

## 实验三 高级查询与视图

### 一、实验目的

- (1) 熟练掌握数据库的连接查询；
- (2) 深入理解 SELECT 语句中嵌套查询的概念；
- (3) 掌握 IN、比较符、ANY、ALL 和 EXISTS 谓词在嵌套查询中的具体应用；
- (4) 掌握视图的创建、查询、更新、修改和删除的方法。

### 二、实验原理

#### 1. 连接的概念

连接可以实现从两个或更多的表中查询数据。

#### 2. 连接的分类：

##### (1) 内连接

是指两个表在连接时，使用比较运算符对表中指定的列进行比较，返回符合连接条件的数据行，从而返回一个由两个表生成的新的记录集。

(2) 自身连接：一个表与其自身进行的比较连接。

##### (3) 外连接

是指两个表在连接时，返回的结果集除了包括复合条件的数据行外，还会返回 FROM 子句中的至少一个表或视图的所有行。外连接有两种类型：

- 左连接：在结果表中包含第一个表中的满足条件的所有记录；如果是在连接条件上匹配的元组，则第二个表返回相应的值，否则第二表返回空值。
- 右连接：在结果表中包含第二个表中的满足条件的所有记录；如果是在连接条件上匹配的元组，则第一个表返回相应的值，否则第一个表返回空值。

#### 3. 连接形式

##### (1) 使用 WHERE 子句连接

在 SELECT 语句的 WHERE 子句中使用比较运算符给出连接条件。其语法格式为：

```
SELECT <列名>[,<列名>]... FROM <表名 1>[,<表名 2>]  
WHERE [ <表名 1>.<列名 1> <比较运算符> <表名 2>.<列名 2>]
```

##### (2) 使用 JOIN 谓词的连接

在 Transact-SQL 中扩展了专门的连接语句，其语法格式为：

```
SELECT <列名>[,<列名>]...  
FROM <表名 1> [连接类型] JOIN <表名 2>  
ON <连接条件> WHERE <查询条件>
```

其中：连接类型的指定方式分别为：

- 内连接: [INNER]
- 外连接: LEFT|RIGHT|FULL|[OUTER]

提示：① JOIN 的顺序和 ON 连接条件的顺序相反。② 外连接只能对两个表进行。

### 4.嵌套查询

一条 SELECT...FROM ...WHERE 语句称为一个查询块。将一个查询块嵌套在另一个查询块的 WHERE 或 HAVING 短语条件中的查询称为嵌套查询。SQL 语句允许多层嵌套查询。

### 5.连接子查询的谓词

#### (1) [NOT] IN 谓词

用于判断一个给定值是否在子查询结果集中，其一般格式为：

<表达式> [NOT] IN （子查询|） 或：<表达式> [NOT] IN (值 1, {值 2, ...})

提示：IN 和 NOT IN 子查询只能返回一列数据。

#### (2) 比较运算符

当用户确切知道内层查询返回值是单值时，可以用比较运算符直接连接子查询。其一般格式为：<表达式> θ （子查询）

其中：θ 代表比较运算符，可以是 =、>、<、>=、<=、!=或<>。

#### (3) SOME、ANY 或 ALL

当内层查询的返回值不是单值时，比较运算符可以与 SOME、ANY 或 ALL 谓词引出的子查询联用。其一般格式为：<表达式> θ {SOME|ANY|ALL} (子查询)

比较运算符与 ANY 或 ALL 搭配使用的取值情况见表 3-1 所示。

表 3-1 比较运算符与 ANY/ALL 搭配的取值

比较运算符	修饰符	取值
>、>=、!<	ALL	最大值
	ANY	最小值
<、<=、!>	ALL	最小值
	ANY	最大值
=	ANY	所有值，相当于 IN
<>	ANY	取所有值

#### (4) [NOT] EXISTS 谓词

由 EXISTS 和 NOT EXISTS 谓词引出子查询的作用是只判断子查询的结果集是否为空，若子查询结果集为非空，则向上层查询返回 TRUE；反之，则返回 FALSE，并不产生其他具体值。其一般格式为：[NOT] EXISTS (子查询)

### 6. 视图的概念及特点

视图是从一个或几个基本表（或视图）导出的表。它可以像表一样使用，但它是一个虚表，它所有的数据是通过引用基表反映出来的。

### 7. 视图的操作



WHERE XS.学号=XS\_KC.学号 AND 专业='计算机'

也可以用 JOIN 谓词实现，具体语句为：

```
SELECT XS.学号,姓名,课程号,成绩 FROM XS JOIN XS_KC  
ON (XS.学号=XS_KC.学号) WHERE 专业='计算机'
```

## 2.外连接

【例 2】以 XS 表和 XS\_KC 表为例，比较理解左连接、右连接两者之间的区别，并理解外连接与内连接的区别。

- (1) 查询 XS 表中学生的基本信息以及他们的选课信息。

```
SELECT XS.学号,XS.姓名,性别,XS_KC.课程号,XS_KC.成绩 FROM XS  
LEFT JOIN XS_KC ON XS.学号=XS_KC.学号
```

- (2) 列出与所有已选课学生的基本信息以及选课信息。

```
SELECT XS.学号,XS.姓名,性别,XS_KC.课程号,XS_KC.成绩 FROM XS  
RIGHT JOIN XS_KC ON XS.学号=XS_KC.学号
```

## 3.使用比较运算符引出子查询

【例 3】找出 GZGL 数据库中年龄大于学号为“0306201”的学生学号和姓名。

由于查询“0306201”学生的年龄返回值一定是单值，可以用比较运算符直接引出子查询来。

```
SELECT 学号,姓名 FROM XS X1 WHERE X1.出生日期 <  
(SELECT X2.出生日期 FROM XS X2 WHERE X2.学号='0306201')
```

【例 4】找出考试成绩高于“0312118”学生的所有课程成绩的学生学号及成绩。

由于本例子查询的返回值不一定是单值，因此要用比较运算符和 ALL 搭配使用。

```
SELECT 学号,课程号,成绩 FROM XS_KC K1 WHERE K1.成绩 > ALL  
(SELECT 成绩 FROM XS_KC K2 WHERE K2.学号='0312118')
```

## 5.使用连接谓词 IN、EXISTS

【例 5】找出选修了课程的学生学号、姓名和专业。

```
SELECT 学号,姓名,专业 FROM XS  
WHERE 学号 IN (SELECT 学号 FROM XS_KC)
```

用谓词 EXISTS 也可以实现上述查询，如：

```
SELECT 学号,姓名,专业 FROM XS WHERE EXISTS  
(SELECT * FROM XS_KC WHERE XS_KC.学号=XS.学号)
```

## 6.基于基表或视图的视图

【例 6】在 GZGL 数据库中创建计算机专业学生的基本信息视图

```
CREATE VIEW J_XS AS  
SELECT 学号,姓名,性别,出生日期 FROM XS WHERE 专业='计算机'
```

【例 7】创建计算机专业男生选课情况的视图。

```
CREATE VIEW J_XS_M AS
SELECT J_XS.学号,姓名,课程号,成绩 FROM J_XS
LEFT JOIN XS_KC ON J_XS.学号= XS_KC.学号 WHERE 性别='男'
```

## 7.带表达式的视图

【例 8】创建一个含有学生学号、姓名、年龄信息的视图。

```
CREATE VIEW J_XS_A (学号,姓名,年龄) AS
SELECT 学号,姓名, year(getdate())- year(出生日期) FROM XS
```

## 四、实验内容

**实验前在 eshop 数据库的 members 表中增加 1 条记录**

内容如下：‘jin’，‘刘津’，‘女’，‘北京’，8200，‘jin’

**1、使用 exists 查询购买了“101”商品的会员号和姓名。**

**2、使用 in 查询与“lfz”购买至少同一种商品的会员号和商品号。**

**3、将 members 表和 orders 表进行左外联接，查询所有会员的信息，包括没有购买商品的会员。**

**4、在 members 和 orders 表上创建购买了商品号为“102”商品的会员信息视图 View1。(使用**

**JOIN 与 WHERE 谓词两种方式实现)**

**5、在视图 View1 中增加一条记录 (内容如下)，并查看 members 表中记录的改变情况。记录内容**

**如下：**

‘fengx’，‘冯向’，‘男’，‘北京’，5000，‘fx’

**6、将视图 View1 中会员号为“liuzc”的会员的密码修改为“liu”，并查看 members 中记录的改变情**

**况。**

## 五、注意事项

1. 连接的类型（内链接，左外连接，右外连接）。
2. 链接查询和子查询的相互转换。
3. 使用JOIN 与使用WHERE 子句的区别。
4. 视图（虚表）和基表的操作的区别。