

1

编写基本的 SQL SELECT 语句

中国科学院西安网络中心 编译 2005

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

进度表:	时间	主题
	40 分钟	讲演
	25 分钟	练习
	65 分钟	总共

目标

完成本课后，您应当能够：

- 列出 SQL **SELECT** 语句的功能
- 执行基本的 **SELECT** 语句
- 区分 SQL 语句和 *iSQL*Plus* 命令

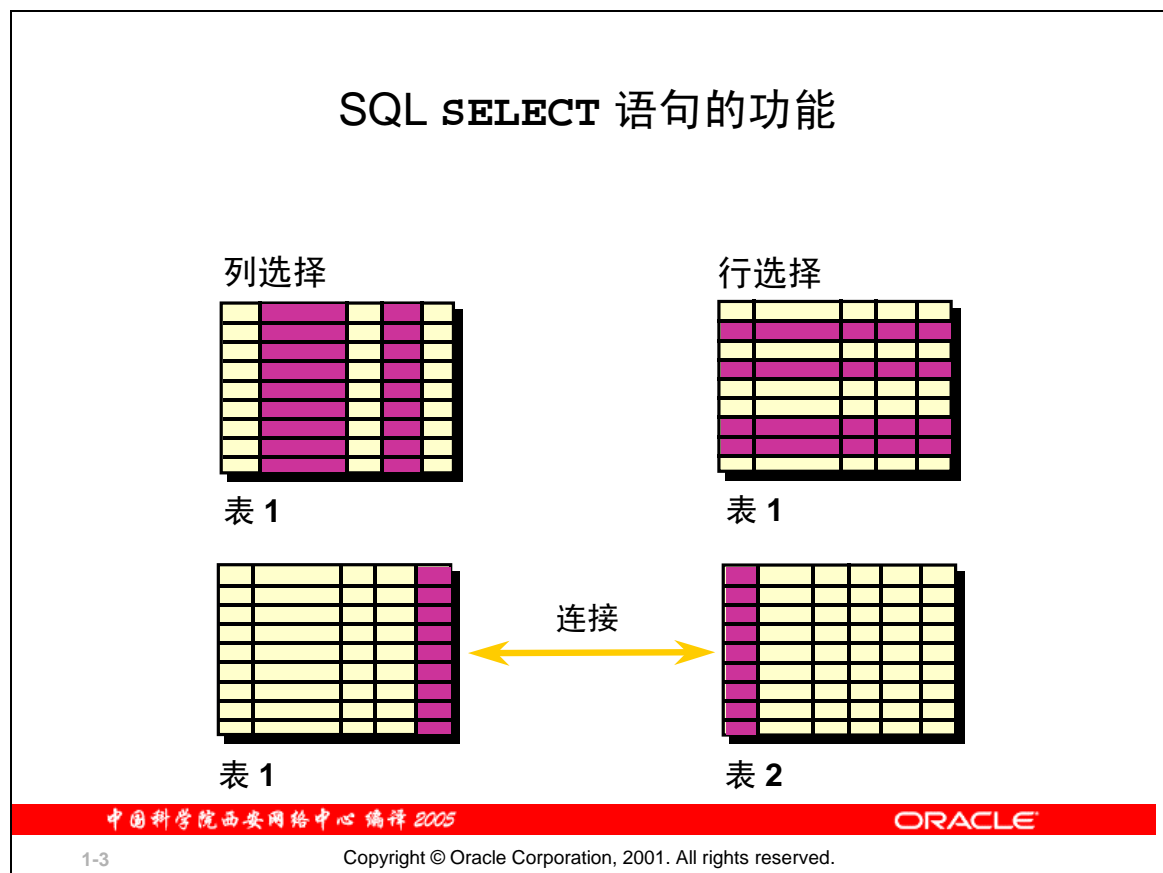
课程目标

为了从数据库中摘录数据，你需要使用结构化查询语言 (Structured Query Language SQL) 的 **SELECT** 语句，你还可能需要限制所显示的行和列。本课程描述了用于执行这个任务所需要的所有 SQL 语句。

你也能够创建可以多次使用的 **SELECT** 语句。

本课中还包括用来执行 SQL 语句的 *iSQL*Plus* 环境的使用介绍。

注：*iSQL*Plus* 是在 Oracle9i 产品中新增的，并在 Oracle10g 中得到增强，它是一个执行 SQL 命令的浏览器环境。Oracle 以前发布的 SQL*Plus 是执行 SQL 命令的默认的环境。SQL*Plus 现在任然可用，并且在附录 C 中讲述。



SQL SELECT 语句的功能

SELECT 语句从数据库中返回信息。使用一个 SELECT 语句，你可以做下面的事：

- **列选择：**你能够使用 SELECT 语句的列选择功能选择表中的列，这些列是你想要用查询返回的。当你查询时，你能够选择你查询的表中指定的列。
- **行选择：**你能够使用 SELECT 语句的行选择功能选择表中的行，这些行是你想要用查询返回的。你能够使用不同的标准限制你看见的行。
- **连接：**你能够使用 SELECT 语句的连接功能来集合数据，这些数据被存储在不同的表中，在它们之间可以创建连接。在后面的课程中你将学到更多关于连接的内容。

教师注释

告诉学生行选择和列选择通常被认为是水平和垂直分割。

基本 SELECT 语句

```
SELECT    *|{[DISTINCT] column|expression [alias],...}  
FROM      table;
```

- **SELECT** 确定哪些列
- **FROM** 确定哪张表

基本 SELECT 语句

在最简单的形式中，SELECT 语句必须包含下面的内容：

- 一个 SELECT 子句，指定被显示的列
- 一个 FROM 子句，指定表，该表包含 SELECT 子句中的字段列表

在语法中：

SELECT	是一个或多个字段的列表
*	选择所有的列
DISTINCT	禁止重复
column expression	选择指定的字段或表达式
alias	给所选择的列不同的标题
FROM table	指定包含列的表

注：在本课程中，单词 *keyword*（关键字），*clause*（子句）和 *statement*（语句）被如下使用：

- *keyword* 引用一个单个的 SQL 元素。
例如，SELECT 和 FROM 是关键字。
- *clause* 是 SQL 语句的一个部分。
例如，SELECT employee_id, last_name, ... 是一个子句。
- *statement* 是两个或多个子句的组合。
例如，SELECT * FROM employees 是一个 SQL 语句。

选择所有列

```
SELECT *  
FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

选择所有行的所有列

用跟在 SELECT 关键字后面的星号 (*), 你能够显示表中数据的所有列。在幻灯片上的例子中, department 表包含 4 个列: DEPARTMENT_ID、DEPARTMENT_NAME、MANAGER_ID 和 LOCATION_ID。该表包含 8 行, 每个部门一行。

在 SELECT 关键字后面列出所有列, 你也能够显示所有的列。例如, 跟着 SQL 语句, 像幻灯片中的例子, 显示 DEPARTMENTS 表的所有列和所有行:

```
SELECT department_id, department_name, manager_id,  
location_id  
FROM departments;
```

教师注释

让学生知道在附录 B 中给出的所有表的细节。

选择指定的列

```
SELECT department_id, location_id
FROM departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

8 rows selected.

选择所有列中指定列

你能够用 SELECT 语句来显示表的指定列，指定列名之间用逗号分隔。幻灯片中例子显示 DEPARTMENTS 表中所有的部门号和位置号。

在 SELECT 子句中，指定你想要的列，其顺序是你想要在输出中呈现的。例如，为了从左到右在部门号之前显示位置号，你可以用下面的语句：

```
SELECT location_id, department_id
FROM departments;
```

LOCATION_ID	DEPARTMENT_ID
1700	10
1800	20
1500	50

8 rows selected.

教师注释

你也能够从伪列中选择。一个伪列的行为像表中的列一样，但不是真的存储在表中。你不能插入或删除伪列的值。一些可用的伪列是 CURRVAL、NEXTVAL、LEVEL、ROWID 和 ROWNUM。

写 SQL 语句

- SQL 语句对大小写不敏感
- SQL 语句可以写成一行或多行
- 关键字不能简写或分开折行
- 子句通常放在不同的行
- 缩进用于增强可读性

写 SQL 语句

用下面的简单规则和原则，你能够构造正确的语句，它们即容易阅读也容易编辑：

- SQL 语句对大小写不敏感，除非指出。
- SQL 语句可以写成一行或多行。
- 关键字不能简写或分开折行。
- 为了易于阅读和编辑，子句通常放在不同的行。
- 应该使用缩进，以使代码更易阅读。
- 关键字一般用大写输入；所有其它的名字和列，用小写输入。

执行 SQL 语句

使用 iSQL*Plus，单击 执行 按钮来运行编辑窗口中的命令。

教师注释

尽管在 iSQL*Plus 中，在子句的最后，分号 (;) 不是必须的，但还是推荐使用。在其它环境中，例如，PL/SQL 程序，需要在每条语句的结尾加一个分号。

列标题默认属性

- **iSQL*Plus:**
 - 默认列标题位置: 居中
 - 默认列标题显示: 大写
- **SQL*Plus:**
 - 字符和日期列标题位置居左
 - 数字列标题居右
 - 默认列标题显示: 大写

列标题默认属性

在 *iSQL*Plus* 中, 列标题用大写字母居中显示。

```
SELECT last_name, hire_date, salary
FROM   employees;
```

LAST_NAME	HIRE_DATE	SALARY
King	17-JUN-87	24000
Kochhar	21-SEP-89	17000
De Haan	13-JAN-93	17000
Hunold	03-JAN-90	9000
Ernst	21-MAY-91	6000
Higgins	07-JUN-94	12000
Gietz	07-JUN-94	8300

20 rows selected.

你能够用别名覆盖列标题显示, 列别名在本课稍后讲述。

算术表达式

用算术运算符创建数字和日期数据的表达式

操作	说明
+	加
-	减
*	乘
/	除

算术表达式

你可能需要修改数据显示方式，如执行计算，或者作假定推测，这些都可能用到算术表达式。一个算术表达式可以包含列名、固定的数字值和算术运算符。

算术运算符（Arithmetic Operators）

幻灯片中列出了 SQL 提供的算术运算符。除了 FROM 子句，你可以在任何一个 SQL 语句的子句中使用使用算术运算符。

教师注释

你只能对 DATE 和 TIMESTAMP 数据类型使用加和减操作。

使用算术运算符

```
SELECT last_name, salary, salary + 300  
FROM employees;
```

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300
...		
Hartstein	13000	13300
Fay	6000	6300
Higgins	12000	12300
Gietz	8300	8600

20 rows selected.

使用算术运算符

幻灯片中的例子用加法运算计算所有雇员的薪水增加\$300 元,并且在输出中显示一个新的列。

注意,合成的计算列 **SALARY+300** 不是 **EMPLOYEES** 表中的一个新列;显示的新列来自产生它的计算,在这里是 **salary+300**。

注: Oracle9i/10g 服务器忽略算术运算前后的空白。

运算符的优先级



- 乘法和除法比加法和减法的优先级高
- 相同优先级的运算符从左到右计算
- 圆括号用于强制优先计算，并且使语句更清晰

运算符优先级

如果算术表达式包含有一个以上的运算，乘法和除法先计算。如果在一个表达式中的运算符优先级相同，计算从左到右进行。

你可以用圆括号强制其中的表达式先计算。

运算符的优先级

```
SELECT last_name, salary, 12*salary+100
FROM   employees;
```

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100
Hunold	9000	108100
Ernst	6000	72100
...		
Hartstein	13000	156100
Fay	6000	72100
Higgins	12000	144100
Gietz	8300	99700

20 rows selected.

运算符的优先级 (续)

幻灯片中的例子显示名字、薪水和雇员的年薪。年薪的计算是 12 乘以月工资，再加一次性的 100 元奖金。注意乘法比加法先执行。

注：使用圆括号可以增强标准顺序，并且改善可读性。例如，幻灯片中的表达式可以写成 $(12 * salary) + 100$ 而不会改变结果。

教师注释

示范: 1_prec1.sql, 1_prec2.sql

目的: 举例说明无圆括号的查询，再用圆括号覆盖优先级规则。

使用圆括号

```
SELECT last_name, salary, 12*(salary+100)
FROM employees;
```

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200
Hunold	9000	109200
Ernst	6000	73200
...		
Hartstein	13000	157200
Fay	6000	73200
Higgins	12000	145200
Gietz	8300	100800

20 rows selected.

使用圆括号

你可以用圆括号覆盖优先级规则来指定运算符执行的顺序。

幻灯片中的例子显示名字、薪水和雇员的年薪。年薪的计算是以月工资加每月的 100 元奖金，再乘以 12。因为有圆括号，加法的优先级就高于乘法。

定义空值

- null 是一个未分配的、未知的，或不适用的值
- null 不是 0，也不是空格

```
SELECT last_name, job_id, salary, commission_pct
FROM employees;
```

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	
Kochhar	AD_VP	17000	
...			
Zlotkey	SA_MAN	10500	.2
Abel	SA_REP	11000	.3
Taylor	SA_REP	8600	.2
...			
Gietz	AC_ACCOUNT	8300	

20 rows selected.

中国科学院西安网络中心 编译 2005

ORACLE

1-14

Copyright © Oracle Corporation, 2001. All rights reserved.

空值

如果一行中的某个列缺少数据值，该值被置为 *null*，或者说包含一个空。

空是一个难以获得的、未分配的、未知的，或不适用的值。空和 0 或者空格不相同。0 是一个数字，而空格是一个字符。

任何数据类型的列都可以包含空。可是，某些约束，如，NOT NULL 和 PRIMARY KEY，防止在列中使用空。

在 EMPLOYEES 表的 COMMISSION_PCT 列中，我们注意到只有销售经理或销售代表才有佣金，其他的雇员没有佣金，空表示了这个事实。

教师注释

示范: 1_null.sql

目的: 举例说明带空值的计算。

算术表达式中的空值

包含空值的算术表达式计算结果为空

```
SELECT last_name, 12*salary*commission_pct  
FROM employees;
```

LAST_NAME	12*SALARY*COMMISSION_PCT
King	
Kochhar	
...	
Zlotkey	25200
Abel	39600
Taylor	20640
...	
Gietz	

20 rows selected.

中国科学院西安网络中心 编译 2005

ORACLE

空值 (续)

如果在一个算术表达式中的列值为空，例如，用零做除数，会出错。可是如果除数为空，计算结果也为空或未知。

在幻灯片的例子中，雇员 King 没有佣金，因为在算术表达式中的 COMMISSION_PCT 列为空，结果也为空。

更多的信息，见 *Oracle9i SQL Reference*, “SQL 的基本原理”。

定义列别名

列别名：

- 改变列标题的名字
- 可用于计算结果
- 紧跟在列名后面 – 在列名和别名之间可以有选项 AS 关键字
- 如果别名中包含有空格、或者特殊字符、或者大小写敏感，要求用双引号

列别名

在显示查询结果时，iSQL*Plus 通常用被选择列的名字作为列标题，该标题的描述可能不确切，因此可能难于理解，这时你可以用列别名改变列标题。

在 SELECT 列表中的列名后面指定别名，列名和别名之间用空格分开。默认情况下，别名标题用大写字母显示。如果别名中包含空格或者特殊字符（例如 # 或 &），或者大小写敏感，将别名放在双引号（“”）中。

教师注释

在一个 SQL 语句中，一个列别名既能用在 SELECT 子句也能用在 ORDER BY 子句中。你不能在 WHERE 子句中使用列别名。别名遵循 ANSI SQL 92 标准。

示范：1_alias.sql

目的：举例说明在表达式中的别名的使用。

使用列别名

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

NAME	COMM
King	
Kochhar	
De Haan	

...

20 rows selected.

```
SELECT last_name "Name", salary*12 "Annual Salary"
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

...

20 rows selected.

列别名 (续)

第一个例子显示所有雇员的名字和佣金的百分比。注意可选项 AS 关键字被用于别名 name 前面。查询结果与是否使用 AS 关键字无关。列别名 name 和 comm 用小写字母定义，但是在查询显示结果中，列标题用大写字母显示。因为前面说过，在默认情况下，列标题用大写字母显示。

第二个例子显示所有雇员的名字和年薪。因为 Annual Salary 中包含一个空格，所以必须放在双引号中，注意，输出的列标题与列名的定义完全相同。

教师注释

在第一个例子中，使用了可选项 AS 关键字，在第二个例子中使用了双引号。通常别名用大写显示，除非放在双引号中。

连字运算符

连字运算符：

- 连接列或者字符串到其它的列
- 用两个竖线表示 (||)
- 构造一个字符表达式的合成列

连字运算符

你能够用连字运算符 (||)，进行列与列之间、列与算术表达式之间或者列与常数值之间的连接，来创建一个字符表达式。连字运算符两边的列被合并成一个单个的输出列。

使用连字运算符

```
SELECT last_name||job_id AS "Employees"  
FROM employees;
```

Employees
KingAD_PRES
KochharAD_VP
De HaanAD_VP
HunoldIT_PROG
ErnstIT_PROG
LorentzIT_PROG
MourgosST_MAN
RajsST_CLERK

...

20 rows selected.

连字运算符 (续)

在例子中，LAST_NAME 和 JOB_ID 被连接，并且指定列别名 Employees。注意雇员的名字和工作代码被合并到一个单个的输出列中。

别名之前的 AS 关键字使得 SELECT 子句易于阅读。

文字字符串

- 文字字符串是包含在 **SELECT** 列表中的一个字符串，一个数字或者一个日期
- 日期和字符的文字字符串值必须用单引号括起来
- 每个文字字符串在每行输出一次

文字字符串

文字字符串是包含在 **SELECT** 列表中的一个字符串，一个数字或者一个日期，并且不是列名或别名。对每个返回行打印一次。任意格式文本的文字字符串能够被包含在查询结果中，并且作为 **SELECT** 列表中的列处理。

日期和字符文字 *必须* 放在单引号 (') 中；数字不需要。

使用文字字符串

```
SELECT last_name || ' is a ' || job_id
      AS "Employee Details"
FROM   employees;
```

Employee Details
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG
Lorentz is a IT_PROG
Mourgos is a ST_MAN
Rajs is a ST_CLERK
...

20 rows selected.

文字字符串 (续)

幻灯片中的例子显示所有雇员的名字和工作代码，列标题用 Employee Details。注意 SELECT 语句中单引号中的空格，这些空格增强了输出的可读性。

在下面的例子中，每个雇员的 last name 和 salary 用文字字符串连接，使得返回行的意思更清楚。

```
SELECT last_name || ': 1 Month salary = ' || salary Monthly
FROM   employees;
```

MONTHLY
King: 1 Month salary = 24000
Kochhar: 1 Month salary = 17000
De Haan: 1 Month salary = 17000
Hunold: 1 Month salary = 9000
Ernst: 1 Month salary = 6000
Lorentz: 1 Month salary = 4200
Mourgos: 1 Month salary = 5800
Rajs: 1 Month salary = 3500

20 rows selected.

相同的行

SELECT 默认显示所有的行，包括相同的行

```
SELECT department_id  
FROM employees;
```

DEPARTMENT_ID	
	90
	90
	90
	60
	60
	60
	50
	50
	50

...

20 rows selected.

相同的行

除非你另外指出，iSQL*Plus 显示的查询结果中不限制重复的行，在幻灯片上的例子中从 EMPLOYEES 表中显示了所有部门号。注意，部门号是有重复的。

除去相同的行

在 **SELECT** 子句中用 **DISTINCT** 关键字除去相同的行

```
SELECT DISTINCT department_id
FROM employees;
```

DEPARTMENT_ID	
	10
	20
	50
	60
	80
	90
	110

8 rows selected.

相同的行 (续)

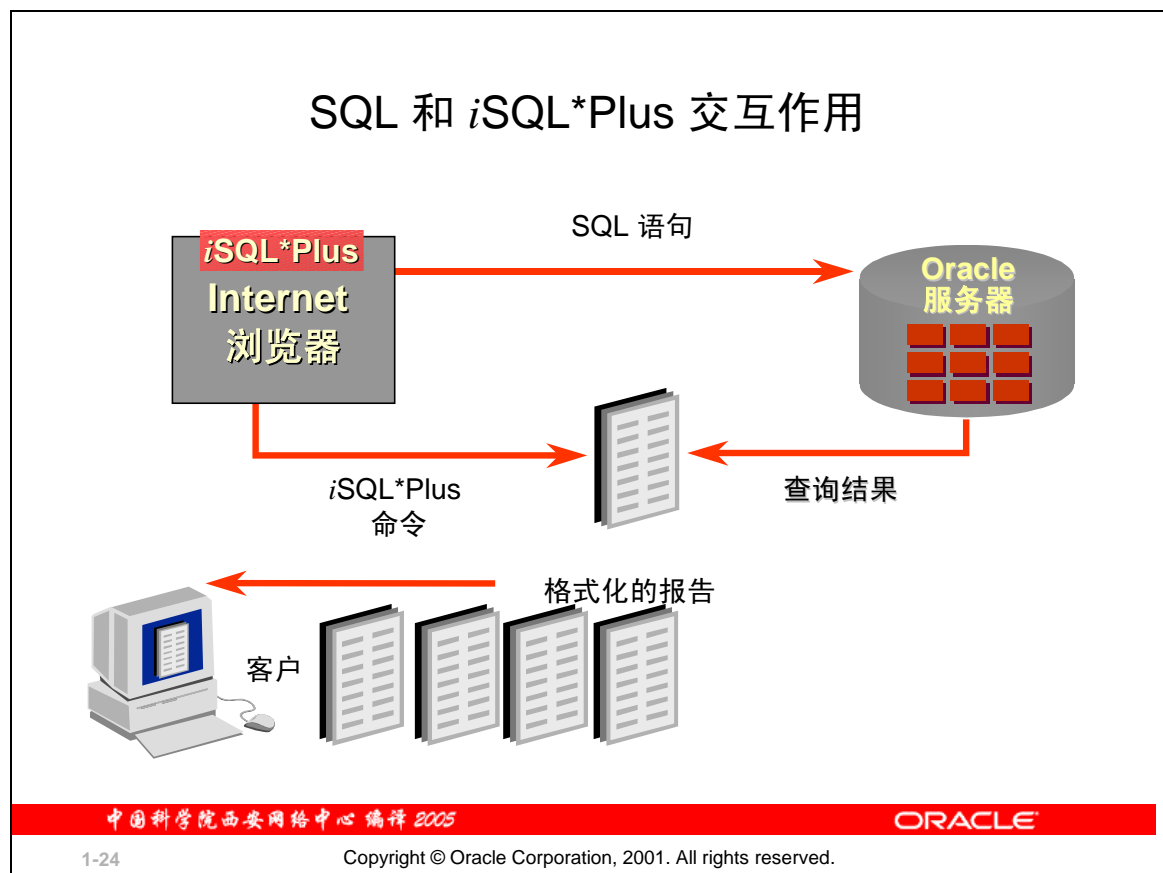
为了在结果中除去相同的行，在 **SELECT** 子句中的 **SELECT** 关键字后面紧跟 **DISTINCT** 关键字。在幻灯片上的例子中，**EMPLOYEES** 表实际有 20 行，但表中只有 7 个不同的部门。

在 **DISTINCT** 限定词后面，你可以指定多个列。**DISTINCT** 限定词影响所有被选定的列，并且结果是每个不相同的列的组合。

```
SELECT DISTINCT department_id, job_id
FROM employees;
```

DEPARTMENT_ID		JOB_ID
	10	AD_ASST
	20	MK_MAN
	20	MK_REP
	50	ST_CLERK
	50	ST_MAN
	60	IT_PROG
	80	SA_MAN
	80	SA_REP

13 rows selected.



SQL 和 iSQL*Plus

SQL 是从一个工具或应用程序与 Oracle 服务器通信的一种命令语言，Oracle SQL 包含许多扩展。

iSQL*Plus 是一个 Oracle 工具，用来识别并发送 SQL 语句到 Oracle 服务器执行。

iSQL*Plus 有它自己的命令语言。

SQL 的特性

- 能够被一个用户范围使用，包括那些有一点或根本没有编程经验的人
- 是一种非过程语言
- 减少创建和维护系统所需的时间
- 是一种类似英语的语言

iSQL*Plus 的特性

- 从浏览器访问
- 接受语句的输入
- 提供联机编辑修改 SQL 语句
- 控制环境设置
- 格式化查询结果到一个基本的报告中
- 访问本地和远程数据库

SQL 语句与 iSQL*Plus 命令

SQL

- 一种语言
- ANSI 标准
- 关键字不能缩写
- SQL 语句操纵定义在数据库中的数据 and 表

SQL
语句

iSQL*Plus

- 一个环境
- Oracle 所有
- 关键字可以缩写
- 命令不允许操纵数据库中的值
- 运行在浏览器上
- 处理集中在服务器上，不在客户机上实现

iSQL*Plus
命令

中国科学院西安网络中心 编译 2005

ORACLE

1-25

Copyright © Oracle Corporation, 2001. All rights reserved.

SQL 和 iSQL*Plus (续)

下面的表比较 SQL 和 iSQL*Plus。

SQL	iSQL*Plus
是一种与Oracle服务器通信以访问数据的语言	识别SQL语句并将它们发送到服务器
是基于美国国家标准协会 (ANSI) 标准的SQL语句	是Oracle所有的，用来执行SQL语句的接口
操纵定义在数据库中的数据 and 表	不允许操纵数据库中的值
没有延长符	如果命令长于一行，用破折号 (-) 折行
不能缩写	可以缩写
用函数完成一些格式化	用命令格式化数据

iSQL*Plus 概览

在登录进入 iSQL*Plus 后，你能够：

- 描述表结构
- 编辑 SQL 语句
- 从 iSQL*Plus 中执行 SQL
- 保存 SQL 语句到文件中，或者添加 SQL 语句到文件中
- 执行保存在文件中的语句
- 从文本文件中装载命令到 iSQL*Plus 编辑窗口中

中国科学院西安网络中心 编译 2005

ORACLE

1-26

Copyright © Oracle Corporation, 2001. All rights reserved.

iSQL*Plus

iSQL*Plus 是一个环境，它可以：

- 执行 SQL 语句，从数据库中取回、修改、添加和删除数据
- 格式化，执行计算，存储和打印在表格或报告中的查询结果
- 创建脚本文件来存储 SQL 语句，以后可以重复使用脚本文件

iSQL*Plus 命令能够被分为下面几种：

种类	目的
环境	影响会话期间 SQL 语句的总体行为
格式化	格式化查询结果
文件处理	保存语句到脚本文件中，从文本脚本文件中运行语句
执行	从浏览器发送 SQL 语句到 Oracle 服务器
编辑	在编辑窗口中修改 SQL 语句
交互	允许你创建并传递变量到 SQL 语句，打印变量值和打印信息到屏幕上
其他	还有各种连接数据库、操作 iSQL*Plus 环境和显示字段定义的命令

登录到 iSQL*Plus

在 Windows 浏览器环境中:



中国科学院西安网络中心 编译 2005

ORACLE

1-27

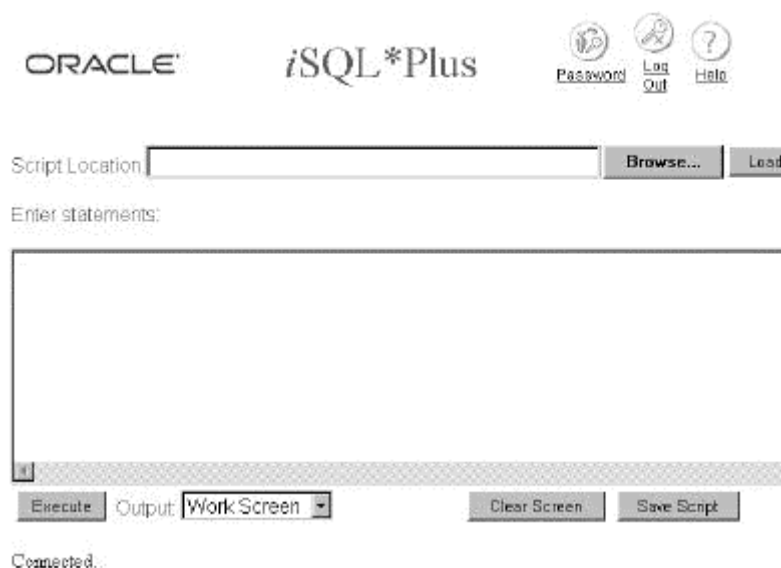
Copyright © Oracle Corporation, 2001. All rights reserved.

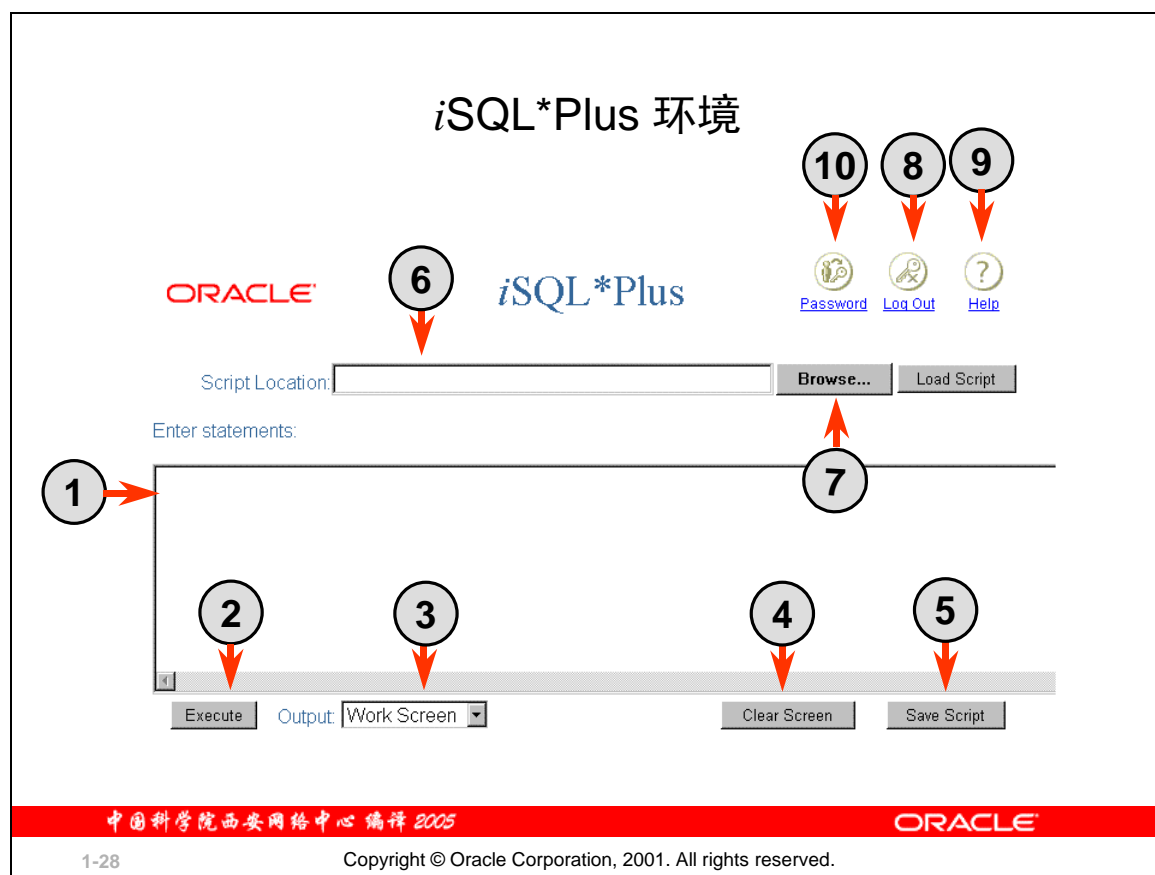
登录 iSQL*Plus

在浏览器环境中登录:

1. 启动浏览器。
2. 输入 iSQL*Plus 环境的 URL 地址。
3. 填写用户名, 密码和 Oracle 连接标识符域。

在你成功登录到 iSQL*Plus 后, 你可以看到下面的画面:





iSQL*Plus 环境

在 Windows 浏览器中，iSQL*Plus 窗口有几个关键区域：

1. 编辑窗口：该区域是用来输入 SQL 语句和 iSQL*Plus 命令的地方。
2. 执行按钮：单击执行编辑窗口中的语句和命令。
3. 输出选项：默认输出到工作屏幕上，在编辑窗口下面显示 SQL 语句的结果，其选项是文件或窗口。如果选择文件，将内容保存到指定的文件；如果选择窗口，内容输出到屏幕上，但在另外的窗口中。
4. 清除屏幕按钮：单击清除编辑窗口中的文本。
5. 保存脚本按钮：将编辑窗口中的内容保存到文件。
6. 脚本位置：指定要执行的脚本文件的名称和位置。
7. 浏览按钮：用 Windows 打开文件对话框查找脚本文件。
8. 退出图标：单击退出图标结束 iSQL*Plus 会话，并且返回到 iSQL*Plus 登录窗口。
9. 帮助图标：提供对 iSQL*Plus 帮助文档的访问。
10. 口令按钮：用于改变用户口令。

显示表结构

使用 *iSQL*Plus* DESCRIBE 命令显示表结构

```
DESC[RIBE] tablename
```

显示表结构

在 *iSQL*Plus* 中，可以用 DESCRIBE 命令显示表的结构。该命令显示列名和数据类型，以及是否列中必须有数据。

在语法中：

tablename 是对该用户任意存在的表、视图或同义词的名字

显示表结构

```
DESCRIBE employees
```

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

中国科学院西安网络中心 编译 2005

ORACLE

1-30

Copyright © Oracle Corporation, 2001. All rights reserved.

显示表结构 (续)

在幻灯片中的例子显示显示了关于 DEPARTMENTS 表的结构信息。

在结果中：

Null? 指示一个列是否必须包含数据；NOT NULL 指示一个列必须包含数据

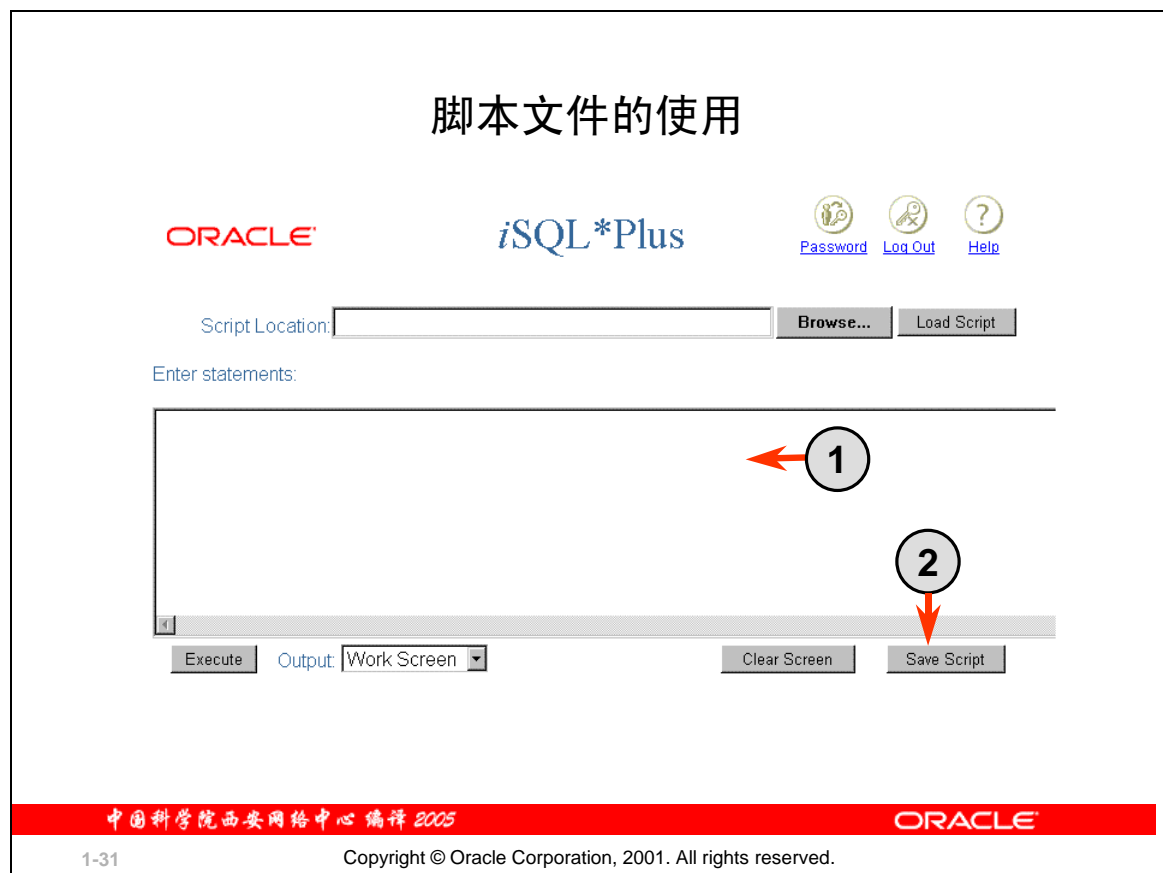
Type 显示列的数据类型

数据类型在下面的表中描述：

数据类型	说明
NUMBER(<i>p</i> , <i>s</i>)	数字值， <i>p</i> 是最大数字数目， <i>s</i> 是小数点右边的数字
VARCHAR2(<i>s</i>)	最大长度为 <i>s</i> 的可变长度字符值
DATE	在 BC 4712 年 1 月 1 日到 AD 9999 年 12 月 31 日之间的日期和时间
CHAR(<i>s</i>)	长度为 <i>s</i> 的固定长度字符值

教师注释

告诉学生在 DESCRIBE *tablename* 中的列顺序和在 SELECT * FROM *tablename* 中的顺序相同。列的显示顺序是创建表时列的定义顺序。

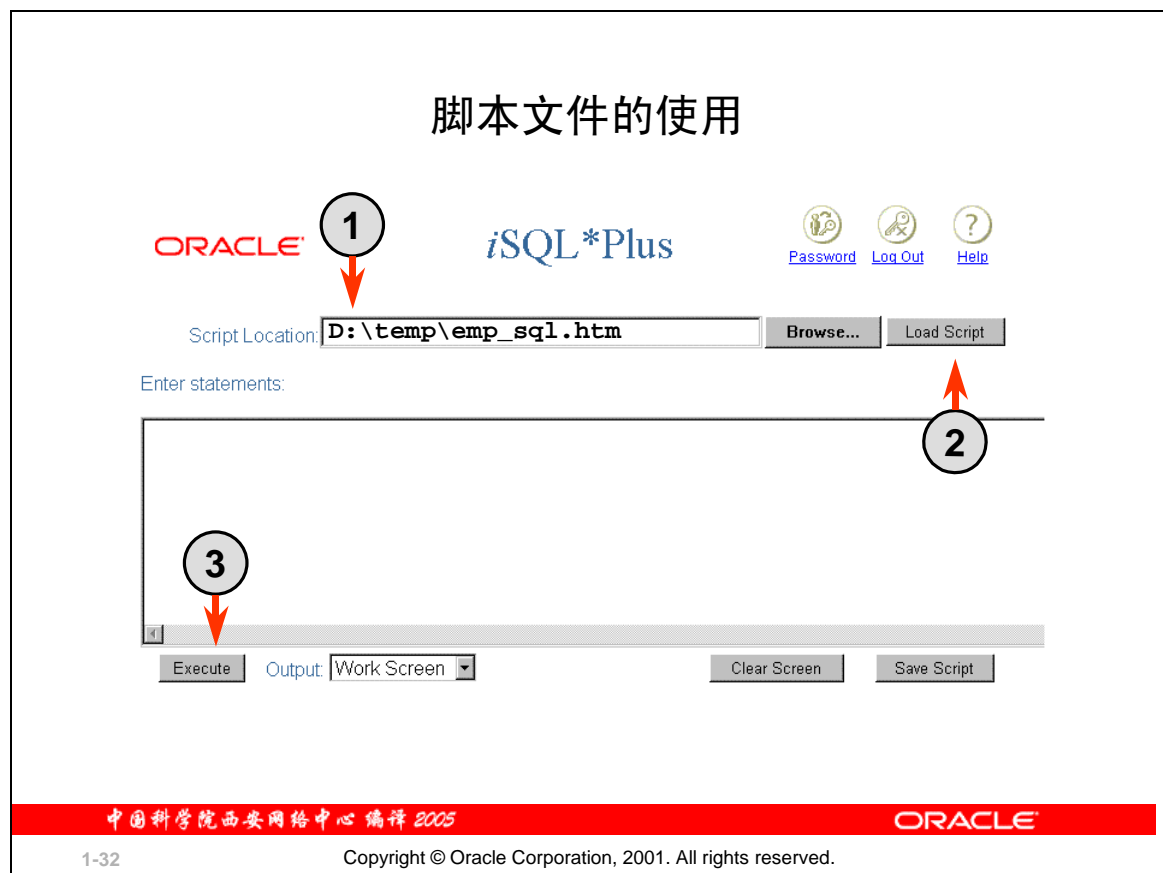


脚本文件的使用

你可以从 iSQL*Plus 的编辑窗口保存命令和语句到一个文本脚本文件中,步骤如下:

1. 在 iSQL*Plus 的编辑窗口中输入 SQL 语句。
2. 单击 Save Script 按钮,这一操作将打开 Windows 文件保存对话框,指定文件的名字,默认扩展名是 .html。你可以改变文件类型为一个文本文件或者将它保存为一个 .sql 文件。





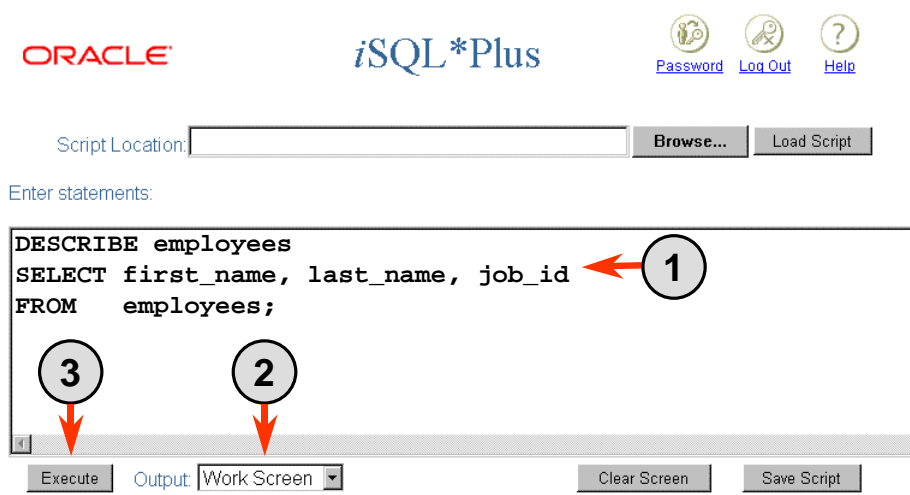
脚本文件的使用

在 iSQL*Plus 中执行脚本文件中的语句和命令

你能够从一个在 iSQL*Plus 中执行脚本文件使用以前保存的命令和语句：

1. 输入脚本名称和位置，或者，单击浏览按钮找到脚本名称和位置。
2. 单击装载脚本按钮。文件的内容被装载到 iSQL*Plus 编辑窗口中。
3. 单击执行按钮来运行 iSQL*Plus 编辑窗口中内容。

脚本文件的使用



ORACLE iSQL*Plus

Script Location: Browse... Load Script

Enter statements:

```
DESCRIBE employees
SELECT first_name, last_name, job_id
FROM employees;
```

Execute Output: Work Screen Clear Screen Save Script

中国科学院西安网络中心 编译 2005

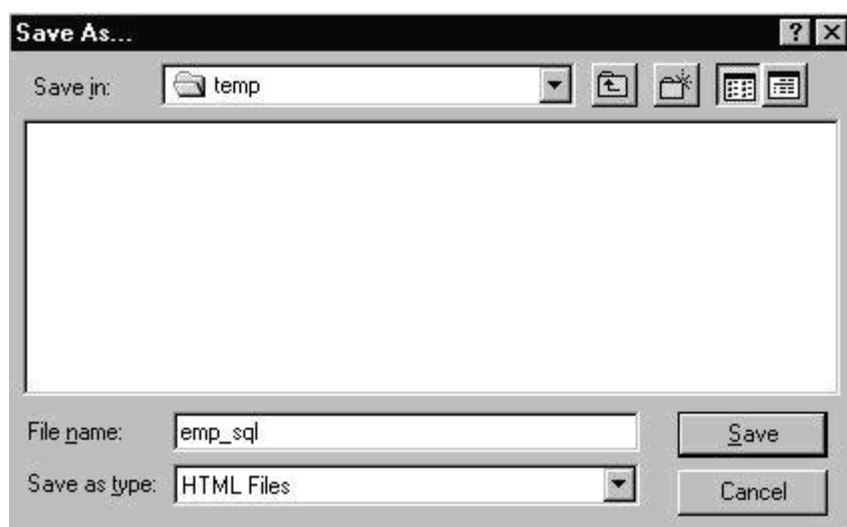
ORACLE

1-33 Copyright © Oracle Corporation, 2001. All rights reserved.

脚本文件的使用

你能够保存由 SQL 语句或 iSQL*Plus 命令产生的结果到一个文件中：

1. 输入 SQL 语句和 iSQL*Plus 命令到 iSQL*Plus 的编辑窗口中。
2. 改变输出选项到保存。
3. 单击执行按钮来运行 iSQL*Plus 编辑窗口中的内容，此操作打开 Windows 文件保存对话框。确定文件的名字，默认有一个 .html 扩展名，你能够改变文件类型，结果被发送到指定的文件。



小结

在本课中，您应该已经学会如何：

- 写一个 **SELECT** 语句，实现：
 - 从一个表中返回所有行和列
 - 从一个表中返回指定的列
 - 使用列别名给列标题以描述
- 使用 *iSQL*Plus* 环境写、保存和执行 SQL 语句和 *iSQL*Plus* 命令

```
SELECT    *|{[DISTINCT] column/expression [alias],...}
FROM      table;
```

SELECT 语句

在本课中，你应该已经学会怎样用 **SELECT** 语句从数据库的表中取回数据。

```
SELECT    *|{[DISTINCT] column [alias],...}
FROM      table;
```

在语法中：

SELECT	一个过多个字段的列表
*	选择所有列
DISTINCT	禁止重复
column/expression	选择列或表达式的名字
alias	给所选择的列以不同的标题
FROM table	指定包含相关列的表

iSQL*Plus

*iSQL*Plus* 是一个执行环境，你能用它来发送 SQL 语句到数据库服务器，并且能编辑和保存 SQL 语句。语句能够从 SQL 提示执行，也能够从一个脚本文件中执行。

注：SQL*Plus 环境在附录 C 中。

练习 1 概览

本章练习包括下面的主题：

- 从不同的表中选择所有的数据
- 描述表的结构
- 执行算术计算并且指定列名
- 使用 iSQL*Plus

练习 1 概览

这是许多练习的开始，答案（如果你需要的话）能够在附录 A 中找到。练习包括了课程中的所有主题，2 - 4 题只需要在卷子上做。

在后面的练习中，可能会有“如果你有时间”或“如果你想要额外的挑战”题目，如果你在分配的时间内已经完成了其它题目，并且想要更进一步的挑战你的技巧时，可以做做这些题目。

如果你完成这些练习的时间比指定的时间长或正好，你可以实验保存并运行命令文件。如果你有任何问题，请随时向教师提问。

判断题

2 - 4 是判断提，圈 True 或 False。

教师注释

让学生知道，为了获得一个他们在上课期间能够访问的表的清单，用命令：

```
SELECT * FROM TAB;
```

练习 1

1. 用教师提供的 user ID 和 password 开始一个 iSQL*Plus 会话。

2. iSQL*Plus 命令访问数据库。

True/False

False

3. 下面的 SELECT 语句是否执行成功:

```
SELECT last_name, job_id, salary AS Sal
FROM employees;
```

True/False

True

4. 下面的 SELECT 语句是否执行成功:

```
SELECT *
FROM job_grades;
```

True/False

True

5. 在下面的语句中有 4 个编码错误，你能找出他们吗？

```
SELECT employee_id, last_name
sal x 12 ANNUAL SALARY
FROM employees;
```

- **EMPLOYEES** 表不包含 **sal** 列，该列是 **SALARY**。
- 在第二行中，乘法运算符是 *****，而不是 **x**。
- **ANNUAL SALARY** 别名不能包含空格，别名应该是 **ANNUAL_SALARY** 或放在双引号中。
- 列 **LAST_NAME** 后面少一个逗号。

6. 显示 DEPARTMENTS 表的结构。选择表中的所有数据。

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

DESCRIBE departments

SELECT *

FROM departments;

- 显示 EMPLOYEES 表的结构。创建一个查询，显示每个雇员的 last name, job code, hire date, 和 employee 号, employee 号显示在第一列, 给 HIRE_DATE 列指定一个别名 STARTDATE。保存 SQL 语句到文件 lab1_7.sql 中。

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

DESCRIBE employees

```
SELECT employee_id, last_name, job_id, hire_date
FROM employees;
```

8. 运行在 lab1_7.sql 文件中的查询。

EMPLOYEE_ID	LAST_NAME	JOB_ID	StartDate
100	King	AD_PRES	17-JUN-87
101	Kochhar	AD_VP	21-SEP-89
102	De Haan	AD_VP	13-JAN-93
103	Hunold	IT_PROG	03-JAN-90
104	Ernst	IT_PROG	21-MAY-91
107	Lorentz	IT_PROG	07-FEB-99
124	Mourgos	ST_MAN	16-NOV-99
141	Rajs	ST_CLERK	17-OCT-95
142	Davies	ST_CLERK	29-JAN-97
143	Matos	ST_CLERK	15-MAR-98
144	Vargas	ST_CLERK	09-JUL-98
149	Zlotkey	SA_MAN	29-JAN-00
205	Higgins	AC_MGR	07-JUN-94
206	Gietz	AC_ACCOUNT	07-JUN-94

20 rows selected.

```
SELECT employee_id, last_name, job_id, hire_date
FROM employees;
```

9. 创建一个查询从 EMPLOYEES 表中显示唯一的工作代码。

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
IT_PROG
MK_MAN
MK_REP
SA_MAN
SA_REP
ST_CLERK
ST_MAN

12 rows selected.

```
SELECT DISTINCT job_id
FROM employees;
```

如果有时间，完成下面的练习：

10. 从 lab1_7.sql 复制语句到 iSQL*Plus 编辑窗口中。分别命名列标题 Emp #, Employee, Job 和 Hire Date, 再运行查询。

Emp #	Employee	Job	Hire Date
100	King	AD_PRES	17-JUN-87
101	Kochhar	AD_VP	21-SEP-89
102	De Haan	AD_VP	13-JAN-93
103	Hunold	IT_PROG	03-JAN-90
104	Ernst	IT_PROG	21-MAY-91
107	Lorentz	IT_PROG	07-FEB-99
124	Mourgos	ST_MAN	16-NOV-99
141	Rajs	ST_CLERK	17-OCT-95
142	Davies	ST_CLERK	29-JAN-97
143	Matos	ST_CLERK	15-MAR-98
144	Vargas	ST_CLERK	09-JUL-98
206	Gietz	AC_ACCOUNT	07-JUN-94

20 rows selected.

```
SELECT employee_id "Emp #", last_name "Employee",
       job_id "Job", hire_date "Hire Date"
FROM employees;
```

11. 显示 last_name, 用 job_ID 连接, 用逗号和空格分开, 用 Employee and Title 作为列名。

Employee and Title
King, AD_PRES
Kochhar, AD_VP
De Haan, AD_VP
Hunold, IT_PROG
Ernst, IT_PROG
Lorentz, IT_PROG
Mourgos, ST_MAN
Rajs, ST_CLERK
Davies, ST_CLERK
Gietz, AC_ACCOUNT

20 rows selected.

```
SELECT last_name||', '||job_id "Employee and Title"
FROM   employees;
```

如果你想要额外的挑战，完成下面的练习：

12. 创建一个查询从 EMPLOYEES 表中显示所有数据。用逗号分开列，命名列 THE_OUTPUT。

THE_OUTPUT									
100	Steven	King	SKING	515.123.4567	AD_PRES		17-JUN-87	24000	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	AD_VP	100	21-SEP-89	17000	90
102	Lex	De Haan	LDEHAAN	515.123.4569	AD_VP	100	13-JAN-93	17000	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	IT_PROG	102	03-JAN-90	9000	60
104	Bruce	Ernst	BERNST	590.423.4568	IT_PROG	103	21-MAY-91	6000	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	IT_PROG	103	07-FEB-99	4200	60
124	Kevin	Mourgos	KMOURGOS	650.123.5234	ST_MAN	100	16-NOV-99	5800	50
141	Trenna	Rajs	TRAJS	650.121.8009	ST_CLERK	124	17-OCT-95	3500	50
206	William	Gietz	WGIETZ	515.123.8181	AC_ACCOUNT	205	07-JUN-94	8300	110

20 rows selected.

```
SELECT employee_id || ', ' || first_name || ', ' || last_name
       || ', ' || email || ', ' || phone_number || ', ' || job_id
       || ', ' || manager_id || ', ' || hire_date || ', ' ||
       salary || ', ' || commission_pct || ', ' || department_id
       THE_OUTPUT
FROM   employees;
```