

# 实验 3 简单查询-单表查询和视图

## 一、目的和要求

- (1) 掌握简单查询的操作，如：单表查询，模糊查询，分组查询，重新设置输出列的名称，输出列的取值与原字段的关系，简单统计的意义，对查询结果进行排序等；
- (2) 掌握分组查询的使用，理解分组查询的目的；
- (1) 掌握分页浏览数据的方法
- (2) 掌握创建视图的方法
- (3) 掌握利用 AS 为字段重新命名的方法

## 二、背景知识

SELECT 语句的语法格式见表 3.1，共包括 SELECT、FROM、WHERE、GROUP BY、HAVING 和 ORDER BY 等子句。其中，SELECT 子句指明需要查询的项目；FROM 子句指明被查询的基表或视图；WHERE 子句说明查询条件；GROUP BY 子句和 HAVING 对查询结果进行分组和过滤；ORDER BY 子句对查询结果进行排序。SELECT 子句和 FROM 子句是每个 SQL 查询语句所必需的，其他子句是可选的。

表 3.1 SELECT 语句语法格式

语句关键字	语法格式	说 明
SELECT	SELECT [ALL   DISTINCT] <列表表达式> [{,<列表表达式>}] FROM <表标识> [<别名>] [{,<表标识> [<别名>]}] [WHERE <查询条件>] [GROUP BY <列标识>[{,<列标识>}] [HAVING <分组条件>]] [ORDER BY <列标识> <序号> [ASC   DESC] [{,<列标识> <序号> [ASC   DESC]}]]];	SELECT 子句中<列表表达式>是算术表达式，用于投影表中的列，或对列值进行计算。ALL(默认)表明返回查询结果的所有行，不去掉重复行。DISTINCT 对重复行只返回其中一行。
		FROM 子句指明查询的数据来源(基表或视图)。
		WHERE 中的<查询条件>是逻辑表达式。简单条件有比较、BETWEEN、LIKE、IN 和 EXISTS；复合条件是由简单条件、逻辑运算符及括号所组成的逻辑表达式。条件中允许嵌入子查询。
		GROUP BY 子句对已选择的行进行分组；HAVING 子句进一步选择已分的组，对每个已选中的组在查询结果中只返回其单行总计信息。
		ORDER BY 子句对查询结果的显示输出进行排序，ASC(默认)为升序，DESC 为降序，排序时遵循“NULL 值最大”原则。

注：表中有下划线的关键字可省略，表示是缺省值。

SELECT 语句中涉及的条件表达式中可以用表 3.2 所示的运算符。

表 3.2 条件表达式中可使用的运算符

运算符分类	运算符	意义
比较运算符	>、>=、<、<=、=、!=、<>、!<	大小比较
范围运算符	BETWEEN.....-AND NOT BETWEEN.....AND	判断表达式的值是否在指定的范围内
列表运算符	IN   NOT IN	判断表达式值是否为列表中的指定值
模式匹配符	LIKE   NOT LIKE	判断列值是否与指定的字符通配格式相符
空值判断符	IS NULL   IS NOT NULL	判断表达式的值是否为空
逻辑运算符	AND   OR   NOT	用于多个条件的逻辑连接 优先级：NOT>AND>OR

除此之外 SQL 还提供了很多聚集函数，以实现数据的统计。主要的聚焦函数功能如下：

**COUNT([DISTINCT|ALL]\*)** 统计元组个数

**COUNT([DISTINCT|ALL]<列名>)** 统计一列中值的个数

**SUM([DISTINCT|ALL] <列名>)** 计算一列值的总和（此列必须是数值列）

**AVG([DISTINCT|ALL] <列名>)** 计算一列值的平均值（此列必须是数值列）

**MAX([DISTINCT|ALL] <列名>)** 求一列值中的最大值

**MIN([DISTINCT|ALL] <列名>)** 求一列值中的最小值

如果指定 DISTINCT 短语，则表示在计算时要取消指定列的重复值；ALL 为缺省值。

在聚集函数遇到空值时，除 COUNT（\*）外，都跳过空值而只处理非空值。

### 三、实验内容

要求在查询编辑器窗口中选择 XSQL 数据库为当前数据库，且使用 SQL 语句练习单表查询和视图操作。

### 四、实验步骤

#### 1.求学院编号为' 0001' 的学生的学号、姓名、性别

```
SELECT SNO,SNAME,SEX
FROM student
WHERE DNO='0001'
```

#### 2.求学院编号为' 0001' 的男生的学号、姓名、性别

```
SELECT SNO,SNAME,SEX
FROM student
WHERE DNO='0001' AND SEX='男'
```

#### 3.求选修授课班号为‘327401’且成绩在80~90之间的学生学号和成绩，并将成绩乘以系数0.8输出，且将SNO列更名为学号，成绩列更名为处理成绩。

```
SELECT SNO AS 学号,GRADE*0.8 AS 处理成绩
FROM sc
WHERE CNO='327401' AND GRADE BETWEEN 80 AND 90
```

#### 4. 求每个学生的年龄，并输出姓名和年龄

```
SELECT SNAME,YEAR(GETDATE())-YEAR(BIRTHDAY) AS 年龄
FROM student
```

#### 5. 求选修了课程的学生学号

```
SELECT DISTINCT SNO FROM sc
```

说明：注意在 SNO 前加 DISTINCT 与不加 DISTINCT 的区别

#### 6. 求选修授课班号为‘327401’的学生学号和成绩，并要求对查询结果按成绩的降序排列，如果成绩相同则按学号的升序排列

```
SELECT SNO,GRADE
FROM sc
WHERE CNO='327401'
ORDER BY GRADE DESC,SNO ASC
```

7. 求缺少了成绩的学生学号和课程号。

```
SELECT SNO,CNO FROM sc WHERE GRADE IS NULL
```

8. 统计选课学生人数及最高分成绩和最低分成绩

```
SELECT COUNT(DISTINCT SNO) ,MAX(GRADE),MIN(GRADE)
FROM sc
```

9. 求学院编号为'0001'或'0002'中姓张的学生的信息。

```
SELECT *
FROM student
WHERE (DNO='0001' OR DNO='0002') AND SNAME LIKE '张%'
```

10. 求姓名中包含'丽'的学生信息

```
SELECT *
FROM student
WHERE SNAME LIKE '%丽%'
```

11. 求姓名只有两个字,且第二个字为'丽'的学生信息

```
SELECT *
FROM student
WHERE SNAME LIKE '_丽'
```

12. 求信息学院计算机专业的学生名单

```
SELECT *
FROM student
WHERE SUBSTRING(SNO,5,2)='28'
```

13. 统计各个学院的人数

```
SELECT COUNT(SNO) AS 学院总人数 ,DNO AS 部门编号
FROM student
GROUP BY DNO
```

14. 按授课班号统计选修该课程的人数,并按照人数升序排列。

```
SELECT COUNT(SNO) AS 人数统计,CNO
FROM sc
GROUP BY CNO
ORDER BY 人数统计
```

15. 统计平均成绩超过80分的学生的学号及平均成绩

```
SELECT SNO,AVG(GRADE)
FROM sc
GROUP BY SNO
HAVING AVG(GRADE)>= 80
ORDER BY AVG(GRADE)
```

16. 求选修课程超过 6 门课的学生学号，并按选修课程数目升序排列。

```
SELECT SNO,COUNT(*)  
FROM sc  
GROUP BY SNO  
HAVING COUNT(SNO)>6  
ORDER BY COUNT(SNO)
```

17. 求每个学院学生的平均年龄，并把结果存入当前数据库‘系平均年龄’临时表中。

```
SELECT DNO,AVG(AGE) AS 平均 INTO 系平均年龄  
FROM student  
GROUP BY DNO
```

说明：系统自动生成表:系平均年龄临时表，在执行该语句之前，数据库中不能存在‘系平均年龄’表

18. 分页浏览数据方法：

(1) 查询学生库中的第 1-10 名同学的信息

```
SELECT top 10 * FROM student
```

(2) 查询学生库中的第 11-20 名同学的信息

```
SELECT top 10 * FROM student WHERE SNO not IN(SELECT top 10 SNO FROM student )
```

说明：我们在浏览网页时，经常可以看到多行数据可以分页显示，上述方法加上适当的变量控制就能实现

19. 查询‘1987-1-1’号以后出生的女生的学生信息

```
SELECT *  
FROM student  
WHERE birthday>'1987-1-1' AND SEX='女'
```

20. 创建‘计算机系学生’视图，用于浏览计算机系学生的学号、姓名和年龄

```
CREATE VIEW 计算机系学生  
AS SELECT SNO,SNAME,AGE FROM student WHERE SUBSTRING(SNO,5,2)='28'
```

思考模仿题：

1.查询分数在 70 和 90 之间的学生学号

```
SELECT SNO  
FROM sc  
GROUP BY SNO  
HAVING MIN(GRADE)>70 AND MAX(GRADE)<90
```

2.查询少于 10 名同学选修的授课班号

```
SELECT CNO  
FROM sc  
GROUP BY CNO  
HAVING COUNT(*)<10  
ORDER BY CNO
```

### 3.查询选课表中的最高分

```
SELECT MAX(GRADE) AS 最高分
FROM sc
```

### 4.查询授课编号为‘153701’的课程平均分

```
SELECT AVG(GRADE) AS '课程平均分'
FROM sc
WHERE CNO='153701'
```

### 5.查询课程平均分超过 85 的授课班号，输出结果按课程平均分升序排列

```
SELECT CNO,AVG(GRADE)
FROM sc
GROUP BY CNO
HAVING AVG(GRADE)>85
ORDER BY AVG(GRADE)
```

### 6.查询课程名称为‘线性代数’的排课情况

```
SELECT *
FROM course
WHERE CNAME='线性代数'
```

### 7.查询选修授课班号为‘218801’的学生学号

```
SELECT SNO
FROM sc
WHERE CNO='218801'
ORDER BY SNO
```

### 8.按授课班号查询课程的平均分，输出授课班号和平均成绩

```
SELECT CNO,AVG(GRADE)
FROM sc
GROUP BY CNO
```

### 9.在 sc 中输出成绩在 90-100 之间的学生信息

```
SELECT * FROM sc WHERE CONVERT(char(20),GRADE) LIKE '9%'
```

## 思考题

#### 1、在学生管理数据库中，完成以下查询：

- (1) 查询‘周芬’老师，这个学期的上课安排情况
- (2) 查询姓‘周’的教师的排课情况
- (3) 按教室分组统计排课门数情况
- (4) 查询排课门数超过 8 门的教室名单及其排课门数
- (5) 创建机电学院女生的视图
- (6) 查询学分超过 4 分的课程，输出课程名和学分，并要求按学分升序
- (7) 按教室明细并汇总排课情况

- (8) 查询课程号为“203402”的成绩最高的前 5 名学生的学号及成绩，结果按成绩降序
- (9) 查询年龄小于 20 岁的学生学号
- (10) 查询有 90 人以上选修的课程号
- (11) 查询全体男生的姓名，要求查询结果按所在系升序排列，对相同系的学生按姓名升序排列
- (12) 查询成绩在 70-90 范围内的学生学号

## 2、完成书后作业的查询。