	用户定义例外	(107)
5.3.1	用户定义例外的说明	(107)
5.3.2	用户定义例外的作用范围	(107)
5.3.3	内部错误命名	(107)
5.3.4	生成例外	(108)
§ 5.4	例外的传播	(109)
§ 5.5	例外处理	(109)
§ 5.6	例外处理实例	(111)
第六章	子程序	(112)
§ 6.1	子程序的概念	(112)
§ 6.2	过程	(112)
§ 6.3	函数	(114)
§ 6.4	正向说明	(115)
§ 6.5	实参数	(116)
§ 6.6	重载	(117)
§ 6.7	递归	(118)
§ 6.8	存储子程序	(119)
第七章	包	(121)
§ 7.1	包的构成和定义	(121)
§ 7.2	包的引用	(123)
§ 7.3	标准包和产品特有包	(123)
7.3.1	标准包	(123)
7.3.2	产品特有包	(124)
第八章	PL/SQL 的使用环境	(125)
§ 8.1	SQL * DBA 和 SQL * Plus 环境	(125)
8.1.1	输入和存储一个 PL/SQL 匿名块	(125)
8.1.2	运行一个 PL/SQL 匿名块	(126)
8.1.3	生成命令文件	(126)

8.1.4 加载和运行命令文件	(127)
8.1.5 调用存储子程序和引用包	(127)
§ 8.2 SQL * Forms 环境	(127)
§ 8.3 SQL * Menu 环境	(128)
§ 8.4 PRO * C 和 OCI 环境	(128)
§ 8.5 其它环境	(129)
附录 A. SQL 命令和 SQL * Plus 命令	(130)
附录 B. PL/SQL 保留字	(134)

第一篇 SQL * Plus 3.1

第一章 SQL * Plus 概述

§ 1.1 数据库的基本概念

1.1.1 什么是数据库和数据库管理系统

数据库是数据的集合,数据库管理系统是为管理数据库而配置的软件工具,数据库管理系统的主要功能有:

- 存储、检索和修改数据
- 保护数据的一致性
- 解决并发性问题
- 为数据提供一个通用的界面
- 限制用户访问数据库的权限

1.1.2 什么是表和视图

每个 ORACLE 数据库都具有相应的物理结构和逻辑结构。物理结构决定了数据库信息在磁盘上的真正的物理存储结构,而逻辑结构是用户所涉及的数据库结构。ORACLE 数据库的逻辑结构由两类因素决定:

- 逻辑存储结构(如表空间、段和范围等)
- 数据库模式对象(如表、视图等)

逻辑存储结构将支配一个数据库的物理空间如何使用,而数据库模式对象及它们之间的相互联系组成了一个数据库的关系设计。

表和视图是比较常用的两种数据库模式对象。在多数情况下,用户在建立、检索和修改数据库时,直接面对的就是表和视图。

表是数据库中数据存储的基本单位,其数据按行和列存储。每个表都有一个表名,表中的每一列都有一个列名,每一列中各行的数据都具有相同数据类型和宽度,而每一行中的数据则是对应于某个单个记录的列的信息的集合。

下面给出两个表的例子,它们在本书后面的章节中将多次被用到。

例 1.1 职工情况表(EMP 表)

EMP 表:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7638	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

该例中,表名为 EMP,各列的列名分别为 EMPNO、ENAME、JOB、MGR、HIREDATE、SAL、COMM 和 DEPTNO。每一列的数据都具有相同的数据类型,如 HIREDATE 列的数据均为日期型,而每一行的数据则记录了每个特定职工的职工号及相应的姓名、工作、工资等个人情况。

例 1.2 部门情况表

DEPT 表:

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

本例中,表名为 DEPT,表中的数据记录了每个部门的部门号及相应的名称和地址。

视图是另一种常用的模式对象。视图是一种虚拟表,其数据是由其它表和视图中提取的。视图具有很多与表相类似的特点,它也可以被查询、修改或删除,但是视图和表的一个重要差别在于:视图并不真正的包含数据,一个视图不分配任何存储空间。通常,将视图所基于的表称为基表。

引入视图的好处在于:

(1) 提供了附加的安全级

提供视图,使得用户只能通过视图看到其有权看到的信息,而无权看到基表中的全部信息。

(2) 隐藏数据的复杂性

通过构造视图,使得用户只看到所需的信息,而不必查看所有的信息,从而使所提供的信息更加简单。

可以提供变换列名的手段,而无需真正地改变基表的定义。

在第 5 章中将详细介绍如何建立视图,从视图的建立过程中,可以更清楚地理解视图的含义和优点。

1.1.3 什么是数据字典

数据字典是 ORACLE 数据库的重要组成部分之一。数据字典由一组只读的表及其视图组成,它提供有关该数据库的信息,如每个 ORACLE 用户的名字及被授予的特权和角色、每个模式对象的名字、数据库中对象的空间分布信息及当前的使用情况等等。数据字典是只读的,其信息只允许被查询。

§ 1.2 SQL 语言概述

1.2.1 什么是 SQL 语言

SQL(结构化查询语言)语言是一种标准的关系数据库语言,用于建立、存储、修改、检索和管理 ORACLE 数据库中的信息。它有三个显著的优点:

(1)它是一种非过程化的语言。因为它每次处理一个记录集而不是每次处理一个单个记录,而且 SQL 对数据提供导航,这意味着用户可以在高层的数据结构上工作,而不必指定对数据的存取方法;

(2)SQL 是一种所有用户都可以使用的统一语言,包括系统管理员、数据库管理员、应用程序员、决策支持系统人员及许多其它类型的终端用户,SQL 可以用于各种数据库活动,包括数据查询、在表中插入、修改和删除行、建立、修改和删除数据对象、控制对数据库和数据对象的存取,保证数据库的一致性和完整性等。

(3)SQL 是所有关系数据库的公共语言,用户可以方便地移植用 SQL 语言编写的程序。

1.2.2 SQL 语言简介

本节介绍 SQL 语言的一些基本概念。

(1)模式对象

数据库模式对象及它们之间的相互联系组成了一个数据库的关系模型。在模式对象中最常用的是表和视图。

(2)SQL 语言的常量

SQL 语言的常量包括文本常量、整数常量和数值常量。文本常量用于指定一个文本或字符串,例如 'JACK'、'101' 均为文本常量;整数常量用于指定一个正整数;数值常量用于指定一个整数或实常数。

(3)SQL 语言的数据类型

由 ORACLE 操作的每一个常量或列值都属于某一种数据类型,一个值的数据类型将一组固定的特性与该值相联系起来,这些特性使得 ORACLE 以不同的方式来处理不同类型的数据。

ORACLE 支持的内部数据类型如下:

VARCHAR2(SIZE):变长字符串,长度 SIZE 由用户指定,SIZE 最大为 2000 字节。

NUMBER:数值型。

LONG:变长字符数据,最长为 2G 字节。

DATE:日期型,有效日期范围从 B. C. 4712 年 1 月 1 日到 A. D. 4712 年 12 月 31 日。

RAW(SIZE):长度为 SIZE 的二进制数据,SIZE 由用户指定,最长不能超过 255 字节。

LONG RAW:变长二进制数据,最长为 2G 字节。

RAWID:十六进制串,用于表示表的行的唯一地址。

CHAR(SIZE):固定长度的字符数据,长度为 SIZE 字节,SIZE 最大为 255,缺省值为 1。

(4)空值(NULL)

如果一行数据在某列没有值,则称该列是空值(NULL)或称该列包含一个空值(NULL)。应当注意,不能用空值来表示取值为 0,它们并不等价。任何包含空值的表达式的计算结果仍为空值。

(5)SQL 语言的运算符

SQL 语言的运算符包括算术运算符、字符运算符、比较运算符、逻辑运算符、集合运算符以及(+)等其它类型的运算符。

算术运算符:

符 号	功 能	例 子
+, -	表示正的或负的表达式	-SAL
*, /	乘除运算	SAL * 1.1
+, -	加减运算	SAL+COMM

字符运算符:

符 号	功 能	例 子
	用于连接两个字符串	'NAME IS' 'JOHN' 等价于 'NAME IS JOHN'

比较运算符:

符号	含 义
=	相等测试
!=, <>	不等测试
>	大于
<	小于
>=	大于或等于
<=	小于或等于

此外,比较运算符还包括 IN、BETWEEN...AND、LIKE、IS NULL、NOT 等,它们的作用将在第 3 章中详细介绍。

逻辑运算符:

符号	含义
NOT	逻辑非
AND	逻辑与
OR	逻辑或

集合运算符包括 UNION、INTERSECT 和 MINUS 等,它们的作用将在第 3 章中详细介绍。

(6)SQL 语言的函数

SQL 语言提供了各种各样的内部函数供用户使用,第 6 章中将详细介绍 SQL 语言的函数。

(7)SQL 语言的表达式

表达式由一个或多个值(常量或列值)、运算符和函数组合而成,并且可以计算出一个值,表达式的计算结果的数据类型通常与它的成分的数据类型相同。例如,COMM/SAL 就是一个表达式。

(8)SQL 语言的条件

条件是由一个或多个含有比较运算符的表达式及逻辑运算符组合而成的,其计算值是 TRUE、FALSE 或 NULL。例如,SAL>1000 AND COMM<1000 就是一个条件。

(9)SQL 语言的命令

SQL 语言提供了用于建立、存储、修改、检索和管理 ORACLE 数据库的多种命令,本书将在后面的章节中予以详细介绍。

§ 1.3 SQL * Plus 和 SQL、PL/SQL

SQL 是一种关系数据库语言,而 PL/SQL 是 SQL 的过程化扩充语言,它将非过程化的关系数据库语言 SQL 和过程化的程序语言结合起来,使得用户可以用过程化的逻辑将 SQL 语言的若干命令连接起来,从而使用过程化方法(如循环、分支等)来处理数据。SQL * Plus 是 ORACLE 提供的一种开发工具,在 SQL * Plus 环境中,可以输入、修改、编辑和运行关系数据库语言 SQL 及其过程化扩充语言 PL/SQL 的全部命令,还可以完成许多附加的功能,如格式化 SQL 的查询结果、在 SQL 数据库之间拷贝数据等。

SQL 命令和 PL/SQL 块都可以在 SQL * Plus 这一开发工具中使用,除此之外,SQL * Plus 还提供了不同于 SQL 命令和 PL/SQL 块的专有命令,用于完成一些附加功能。

SQL * Plus 的主要功能有:

- 输入、编辑、存储、检索和运行 SQL 命令及 PL/SQL 块;
- 对查询结果进行计算并且以报表的形式显示和输出;
- 显示表的列定义;
- 在 SQL 数据库中拷贝数据;
- 向终端用户发送信息或从终端用户接收应答。

SQL 语言及其过程化扩充语言 PL/SQL 是使用 SQL * Plus 的重要基础。

在 SQL * Plus 中,SQL 命令、PL/SQL 块和 SQL * Plus 命令在输入、编辑和运行方法上有一些差异。因此,在使用 SQL * Plus 时,应当区分运行的是哪一种类型的命令。本书将在后面的有关章节中详细介绍。

参考文献

- (1) SQL * Plus User's Guide and Reference Version 3.1 , Part No. 5142-31-1192
- (2) ORACLE 7 Server SQL Language Reference Manual ,Part No. 778-70-1292
- (3) ORACLE 7 Server SQL Language Quick Reference ,Part No. 5421-70-1292
- (4) SQL * Plus Quick Reference Version 3.1 Part No. 3703-31-1192

第二章 SQL * Plus 界面

SQL * Plus 为用户提供了很方便的界面环境,使得用户可以在 SQL * Plus 环境中输入、编辑和运行 SQL、SQL * Plus 命令和 PL/SQL 块,还可以使用户随时获得帮助信息。本章将介绍如何利用 SQL * Plus 的界面环境进行最基本的操作,包括登录和退出 SQL * Plus、命令的编辑与运行及如何获得帮助信息。

§ 2.1 登录和退出 SQL * Plus

2.1.1 登录 SQL * Plus

登录 SQL * Plus 的具体步骤如下:

(1)在操作系统提示符下,输入 SQLPLUS,然后回车。

这时屏幕将显示版本号、日期和版权信息,并提示用户输入用户名,如下所示。

```
SQL * Plus; Version 3.1.3 - Production on Fri April 10 09:39:26 1992
```

```
Copyright (c) Oracle Corporation 1979,1992. All rights reserved.
```

```
Enter user-name:
```

(2)输入正确的 ORACLE 用户名并回车,这时 SQL * Plus 将提示用户输入口令,即显示:

```
Enter password:
```

(3)输入用户口令并回车;

这时 SQL * Plus 将检查用户名和口令,如果正确,SQL * Plus 将显示用户所连接的 ORACLE 版本号及用户可使用的开发工具的版本,然后 SQL * Plus 显示命令提示符:

```
SQL>
```

这意味着用户已经成功地登录到 SQL * Plus 中,可以使用各种 SQL * Plus 允许的命令。

如果输入的用户名或口令不正确,SQL * Plus 将提示用户重新输入用户名和口令,如果三次登录失败,SQL * Plus 将自动返回到操作系统下。

用户也可以用另一种简化的方法登录 SQL * Plus,即在系统提示符下直接输入:

```
SQLPLUS 用户名/口令 (回车)
```

这样可以直接进入 SQL * Plus 提示符状态。

2.1.2 退出 SQL * Plus

在 SQL * Plus 提示符后输入 EXIT,可以退出 SQL * Plus,返回到操作系统。

§ 2.2 命令的编辑与运行

在 SQL * Plus 中,编辑和运行 SQL 命令、PL/SQL 块和 SQL * Plus 命令的方法并不完

全相同。对于每一种命令,SQL * Plus 都提供了多种编辑和运行方法。下面详细介绍。

2.2.1 编辑和运行 SQL 命令

(1)在命令提示符后输入 SQL 命令并运行

为了使命令更具可读性,SQL * Plus 允许将每一个 SQL 命令划分为多行,划分点可以在任意两个单词之间,但不能把一个单词分写在两行中,输入完一行并回车后,SQL * Plus 将显示新的行号并允许在新的一行上输入命令,直到输入某个特殊的字符表示结束一个 SQL 命令的输入过程。SQL * Plus 可以采用下述三种方法结束一个 SQL 命令:

- 在命令的最后一行的尾部加上分号(;)并回车,这时系统将所输入命令的全部内容存入 SQL 缓冲区中以备用并运行该命令,然后显示运行结果。运行完毕后显示新的命令提示符

```
SQL>
```

这时可以输入新的命令。

- 在独自一行输入斜杠 / 并回车(即在命令的最后一行输完后回车,在新的行号提示符后输入斜杠/并回车),则系统将所有输入的命令存入 SQL 缓冲区并运行该命令,然后显示运行结果。运行完毕后显示新的命令提示符

```
SQL>
```

- 在新的输入行输入一空行并回车,则系统将所有输入的命令存入 SQL 缓冲区,并显示新的命令提示符。与前面两种结束 SQL 命令的方法不同,这种方法并不运行 SQL 命令。

例 2.1 从 EMP 表中找出所有工资大于 2500 元的职工的职工号、职工姓名、工作和工资记录。这时需要输入一个 SQL 命令:

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

我们将这个命令分为多行输入,并以上述三种方式结束命令的输入过程。

- 以分号结束输入

```
1 SQL>SELECT
2 EMPNO,ENAME,JOB,SAL
3 FROM EMP
4 WHERE SAL>2500;
```

- 以斜杠结束输入

```
SQL>SELECT
2 EMPNO,ENAME,JOB,SAL
3 FROM EMP
4 WHERE SAL>2500
5 /
```

- 以空行结束输入

```
SQL>SELECT
2 EMPNO,ENAME,JOB,SAL
3 FROM EMP
4 WHERE SAL>2500
```