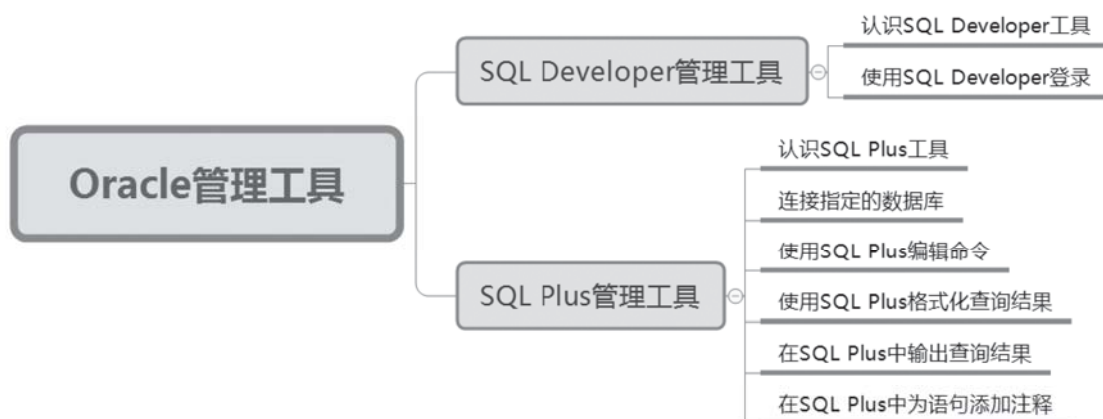


第3章 Oracle管理工具

本章导读

Oracle 数据库中常用的管理工具包括 SQL Developer 工具与 SQL Plus 工具。SQL Developer 是以图形化方式管理数据库数据的工具，而 SQL Plus 是与 Oracle 进行交互的客户端工具，在 SQL Plus 中，可以运行 SQL Plus 命令和 SQL 语句。本章就来介绍 Oracle 管理工具的应用。

知识导图





3.1 SQL Developer 管理工具

SQL Developer 是针对 Oracle 数据库的交互式开发环境（IDE）。使用 SQL Developer 可以浏览数据库对象、运行 SQL 语句和脚本、编辑和调试 PL/SQL 语句等。

3.1.1 认识 SQL Developer 工具

Oracle SQL Developer 是 Oracle 公司出品的一个免费的集成开发环境，该工具简化了 Oracle 数据库的开发和管理操作。SQL Developer 提供了 PL/SQL 程序的端到端开发、运行查询工作表的脚本、管理数据库的 DBA 控制台、报表接口、完整的数据建模的解决方案，并且能够支持将第三方数据库迁移至 Oracle。

要想使用 SQL Developer 工具管理 Oracle 数据库，首先需要下载并安装 SQL Developer 工具，具体操作步骤如下。

01 在地址栏中输入 SQL Developer 工具的下载地址：<https://www.oracle.com/tools/downloads/sqldev-v192-downloads.html>，进入下载页面，如图 3-1 所示。

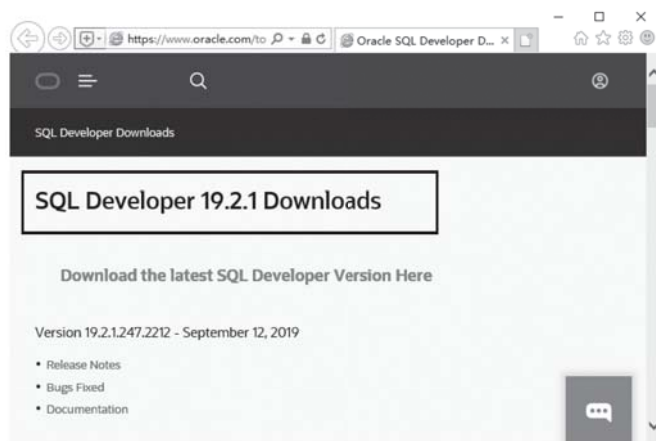


图 3-1 SQL Developer 下载页面

02 选择适合自己电脑的版本进行下载。这里选择第一项，包含 JDK 8 安装包，如图 3-2 所示。

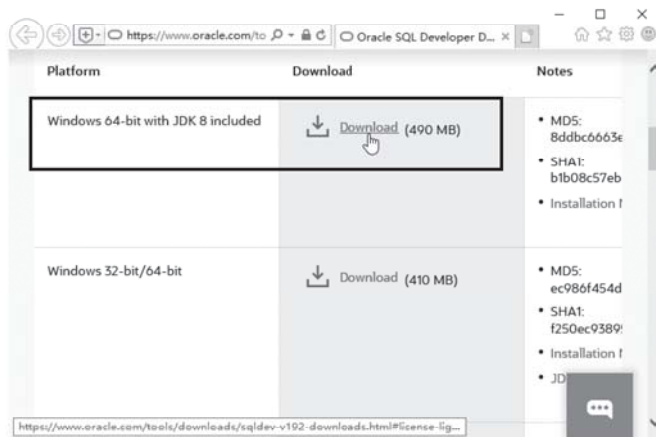


图 3-2 选择需要下载的版本

03 下载完成后，直接解压进行安装就可以用了。在解压包下找到 `sqldeveloper.exe` 双击启动，启动后默认只能连接 Oracle 数据库，如图 3-3 所示。



图 3-3 启动 SQL Developer

04 启动的过程中会弹出【确认导入首选项】对话框，在其中单击【否】按钮，如图 3-4 所示。



图 3-4 【确认导入首选项】对话框

05 开始启动 SQL Developer，并显示启动的进度，如图 3-5 所示。

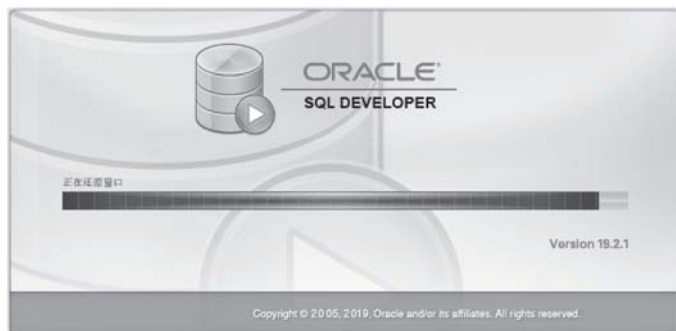


图 3-5 SQL Developer 启动的进度

06 启动完毕后，会弹出【Oracle 使用情况跟踪】对话框，选中【允许自动将使用情况报告给 Oracle】复选框，如图 3-6 所示。



图 3-6 【Oracle 使用情况跟踪】对话框

07 单击【确定】按钮，即可进入 SQL Developer 工具的工作界面，如图 3-7 所示。

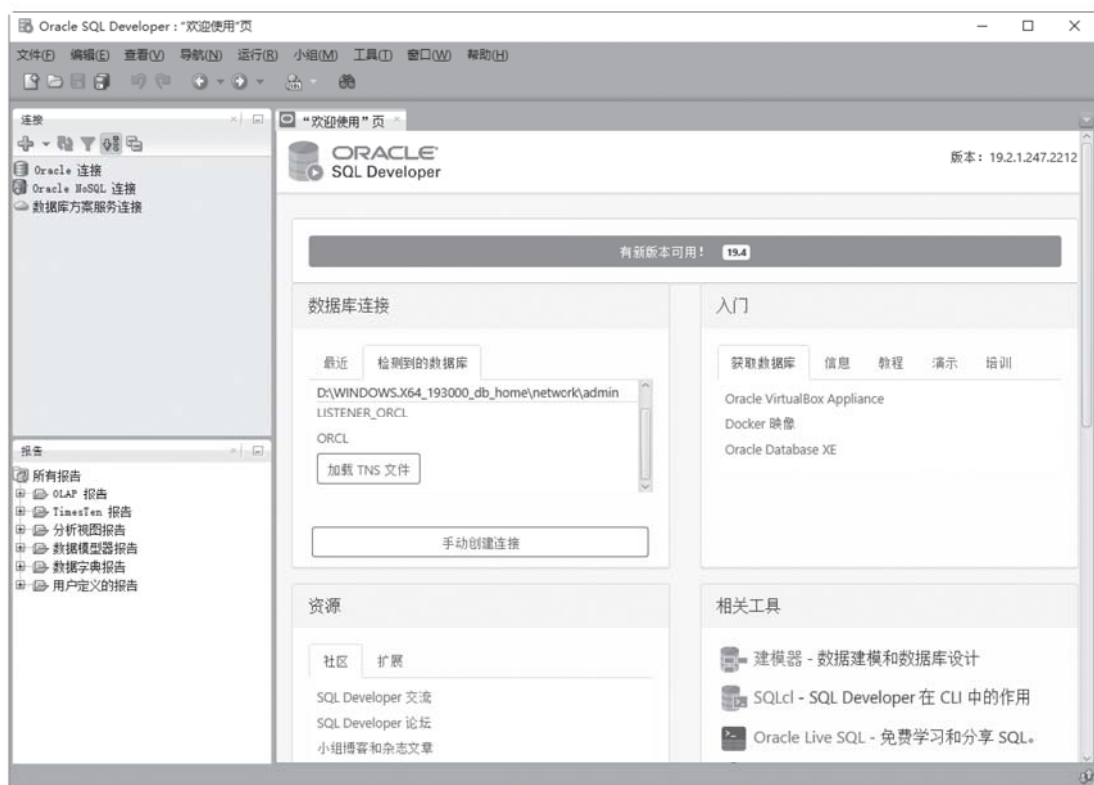


图 3-7 Oracle SQL Developer 工作界面

3.1.2 使用 SQL Developer 登录

启动 SQL Developer 工具后，还需要使用 SQL Developer 登录数据库，才能使用该工具管理 Oracle 数据库中的数据信息。具体操作步骤如下。

01 在 Oracle SQL Developer 窗口中，单击【连接】窗格中的下拉按钮，在弹出的下拉菜单中选择【新建数据库连接】命令，如图 3-8 所示。

02 打开【新建 / 选择数据库连接】对话框，输入连接名为“OracleConnect”，设置【验证类型】为【默认值】，并输入用户名与密码，设置【角色】为 SYSDBA，设置【连接类型】为【基本】、【主机名】为 localhost、【端口】为 1521、SID 为 orcl，如图 3-9 所示。

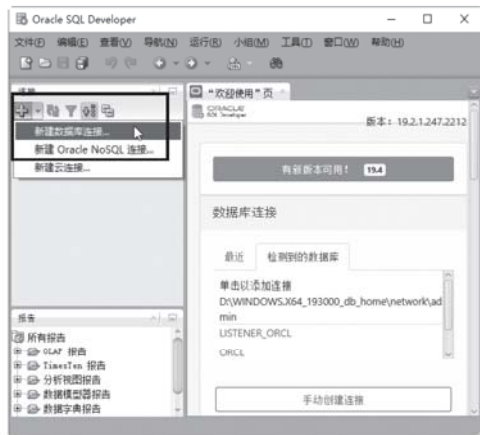


图 3-8 选择【新建数据库连接】命令



图 3-9 【新建 / 选择数据库连接】对话框

03 单击【连接】按钮，打开【连接信息】对话框，在其中输入用户名与密码，如图 3-10 所示。



图 3-10 【连接信息】对话框

04 单击【确定】按钮，即可打开 SQL Developer 主界面窗口，在打开的窗口中输入 SQL 命令，即可进行相关数据库文件的操作，如图 3-11 所示。

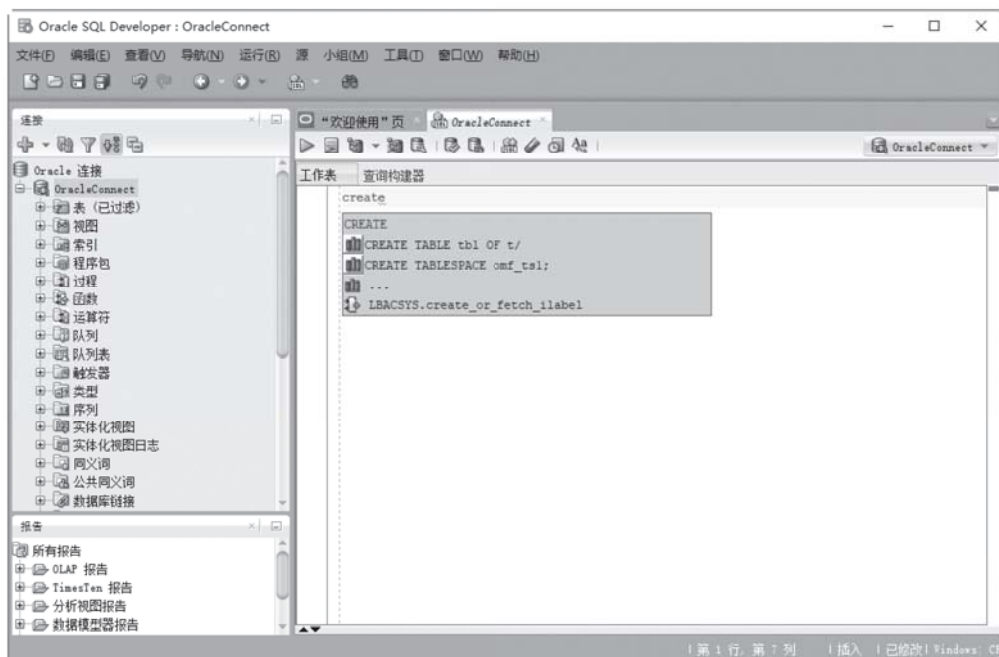


图 3-11 连接数据库后的 SQL Developer 主界面

3.2 SQL Plus 管理工具

SQL Plus 是与 Oracle 进行交互的客户端工具，借助 SQL Plus 可以查看和修改数据库记录。在 SQL Plus 中，可以运行 SQL Plus 命令与 SQL 语句。



3.2.1 认识 SQL Plus 工具

SQL Plus 是一个最常用的工具，具有很强的功能，主要功能如下。

- (1) 维护数据库，如启动、关闭等，一般在服务器上操作。
- (2) 执行 SQL 语句和 PL/SQL 程序。
- (3) 执行 SQL 脚本。
- (4) 导入和导出数据。
- (5) 开发应用程序。
- (6) 生成新的 SQL 脚本。
- (7) 供应用程序调用。
- (8) 用户管理及权限维护等。

3.2.2 连接指定的数据库

连接默认的数据库之后，用户还可以连接指定的数据库。命令如下：

```
SQL>connect username/password @Oracle net 名称
```

用户也可以在进入 SQL Plus 时直接连接其他数据库，命令如下：

```
C: >sqlplus username/password @Oracle net 名称
```

例如，连接 mytest 数据库，可以执行如下语句：

```
connect scott/Password123 @Oracle net mytest
```

或执行如下语句：

```
C: >sqlplus scott/Password123 @Oracle net mytest
```

3.2.3 使用 SQL Plus 编辑命令

在 SQL Plus 中，运行语句的方法比较简单，只需要输入语句后按 Enter 键即可。如果需要运行上一次的语句，直接按键盘上的 “/” 键即可。

为了演示如何使用 SQL Plus 命令，下面创建数据表 student，执行语句如下：

```
CREATE TABLE student
(
  id      NUMBER(9),
  name    VARCHAR2(10),
  class   VARCHAR2(10),
  addr    VARCHAR2(10)
);
```

执行结果如图 3-12 所示。

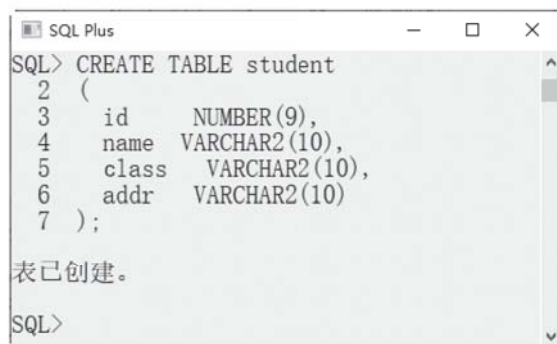


图 3-12 创建数据表 student

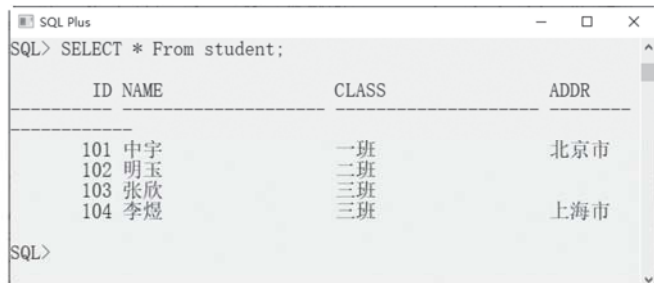
然后在该数据表中插入记录，执行语句如下：

```
INSERT INTO student (id ,name, class , addr) VALUES (101,'中宇', '一班', '北京市');
INSERT INTO student (id ,name, class ) VALUES (102,'明玉', '二班' );
INSERT INTO student (id ,name, class ) VALUES (103,'张欣', '三班');
INSERT INTO student (id ,name, class , addr) VALUES (104,'李煜', '三班', '上海市');
```

最后执行查询数据记录，执行语句如下：

```
SELECT * From student;
```

执行结果如图 3-13 所示。



The screenshot shows a SQL Plus window with the command 'SQL> SELECT * From student;' entered. The output is a table with four columns: ID, NAME, CLASS, and ADDR. The data is as follows:

ID	NAME	CLASS	ADDR
101	中宇	一班	北京市
102	明玉	二班	
103	张欣	三班	
104	李煜	三班	上海市

图 3-13 查询数据记录

1. 追加文本

一个语句执行完成后，如果想在此语句基础上增加一些内容，可以使用追加文本的命令来完成，具体语法如下：

```
Append text;
```

其中，text 就是需要追加的内容。

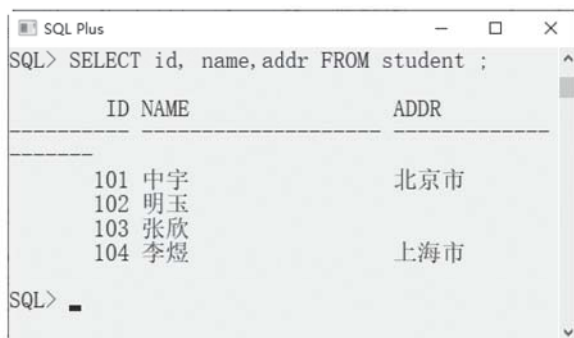
实例 1：通过追加文本查询数据

首先查询 student 表中指定的字段，然后追加 addr 不为空的条件。

首先查询指定的字段，执行语句如下：

```
SELECT id, name,addr FROM student;
```

查询结果如图 3-14 所示。



The screenshot shows a SQL Plus window with the command 'SQL> SELECT id, name,addr FROM student;' entered. The output is a table with three columns: ID, NAME, and ADDR. The data is as follows:

ID	NAME	ADDR
101	中宇	北京市
102	明玉	
103	张欣	
104	李煜	上海市

图 3-14 查询指定字段数据信息

然后追加 addr 不为空的条件，执行语句如下：

```
SQL> Append WHERE addr IS NOT NULL;
```

执行追加命令之后，会出现追加后的一条语句，然后输入“/”，即可运行该语句，结果如图 3-15 所示。

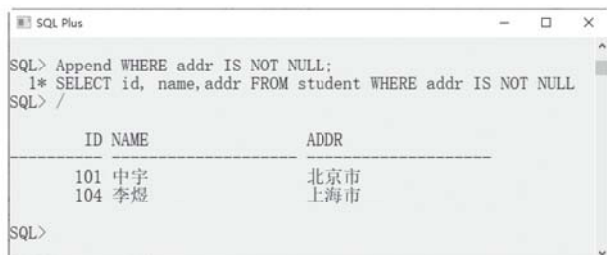


图 3-15 追加查询条件

2. 替换文本

在编辑时，如果需要对原有的语句进行更改，可以使用文本替换语句，如下所示：

```
Change / old text / new text
```

其中，old text 为需要被替换的内容；new text 为替换后的新内容。

实例 2：通过替换文本查询数据

首先查询 student 表中的 id 字段和 name 字段，然后修改为查询 name 字段和 class 字段。首先查询 id 字段和 name 字段，执行语句如下：

```
SQL> SELECT id, name FROM student ;
```

执行结果如图 3-16 所示。

然后修改为查询 name 字段和 class 字段，执行语句如下：

```
SQL> Change / name / class ;
```

然后修改命令之后，会出现替换后的一条语句，输入“/”，即可运行该语句，查询结果如图 3-17 所示。

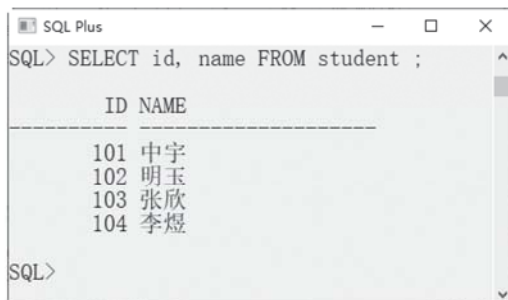


图 3-16 查询 id 字段和 name 字段数据信息

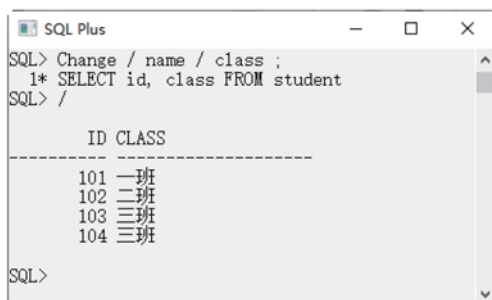


图 3-17 替换查询条件

注意：如果替换时没有输入 new text 的内容，则相当于把原来的字符串删除。

3. 删除命令

删除命令主要包括删除当前行、指定要删除的行数和删除的范围等命令。

删除指定行可以使用如下命令：

```
DEL n
```

其中, n 为要删除的行数。

实例 3: 通过删除指定行查询数据

首先查询 `student` 表中包含特定条件的 `id` 字段和 `addr` 字段, 然后删除特定条件查询 `id` 字段和 `addr` 字段。

首先查询 `addr` 不为空的 `id` 字段和 `addr` 字段, 执行语句如下:

```
SELECT id, addr FROM student  
WHERE addr IS NOT NULL;
```

执行结果如图 3-18 所示。

然后执行删除第二行的命令, 如下所示:

```
DEL 2;
```

执行结果如图 3-19 所示。输入 “/” 即可运行该语句。

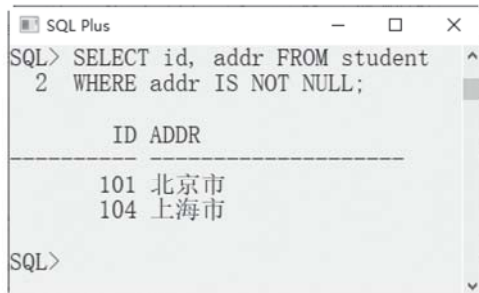


图 3-18 查询 `id` 字段和 `addr` 字段



图 3-19 删除条件进行查询

实例 4: 删除缓存区中的全部内容

删除缓存区中的全部内容, 命令如下:

```
CLEAR BUFFER;
```

执行结果如图 3-20 所示。从结果可以看出, 缓存区中的程序已经被全部删除了。

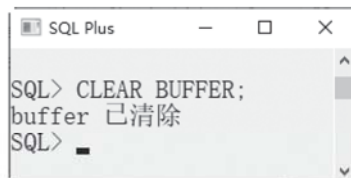


图 3-20 删除缓冲区中的内容

4. 添加行

添加行操作的命令如下:

```
INPUT text
```

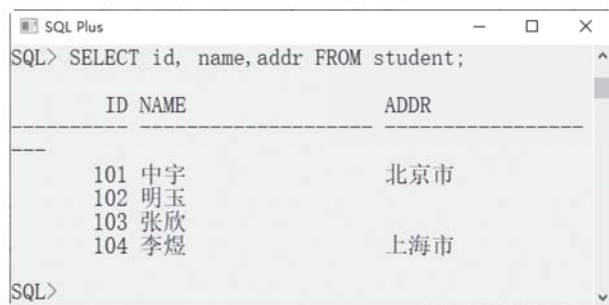
实例 5: 通过添加行查询数据

首先查询 `student` 表中指定的字段, 然后添加行, 内容为 `addr` 不为空的条件。

首先查询指定的字段, 执行语句如下:

```
SQL> SELECT id, name, addr FROM student;
```

执行结果如图 3-21 所示。



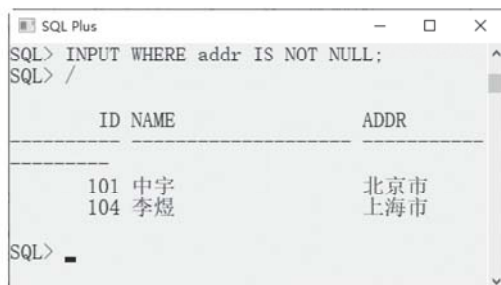
ID	NAME	ADDR
101	中宇	北京市
102	明玉	北京市
103	张欣	上海市
104	李煜	上海市

图 3-21 查看指定字段数据信息

然后添加行，执行语句如下：

```
INPUT WHERE addr IS NOT NULL;
```

执行结果如图 3-22 所示。



ID	NAME	ADDR
101	中宇	北京市
104	李煜	上海市

图 3-22 添加行后的查询结果

5. 显示缓存区中的内容

如果要显示缓存区中的内容，可以使用如下命令：

```
LIST [n/LAST]
```

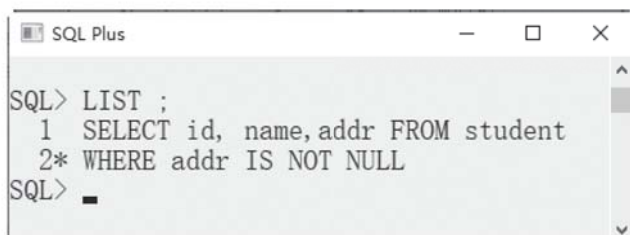
其中，**n** 表示显示缓存区中指定行的内容；**LAST** 表示缓存区中最后一行语句。

实例 6：显示缓存区中的内容

（1）显示缓存区中的全部内容，命令如下：

```
LIST ;
```

执行结果如图 3-23 所示。



```
SQL> LIST ;
1  SELECT id, name, addr FROM student
2*  WHERE addr IS NOT NULL
SQL>
```

图 3-23 显示缓存区中的全部内容

(2) 显示缓存区中第 2 行的内容，命令如下：

```
LIST 2;
```

执行结果如图 3-24 所示。

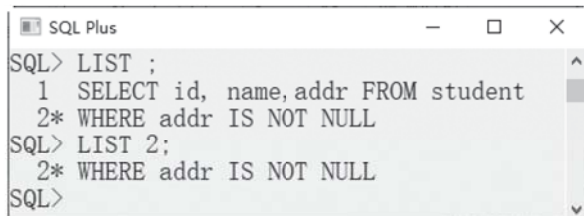


图 3-24 显示缓存区中第 2 行的内容

(3) 显示缓存区中最后一行的内容，命令如下：

```
LIST LAST;
```

结果如图 3-25 所示。

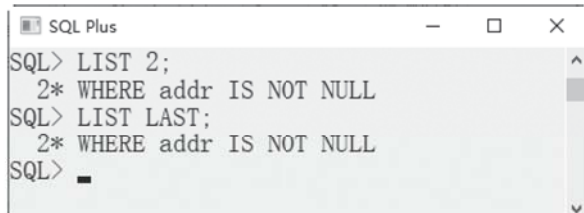


图 3-25 显示缓存区中最后一行的内容

3.2.4 使用 SQL Plus 格式化查询结果

默认情况下，如果查询的列比较多，会出现显示混乱的情况，此时可以使用 SQL Plus 提供的多种设置查询结果显示的方法。

1. 使用 COLUMN 命令设置列的别名和格式

使用 COLUMN 命令可以设置列的别名，具体语法如下：

```
COLUMN oldname HEADING newname
```

其中，oldname 表示要查询表的列名；newname 为列的别名。

实例 7：使用 COLUMN 命令设置列的别名

查询 student 表中指定的字段，并设置 name 的别名为“姓名”，执行语句如下：

```
SQL>COLUMN name HEADING 姓名  
SQL>SELECT id, name,class,addr FROM student;
```

执行结果如图 3-26 所示。

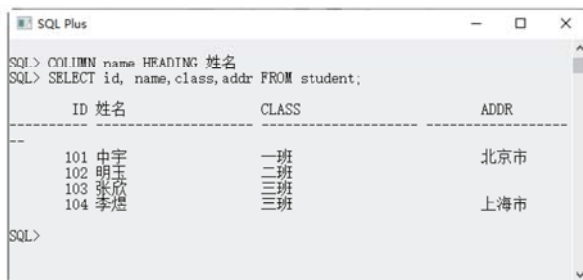


图 3-26 设置列的别名

使用 COLUMN 命令可以设置列的显示格式，具体语法如下：

```
COLUMN column_name FORMAT dataformat
```

其中，column_name 表示需要格式化查询结果的列名；dataformat 表示格式化后显示的格式。常见的数据格式如下。

- (1) 0: 在指定的位置显示前导 0 或者后置 0。
- (2) 9: 代表一个数字字符。
- (3) B: 显示一个空格。
- (4) MI: 显示负号。
- (5) \$: 美元货币符号。
- (6) G: 显示千分之分组符号。
- (7) L: 显示本地区区域的货币符号。
- (8) .: 显示小数点。

实例 8: 使用 COLUMN 命令设置列的格式

为了演示如何使用 SQL Plus 命令，下面创建数据表 fruits，执行语句如下：

```
CREATE TABLE fruits
(
  f_id      varchar2(10)      NOT NULL,
  s_id      number(6)         NOT NULL,
  f_name    varchar2(10)      NOT NULL,
  f_price   number(8,2)       NOT NULL
);
```

执行结果如图 3-27 所示，即可完成数据表的创建。

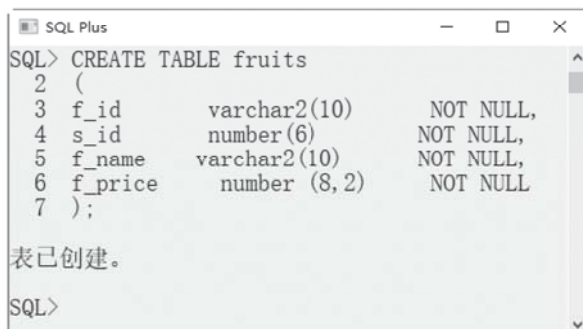


图 3-27 创建数据表 fruits

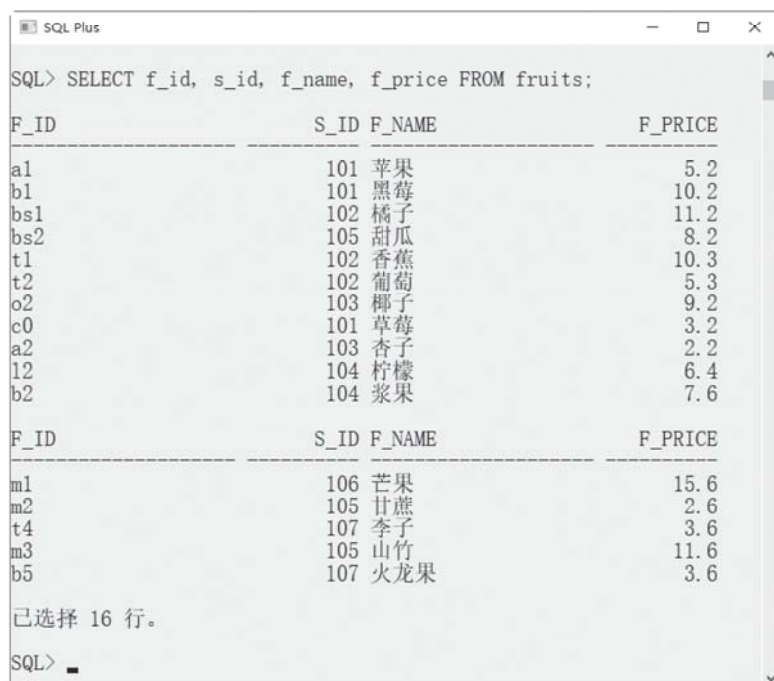
创建好数据表后，向 **fruits** 表中输入表数据，执行语句如下：

```
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('a1', 101, '苹果', 5.2);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('b1', 101, '黑莓', 10.2);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('bs1', 102, '橘子', 11.2);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('bs2', 105, '甜瓜', 8.2);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('t1', 102, '香蕉', 10.3);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('t2', 102, '葡萄', 5.3);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('o2', 103, '椰子', 9.2);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('c0', 101, '草莓', 3.2);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('a2', 103, '杏子', 2.2);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('l2', 104, '柠檬', 6.4);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('b2', 104, '浆果', 7.6);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('m1', 106, '芒果', 15.6);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('m2', 105, '甘蔗', 2.6);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('t4', 107, '李子', 3.6);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('m3', 105, '山竹', 11.6);
INSERT INTO fruits (f_id, s_id, f_name, f_price) VALUES ('b5', 107, '火龙果', 3.6);
```

使用 **SELECT** 语句查询 **fruits** 表中所有字段的数据，执行语句如下：

```
SELECT f_id, s_id, f_name, f_price FROM fruits;
```

即可完成数据的查询，并显示查询结果，如图 3-28 所示。



F_ID	S_ID	F_NAME	F_PRICE
a1	101	苹果	5.2
b1	101	黑莓	10.2
bs1	102	橘子	11.2
bs2	105	甜瓜	8.2
t1	102	香蕉	10.3
t2	102	葡萄	5.3
o2	103	椰子	9.2
c0	101	草莓	3.2
a2	103	杏子	2.2
l2	104	柠檬	6.4
b2	104	浆果	7.6
m1	106	芒果	15.6
m2	105	甘蔗	2.6
t4	107	李子	3.6
m3	105	山竹	11.6
b5	107	火龙果	3.6

已选择 16 行。

SQL> _

图 3-28 显示数据表中的全部记录

查询 **fruits** 表中指定的字段，并将价格的格式设置为 “\$0.0” 的形式，执行语句如下：

```
SQL> COLUMN f_price FORMAT $0.0
SQL> SELECT f_price FROM fruits where f_name='苹果';
```

执行结果如图 3-29 所示。

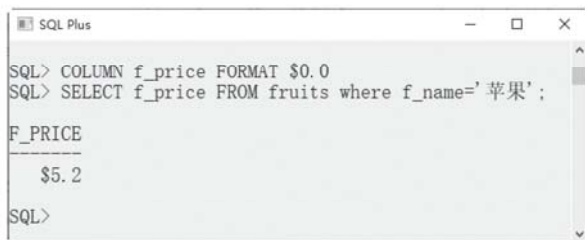


图 3-29 设置列的格式

2. 使用 SET 命令设置格式

使用 SET 命令可以设置查询结果的格式，包括每一页显示的行数、每行显示的字符数、显示查询数据所用的时间、是否显示列标题、是否显示“已选择行数”等。

语法格式如下：

```
SET PAGESIZE/NEWPAGE/LINESIZE n
```

其中，PAGESIZE 表示设置显示页的格式；NEWPAGE 表示设置每页之间间隔的格式；LINESIZE 表示设置每行字符数的格式；n 表示对应格式的数量。

默认每页显示的行数为 24，如果需要设置为 20，命令如下：

```
SET PAGESIZE 20
```

设置每页之间间隔 2 个空格，命令如下：

```
SET NEWPAGE 2
```

实例 9：使用 SET 命令设置每行显示的字符数

查询目前每行显示的字符数，然后设置每行的显示字符数为 60，执行语句如下：

```
SQL> SHOW LINESIZE;
SQL> SET LINESIZE 60;
```

结果如图 3-30 所示。

再次查询每行显示的字符数，命令如下：

```
SQL> SHOW LINESIZE;
```

结果如图 3-31 所示，从结果中可以看出，设置已经生效。

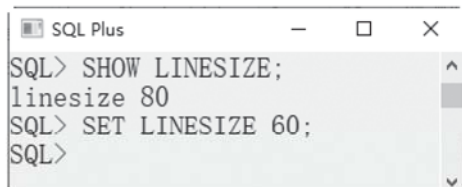


图 3-30 设置每行的显示字符数为 60

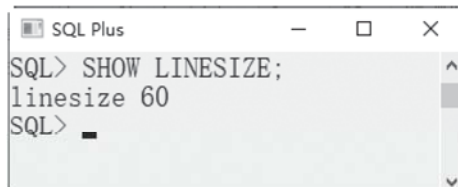


图 3-31 查询每行显示的字符数

使用 SET 命令还可以设置是否显示数据所用的时间、列标题、“已选择行”。语法格式如下：

```
SET TIMING/HEADING/FEEDBACK/ ON/OFF
```

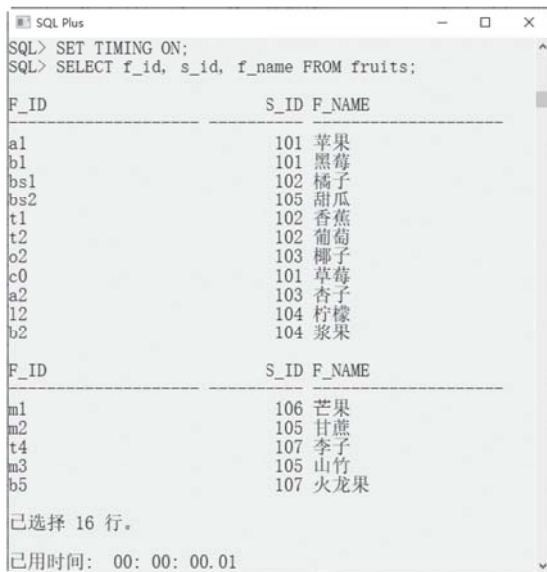
其中, TIMING 用于设置是否显示数据所用的时间; HEADING 用于设置是否显示列标题; FEEDBACK 用于设置是否显示“已选择行”; ON 表示开启对应的功能, OFF 表示关闭对应的功能。

实例 10: 使用 SET 命令设置显示数据所用的时间

设置显示数据所用的时间, 然后测试是否成功, 执行语句如下:

```
SQL> SET TIMING ON;  
SQL> SELECT f_id, s_id, f_name FROM fruits;
```

执行结果如图 3-32 所示。



```
SQL Plus  
SQL> SET TIMING ON;  
SQL> SELECT f_id, s_id, f_name FROM fruits;  
  
F_ID          S_ID F_NAME  
-----  
a1            101 苹果  
b1            101 黑莓  
bs1           102 橘子  
bs2           105 甜瓜  
t1            102 香蕉  
t2            102 葡萄  
o2            103 椰子  
c0            101 草莓  
a2            103 杏子  
l2            104 柠檬  
b2            104 浆果  
  
F_ID          S_ID F_NAME  
-----  
m1            106 芒果  
m2            105 甘蔗  
t4            107 李子  
m3            105 山竹  
b5            107 火龙果  
  
已选择 16 行。  
已用时间: 00: 00: 00.01
```

图 3-32 设置显示数据所用的时间

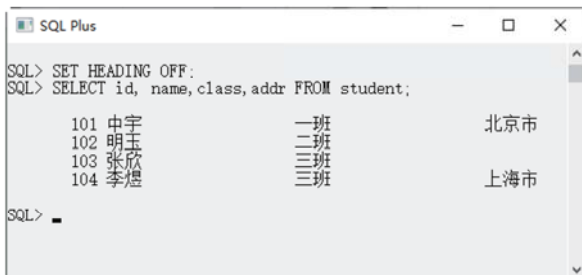
有时为了提高数据库的安全, 会隐藏每列的标题。下面通过案例来理解。

实例 11: 使用 SET 命令隐藏查询结果中每列的标题

隐藏查询结果中每列的标题, 然后测试是否成功, 命令如下:

```
SQL> SET HEADING OFF;  
SQL> SELECT id, name, class, addr FROM student;
```

查询结果如图 3-33 所示。



```
SQL Plus  
SQL> SET HEADING OFF;  
SQL> SELECT id, name, class, addr FROM student;  
  
101 中宇          一班          北京市  
102 明玉          二班          北京市  
103 张欣          三班          上海市  
104 李煜          三班          上海市  
  
SQL>
```

图 3-33 隐藏查询结果中每列的标题

如果需要显示每列的标题，命令如下：

```
SQL> SET HEADING ON;
```

实例 12：使用 SET 命令设置显示已选择行数

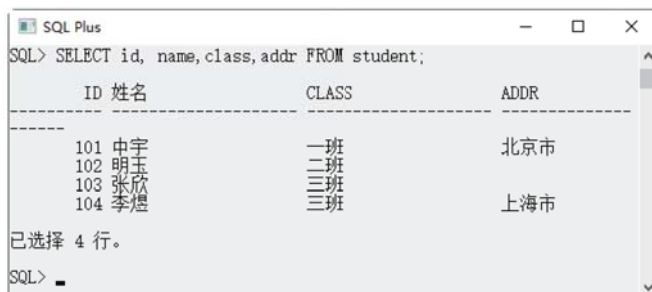
如果需要显示“已选择行数”，命令如下：

```
SET FEEDBACK ON;
```

运行查询命令进行测试，命令如下：

```
SQL> SELECT id, name, class, addr FROM student;
```

结果如图 3-34 所示。



The screenshot shows a SQL Plus window with the following content:

```
SQL> SELECT id, name, class, addr FROM student;
```

ID	姓名	CLASS	ADDR
101	中宇	一班	北京市
102	明玉	二班	
103	张欣	三班	
104	李煜	三班	上海市

已选择 4 行。

```
SQL>
```

图 3-34 设置显示“已选择行数”

如果不再需要显示“已选择行数”，命令如下：

```
SET FEEDBACK OFF;
```

3.2.5 在 SQL Plus 中输出查询结果

在 SQL Plus 中经常需要把查询结果保存到文件中。使用 SPOOL 命令可以完成保存的操作，具体语法格式如下：

```
SPOOL filename  
SPOOL OFF
```

其中，filename 为保存输出结果的文件名，扩展名为 .sql，可以包含保存路径；SPOOL OFF 的作用是开始把查询结果真正写入指定文件中。

实例 13：把查询结果写入指定文件中

把查询结果写入 myrcst.sql 文件中，执行语句如下：

```
SQL>SPOOL d:/myrcst.sql  
SQL>SELECT id, name,addr FROM student;  
SQL>SPOOL OFF;
```

执行结果如图 3-35 所示。



图 3-35 将查询结果写入 myrest.sql 文件

用记事本打开保存的文件 myrest.sql，内容如图 3-36 所示。

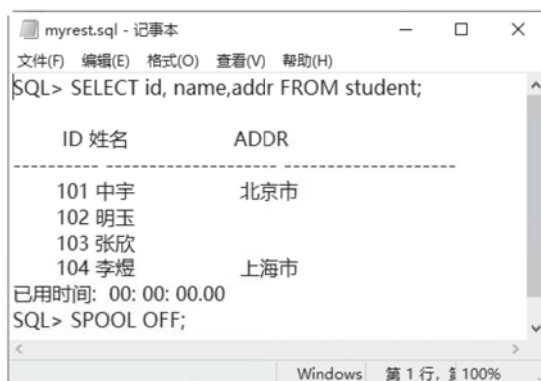


图 3-36 打开 myrest.sql 文件

3.2.6 在 SQL Plus 中为语句添加注释

添加注释可以提高程序的可读性，在 SQL Plus 中也可以为语句添加注释，主要方法如下。

1. 使用 /*.....*/ 方法

通过 /*.....*/ 方法可以为语句添加注释内容。注释在编译时不会被执行，并且注释可以写在语句中的任何地方，如图 3-37 所示。

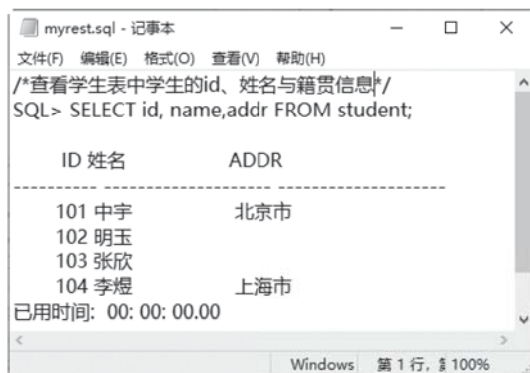


图 3-37 在记事本中添加注释

2. 使用 REMARK 命令添加注释

在 SQL Plus 中，可以使用 REMARK 命令添加注释，语法格式如下：

```
REMARK comment
```

其中，comment 为添加的注释信息。

实例 14: 使用 REMARK 命令添加注释

实例执行语句如下:

```
SQL> SPOOL e:\mytest.sql
SQL> REMARK '查询学生的姓名与班级信息'
SQL> SELECT name,class FROM student;
SQL> SPOOL OFF;
```

执行结果如图 3-38 所示。

这样注释信息被添加到 mytest.sql 文件中, 其内容如图 3-39 所示。

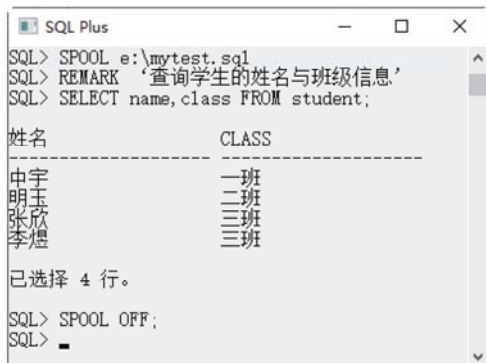


图 3-38 使用 REMARK 命令添加注释

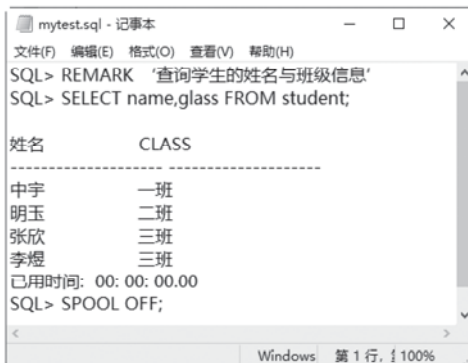


图 3-39 查看注释信息

3.3 疑难问题解析

疑问 1: 如何隐藏查询结果?

答: 在 SQL Plus 中, 可以隐藏查询结果, 命令如下:

```
SET TERM ON/OFF;
```

其中, ON 表示显示查询结果, OFF 表示隐藏查询结果。

疑问 2: 如何让 SQL Plus 中的查询结果不混乱?

答: 在 SQL Plus 中, 如果发现查询结果比较混乱, 可以采用如下命令进行调整:

```
SQL> set wrap off;
SQL> set linesize 180;
```

3.4 实战训练营

实战 1: SQL Developer 管理工具的应用

- (1) 使用 SQL Developer 登录数据库。
- (2) 在 SQL Developer 窗口中创建数据表 student。
- (3) 在 SQL Developer 窗口中为数据表 student 插入数据。

（4）在 SQL Developer 窗口中查询数据表 student。

■ 实战 2：SQL Plus 管理工具的应用

- （1）使用 SQL Plus 连接指定数据库。
- （2）使用 SQL Plus 编辑命令查询数据记录。
- （3）使用 SQL Plus 格式化命令设置查看结果的格式。
- （4）使用 SQL Plus 命令输出查询结果到记事本。
- （5）在 SQL Plus 窗口中添加注释信息。