

## 第 5 章 数据库的查询和视图

数据查询是数据库系统应用的主要内容,保存数据就是为了使用,使用数据首先要查找到需要的数据。在 T-SQL 中使用 SELECT 语句实现数据查询。用户通过 SELECT 语句可以从数据库中搜寻用户所需要的数据,也可以进行数据的统计汇总并返回给用户。SELECT 语句是数据库操作中使用频率最高的语句,是 SQL 的“灵魂”。

通过学习本章,读者应掌握以下内容:

- 掌握各种查询方法的语法格式和使用,包括单表条件查询、单表多条件查询、多表多条件查询、嵌套查询,并能对查询结果进行排序、分组和汇总操作;
- 掌握视图的建立、修改、使用和删除操作,并能通过视图查询数据、修改数据、更新数据和删除数据。

本章将以 jxgl 数据库为例,在“学生信息”“课程”“成绩”“系部”等表的基础上介绍有关数据查询的技术,包括基本查询和高级查询。在进行查询之前,各表中应当已输入相应记录,各表中的记录分别如图 5.1~图 5.5 所示。

| stu.id     | stu.name | stu.sex | stu.birth       | stu.birthplace | dept.id | stu.telephone | credit | photo |
|------------|----------|---------|-----------------|----------------|---------|---------------|--------|-------|
| 1801010101 | 秦建兴      | 男       | 2000-05-05 0... | 北京市            | 0101    | 18401101456   | 13     | NULL  |
| 1801010102 | 张吉智      | 男       | 2000-12-05 0... | 上海市            | 0101    | 13802104456   | 13     | NULL  |
| 1801010103 | 王胜男      | 女       | 1999-11-08 0... | 广州市            | 0101    | 18624164512   | 13     | NULL  |
| 1801010104 | 李楠楠      | 女       | 2000-08-25 0... | 重庆市            | 0101    | 13902211423   | 4      | NULL  |
| 1801010105 | 耿明       | 男       | 2000-07-15 0... | 北京市            | 0101    | 18501174581   | 13     | NULL  |
| 1801020101 | 贾志强      | 男       | 2000-04-29 0... | 天津市            | 0102    | 15621010025   | 13     | NULL  |
| 1801020102 | 朱凡       | 男       | 2000-05-01 0... | 石家庄市           | 0102    | 13896308457   | 0      | NULL  |
| 1801020103 | 沈柯幸      | 女       | 1999-12-31 0... | 哈尔滨市           | 0102    | 15004511439   | 13     | NULL  |
| 1801020104 | 牛不文      | 女       | 2000-02-14 0... | 长沙市            | 0102    | 13316544789   | 13     | NULL  |
| 1801020105 | 王东东      | 男       | 2000-03-05 0... | 北京市            | 0102    | 18810111256   | 13     | NULL  |
| 1902030101 | 耿娇       | 女       | 2001-05-25 0... | 广州市            | 0203    | 15621014488   | 10     | NULL  |
| 1902030102 | 王向阳      | 男       | 2001-03-15 0... | 北京市            | 0203    | 18810101014   | 10     | NULL  |
| 1902030103 | 郭波       | 女       | 2001-10-05 0... | 重庆市            | 0203    | 18940110111   | 10     | NULL  |
| 1902030104 | 李虹       | 女       | 2001-09-05 0... | 上海市            | 0203    | 13802101458   | 3      | NULL  |
| 1902030105 | 王光伟      | 男       | 2001-01-25 0... | 哈尔滨市           | 0203    | 18945103256   | 10     | NULL  |
| 2002030101 | 于友荣      | 男       | 2002-05-06 0... | 上海市            | 0203    | 15567238922   | 0      | NULL  |
| NULL       | NULL     | NULL    | NULL            | NULL           | NULL    | NULL          | NULL   | NULL  |

图 5.1 “学生信息”表记录

| course.id | course.name | course.credit | course.hour |
|-----------|-------------|---------------|-------------|
| 010001    | 大学计算机基      | 3             | 54          |
| 010002    | 数据结构        | 3             | 54          |
| 010003    | 数据库原理       | 4             | 68          |
| 010004    | 操作系统原理      | 4             | 68          |
| 020001    | 金融学         | 4             | 68          |
| 100101    | 马克思主义基      | 3             | 54          |
| 200101    | 大学英语        | 3             | 54          |
| NULL      | NULL        | NULL          | NULL        |

图 5.2 “课程”表记录

| MS-SDUZHSTNECA.jxgl - dbo.成绩 |           |       |  | MS-SDUZHSTNECA.jxgl - dbo.成绩 |           |       |  |
|------------------------------|-----------|-------|--|------------------------------|-----------|-------|--|
| stu_id                       | course_id | score |  | stu_id                       | course_id | score |  |
| 1801010101                   | 010002    | 78    |  | 1801020102                   | 200101    | 52    |  |
| 1801010101                   | 010003    | 69    |  | 1801020103                   | 010002    | 65    |  |
| 1801010101                   | 100101    | 89    |  | 1801020103                   | 010004    | 65    |  |
| 1801010101                   | 200101    | 77    |  | 1801020103                   | 100101    | 87    |  |
| 1801010102                   | 010002    | 88    |  | 1801020103                   | 200101    | 67    |  |
| 1801010102                   | 010003    | 84    |  | 1801020104                   | 010002    | 84    |  |
| 1801010102                   | 100101    | 75    |  | 1801020104                   | 010004    | 84    |  |
| 1801010102                   | 200101    | 85    |  | 1801020104                   | 100101    | 92    |  |
| 1801010103                   | 010002    | 75    |  | 1801020104                   | 200101    | 90    |  |
| 1801010103                   | 010003    | 81    |  | 1801020105                   | 010002    | 76    |  |
| 1801010103                   | 100101    | 78    |  | 1801020105                   | 010004    | 76    |  |
| 1801010103                   | 200101    | 76    |  | 1801020105                   | 100101    | 64    |  |
| 1801010104                   | 010002    | 45    |  | 1801020105                   | 200101    | 66    |  |
| 1801010104                   | 010003    | 61    |  | 1902030101                   | 020001    | 85    |  |
| 1801010104                   | 100101    | 55    |  | 1902030101                   | 100101    | 85    |  |
| 1801010104                   | 200101    | 56    |  | 1902030101                   | 200101    | 83    |  |
| 1801010105                   | 010002    | 83    |  | 1902030102                   | 020001    | 66    |  |
| 1801010105                   | 010003    | 86    |  | 1902030102                   | 100101    | 66    |  |
| 1801010105                   | 100101    | 81    |  | 1902030102                   | 200101    | 61    |  |
| 1801010105                   | 200101    | 81    |  | 1902030103                   | 020001    | 79    |  |
| 1801020101                   | 010002    | 91    |  | 1902030103                   | 100101    | 79    |  |
| 1801020101                   | 010004    | 91    |  | 1902030103                   | 200101    | 89    |  |
| 1801020101                   | 100101    | 77    |  | 1902030104                   | 020001    | 49    |  |
| 1801020101                   | 200101    | 87    |  | 1902030104                   | 100101    | 49    |  |
| 1801020102                   | 010002    | 56    |  | 1902030104                   | 200101    | 69    |  |
| 1801020102                   | 010004    | 56    |  | 1902030105                   | 020001    | 69    |  |
| 1801020102                   | 100101    | 45    |  | 1902030105                   | 100101    | 69    |  |
| 1801020102                   | 200101    | 52    |  | 1902030105                   | 200101    | 62    |  |
| 1801020103                   | 010002    | 65    |  | NULL                         | NULL      | NULL  |  |

图 5.3 “成绩”表记录

| MS-SDUZHSTNECA.jxgl - dbo.系部 |           |           |  |
|------------------------------|-----------|-----------|--|
| dept_id                      | dept_name | dept_head |  |
| 0101                         | 数学系       | 张光明       |  |
| 0102                         | 计算机系      | 王新亮       |  |
| 0203                         | 金融系       | 刘娟        |  |
| 0301                         | 外语系       | 辛英红       |  |
| NULL                         | NULL      | NULL      |  |

图 5.4 “系部”表记录

| MS-SDUZHSTNECA.jxgl - dbo.教师信息 |              |             |         | MS-SDUZHSTNECA.jxgl - dbo.系部 |           | MS-SDUZHSTNECA.jxgl - dbo.成绩 |  |
|--------------------------------|--------------|-------------|---------|------------------------------|-----------|------------------------------|--|
| teacher_id                     | teacher_name | teacher_sex | dept_id | teacher_birth                | dept_head | teacher_telep...             |  |
| 010101                         | 张光明          | 男           | 0101    | 1971-11-02                   | True      | 13901023415                  |  |
| 010102                         | 李秀丽          | 女           | 0101    | 1988-02-24                   | False     | 18720119821                  |  |
| 010201                         | 王新亮          | 男           | 0102    | 1972-04-08                   | True      | 18801214533                  |  |
| 010202                         | 张诗涵          | 女           | 0102    | 1992-07-16                   | False     | 13801215627                  |  |
| 010203                         | 吴建设          | 男           | 0102    | 1989-09-22                   | False     | 15501128976                  |  |
| 020301                         | 刘娟           | 女           | 0203    | 1980-07-03                   | True      | 13301221789                  |  |
| 020302                         | 赵立勇          | 男           | 0203    | 1985-05-01                   | False     | 18701227899                  |  |
| 020305                         | 付华           | 女           | 0203    | 1990-09-15                   | False     | 18514510011                  |  |
| NULL                           | NULL         | NULL        | NULL    | NULL                         | NULL      | NULL                         |  |

图 5.5 “教师信息”表记录



简单 SELECT 语句

## 5.1 简单 SELECT 语句

简单查询是按照一定的条件在单表上查询数据,还包括汇总查询以及查询结果的排序与保存。

### 5.1.1 SELECT 语句概述

查询是指对已经存在于数据库中的数据按特定的组合、条件或次序进行检索,从数据库中获取数据和操作数据的过程。查询功能是数据库最基本也是最重要的功能。

查询是 SQL 中最主要、最核心的部分。查询语言用来对已经存在于数据库的数据按照特定组合、条件表达式或者一定的次序进行检索。

数据查询命令是 SQL 最常用的命令,由于查询要求不同而有各种变化,因此查询命令也是最复杂的命令。其基本格式是由 SELECT 子句、FROM 子句和 WHERE 子句组成的 SQL 查询语句。

### 5.1.2 完整的 SELECT 语句的基本语法格式

SELECT 语句的完整语法格式如下:

```
SELECT [ALL|DISTINCT] [TOP n [PERCENT]] select_list
    [INTO new_table_name]
    FROM <table_name/view_name>
    [WHERE search_condition]
    [GROUP BY group_by_expression]
    [HAVING search_condition]
    [ORDER BY order_expression[ASC|DESC]][,...n]
```

对其中各参数的说明如下。

- (1) ALL: 表示输出所有记录,包括重复记录。
- (2) DISTINCT: 表示去掉重复的记录。
- (3) TOP  $n$ : 查询结果只显示表中的前  $n$  条记录, TOP  $n$  PERCENT 关键字表示查询结果只显示前面  $n\%$  条记录。
- (4) select\_list: 所要查询的选项的集合,多个选项之间用逗号分开。
- (5) INTO new\_table\_name: 用于指定使用结果集创建一个新表,new\_table\_name 是新表名。
- (6) FROM <table\_name/view\_name>: 结果集数据来源于哪些表或视图, FROM 子句还可包含连接的定义。
- (7) WHERE search\_condition: 一个条件筛选,只有符合条件的行才向结果集提供数据,不符合条件的行中的数据不会被使用。

(8) GROUP BY group\_by\_expression: 根据 group\_by\_expression 列中的值将查询结果进行分组。

(9) HAVING search\_condition: 应用于结果集的附加筛选。从逻辑上讲,HAVING 子句从中间结果集对行进行筛选,这些中间结果集是用 SELECT 语句中的 FROM、WHERE 或 GROUP BY 子句创建的。HAVING 子句通常与 GROUP BY 子句一起使用。

(10) ORDER BY order\_expression[ASC|DESC]: 定义结果集中行的排列顺序。order\_expression 指定组成排序列表的结果集的列。ASC 指定行按升序排序,DESC 指定行按降序排序。

SELECT 语句可以完成以下工作。

- 投影: 用来选择表中的列。
- 选择: 用来选择表中的行。
- 连接: 将两个关系拼接成一个关系。

SELECT 语句的功能为从 FROM 列出的数据源表中找出满足 WHERE 检索条件的记录,按 SELECT 子句的字段列表输出查询结果表,在查询结果表中可进行分组与排序。

**说明:** 在 SELECT 语句中,SELECT 子句和 FROM 子句是不可缺少的,其余的是可选的。

### 5.1.3 基本的 SELECT 语句

SELECT 语句的基本形式如下:

```
SELECT [ALL|DISTINCT] [TOP n [PERCENT]] select_list
      FROM <table_name/view_name>
      [WHERE search_condition]
```

#### 1. 查询表中的若干列

在很多情况下,用户只对表中的一部分属性感兴趣,这时可以在 SELECT 子句的<字段列表>中指定要查询的属性。

**【例 5.1】** 在“学生信息”表中查询学生的学号及姓名。

```
USE jxgl
GO
SELECT stu_id,stu_name
      FROM 学生信息
GO
```

查询结果如图 5.6 所示。

#### 2. 查询表中的全部列

将表中的所有属性都选出来有两种方法:一种方法是在 SELECT 命令后面列出所有列名;另一种方法是如果列的显示顺序与其在基表中的顺序相同,也可以简单地将<字

段列表>简写为“\*”。

**【例 5.2】** 查询“课程”表的所有信息。

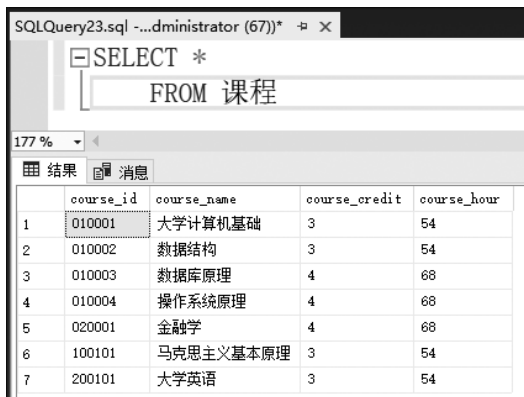
```
USE jxgl
GO
SELECT *
    FROM 课程
GO
```

查询结果如图 5.7 所示。



|    | stu_id     | stu_name |
|----|------------|----------|
| 1  | 1801010101 | 秦建兴      |
| 2  | 1801010102 | 张吉哲      |
| 3  | 1801010103 | 王胜男      |
| 4  | 1801010104 | 李楠楠      |
| 5  | 1801010105 | 耿明       |
| 6  | 1801020101 | 贾志强      |
| 7  | 1801020102 | 朱凡       |
| 8  | 1801020103 | 沈柯辛      |
| 9  | 1801020104 | 牛不文      |
| 10 | 1801020105 | 王东东      |
| 11 | 1902030101 | 耿娇       |
| 12 | 1902030102 | 王向阳      |
| 13 | 1902030103 | 郭波       |
| 14 | 1902030104 | 李红       |
| 15 | 1902030105 | 王光伟      |

图 5.6 查询“学生信息”指定列的结果



|   | course_id | course_name | course_credit | course_hour |
|---|-----------|-------------|---------------|-------------|
| 1 | 010001    | 大学计算机基础     | 3             | 54          |
| 2 | 010002    | 数据结构        | 3             | 54          |
| 3 | 010003    | 数据库原理       | 4             | 68          |
| 4 | 010004    | 操作系统原理      | 4             | 68          |
| 5 | 020001    | 金融学         | 4             | 68          |
| 6 | 100101    | 马克思主义基本原理   | 3             | 54          |
| 7 | 200101    | 大学英语        | 3             | 54          |

图 5.7 查询“课程”表的全部列

### 3. 设置字段别名

T-SQL 提供了在 SELECT 语句中操作别名的方法。用户可以根据实际需要对查询数据的列标题进行修改,或者为没有标题的列加上临时标题。其语法格式如下:

列表表达式 [AS] 别名

或

别名=列表表达式

**【例 5.3】** 查询 jxgl 数据库的“课程”表,列出表中的所有记录,每个记录的名称依次为课程号、课程名、课程学分和学时数。

```
USE jxgl
GO
SELECT course_id AS 课程号, course_name AS 课程名, course_credit AS 课程学分,
```

```

course_hour AS 学时数
FROM 课程
GO

```

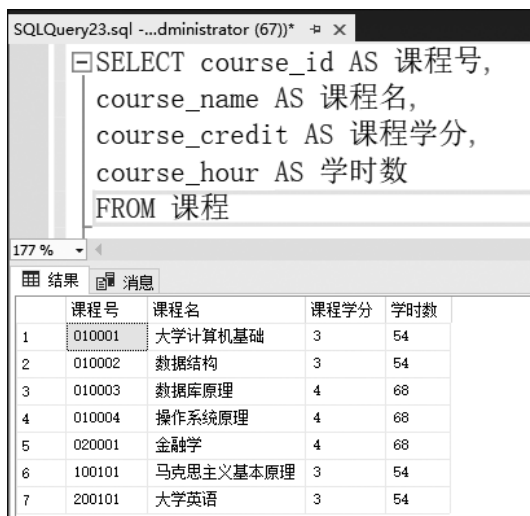
或

```

SELECT 课程号=course_id, 课程名=course_name, 课程学分=course_credit, 学时数=
course_hour
FROM 课程
GO

```

查询结果如图 5.8 所示。



The screenshot shows a SQL query window titled 'SQLQuery23.sql - ...Administrator (67))' with the following query:

```

SELECT course_id AS 课程号,
course_name AS 课程名,
course_credit AS 课程学分,
course_hour AS 学时数
FROM 课程

```

Below the query, the results are displayed in a table with columns: 课程号, 课程名, 课程学分, 学时数. The results are as follows:

|   | 课程号    | 课程名       | 课程学分 | 学时数 |
|---|--------|-----------|------|-----|
| 1 | 010001 | 大学计算机基础   | 3    | 54  |
| 2 | 010002 | 数据结构      | 3    | 54  |
| 3 | 010003 | 数据库原理     | 4    | 68  |
| 4 | 010004 | 操作系统原理    | 4    | 68  |
| 5 | 020001 | 金融学       | 4    | 68  |
| 6 | 100101 | 马克思主义基本原理 | 3    | 54  |
| 7 | 200101 | 大学英语      | 3    | 54  |

图 5.8 显示字段别名

#### 4. 查询经过计算的值

SELECT 子句中的 <select\_list> 不仅可以是表中的属性列,也可以是表达式,包括字符串常量、函数等,其语法格式如下:

计算字段名=表达式

**【例 5.4】** 查询“学生信息”表中所有学生的学号、姓名及年龄。

在本例的查询操作中应该使用“学生信息”表中的 stu\_birth 字段值计算得到,这里需要用到两个函数,一个是取得当前系统日期的函数 GETDATE(),另一个是计算两个日期型量之差的函数 DATEDIFF(),本例计算当前日期与学生出生时间之间年份的差值,通过这种方式得到学生的年龄。

```

USE jxgl
GO
SELECT stu_id AS 学号, stu_name AS 姓名, 年龄=DATEDIFF (YY, stu_birth, GETDATE ())
FROM 学生信息

```

GO

查询结果如图 5.9 所示。

```

SELECT stu_id AS 学号,
       stu_name AS 姓名,
       年龄=DATEDIFF(YY, stu_birth, GETDATE())
FROM 学生信息
  
```

|    | 学号         | 姓名  | 年龄 |
|----|------------|-----|----|
| 1  | 1801010101 | 秦建兴 | 20 |
| 2  | 1801010102 | 张吉哲 | 20 |
| 3  | 1801010103 | 王胜男 | 21 |
| 4  | 1801010104 | 李楠楠 | 20 |
| 5  | 1801010105 | 耿明  | 20 |
| 6  | 1801020101 | 贾志强 | 20 |
| 7  | 1801020102 | 朱凡  | 20 |
| 8  | 1801020103 | 沈柯辛 | 21 |
| 9  | 1801020104 | 牛不文 | 20 |
| 10 | 1801020105 | 王东东 | 20 |
| 11 | 1902030101 | 耿娇  | 19 |
| 12 | 1902030102 | 王向阳 | 19 |
| 13 | 1902030103 | 郭波  | 19 |
| 14 | 1902030104 | 李红  | 19 |
| 15 | 1902030105 | 王光伟 | 19 |

图 5.9 显示经过计算的年龄字段

## 5. 返回全部记录

如果要返回全部记录可在 SELECT 后使用 ALL, ALL 是默认设置, 因此也可以省略。

**【例 5.5】** 查询“学生信息”表中所有学生的系别代码。

```

USE jxgl
GO
SELECT dept_id
FROM 学生信息
GO
  
```

部分查询结果如图 5.10 所示。

## 6. 过滤重复记录

在例 5.5 的执行结果集中显示了重复行。如果让重复行只显示一次, 需在 SELECT 子句中用过滤重复记录 (DISTINCT) 指定在结果集中只能显示唯一行。

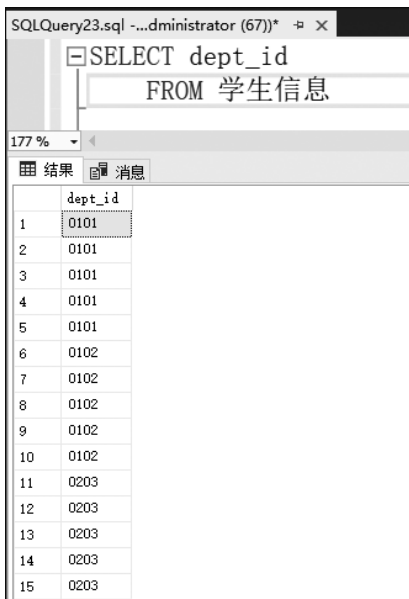
**【例 5.6】** 查询“学生信息”表中的学生所在系别有哪些 (重复专业只显示一次)。

```

USE jxgl
GO
  
```

```
SELECT DISTINCT dept_id
FROM 学生信息
GO
```

查询结果如图 5.11 所示。



|    | dept_id |
|----|---------|
| 1  | 0101    |
| 2  | 0101    |
| 3  | 0101    |
| 4  | 0101    |
| 5  | 0101    |
| 6  | 0102    |
| 7  | 0102    |
| 8  | 0102    |
| 9  | 0102    |
| 10 | 0102    |
| 11 | 0203    |
| 12 | 0203    |
| 13 | 0203    |
| 14 | 0203    |
| 15 | 0203    |

图 5.10 显示所有学生的系别代码



|   | dept_id |
|---|---------|
| 1 | 0101    |
| 2 | 0102    |
| 3 | 0203    |

图 5.11 过滤重复记录后的显示结果

**注意：**在使用 DISTINCT 关键字后,如果表中有多多个为 NULL 的数据,服务器会把这些数据视为相同。

## 7. 仅返回前面若干记录

其语法格式如下：

```
SELECT [TOP n | TOP n PERCENT] select_list
FROM table_name
```

其中,TOP  $n$  表示返回最前面的  $n$  行, $n$  表示返回的行数;

TOP  $n$  PERCENT 表示返回最前面的  $n\%$  行。

**【例 5.7】** 查询“学生信息”表中的前 5 条记录。

```
USE jxgl
GO
SELECT TOP 5 *
FROM 学生信息
GO
```

**【例 5.8】** 查询“学生信息”表中前面的 10% 行记录。

```
USE jxgl
GO
SELECT TOP 10 PERCENT *
    FROM 学生信息
GO
```

运行例 5.7 将返回 5 条记录,运行例 5.8 会返回“学生信息”表最前面 10% 条记录,即前两条记录。

**注意:** TOP 子句不能和 DISTINCT 关键字同时使用。

#### 5.1.4 INTO 子句

使用 INTO 子句允许用户自定义一个新表,并且把 SELECT 子句的数据插入新表中。

在使用 INTO 子句插入数据时,应注意以下 4 点。

- 新表不能存在,否则会产生错误信息。
- 新表中的列和行是基于查询结果集的。
- 使用该子句必须在目的数据库中具有 CREATE TABLE 权限。
- 如果新表名称的开头为“#”,则生成的是临时表。

**注意:** 使用 INTO 子句,通过在 WHERE 子句中设置 FALSE 条件,可以创建一个和源表结构相同的空表。

**【例 5.9】** 创建一个和“学生信息”表结构相同的 xs\_new 表。

```
USE jxgl
GO
SELECT * INTO xs_new
    FROM 学生信息
    WHERE 6>8
GO
```

设置 WHERE 6>8 这样一个明显为逻辑否的条件的目的是只保留“学生信息”表的结构,而不返回任何记录。

**【例 5.10】** 查询所有女生的信息并将结果保存在名为“女生表”的数据表中。

```
USE jxgl
GO
SELECT * INTO 女生表
    FROM 学生信息
    WHERE stu_sex='女'
GO
```

**【例 5.11】** 查询所有男生的信息并将结果存入临时表中。

```
USE jxgl
GO
SELECT * INTO #男生表
```

```

FROM 学生信息
WHERE stu_sex='男'
GO

```

查看临时表的内容可用下面的语句：

```
SELECT * FROM #男生表
```

查询结果如图 5.12 所示。

|   | stu_id     | stu_name | stu_sex | stu_birth               | stu_birthplace | dept_id | stu_telephone | credit | photo |
|---|------------|----------|---------|-------------------------|----------------|---------|---------------|--------|-------|
| 1 | 1801010101 | 秦建兴      | 男       | 2000-05-05 00:00:00.000 | 北京市            | 0101    | 18401101456   | 13     | NULL  |
| 2 | 1801010102 | 张吉哲      | 男       | 2000-12-05 00:00:00.000 | 上海市            | 0101    | 13802104456   | 13     | NULL  |
| 3 | 1801010105 | 耿明       | 男       | 2000-07-15 00:00:00.000 | 北京市            | 0101    | 18501174581   | 13     | NULL  |
| 4 | 1801020101 | 贾志强      | 男       | 2000-04-29 00:00:00.000 | 天津市            | 0102    | 15621010025   | 13     | NULL  |
| 5 | 1801020102 | 朱凡       | 男       | 2000-05-01 00:00:00.000 | 石家庄市           | 0102    | 13896308457   | 0      | NULL  |
| 6 | 1801020105 | 王东东      | 男       | 2000-03-05 00:00:00.000 | 北京市            | 0102    | 18810111256   | 13     | NULL  |
| 7 | 1902030102 | 王向阳      | 男       | 2001-03-15 00:00:00.000 | 北京市            | 0203    | 18810101014   | 10     | NULL  |
| 8 | 1902030105 | 王光伟      | 男       | 2001-01-25 00:00:00.000 | 哈尔滨市           | 0203    | 18945103256   | 10     | NULL  |

图 5.12 查询临时表的结果

### 5.1.5 WHERE 子句

WHERE 子句获取 FROM 子句返回的值(在虚拟表中),并且应用 WHERE 子句中定义的搜索条件。WHERE 子句相当于从 FROM 子句返回结果的筛选器,每行都要根据搜索条件进行评估,评估为真的那些行作为查询结果的一部分返回,评估为未知或假的那些行不出现在结果中。条件查询就是关系运算的选择运算,就是对数据源进行水平分割。

使用 WHERE 子句可以限制查询的记录范围。在使用时,WHERE 子句必须紧跟在 FROM 子句后面。WHERE 子句中的条件是一个逻辑表达式,其中可以包含的运算符如表 5.1 所示。

表 5.1 查询条件中常用的运算符

| 运 算 符                     | 用 途    |
|---------------------------|--------|
| =、<、>、<>、!=、!<、!>、=、<=、!= | 比较大小   |
| AND、OR、NOT                | 设置多重条件 |
| BETWEEN...AND...          | 确定范围   |