

《数据库系统》作业-第三章

姓名：任宇 学号：33920212204567

1. 试述 SQL 的特点

1) 综合统一：

SQL 集数据定义语言、数据操纵语言、数据控制语言的功能于一体，语言风格统一。可以独立完成数据库生命周期中的全部活动。

2) 高度非过程化：

SQL 进行数据操作时，只要提出做什么，而无须指明怎么做，因此无须了解存取路径。存取路径的选择以及 SQL 的操作过程由系统自动完成。

3) 面向集合的操作方式：

SQL 采用集合操作方式，不仅操作对象、查找结果可以是元组的集合，而且一次插入、删除、更新操作的对象也可以是元组的集合。

4) 以同一种语法结构提供多种使用方式

SQL 既是独立的语言，又是嵌入式语言。而在两种不同的使用方式下，SQL 的语法结构基本上是一致的。这种以统一的语法结构提供多种不同使用方式的做法，提供了极大的灵活性和方便性。

5) 语言简洁，易学易用

SQL 功能极强，但由于设计巧妙，语言十分简洁，完成核心功能只用了九个动词。

2. 说明在 DROP TABLE 时，RESTRICT 和 CASCADE 的区别。

RESTRICT：若选择 RESTRICT，则该表的删除是有限制条件的。欲删除的基本表不能被其他表的约束所引用，不能有视图，不能有触发器，不能有存储过程或函数等，如果存在这些依赖该表的对象，则此表不能被删除。

CASCADE：若选择 CASCADE，则该表的删除是没有限制条件的，在删除基本表的同时，相关的依赖对象，例如视图，都将被一起删除。

3. 有两个关系 S (A, B, C, D) 和 T (C, D, E, F)，写出与下列查询等价的 SQL 表达式：

1) `SELECT * FROM S WHERE A=10`

2) `SELECT DISTINCT A,B FROM S`

3) `SELECT A,B,S.C,S.D,E,F FROM S,T WHERE S.C=T.C AND S.D=T.D`

4) `SELECT A,B,S.C,S.D,T.C,T.D,E,F WHERE S.C=T.C`

5) `SELECT A,B,S.C,S.D,T.C,T.D,E,F WHERE A<E`

6) `SELECT S1.C,S1.D,T.C,T.D,E,F FROM T,(SELECT DISTINCT C,D FROM S) AS S1`

4. 用 SQL 语句建立第二章习题六中的 4 个表，针对建立的 4 个表用 SQL 完成第二章习题六中的查询。

建表：

1) S 表：

`CREATE TABLE S(SNO CHAR(3),SNAME CHAR(10),STATUS CHAR(2),CITY`

CHAR(10));

2) P 表:

```
CREATE TABLE P(PNO CHAR(3),PNAME CHAR(10),COLOR CHAR(4),WEIGHT
INT);
```

3) J 表:

```
CREATE TABLE J(JNO CHAR(3),JNAME CHAR(10),CITY CHAR(10));
```

4) SPJ 表:

```
CREATE TABLE SPJ(SNO CHAR(3),PNO CHAR(3),JNO CHAR(3),QTY INT);
```

查询:

1) 求供应工程 J1 零件的供应商号码 SNO:

```
SELECT SNO
FROM SPJ
WHERE JNO= 'J1' ;
```

2) 求供应工程 J1 零件 P1 的供应商号码 SNO:

```
SELECT SNO
FROM SPJ
WHERE JNO= 'J1' AND PNO= 'P1' ;
```

3) 求供应工程 J1 零件为红色的供应商号码 SNO:

```
SELECT SNO
FROM P, SPJ
WHERE JNO= 'J1' AND SPJ.PNO=P.PNO AND COLOR= '红' ;
```

4) 求没有使用天津供应商生产的红色零件的工程号 JNO:

```
SELECT JNO
FROM J
WHERE NOT EXISTS
( SELECT *
FROM SPJ, S, P
WHERE P.COLOR= '红'
AND S.CITY= '天津'
AND SPJ.PNO=P.PNO
AND SPJ.SNO=S.SNO
AND SPJ.JNO=J.JNO);
```

5) 求至少用了供应商 S1 所供应的全部零件的工程号 JNO:

```
SELECT DISTINCT JNO
FROM SPJ SPJZ
WHERE NOT EXISTS
(SELECT *
FROM SPJ SPJX
WHERE SNO= 'S1'
AND NOT EXISTS
(SELECT *
FROM SPJ SPJY
WHERE SPJY.PNO=SPJX.PNO
AND SPJY.JNO=SPJZ.JNO));
```

5. 针对习题 4 的 4 个表用 SQL 完成以下各项操作：

1) 找出所有供应商的姓名和所在城市：

```
SELECT SNAME, CITY  
FROM S;
```

2) 找出所有零件的名称、颜色、重量：

```
SELECT PNAME, COLOR, WEIGHT  
FROM P;
```

3) 找出使用供应商 S1 所供应零件的工程号码：

```
SELECT JNO  
FROM SPJ
```

```
WHERE SNO = 'S1' ;
```

4) 找出工程项目 J2 使用的各种零件的名称及其数量：

```
SELECT P.PNAME, SPJ.QTY  
FROM P, SPJ  
WHERE SPJ.JNO= 'J2' AND P.PNO=SPJ.PNO;
```

5) 找出上海厂商供应的所有零件号码：

```
SELECT DISTINCT PNO  
FROM SPJ  
WHERE SNO IN  
(SELECT SNO FROM S WHERE CITY= '上海' );
```

6) 找出使用上海产的零件的工程名称：

```
SELECT JNAME  
FROM J  
WHERE JNO IN  
(SELECT JNO  
FROM SPJ, S  
WHERE S.CITY= '上海' AND SPJ.SNO=S.SNO);
```

7) 找出没有使用天津产的零件的工程号码：

```
SELECT JNO  
FROM J  
WHERE NOT EXISTS  
(SELECT *  
FROM SPJ, S  
WHERE SPJ.SNO=S.SNO AND SPJ.JNO=J.JNO AND S.CITY= '天津' );
```

8) 把全部红色零件的颜色改成蓝色：

```
UPDATE SPJ  
SET COLOR = '蓝'  
WHERE COLOR = '红' ;
```

9) 由 S5 供给 J4 的零件 P6 改为由 S3 供应：

```
UPDATE SPJ  
SET SNO= 'S3'  
WHERE SNO= 'S5' AND JNO= 'J4' AND PNO= 'P6' ;
```

10) 从供应商关系中删除 S2 的记录，并从供应情况关系中删除相应的记录：

```
DELETE FROM SPJ WHERE SNO= 'S2' ;
```

```
DELETE FROM S WHERE SNO= 'S2' ;
```

11) 请将 (S2, J6, P4, 200) 插入供应情况关系:

```
INSERT INTO SPJ (SNO, JNO, PNO, QTY)
```

```
VALUES (S2, J6, P4, 200) ;
```

6. 什么是基本表? 什么是视图? 两者的区别和联系是什么?

基本表是本身独立存在的表, 在 SQL 中一个关系就对应一个基本表。

视图是从一个或几个基本表导出的表。

视图本身不独立存储在数据库中, 即数据库中只存放视图的定义而不存放视图对应的数据, 这些数据仍存放在导出视图的基本表中, 因此视图是一个虚表。视图在概念上与基本表等同, 用户可以在视图上再定义视图。

7. 试述视图的优点。

1) 视图能够简化用户的操作

2) 视图使用户能以多种角度看待同一数据

3) 视图对重构数据库提供了一定程度的逻辑独立性

4) 视图能够对机密数据提供安全保护

5) 适当利用视图可以更清晰地表达查询

8. 哪些视图是可以更新的, 哪类视图是不可更新的。

基本表的行列子集视图一般是可以更新的。若视图的属性来自聚集函数、表达式, 则该视图肯定是不可以更新的。

9. 请为三建工程项目建立一个供应情况的视图, 包括供应商代码 (SNO)、零件代码 (PNO)、供应数量 (QTY)。针对该视图完成下列查询:

A. 找出三建工程项目使用的各种零件代码及其数量。

B. 找出供应商 S1 供应三建工程的情况。

创建视图:

```
CREATE VIEW VSPJ AS
```

```
SELECT SNO, PNO, QTY
```

```
FROM SPJ
```

```
WHERE JNO IN (SELECT JNO FROM J WHERE JNAME= '三建') ;
```

查询:

A. SELECT PNO, QTY

```
FROM VSPJ;
```

B. SELECT PNO, QTY

```
FROM VSPJ
```

```
WHERE SNO = 'S1' ;
```