实验 3 SQL 语言进阶

3.1 实验目的

1．熟练掌握用 SELECT 书写集合查询；

2．熟练使用 INSERT、UPDATE、DELETE 语句，实现包含子查询的数据更新需求；

3．熟练使用 SQL 语句创建需要的视图；

4．能处理空值和空集对结果的影响。

3.2 内容提要

3.2.1 集合操作

1.并集操作

UNION 操作符用于合并两个或多个 SELECT 语句的结果集。

示例 3-1：查询选修了课程 1 或者选修了课程 2 的学生。

```
SELECT Sno
FROM XUANKE
WHERE Cno='C001'
UNION
SELECT Sno
FROM XUANKE
WHERE Cno= 'C002'
```

2.交集操作

SQL Server 中使用嵌套子查询实现交集操作。

示例 3-2：查询既选修了课程 1 又选修了课程 2 的学生。

```
SELECT   Sno
FROM   XUANKE
WHERE Cno='C001' AND Sno IN
      (SELECT Sno
```

```
        FROM   XUANKE
        WHERE Cno=‘C002');
```

3.差集操作

SQL Server 中使用嵌套子查询实现差集操作。

示例 3-3：查询选修了课程 1 但是没有选修课程 2 的学生。

```
SELECT   Sno
 FROM   XUANKE
 WHERE Cno=‘C001' AND Sno NOT IN
      (SELECT Sno
       FROM   XUANKE
       WHERE Cno=‘C002');
```

3.2.2 数据更新

1.插入单行记录

语句格式：

```
INSERT
INTO <表名> [(<属性列 1>[，<属性列 2 >…)]
      VALUES (<常量 1>  [，<常量 2>]   …)
```

示例 3-4：将一个新学生元组（学号：08660401；姓名：陈冬；性别：男；所在系：YY；年龄：18 岁）插入到 Student 表中。

```
INSERT
INTO   Student (Sno, Sname, Ssex, Sdept, Sage)
       VALUES   (‘08660401','陈冬','男','YY',18)
```

2.插入子查询结果

即将子查询结果插入指定表中，语句格式：

```
INSERT
INTO <表名>  [(<属性列 1> [，<属性列 2>…  )]
子查询；
```

示例 3-5：对每一个系，求学生的平均年龄，并把结果存入数据库 Dept_age 表中。

第一步：建表

```
CREATE  TABLE  Dept_age
    (Sdept  CHAR(15),              /*  系名*/
     Avg_age SMALLINT);          /*学生平均年龄*/
```

第二步：插入数据

```
INSERT
 INTO  Dept_age(Sdept, Avg_age)
      SELECT  Sdept, AVG(Sage)
      FROM  Student
      GROUP BY Sdept;
```

3. 修改操作

语句格式：

```
UPDATE  <表名>
SET  <列名>=<表达式>[，<列名>=<表达式>]…
[[from <表名>] WHERE <条件>];
```

示例 3-6:将计算机科学系学生的成绩置零。

```
UPDATE XUANKE
SET  Grade=60
FROM STUDENT
WHERE XUANKE.SNO = STUDENT.SNO AND SDEPT='JSJ'

UPDATE XUANKE
SET  Grade=0
WHERE  SNO  IN
        (SELECT  SNO  FROM  STUDENT  WHERE SDEPT='JSJ')
```

4. 删除操作

语句格式：

```
DELETE
FROM  <表名>
[WHERE <条件>];
```

示例 3-7:删除计算机科学系学生的选课记录。

```
DELETE
    FROM XUANKE
    WHERE   sno in (SELECT Student.sno
                    FROM Student,xuanke
                    WHERE Student.Sno=XUANKE.Sno AND
                        SDEPT='JSJ');
```

3.3.3 视图操作

语句格式：

```
CREATE VIEW <视图名>  [(<列名>  [，<列名>]…)]
  AS  <子查询>
  [WITH  CHECK  OPTION]；
```

示例 3-8:建立计算机系学生的视图。

```
CREATE VIEW JSJ_Student
AS
SELECT Sno,Sname,Sage
FROM      Student
WHERE   Sdept= 'JSJ'
```

3.3.4 空值和空集处理

NULL 是个常量，仅在数值和字符串类型的列中有意义，代表的是“不确定的值”。由于 NULL 值的特殊性，可能导致特别的效果：

1.对 NULL 值做算术运算的结果还是 NULL；

2. NULL 与比较运算符的运算都返回 FALSE 的值；

3.判断是否为空值的要使用 IS NULL

4.使用排序 ORDER BY 情况,NULL 被当作最小值处理

5.与 DISTINCT 保留字结合使用时,所有的 NULL 被视为相同

6.在使用 GROUP BY 时，所有的 NULL 被视为相同

7.在集合函数中,空值和空集处理情况:

COUNT()返回值 0

其它集合函数忽略 NULL

8.嵌套查询中,空值和空集处理情况:

1) **IN** 后里面的列表可以包括 null，in 相当于用=依次比较，然后去 or，true or null =true，所以 null 是被忽视的(不理睬)。

示例 3-9：查询跟 e3 职工有联系的供应商信息（供应商号，供应商名，地址）

```
select 供应商号,供应商名,地址
from 供应商
where 供应商号 in
    (
    select 供应商号
    from 订购单
    where 职工号='e3'
        )
```

2)**NOT IN** 实际上是用!=依次比较列表，然后去 and，true and null = null。只要列表包括 null 值，就会返回 false。

所以，当 not in 后面是子查询时，只有保证 select 后面的字段有 not null 约束的时候才能使用。

示例 3-10：查询跟 e3 职工没有联系的供应商信息（供应商号，供应商名，地址）

```
select 供应商号,供应商名,地址
from 供应商
where 供应商号 not in
    (
    select 供应商号
    from 订购单
    where 职工号='e3'AND 供应商号 IS NOT NULL
    )
```

3.3 实验任务及步骤

本次试验需要用到实验 1 建立的数据库，参照实验 2 中方法添加，并对照数据验证语句书写是否正确。

1．集合查询

1)检索出和职工 E1 或者 E3 有联系的北京的供应商信息，使用 UNION 实现并集操作。

2)检索出和职工 E1、E3 都有联系的北京的供应商信息。

3)检索出在北京工作并且只向 S4 供应商发出了订购单的职工号。

2．数据更新

1)插入一个新的供应商元组（S9，智通公司，沈阳）。

2)删除目前没有任何订购单的供应商。

3)删除由在上海仓库工作的职工发出的所有订购单。

4)北京的所有仓库增加 100㎡的面积。

5)给低于所有职工平均工资的职工提高 5%的工资。

6)创建一张新表，统计每个职工的订单数量，并将统计结果写入新表中。

3．视图操作

检索出工资低于本仓库平均工资的职工信息,并创建视图。

4．空值和空集处理

1)检索出目前没有任何订购单的供应商信息。

2)检索出与工资在 1220 元以下的职工没有联系的供应商的名称。

5 实验总结

1).记录实验全过程，并写出实验报告。

2).详细记录实验过程中遇到的问题，以及问题的解决方法。