

第1章 关系数据库与SQL语言环境

- 1.1 关系数据库的基本概念
- 1.2 SQL*Plus环境
- 1.3 Oracle的应用开发工具TOAD
- 1.4 操作准备
- 1.5 阶段训练
- 1.6 练习

1.1 关系数据库的基本概念

1.1.1 数据库和数据库管理系统

数据库是在计算机上组织、存储和共享数据的方法，数据库系统是由普通的文件系统发展而来的。数据库系统具有较高的数据独立性，即不依赖于特定的数据库应用程序；数据库系统的数据冗余小，可以节省数据的存储空间；另外数据库系统还很容易实现多个用户的数据共享。

数据库系统成熟的标志就是数据库管理系统的出现。数据库管理系统(DataBase Managerment System, 简称DBMS)是对数据库的一种完整和统一的管理和控制机制。数据库管理系统不仅让我们能够实现对数据的快速检索和维护, 还为数据的安全性、完整性、并发控制和数据恢复提供了保证。数据库管理系统的核心是一个用来存储大量数据的数据库。

一个真正的数据库系统由硬件和软件两个方面构成。比如我们要使用Oracle数据库，需要安装Oracle公司提供的数据库服务器软件和一台用于安装数据库管理系统的高性能的计算机服务器。

数据库系统的发展经历了层次模型、网状模型及关系模型几个阶段。当今应用最普遍的是关系型数据库管理系统。目前，市场上流行的几种大型数据库，如Oracle、DB2、Sybase、MS SQL Server等都是关系型数据库管理系统。Oracle数据库是一种面向对象的关系型数据库管理系统(ORDBMS)，是基于标准SQL语言的数据库产品。

数据库和数据库管理系统实现了信息的存储和管理，还需要开发面向特定应用的数据库应用系统，以完成更复杂的信息处理任务。典型的数据库应用有C/S(客户/服务器)和B/S(浏览器/服务器)两种模式。C/S模式由客户端和服务端构成，客户端是一个运行在客户机上的数据库应用程序，服务端是一个后台的数据库服务器，客户端通过网络访问数据库服务器。B/S模式是基于Internet的一个应用模式，需要一个WEB服务器。客户端分布在Internet上，使用通用的网页浏览器，不需要对客户端进行专门的开发。应用程序驻留在WEB服务器或以存储过程的形式存放在数据库服务器上，服务端是一个后台数据库服务器。

例如一个有代表性的信息检索网站，通常都是一个典型的基于大型数据库的WEB应用。很多这样的网站都采用Oracle的数据库服务器，以获得优越的性能。图1-1给出了典型的WEB数据库应用系统的结构示意图。

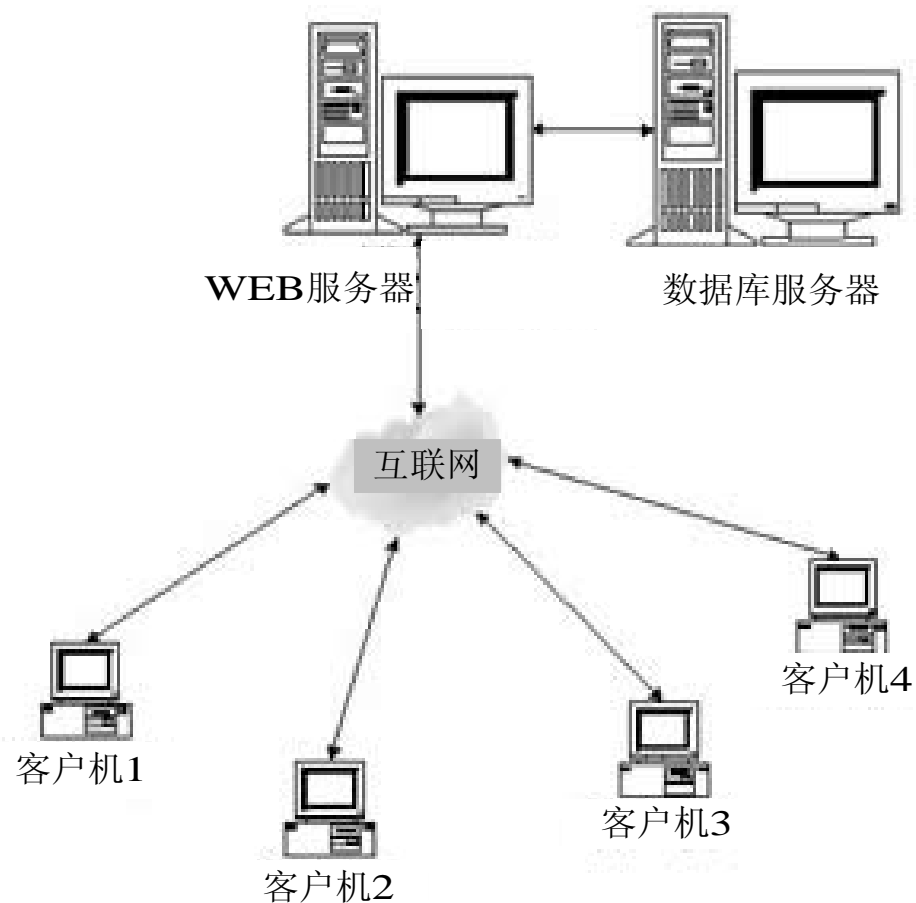


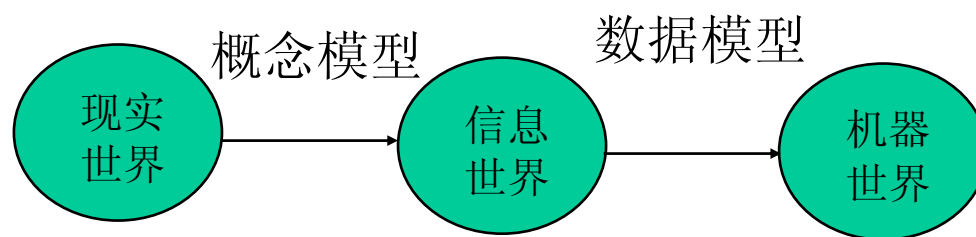
图1-1 WEB数据库应用示意图

1.1.2 实体关系模型

在数据库的设计阶段，需要创建逻辑模型。关系数据库的逻辑模型叫做实体—关系模型。实体模型化最常用的工具是实体关系图，简称E - R(Entity - Relationship)图，它是一种简单的图形技术，用来定义数据库中需要的表、字段和关系。它用于数据库设计的第一步，与我们使用的具体的数据库管理系统无关。ER图有如下优点：

- ☐ 有效地搜集和表示组织的信息需求。
- ☐ 提供一个容易理解的系统描述图。
- ☐ 易于开发和提炼。
- ☐ 明确定义了信息需求的范围。
- ☐ 将业务需求信息与业务执行活动分开。

根据业务说明或描述创建实体关系图。



概念模型-- ER模型基本概念

是现实世界到机器世界的一个中间层次，概念模型中最常用的是ER模型，介绍ER模型（实体联系）中的主要概念。

实体（Entity）：客观存在并可以相互区分的事物叫实体。（例如：一个个学生、一辆辆轿车）

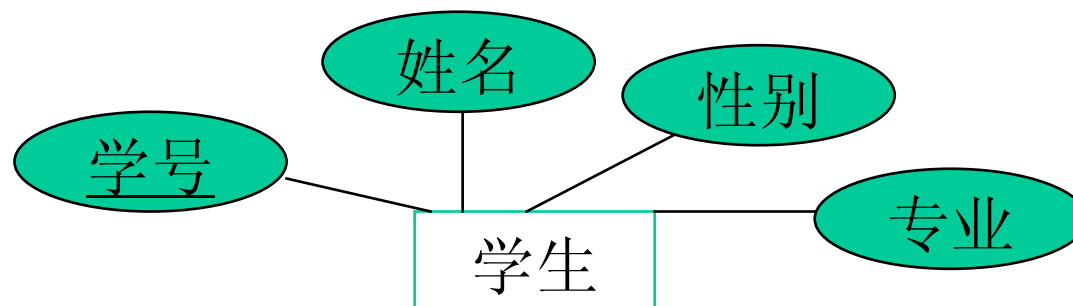
属性（Attribute）：实体一般具有若干特征，称之为实体的属性。例如：学生具有学号、姓名等属性。

域（Domain）：一个属性可能取值的范围称为这个属性的域。

ER模型基本概念

码（Key）：能够唯一标识实体的属性或最小属性组称为码（关键字），可能存在多个候选码，设计者必须指明一个做主码。例如：见图

实体型（Entity type）：具有相同属性的实体具有共同的特征和性质，用实体名及其属性集合来抽象、刻画同类实体，称为实体型。



ER模型基本概念

实体集 (Entity set)：同型实体的集合

联系：现实世界的事物之间是有联系的，这种联系在信息世界中反映为：实体(型)内部的联系和实体(型)之间的联系。

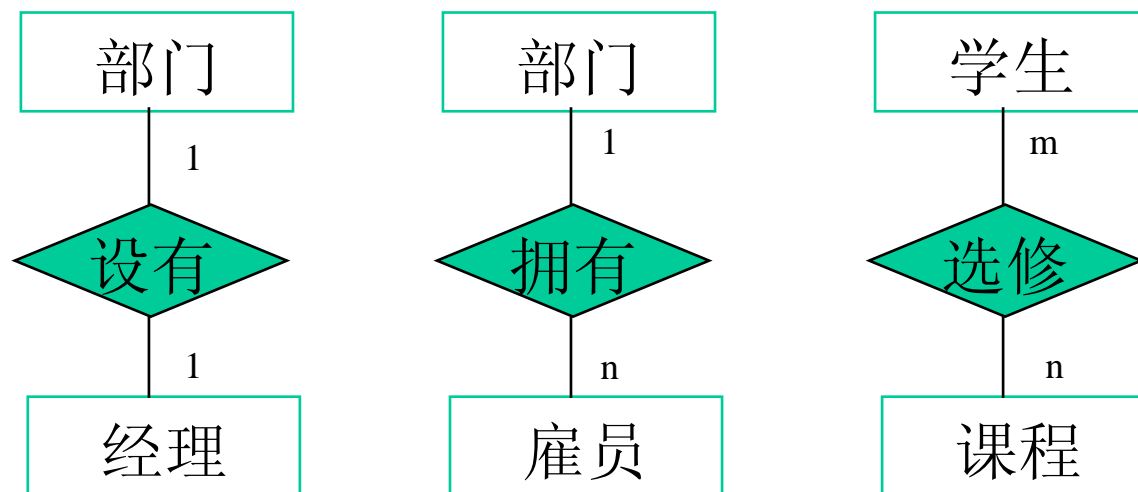
两个实体型之间的联系

一对一联系 (1: 1) 例如：部门、经理

一对多联系 (1: n) 例如：部门、雇员

多对多联系 (m: n) 例如：学生、课程

ER模型基本概念



规范化的意义在于可以从实体中删除冗余信息，通过修改数据模型达到可以惟一地表示实体的每一种情况为止。

规范化是降低或消除数据库中冗余数据的过程。尽管在大多数的情况下冗余数据不能被完全清除，但冗余数据降得越低，就越容易维护数据的完整性，并且可以避免非规范化的数据库中数据的更新异常。数据库的规范化通过范式来验证，但是一味地考虑满足范式，也会对数据库性能产生影响，并给实际的实施带来困难。所以实际的情况是采取折衷的方法。

规范化设计的规则有三个，分别称作第一范式、第二范式和第三范式：

第一范式(1NF): 实体的所有属性必须是单值的并且不允许重复。

第二范式(2NF): 实体的所有属性必须依赖于实体的惟一标识。

第三范式(3NF): 一个非惟一标识属性不允许依赖于另一个非惟一标识属性。

在数据库的设计中,一般都采用第三范式,以保证数据的冗余最小,提高数据的完整性。

图1-4是实际设计的两张表和表间关系的示意图。教师表由教师ID、姓名和系部ID等列构成；系部表由系部ID、系部名称和地点等列构成。其中，教师ID和系部ID分别是这两张表的主键。教师表的系部ID和系部表的系部ID之间建立了外键联系，即教师表的系部ID必须是系部表的某个系部ID。

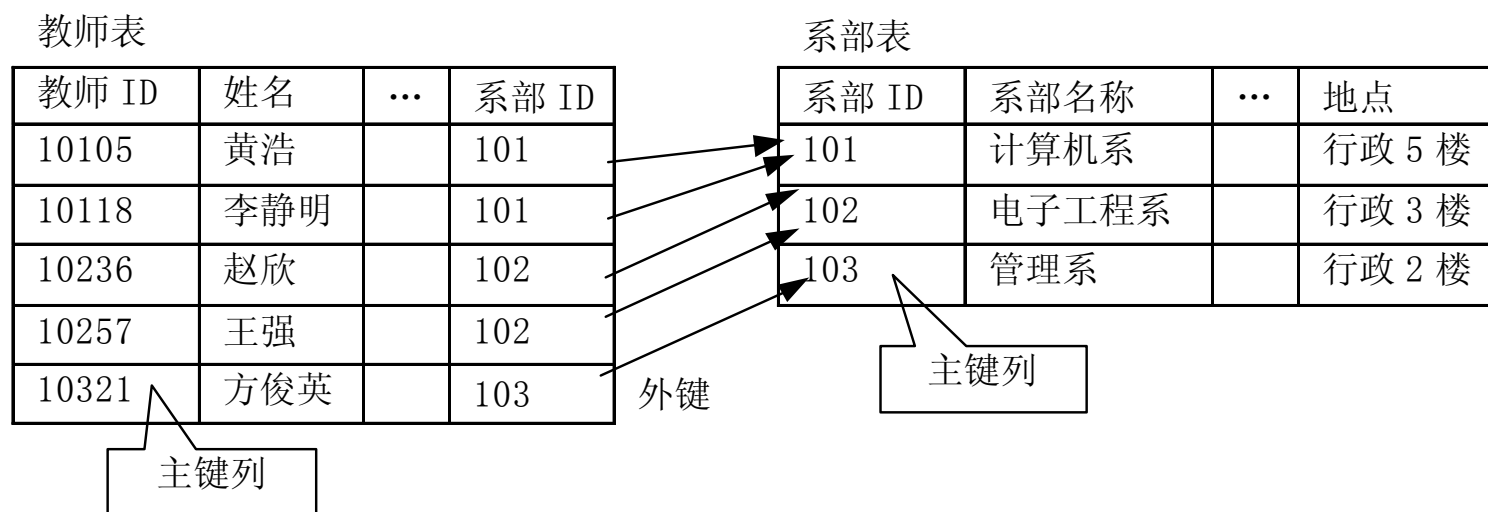


图1-4 表的结构和表间关系示意

1.1.5 开发数据库应用系统的步骤

下面列出了常见的数据库应用系统的开发步骤：

- ☐ 系统需求分析。
- ☐ 设计数据库表。
- ☐ 规划表中的字段。
- ☐ 确定表与表之间的关系。
- ☐ 优化表和表中字段的设计。
- ☐ 输入数据，检测表的设计，如果需要改进可以再次优化表的设计。

- ☐ 创建查询、存储过程、触发器以及其他的数据库对象。
- ☐ 使用数据库分析工具来分析和改进数据库的性能。
- ☐ 设置数据库安全性。

1.2 SQL*Plus环境

1.2.1 SQL*Plus的登录和环境设置

1. 登录SCOTT账户

在登录和使用SQL*Plus的同时，要以数据库用户的身份连接到某个数据库实例。在Oracle数据库创建过程中，选择通用目的安装，会创建一个用于测试和练习目的的账户——SCOTT。其中保存了一些数据库表的实例，主要的两个表是雇员表EMP和部门表DEPT通过登录SCOTT账户就可以访问这些表。

SCOTT账户的默认口令是TIGER。

我们假定Oracle数据库已经安装在局域网中的一台基于Windows操作系统的服务器上，服务器的名称为ORACLE，数据库实例的名称为MYDB。管理客户端和开发工具安装在其他基于Windows操作系统的客户机上，并且该机器通过网络能够访问到Oracle数据库服务器。这时，我们就可以使用管理客户端中的SQL*Plus工具来进行登录了。登录前一般要由管理员使用Oracle的网络配置工具创建一个网络服务名，作为客户端连接名。为了方便记忆，连接名可以与数据库实例名相重。我们假定创建的网络连接服务名为MYDB，则登录过程如下。

【训练1】 使用SQL*Plus工作表，以SCOTT账户登录数据库。

步骤1：启动SQL*Plus。在开始菜单中，找到Oracle菜单目录的“Application Development”子菜单，找到其下的“SQL*Plus WorkSheet”命令。

步骤2：为其在桌面上创建一个快捷方式并启动，出现如图1-5所示的登录界面。

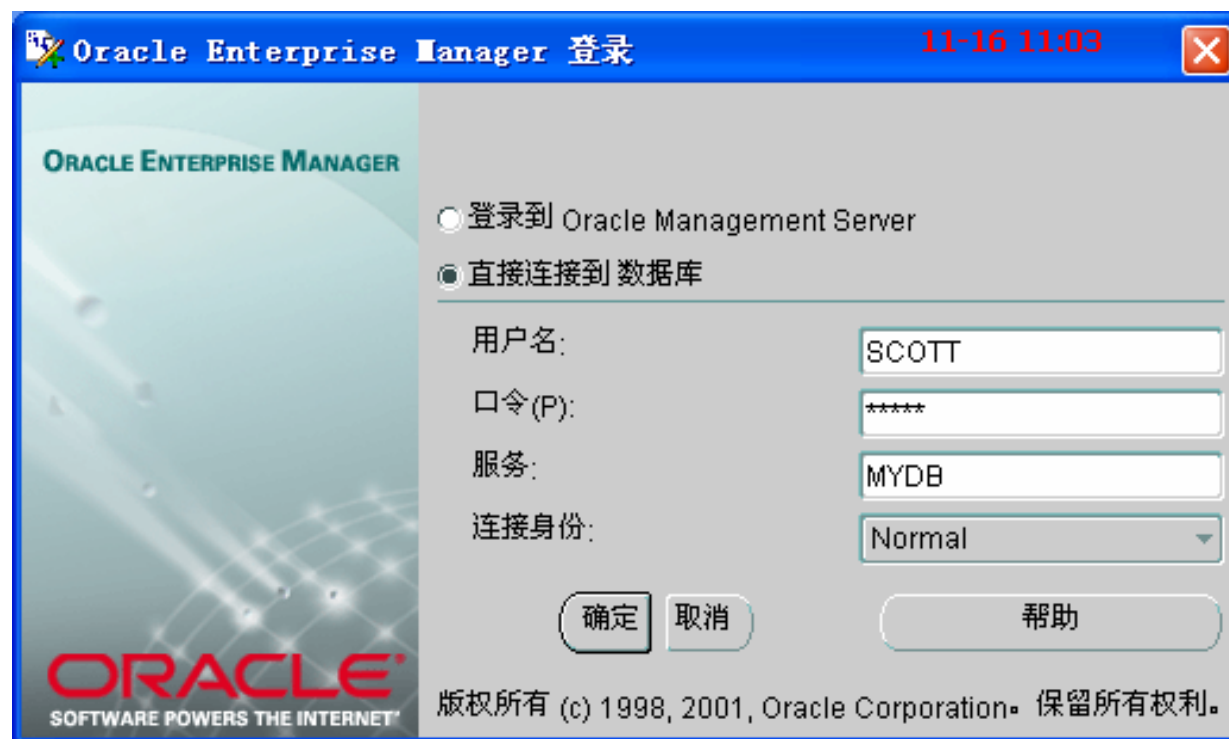


图1-5 SQL*Plus的登录对话框

步骤3：在登录对话框中选择直接连接到数据库，并输入其他必要的参数。

用户名为SCOTT。

口令为TIGER。

服务为MYDB，其中MYDB为由系统管理员创建的网络服务名。

连接身份选为Normal。

点击“确定”按钮即可进行数据库连接了。

在输出区输出结果为：

已连接。

连接成功后，出现如图1-6所示的SQL*Plus工作表的工作窗口，在输出区的信息“已连接”表示数据库连接成功。如果显示登录失败信息，则需要重新检查输入的连接参数是否正确。



图1-6 SQL*PLUS工作表

窗口界面可划分成如下几个区域：菜单区、按钮区、输入区和输出区。通过拖动输入区和输出区中间的分隔线可以调整两个区的大小。输入区为一文本编辑区，可以在其中进行命令的输入和编辑，可以使用通用的文本编辑命令，如“选择”、“剪切”、“复制”和“粘贴”等进行操作。在输入区中可以输入SQL命令或PL/SQL程序。输出区为一只读文本区，显示命令的输出结果。

SQL*Plus可以同时运行多个副本，连接相同或不同的账户，同时进行不同的操作。

Oracle数据库的很多对象，都是属于某个模式(Schema)的，模式对应于某个账户，如SCOTT模式对应SCOTT账户。往往我们对模式和账户不做区分。数据库的表是模式对象中的一种，是最常见和最基本的数据库模式对象。一般情况下，如果没有特殊的授权，用户只能访问和操作属于自己的模式对象。比如以SCOTT账户登录，就只能访问属于SCOTT模式的表。所以通过以不同的用户身份连接，可以访问属于不同用户模式的表。

如果需要重新连接另外一个账户，可以点击“连接”按钮，则重新出现连接对话框，在该对话框中输入新的账户名、口令和其他参数进行连接即可。任何时刻，如果需要运行输入区中的命令，可以点击“执行”按钮。用户可以在输入区中输入和编辑任何命令，在编辑完成后通过点击“执行”按钮(或按F5快捷键)来执行输入区中的命令脚本。

还有一种以命令方式进行重新连接的方法更为便捷，重新进行连接的命令是CONNECT。

以下是该方法的训练。

【训练2】 输入和执行CONNECT命令重新连接数据库。

步骤1：在输入区域清除原有命令，输入新命令。

```
CONNECT SCOTT/TIGER@MYDB
```

步骤2：点击“执行”按钮(或按F5快捷键)执行该命令。

显示结果为：

已连接。

说明：SCOTT为账户名，TIGER为口令，账户名和口令之间用“/”分隔。“@”后面的字符串称为网络服务名或称为连接字符串。

注意：以上方法的口令是显式的，容易被其他人窃取。

要关闭或退出SQL*Plus，可以在输入区域直接输入“EXIT”或“QUIT”命令并执行，或执行“文件”菜单下的“退出”命令。

2. 环境设置命令

在SQL*Plus环境下，可以使用一系列的设置命令来对环境进行设置。如果不进行设置，系统会使用默认值。通过SHOW ALL命令可以查看SQL*Plus的环境参数。设置命令的格式为

SET 参数 [ON|OFF|值]

通常需要对输出的显示环境进行设置，这样可以达到更理想的输出效果。显示输出结果是分页的，默认的页面大小是14行×80列。以下的训练是设置输出页面的大小，用户可以比较设置前后的输出效果。

【训练3】 设置输出页面的大小。

步骤1：输入并执行以下命令，观察显示结果：

```
SELECT * FROM emp;
```

步骤2：在输入区输入并执行以下命令：

```
SET PAGESIZE 100
```

```
SET LINESIZE 120
```

或

```
SET PAGESIZE 100 LINESIZE 120
```

步骤3：重新输入并执行以下命令，观察显示结果：

```
SELECT * FROM emp;
```

说明：命令SET PAGESIZE 100将页高设置为100行，命令SET LINESIZE 120将页宽设置为120个字符。通过页面的重新设置，消除了显示的折行现象。SELECT语句用来对数据库的表进行查询，这将在后面介绍。

如果用户忘记了自己是以什么用户身份连接的，可以用以下的命令显示当前用户。

【训练4】 显示当前用户。

输入并执行命令：

```
SHOW USER
```

执行结果是：

USER为“SCOTT”

说明：显示的当前用户为SCOTT，即用户是以SCOTT账户登录的。

注意：使用SELECT USER FROM dual命令也可以取得用户名。

通过进行适当的设置，可以把操作内容或结果记录到文本文件中。

【训练5】 使用SPPOOL命令记录操作内容。

步骤1：执行命令：

```
SPPOOL C:\TEST
```

步骤2：执行命令：

```
SELECT * FROM emp;
```

步骤3: 执行命令:

```
SELECT * FROM dept;
```

步骤4: 执行命令:

```
SPOOL OFF
```

步骤5: 用记事本打开C:\TEST.LST并查看内容:

```
OEM_sqlplus_input_finished
```

```
SELECT * FROM emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
OMM	DEPTNO				

```
-----
```

```
-----
```

7369	SMITH	CLERK	7902	17-12月-80	1560
		20			
7499	ALLEN	SALESMAN	7698	20-2月 -81	1936
	300		30		
7521	WARD	SALESMAN	7698	22-2 月	-81
	1830	500		30	
7566	JONES	MANAGER	7839	02-4月 -81	2975
		20			
7654	MARTIN	SALESMAN	7698	28-9 月	-81
	1830	1400		30	
7698	BLAKE	MANAGER	7839	01-5 月	-81
	2850		30		

7782	CLARK	MANAGER	7839	09-6	月	-81
	2850			10		
7839	KING	PRESIDENT		17-11	月-81	
	5000			10		
7844	TURNER	SALESMAN	7698	08-9	月	-81
	1997	0		30		
7876	ADAMS	CLERK	7788	23-5	月	-87
	1948			20		
7900	JAMES	CLERK	7698	03-12	月	-81
	1852			30		

7788 SCOTT	ANALYST	7566	19-4 月	-87
3000		20		
7902 FORD	ANALYST	7566	03-12 月	-81
3000		20		
7934 MILLER	CLERK	7782	23-1 月	-82
1903		10		

已选择14行。


```
OEM_sqlplus_input_finished
```

```
SELECT * FROM dept;
```

DEPTNO	DNAME	LOC
--------	-------	-----

-----	-----	-----
-------	-------	-------

--	-----	
----	-------	--

10	ACCOUNTING	NEW YORK
----	------------	----------

20	RESEARCH	DALLAS
----	----------	--------

30	SALES	CHICAGO
----	-------	---------

40	OPERATIONS	BOSTON
----	------------	--------

已选择4行。

说明：以上步骤将输入的命令和输出的结果记录到C盘根目录下的TEST.LST文件中，内容如上所示。SPOOL OFF命令用来关闭记录过程。

可以使用这种方法对操作进行磁盘记录。

还有如下环境设置命令，在这里不做详细介绍：

- ☐ Set heading on/off: 打开/关闭查询结果表头的显示，默认为ON。
- ☐ Set feedback on/off: 打开/关闭查询结果中返回行数的显示，默认为ON。
- ☐ Set echo on/off: 打开/关闭命令的回显，默认为ON。
- ☐ Set time on/off: 打开/关闭时间显示，默认为OFF。

我们可以将一系列的SET命令存入BEGIN.SQL，并放在SQL*Plus启动文件的同一个目录下，这样就可以在启动时自动地进行设置了。SQL*Plus启动文件的目录在Oracle主目录的BIN目录下。

【练习1】 关闭表头和返回结果行数的显示，然后再打开。

1.2.2 认识表的结构

SCOTT账户拥有若干个表，其中主要有一个EMP表，该表存储公司雇员的信息，还有一个DEPT表，用于存储公司的部门信息。表是用来存储二维信息的，由行和列组成。行一般称为表的记录，列称为表的字段。要了解一个表的结构，就要知道表由哪些字段组成，各字段是什么数据类型，有什么属性。要看表的内容，就要通过查询显示表的记录。

ORACLE常用的表字段数据类型有：

- ☐ CHAR: 固定长度的字符串，没有存储字符的位置，用空格填充。
- ☐ VARCHAR2: 可变长度的字符串，自动去掉前后的空格。

☐ NUMBER (M, N): 数字型, M是位数总长度, N是小数的长度。

☐ DATE: 日期类型, 包括日期和时间在内。

☐ BOOLEAN: 布尔型, 即逻辑型。

可以使用DESCRIBE命令 (DESCRIBE可简写成DESC) 来检查表的结构信息。

1. 雇员表EMP的结构

以下训练显示emp表的结构。

【训练1】 显示EMP表的结构。

输入并执行以下命令(emp为要显示结构的表名)：

```
DESCRIBE emp
```

输出区的显示结果如下：

名称	是否为空?	数据类型
----	-------	------

EMPNO	NOT NULL	NUMBER(4)
ENAME		
		VARCHAR2(10)
JOB		
		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		
		NUMBER(7, 2)
COMM		
		NUMBER(7, 2)
DEPTNO		
		NUMBER(2)

说明：以上字段用到了3种数据类型：数值型、字符型和日期型，都是常用的数据类型。

列表显示了字段名、字段是否可以为空、字段的数据类型和宽度。在是否为空域中的“NOT NULL”代表该字段的内容不能为空，即在插入新记录时必须填写；没有代表可以为空。括号中是字段的宽度。日期型数据是固定宽度，无需指明。该表共有8个字段，或者说有8个列，各字段的名称和含义解释如下：

EMPNO是雇员编号，数值型，长度为4个字节，不能为空。

ENAME是雇员姓名，字符型，长度为10个字节，可以为空。

JOB是雇员职务，字符型，长度为9个字节，可以为空。

MGR是雇员经理的编号，数值型，长度为4个字节，可以为空。

HIREDATE是雇员雇佣日期，日期型，可以为空。

SAL是雇员工资，数值型，长度为7个字节，小数位有2位，可以为空。

COMM是雇员津贴，数值型，长度为7个字节，小数位有2位，可以为空。

DEPTNO是雇员所在的部门编号，数值型，长度为2个字节的整数，可以为空。

2. 部门表DEPT的结构

以下训练显示DEPT表的结构。

【训练2】 显示部门表DEPT的结构。

输入以下的命令：

```
DESCRIBE dept
```

结果为：

名称	是否为空?	类
型		

DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

说明：以上字段用到了2种数据类型：数值型和字符型。DEPT表共有3个字段：

DEPTNO代表部门编号，数值型，宽度为2个字节，不能为空。

DNAME代表部门名称，字符型，长度为14个字节，可以为空。

LOC代表所在城市，字符型，长度为13个字节，可以为空。

1.2.3 显示表的内容

已知表的数据结构，还要通过查询命令来显示表的内容，这样就可以了解表的全貌。显示表的内容用查询语句进行。

1. 雇员表EMP的内容

【训练1】 显示EMP表的全部记录。

步骤1：输入并执行以下命令：

```
SELECT * FROM emp;
```

在输出区将显示表的内容。

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
COMM	DEPTNO				

7369	SMITH	CLERK	7902	17-12月-80	800		20
7499	ALLEN	SALESMAN	7698	20-2月 -81	1600	300	30
7521	WARD	SALESMAN	7698	22-2月 -81	1250	500	30
7566	JONES	MANAGER	7839	02-4月 -81	2975		20
7654	MARTIN	SALESMAN	7698	28-9月 -81	1250	1400	30
7698	BLAK	MANAGER	7839	01-5月 -81	2850		30
7782	CLARK	MANAGER	7839	09-6月 -81	2450	10	
7788	SCOTT	ANALYST	7566	19-4月 -87	3000	20	

7839	KING	PRESIDENT	17-11月-81	5000		10	
7844	TURNER	SALESMAN		7698	08-9月 -81	1500	0 30
7876	ADAMS	CLERK		7788	23-5月 -87	1100	20
7900	JAMES	CLERK		7698	03-12月-81	950	30
7902	FORD	ANALYST		7566	03-12月-81	3000	20
7934	MILLER	CLERK		7782	23-1月 -82	1300	10

已选择14行。

说明：观察表的内容，在显示结果中，虚线以上部分(第一行)称为表头，是EMP表的字段名列表。该表共有8个字段，显示为8列。虚线以下部分是该表的记录，共有14行，代表14个雇员的信息。如雇员7788的名字是SCOTT，职务为ANALYST，.....。

这个表在下面的练习中要反复使用，必须熟记字段名和表的内容。

2. 部门表DEPT的内容

【训练2】 显示DEPT表的全部记录。

输入并执行以下查询命令：

```
SELECT * FROM dept;
```

执行结果为：

DEPTNO	DNAME	LOC

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

说明：该表中共有3个字段：部门编号DEPTNO、部门名称DNAME和所在城市LOC。该表共有4个记录，显示出4个部门的信息，如部门10的名称是ACCOUNTING，所在城市是NEW YORK。

这个表在下面的练习中要反复使用，必须熟记字段名和表的内容。

【练习1】根据EMP表和DEPT表的显示结果，说出雇员ADAMS的雇员编号、职务、经理名字、雇佣日期、工资、津贴和部门编号以及该雇员所在的部门名称和所在城市。

【练习2】说出职务为CLERK的工资最高的雇员是哪一位？职务为CLERK、部门在NEW YORK的雇员是哪一位？

1.2.4 SQL*Plus环境的使用

在SQL*Plus环境下，命令可以在一行或多行输入，命令是不分大小写的。SQL命令一般要以“;”结尾。

可以在输入内容中书写注释，或将原有内容变成注释。注释的内容在执行时将被忽略。

注释的方法是：

- * 在一行的开头处书写REM，将一行注释掉。
- * 在一行中插入“--”，将其后的内容注释掉。
- * 使用/*..... */，可以用来注释任何一段内容。

【训练1】 使用注释。

在输入区输入以下内容，按F5执行。

REM 本句是注释语句

--SELECT * FROM emp;该句也被注释

执行后没有产生任何输出。

说明：REM和“--”产生注释作用，语句不执行，所以没有输出。注释后的内容将变成红色显示。

如果需要的话，可以分别将输入区或输出区的内容以文本文件的形式存盘，供以后查看或重新使用。

【训练2】 保存输入区的内容。

步骤1：在输入区重新输入命令：

```
SELECT * FROM emp;
```

```
SELECT * FROM dept;
```

步骤2：选择“文件”菜单下的“将输入另存为”命令，弹出文件存盘对话框。选择正确的磁盘位置，为存盘的文件起一个名字。如果输入区中的内容是SQL命令或命令序列，则使用扩展名.SQL；否则可以省略或以.TXT做扩展名。在本例中，选SELECT.SQL做文件名。

步骤3：按“保存”按钮，将输入区的内容存入磁盘文件。

步骤4：用记事本查看保存的内容(略)。

说明：用同样的方法可以保存输出区的内容。

【练习1】请试着将输出区的内容存入文件RESULT.TXT。

输出区中每一条命令的执行结果都将出现在其中，通过滚动条来显示屏幕之外的内容。当输出区的内容很多，显示混乱时，可以清除区域中的内容。

【训练3】 清除输出区域的显示内容。

将光标置入输出区，执行“编辑”菜单的“全部清除”命令。
结果输入区的内容被清除。

说明：以上菜单命令将清除输出区的全部显示内容。使用同样的方法可以清除输入区的全部内容。

如果某些命令已经存盘，特别是比较长和复杂的命令或命令序列，可以重新调入输入区或直接执行。

【训练4】 调入磁盘文件执行。

步骤1：执行“文件”菜单的“打开”命令，弹出打开文件对话框。

步骤2：选择刚刚存盘的SELECT.SQL文件，按“打开”按钮，将存盘的文件装入输入区。

步骤3：按F5执行该命令。

结果从略。

执行“工作单”菜单下的“运行本地脚本”命令，可以直接运行存盘文件中的SQL命令，请做如下练习。

【练习2】 直接执行SELECT. SQL文件中的语句。

每次用户执行过的命令将存储在内存的“命令历史记录区”中，直到退出SQL*Plus环境。用户可以使用“工作单”菜单下的“命令历史记录”、“上一条”或“下一条”命令（或按Ctrl+H/Ctrl+P/Ctrl+N快捷键）调出执行过的命令脚本。

【练习3】 通过快捷键CTRL+P和CTRL+N调出前一条和后一条命令到输入区。

1.3 Oracle的应用开发工具TOAD

Oracle有很多开发工具可以选用，TOAD 是Quest Software公司的具有图形界面的轻量级开发工具，是Oracle应用开发者工具(Tools for Oracle Application Developer)的缩写。它比SQL*Plus有更多的功能，除了可以自动格式化和执行SQL语句以及支持PL/SQL程序的编程和调试之外，它的数据库对象浏览功能能够让我们直观地看到数据库的模式对象，并进行直观的操作。TOAD提供60天的免费使用版本，在功能上比商业版有所限制，使用到期后需要重新下载和安装。

TOAD的下载网址是<http://www.toadsoft.com>。

TOAD可以用菜单或图标按钮的方式进行操作。在启动后的界面中，它可以建立多个数据库连接，打开多个SQL工作窗口，用于执行SQL语句和PL/SQL程序；TOAD还提供了单独的存储过程编辑调试窗口，用来编辑、调试和执行存储过程、函数和包；还可以打开多个对象浏览窗口，用来观察用户的模式对象信息，如表、视图、约束条件和存储过程等。以上TOAD的主要功能就能够满足本课程的需要。TOAD还有许多功能，这里就不一一介绍了。

图1-7是TOAD的启动登录界面，在输入正确的连接字符串、账户名和口令后，点击“OK”按钮就可以登录了。



图1-7 TOAD的登录对话框

【训练1】 登录TOAD，执行简单查询。

步骤1：启动TOAD，在启动对话框中输入以下登录信息：

用户名：SCOTT

口令：TIGER

服务：MYDB

其中，MYDB为以前创建好的数据库连接字符串。

按确定按钮登录，登录后出现主窗口(如图1-8所示)。TOAD是个多用户窗口界面，可以根据需要在工作区中打开多个子窗口，同时进行操作。



图1-8 TOAD的主界面

步骤2：每次按打开SQL工作窗口按钮都可以打开一个SQL窗口，可测试不同的SQL语句。

在第一个窗口中输入以下查询语句，并按执行按钮：

```
SELECT * FROM emp;
```

在结果区以表格形式显示查询结果，结果如图1-9所示。

步骤3：打开第二个SQL工作窗口，在其中输入另外的查询语句并执行：

```
SELECT * FROM dept;
```

结果从略。

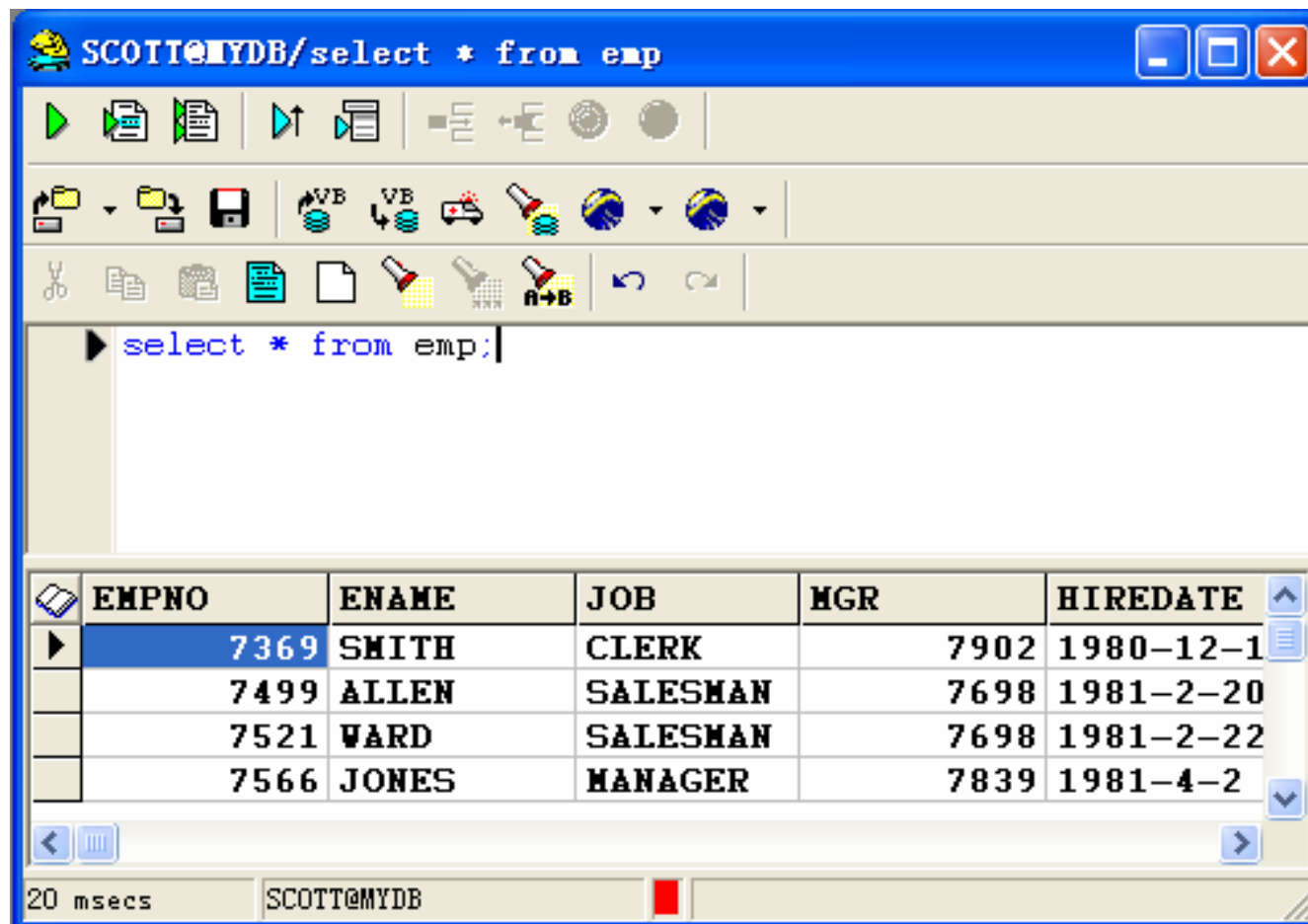


图1-9 SQL子窗口

1.4 操作准备

为了完成后续的练习，需要为每一个练习者准备一个账户，将数据复制到该账户下。先由教师或管理员创建一个公用账户STUDENT，并授予该账户创建用户、创建表等必要权限，通过该账户创建新用户并传递权限给新账户。其次，教师或管理员负责登录SCOTT账户，将EMP和DEPT等表的访问权限(SELECT)授予所有用户(PUBLIC)。

以下是创建新用户的脚本，使用前应由教师或管理员准备好账户STUDENT，该账户的口令为STDUENT。下面以创建USER1账户为例，完成准备工作。

【训练1】 创建新用户USER1，并登录和复制数据。

步骤1：登录STUDENT账户。

```
CONNECT STUDENT/STUDENT@MYDB
```

执行结果：

已连接。

步骤2：创建新用户USER1，口令为ABC123，口令需要以字母开头。

```
CREATE USER USER1 IDENTIFIED BY ABC123
```

执行结果：

用户已创建。

其中，用户名为USER1，口令为ABC123。

步骤3：授予连接数据库权限、创建表权限、创建存储过程和表空间使用权限。

```
GRANT CONNECT TO USER1;
```

```
GRANT CREATE TABLE TO USER1;
```

```
GRANT CREATE PROCEDURE TO USER1;
```

```
GRANT UNLIMITED TABLESPACE TO USER1;
```

执行结果：

授权成功。

其他权限在必要时再添加。

步骤4：使用新账户登录。

```
CONNECT USER1/ABC123@MYDB
```

输出结果：

已连接。

步骤5：以创建表的方式复制数据到新账户。

```
CREATE TABLE EMP AS SELECT * FROM SCOTT.EMP;
```

```
CREATE TABLE DEPT AS SELECT * FROM SCOTT.DEPT;
```

```
CREATE TABLE SALGRADES AS SELECT * FROM SCOTT.SALGRADE;
```

输出结果：

表已创建。

说明：执行完以上脚本后，在USER1账户下复制了SCOTT账户的三个表：EMP、DEPT和SALGRADES。

【练习1】显示当前的账户名，显示当前账户的EMP表的结构，显示EMP表中的数据。



1.5 阶段训练

【练习1】用SQL*Plus登录新创建的用户，熟悉环境设置命令。

【练习2】使用TOAD登录新创建的用户，执行简单的查询。



1.6 练习

1. 以下不是数据库特点的是:
 - A. 高速数据传输
 - B. 较高的数据独立性
 - C. 较小的数据冗余
 - D. 多用户数据共享
2. Oracle数据库属于以下哪种类型:
 - A. 网状数据库
 - B. 层次数据库
 - C. 关系数据库
 - D. 面向对象关系数据库

3. Oracle自带的SQL语言环境称为:

- A. SQL
- B. PL/SQL
- C. SQL*Plus
- D. TOAD

4. 显示登录的用户名, 可以用的命令是:

- A. DESCRIB user
- B. SELECT user
- C. SHOW user
- D. REM user

5. 可变长度的字符串类型, 用以下的哪个关键字表示:

- A. CHAR
- B. VARCHAR2
- C. BOOLEAN
- D. NUMBER