CHAPTER 5 第5章

数据库的查询和视图

Oracle 是一个关系数据库管理系统，关系数据库建立在关系模型基础之上，具有严格的数学理论基础。关系数据库对数据的操作除了包括集合代数的并、差等运算之外，还定义了一组专门的关系运算：选择、投影和连接，关系运算的特点是运算的对象和结果都是表。

在数据库应用中，最常用的操作是查询，它是数据库的其他操作（如统计、插入、删除及修改）的基础。在 Oracle 11g 中，对数据库的查询使用 SELECT 语句，它功能非常强大，使用灵活，可对表进行选择、投影和连接。

本章重点讨论利用 SELEC 语句对数据库进行各种查询的方法。

视图是由一个或多个基本表导出的数据信息，可以根据用户的需要创建视图。视图对于数据库的用户来说很重要，本章将讨论视图概念以及视图的创建与使用方法。

5.1 数据库的查询

使用数据库和表的主要目的是存储数据，以便在需要时进行检索、统计或组织输出，通过 SQL 查询可以从表或视图中迅速、方便地检索数据。SQL 的 SELECT 语句可以实现对表的选择、投影及连接操作，其功能十分强大。SELECT 语句比较复杂，其主要的子句语法格式如下：

```
SELECT<列>                                  /*指定要选择的列及其限定*/
    FROM  <表或视图>                        /*FROM子句,指定表或视图*/
    [WHERE  <条件表达式>]                   /*WHERE子句,指定查询条件*/
    [GROUP BY<分组表达式>]                  /*GROUP BY子句,指定分组表达式*/
    [HAVING<分组条件表达式>]                /*HAVING子句,指定分组统计条件*/
    [ORDER BY<排序表达式>[ASC | DESC]]      /*ORDER子句,指定排序表达式和顺序*/
```

指定要选择的列可实现关系运算中的投影操作，指定要选择的行可实现关系运算中的选择操作，通过 FROM、WHERE 及其配合可以实现关系运算中的连接操作。

下面讨论 SELECT 的基本语法和主要功能。

5.1.1 选择列

选择表中的列组成结果表，语法格式为

```
SELECT [ALL | DISTINCT]<列名列表>
```

其中，<列名列表>指出了结果的形式，其主要格式为：

```
{    *                                  /* 选择当前表或视图的所有列 */
     | {<表名>|<视图>} . *               /* 选择指定的表或视图的所有列 */
     | {<列名>|<表达式>}
     [[AS]<列别名>]                     /* 选择指定的列 */
     |<列标题>=<列名表达式>             /* 选择指定列并更改列标题 */
} [,...n]
```

1. 选择指定的列

使用 SELECT 语句选择一个表中的某些列，各列名之间要以逗号分隔，语法格式为：

```
SELECT<列名 1>[,<列名 2>[,...n]]
    FROM<表名>
    [WHERE<条件表达式>]
```

其功能是在 FROM 子句指定的表中检索符合条件的列。

当在 SELECT 语句指定列的位置上使用 * 号时，表示选择表的所有列。

【例 5.1】 查询 XSCJ 数据库的 XSB 表中各个学生的学号、姓名和总学分。

在 SQL Developer 中 myorcl 连接的命令编辑区输入如下语句：

```
SELECT 学号, 姓名, 总学分
    FROM XSB;
```

将光标定位到语句第一行，单击“执行”按钮▷，执行结果如图 5.1 所示。执行完后，“结果”选项页中将列出所有结果数据。

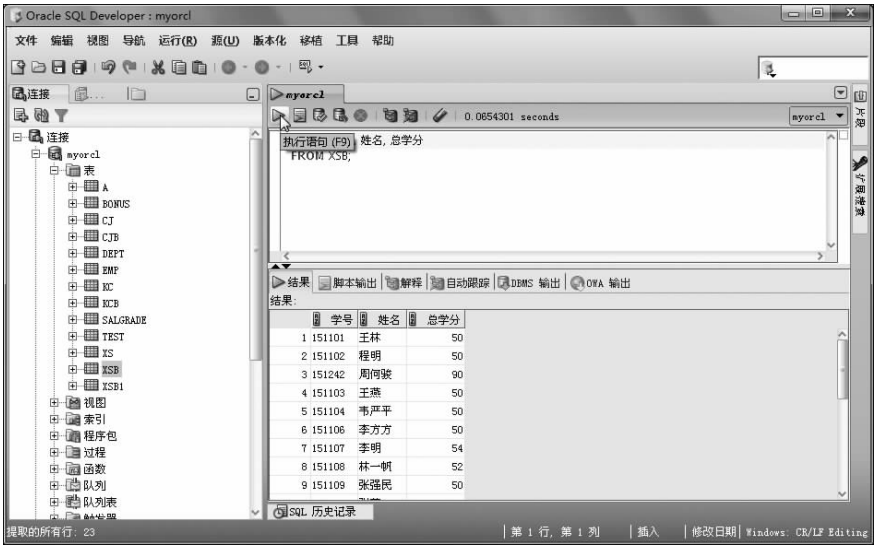

图 5.1　在 XSB 表中选择列

【例 5.2】 查询 XSB 表中总学分大于 50 的学生的学号、姓名和总学分。

```
SELECT 学号, 姓名, 总学分
    FROM XSB
    WHERE 总学分>50;
```

查询结果如图 5.2 所示。

【例 5.3】 查询 XSB 表中的所有列。

```
SELECT *
    FROM XSB;
```

该语句等价于：

```
SELECT 学号姓名, 性别, 出生时间, 专业, 总学分, 备注
    FROM XSB;
```

执行后将列出 XSB 表中的所有数据。

2. 修改查询列标题

如果希望查询结果中的某些列，或者所有列显示时使用自己选择的列标题，则可以在列名之后使用 AS 子句指定一个列别名来替代查询结果的列标题名。

【例 5.4】 查询 XSB 表中计算机专业学生的学号、姓名和总学分，结果中各列的标题分别指定为 num、name 和 score。

```
SELECT 学号 AS num, 姓名 AS name, 总学分 AS score
    FROM XSB
    WHERE 专业='计算机';
```

也可以省略 AS 关键字，写成：

```
SELECT 学号 num, 姓名 name, 总学分 score
    FROM XSB
    WHERE 专业='计算机';
```

查询结果如图 5.3 所示。

	学号	姓名	总学分
1	151107	李明	54
2	151108	林一帆	52
3	151242	周何骏	90

图 5.2 例 5.2 查询结果

	NUM	NAME	SCORE
1	151101	王林	50
2	151102	程明	50
3	151103	王燕	50
4	151104	韦严平	50
5	151106	李方方	50
6	151107	李明	54
7	151108	林一帆	52
8	151109	张强民	50
9	151110	张蔚	50
10	151111	赵琳	50
11	151113	严红	48

图 5.3 例 5.4 查询结果

3. 计算列值

使用 SELECT 对列进行查询时，在结果中可以输出对列值计算后的值，即 SELECT 语句可以使用表达式作为结果，格式为：

```
SELECT<表达式>[,<表达式>]
```

【例 5.5】 创建产品表 CP,其结构如表 5.1 所示。

表 5.1　CP 表结构

列名	数据类型	是否允许为空值	默认值	说明
产品编号	char(8)	×	无	主键
产品名称	char(12)	×	无	
价格	number(8)	×	无	
库存量	number(4)	√	无	

假设 CP 表中已有如表 5.2 所示的数据。

表 5.2　CP 表数据

产品编号	产品名称	价格	库存量
10001100	冰箱 A_100	1500.00	500
10002120	冰箱 A_200	1850.00	200
20011001	空调 K_1200	2680.00	300
20012000	空调 K_2100	3200.00	1000
30003001	冰柜 L_150	5000.00	100
10001200	冰箱 B_200	1600.00	1200
10001102	冰箱 C_210	1890.00	600
30004100	冰柜 L_210	4800.00	200
20001002	空调 K_3001	3800.00	280
20011600	空调 K_1600	4200.00	1500

下列语句将列出产品名称和产品总值：

```
SELECT 产品名称,  价格 * 库存量 AS 产品总值
    FROM CP;
```

执行结果如图 5.4 所示。

	产品名称	产品总值
1	冰箱A_100	750000
2	冰箱A_200	370000
3	空调K_1200	804000
4	空调K_2100	3200000
5	冰柜L_150	500000
6	冰箱B_200	1920000
7	冰箱C_210	1134000
8	冰柜L_210	960000
9	空调K_3001	1064000
10	空调K_1600	6300000

图 5.4　例 5.5 执行结果

计算列值使用算术运算符：+(加)、-(减)、*(乘)、/(除),它们均可用于数值类型的列的计算。例如,语句“SELECT 产品编号, 价格 * 0.8 FROM CP”列出的是每种产品的编号和其打 8 折后的单价。

4. 消除结果集里的重复行

对表只选择某些列时,可能会出现重复行。例如,若对 XSCJ 数据库的 XSB 表只选择专业和总学分,就会出现多行重复的情况。可以使用 DISTINCT 关键字消除结果集里的重复行,其格式是：

```
SELECT DISTINCT<列名>[,<列名>...]
```

关键字 DISTINCT 的含义是对结果集里的重复行只选择一个,保证行的唯一性。

【例 5.6】 对 XSCJ 数据库的 XSB 表只选择专业和总学分,消除结果集里的重复行。

```
SELECT DISTINCT 专业,总学分
    FROM XSB;
```

执行结果如图 5.5 所示。

	专业	总学分
1	计算机	50
2	通信工程	40
3	通信工程	44
4	通信工程	50
5	计算机	52
6	通信工程	90
7	计算机	54
8	计算机	48
9	通信工程	42

图 5.5　例 5.6 执行结果

与 DISTINCT 相反,当使用关键字 ALL 时,将保留结果集的所有行。当 SELECT 语句中不写 ALL 与 DISTINCT 时,默认值为 ALL。

5.1.2　选择行

在 Oracle 中,选择行是使用 WHERE 子句指定选择的条件来实现的。

上面已经列举了使用 WHERE 子句给出查询条件的例子,本节将详细讨论 WHERE 子句中查询条件的构成,它们同样适合于选择列的查询。WHERE 子句必须紧跟 FROM 子句之后,其基本格式为:

```
WHERE<条件表达式>
```

其中<*条件表达式*>为查询条件,格式为:

```
<条件表达式>::=
    { [NOT]<判定运算>| (<条件表达式>) }
      [{ AND | OR } [NOT] {<判定运算>| (<条件表达式>) }]
    } [,...n]
```

其中:

```
<判定运算>::=
{    <表达式 1>{=|<|<=| >| >=|<>| !=}<表达式 2>                    /* 比较运算 */
     |<字符串表达式 1>[NOT] LIKE<字符串表达式 2>[ESCAPE '<转义字符>']
                                                                   /* 字符串模式匹配 */
     |<表达式>[NOT] BETWEEN<表达式 1>AND<表达式 2>                  /* 指定范围 */
     |<表达式>IS [NOT] NULL                                        /* 是否空值判断 */
     |<表达式>[NOT] IN (<子查询>|<表达式>[,...n])                  /* IN 子句 */
     | EXIST (<子查询>)                                            /* EXIST 子查询 */
}
```

<*判定运算*>的结果为 TRUE、FALSE 或 UNKNOWN。

从查询条件的构成可以看出,可以将多个判定运算的结果通过逻辑运算符再组成更为复杂的查询条件。判定运算包括比较运算、模式匹配、范围比较、空值比较和子查询。

在使用字符串和日期数据进行比较时,注意要符合以下限制。

(1) 字符串和日期必须用单引号括起来。

(2) 字符串数据区分大小写。

(3) 日期数据的格式是敏感的,默认的日期格式是 DD-MM 月-YY,可以使用之前的 ALTER SESSION 语句将默认日期修改为 YYYY-MM-DD。

说明:IN 关键字既可以指定范围,也可以表示子查询。

在 SQL 中，返回逻辑值(TRUE 或 FALSE)的运算符或关键字都可称为谓词。

1. 表达式比较

比较运算符用于比较两个表达式值，共有 7 个：=(等于)、<(小于)、<=(小于等于)、>(大于)、>=(大于等于)、<>(不等于)、!=(不等于)。

比较运算的格式为：

```
<表达式 1>{=|<|<=|>|>=|<>|!=}<表达式 2>
```

当两个表达式值均不为空值(NULL)时，比较运算返回逻辑值 TRUE(真)或 FALSE(假)；而当两个表达式值中有一个为空值或都为空值时，比较运算将返回 UNKNOWN。

【例 5.7】 比较运算符的应用。

① 查询 CP 表中库存量在 500 以上的产品情况。

```
SELECT *
    FROM CP
    WHERE 库存量>500;
```

执行结果如图 5.6 所示。

	产品编号	产品名称	价格	库存里
1	20012000	空调K_2100	3200	1000
2	10001200	冰箱B_200	1600	1200
3	10001102	冰箱C_210	1890	600
4	20011600	空调K_1600	4200	1500

图 5.6　例 5.7①执行结果

② 查询 XSB 表中通信工程专业总学分大于等于 44 的学生的情况。

```
SELECT *
    FROM XSB
    WHERE 专业='通信工程' AND 总学分>=44;
```

执行结果如图 5.7 所示。

	学号	姓名	性别	出生时间	专业	总学分	备注
1	151210	李红庆	男	1996-05-01	通信工程	44	已提前修完一门课，并获得学分
2	151241	罗林琳	女	1997-01-30	通信工程	50	转专业学习
3	151242	周何骏	男	1998-09-25	通信工程	90	辅修计算机专业

图 5.7　例 5.7②执行结果

2. 模式匹配

LIKE 谓词用于指出一个字符串是否与指定的字符串相匹配，其运算对象可以是 CHAR、VARCHAR2 和 DATE 类型的数据，返回逻辑值 TRUE 或 FALSE。LIKE 谓词表达式的格式为：

```
<字符串表达式 1>[NOT] LIKE<字符串表达式 2>[ESCAPE '<转义字符>']
```

在使用 LIKE 时，可以使用两个通配符：%和_。若使用带%通配符的 LIKE 进行字符串比较，模式字符串中的所有字符都有意义，包括起始或尾随空格。如果只是希望在模糊条件中表示一个字符，那么可以使用_通配符，使用 NOT LIKE 与 LIKE 的作用相反。

ESCAPE 子句可指定转义字符，转义字符必须为单个字符。当模式串中含有与通配符相同的字符时，应通过转义字符指明其为模式串中的一个匹配字符。

【例 5.8】 查询 CP 表中产品名含有“冰箱”的产品情况。

```
SELECT *
```

```
    FROM CP
    WHERE 产品名称 LIKE '%冰箱%';
```

执行结果如图 5.8 所示。

	产品编号	产品名称	价格	库存里
1	10001100	冰箱A_100	1500	500
2	10002120	冰箱A_200	1850	200
3	10001200	冰箱B_200	1600	1200
4	10001102	冰箱C_210	1890	600

图 5.8　例 5.8 执行结果

【例 5.9】 查询 XSB 表中姓“王”且单名的学生情况。

```
SELECT *
    FROM XSB
    WHERE 姓名 LIKE '王_';
```

执行结果如图 5.9 所示。

	学号	姓名	性别	出生时间	专业	总学分	备注
1	151101	王林	男	1997-02-10	计算机	50	(null)
2	151103	王燕	女	1996-10-06	计算机	50	(null)
3	151201	王敏	男	1996-06-10	通信工程	42	(null)
4	151202	王林	男	1996-01-29	通信工程	40	有一门课不及格，待补考

图 5.9　例 5.9 执行结果

3. 范围比较

用于范围比较的关键字有两个：BETWEEN 和 IN。

当要查询的条件是某个值的范围时，可以使用 BETWEEN 关键字。BETWEEN 关键字指出查询范围，格式为：

```
<表达式>[NOT] BETWEEN<表达式 1>AND<表达式 2>
```

当不使用 NOT 时，若表达式的值在表达式 1 与表达式 2 之间(包括这两个值)，则返回 TRUE，否则返回 FALSE；使用 NOT 时，返回值刚好相反。

注意：表达式 1 的值不能大于表达式 2 的值。

【例 5.10】 指定查询的范围。

① 查询 CP 表中价格为 2000～4000 元的产品情况。

```
SELECT *
    FROM CP
    WHERE 价格 BETWEEN 2000 AND 4000;
```

执行结果如图 5.10 所示。

	产品编号	产品名称	价格	库存里
1	20011001	空调K_1200	2680	300
2	20012000	空调K_2100	3200	1000
3	20001002	空调K_3001	3800	280

图 5.10　例 5.10①执行结果

② 查询 XSB 表中不在 1996 年出生的学生情况。

```
SELECT *
    FROM XSB
    WHERE 出生时间  NOT  BETWEEN  TO_DATE('19960101', 'YYYYMMDD')
                  AND TO_DATE('19961231', 'YYYYMMDD');
```

执行结果如图 5.11 所示。

使用 IN 关键字可以指定一个值表，值表中列出所有可能的值，当表达式与值表中的任意一个匹配时，即返回 TRUE，否则返回 FALSE。使用 IN 关键字指定值表的格式为：

	学号	姓名	性别	出生时间	专业	总学分	备注
1	151101	王林	男	1997-02-10	计算机	50	(null)
2	151104	韦严平	男	1997-08-26	计算机	50	(null)
3	151102	程明	男	1998-02-01	计算机	50	(null)
4	151106	李方方	男	1997-11-20	计算机	50	(null)
5	151107	李明	男	1997-05-01	计算机	54	提前修完数据结构课，并获学分
6	151110	张蔚	女	1998-07-22	计算机	50	三好生
7	151111	赵琳	女	1997-03-18	计算机	50	(null)
8	151203	王玉民	男	1997-03-26	通信工程	42	(null)
9	151218	孙研	男	1997-10-09	通信工程	42	(null)
10	151220	吴薇华	女	1997-03-18	通信工程	42	(null)
11	151241	罗林琳	女	1997-01-30	通信工程	50	转专业学习
12	151242	周何骏	男	1998-09-25	通信工程	90	辅修计算机专业

图 5.11　例 5.10②执行结果

```
<表达式>IN (<表达式>[,...n])
```

【例 5.11】 查询 CP 表中库存量为 200、300 和 500 的产品情况。

```
SELECT *
    FROM CP
    WHERE 库存量 IN(200,300,500);
```

该语句与下列语句等价：

```
SELECT *
    FROM CP
    WHERE 库存量=200 OR 库存量=300 OR 库存量=500;
```

执行结果如图 5.12 所示。

	产品编号	产品名称	价格	库存量
1	10001100	冰箱A_100	1500	500
2	10002120	冰箱A_200	1850	200
3	20011001	空调K_1200	2680	300
4	30004100	冰柜L_210	4800	200

图 5.12　例 5.11 执行结果

说明：IN 关键字最主要的作用是表达子查询，关于子查询的内容稍后具体介绍。

4. 空值比较

当需要判定一个表达式的值是否为空值时，使用 IS NULL 关键字，格式为：

```
<表达式>IS [NOT] NULL
```

当不使用 NOT 时，若表达式的值为空值，返回 TRUE，否则返回 FALSE；当使用 NOT 时，结果刚好相反。

【例 5.12】 查询 XSB 表中拥有备注信息的学生情况。

```
SELECT *
    FROM XSB
    WHERE 备注 IS NOT NULL;
```

执行结果如图 5.13 所示。

5. 子查询

在查询条件中，可以使用另一个查询的结果作为条件的一部分，例如判定列值是否与某个查询的结果集里的值相等，作为查询条件一部分的查询称为子查询。SQL 允许 SELECT 多层嵌套使用，用来表示复杂的查询。子查询除了可以用在 SELECT 语句中外，还可以用

	学号	姓名	性别	出生时间	专业	总学分	备注
1	151107	李明	男	1997-05-01	计算机	54	提前修完数据结构课，并获学分
2	151108	林一帆	男	1996-08-05	计算机	52	已提前修完一门课
3	151110	张蔚	女	1998-07-22	计算机	50	三好生
4	151113	严红	女	1996-08-11	计算机	48	有一门课不及格，待补考
5	151202	王林	男	1996-01-29	通信工程	40	有一门课不及格，待补考
6	151210	李红庆	男	1996-05-01	通信工程	44	已提前修完一门课，并获得学分
7	151241	罗林琳	女	1997-01-30	通信工程	50	转专业学习
8	151242	周何骏	男	1998-09-25	通信工程	90	辅修计算机专业

图 5.13　例 5.12 执行结果

在 INSERT、UPDATE 以及 DELETE 语句中。

子查询通常与谓词 IN、EXIST 及比较运算符结合使用。

(1) IN 子查询。IN 子查询用于进行一个给定值是否在子查询结果集中的判断，格式为：

```
<表达式>[NOT] IN  (<子查询>)
```

当表达式与子查询的结果表中的某个值相等时，IN 谓词返回 TRUE，否则返回 FALSE；若使用了 NOT，则返回的值刚好相反。

【例 5.13】 在 XSCJ 数据库中查找选修了课程号为 102 的课程的学生的情况。

```
SELECT *
    FROM XSB
    WHERE 学号 IN
    (SELECT 学号 FROM CJBWHERE 课程号='102');
```

在执行包含子查询的 SELECT 语句时，系统先执行子查询，产生一个结果表，再执行查询。本例中，先执行子查询：

```
SELECT 学号
    FROM CJB
    WHERE 课程号='102';
```

得到一个只含有学号列的表，CJB 中课程名列值为 102 的行在结果表中都有一行。再执行外查询，若 XSB 表中某行的学号列值等于子查询结果表中的任一个值，则该行就被选择。

执行结果如图 5.14 所示。

	学号	姓名	性别	出生时间	专业	总学分	备注
1	151101	王林	男	1997-02-10	计算机	50	(null)
2	151104	韦严平	男	1997-08-26	计算机	50	(null)
3	151102	程明	男	1998-02-01	计算机	50	(null)
4	151103	王燕	女	1996-10-06	计算机	50	(null)
5	151106	李方方	男	1997-11-20	计算机	50	(null)
6	151107	李明	男	1997-05-01	计算机	54	提前修完数据结构课，并获学分
7	151108	林一帆	男	1996-08-05	计算机	52	已提前修完一门课
8	151109	张强民	男	1996-08-11	计算机	50	(null)
9	151110	张蔚	女	1998-07-22	计算机	50	三好生
10	151111	赵琳	女	1997-03-18	计算机	50	(null)
11	151113	严红	女	1996-08-11	计算机	48	有一门课不及格，待补考

图 5.14　例 5.13 执行结果

注意：IN 和 NOT IN 子查询只能返回一列数据。对于较复杂的查询，可使用嵌套的子查询。

【例 5.14】　查找未选修离散数学的学生的情况。

```
SELECT 学号, 姓名, 专业, 总学分
    FROM XSB
    WHERE 学号 NOT IN
        (SELECT 学号
            FROM CJB
            WHERE 课程号 IN
                (SELECT 课程号
                FROM KCB
                WHERE 课程名='离散数学'
            )
        );
```

执行结果如图 5.15 所示。

	学号	姓名	专业	总学分
1	151201	王敏	通信工程	42
2	151202	王林	通信工程	40
3	151203	王玉民	通信工程	42
4	151204	马琳琳	通信工程	42
5	151206	李计	通信工程	42
6	151210	李红庆	通信工程	44
7	151216	孙祥欣	通信工程	42
8	151218	孙研	通信工程	42
9	151220	吴薇华	通信工程	42
10	151221	刘燕敏	通信工程	42
11	151241	罗林琳	通信工程	50
12	151242	周何骏	通信工程	90

图 5.15　例 5.4 执行结果

（2）比较子查询。这种子查询可以认为是 IN 子查询的扩展，它使表达式的值与子查询的结果进行比较运算，格式为：

```
<表达式>{<|<=|=|>|>=|!=|<>  }{ ALL | SOME | ANY }
(<子查询>)
```

其中，ALL、SOME 和 ANY 关键字说明对比较运算的限制。ALL 指定表达式要与子查询结果集中的每个值都进行比较，当表达式与每个值都满足比较的关系时，才返回 TRUE，否则返回 FALSE；SOME 或 ANY 表示表达式只要与子查询结果集中的某个值满足比较的关系时，就返回 TRUE，否则返回 FALSE。

【例 5.15】　查找比所有计算机系学生年龄都大的学生。

```
SELECT *
    FROM XSB
    WHERE 出生时间<ALL
        (SELECT 出生时间
            FROM XSB
            WHERE 专业='计算机'
        );
```

执行结果如图 5.16 所示。

	学号	姓名	性别	出生时间	专业	总学分	备注
1	151201	王敏	男	1996-06-10	通信工程	42	(null)
2	151210	李红庆	男	1996-05-01	通信工程	44	已提前修完一门课，并获得学分
3	151216	孙祥欣	男	1996-03-19	通信工程	42	(null)
4	151204	马琳琳	女	1996-02-10	通信工程	42	(null)
5	151202	王林	男	1996-01-29	通信工程	40	有一门课不及格，待补考

图 5.16　例 5.15 执行结果

【例 5.16】　查找课程号 206 的成绩不低于课程号 101 的最低成绩的学生的学号、姓名。