

# 数据库第三章作业

陈鑫蕾 22920202202877

## 1. 试述 SQL 的特点

综合统一

高度过程化

面向集合的操作方式

以同一种语言结构提供多种使用方式

语言简洁，易学易用

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

RESTRICT 的删除是有限制条件的，即删除的基本表不能被其他表的约束所引用（如 CHECK, FOREIGN KEY 等约束），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 不仅删除表中的数据 and 此表的定义，同时也删除此表上建立的索引，触发器等对象一般也都被删除。有的关系数据库管理系统还会同时删除在此表上建立的视图。如果欲删除的基本表被其他基本表所引用，则这些表也可能被删除。

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

(1) SELECT \* FROM S WHERE A=10;

(2) SELECT A,B FROM S;

(3) SELECT \* FROM S INNER JOIN T ON S.C=T.C AND S.D=T.D;

(4) SELECT \* FROM S INNER JOIN T ON S.C=T.C;

(5) SELECT \* FROM S INNER JOIN T ON S.A<T.E;

(6) SELECT \* FROM S,T WHERE S.C=T.C AND T.D=S.D;

## 4. 用 SQL 语句建立第二章习题 6 中的四个表；针对建立的四个表用 SQL 完成第二章习题六中的查询。

(1) CREATE TABLE S(  
SNO CHAR(3),  
SNAME CHAR(10),  
STATUS INT,  
CITY CHAR(10));

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

(3) CREATE TABLE J(  
JNO CHAR(3),

```
JNAME CHAR(10),  
CITY CHAR(10));
```

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

```
(5) SELECT SNO  
FROM SPJ  
WHERE JNO=' J1' ;
```

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

```
(7) SELECT SNO  
FROM SPJ  
WHERE JNO=' J1' AND PNO IN(  
SELECT PNO  
FROM P  
WHERE COLOR=' 红' );
```

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

```
(9) SELECT JNO  
FROM SPJ SPJ1  
WHERE NOT EXISTS  
SELECT *  
FROM SPJ SPJ2  
WHERE SPJ2.SNO='S1' AND NOT EXISTS  
SELECT *  
FROM SPJ SPJ3  
WHERE  
SPJ3.PNO=SPJ2.PNO AND SPJ3.JNO=SPJ1.JNO
```

5. 针对习题 4 中的 4 个表用 SQL 完成以下各项操作

(1) SELECT Sname, City  
FROM S

(2) SELECT PNAME, COLOR, WEIGHT  
FROM P

(3) SELECT DISTINCT JNO  
FROM SPJ  
WHERE SNO= 'S1'

(4) SELECT QTY, PNAME  
FROM SPJ, P  
WHERE SPJ. JNO= 'J2' AND P. PNO=SPJ. PNO

(5) SELECT DISTINCT PNO  
FROM S, SPJ  
WHERE S. CITY= '上海' AND S. SNO=SPJ. SNO

(6) SELECT JNAME  
FROM J  
WHERE JNO=  
                  SELECT DISTINCT JNO  
                  FROM SPJ  
                  WHERE SNO=  
                                  SELECT SNO  
                                  FROM S  
                                  WHERE CITY= '上海'

(7) SELECT JNO  
FROM SPJ  
WHERE SNO NOT IN(  
                  SELECT SNO  
                  FROM S  
                  WHERE S. CITY= '天津' )

(8) UPDATE P SET COLOR=' 蓝'  
      WHERE COLOR=' 红'

(9) UPDATE SPJ SET SNO=' S3'  
WHERE SNO=' S5' AND JNO=' J4' AND PNO=' P6'

(10) DELETE FROM S WHERE SNO=' S2'  
DELETE FROM SPJ WHERE SNO=' S2'

(11) INSERT INTO SPJ 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表。

视图是从一个或者多个基本表中导出的虚表，不独立存储在数据库当中

区别：视图是虚表，不独立存储在数据库中，数据库只存放视图的定义，数据存放在基本表当中

联系：视图在概念上与基本表相同，用户可以如同使用表一样使用视图，可以在视图上在定义视图。视图依据基本表产生，会随着基本的数据变化而变化。

7. 试述视图的优点

(1) 视图能够简化用户的操作：视图机制使用户注意力集中在数据上，通过定义视图，可以使数据库看起来结构简单、清晰，并且简化用户的数据查询操作；

(2) 视图使用户能够以多种角度看待同一数据：视图机制使不同的用户以不同的角度看待同一数据，当多个用户使用同一数据库时，这种机制十分重要；

(3) 视图对重构数据库提供了一定程度的独立性：当数据库重构时，用户的应用程序不受到影响；

(4) 视图对机密数据提供了安全保护：有了视图机制，在设计数据库应用系统的时候，对不同的用户定义不同的视图，使机密数据收到保护；

(5) 使用视图可以更清晰的表达查询。

8. 哪类视图是可以更新的？哪类视图是不可更新的？各举一例说明。

```
CREATE TABLE TEST (
```

```
A VARCHAR(2),
```

```
B INT,
```

```
C INT);
```

```
CREATE VIEW V_A
```

```
AS
```

```
SELECT A,B FROM TEST;
```

```
CREATE VIEW V_B
```

```
AS
```

```
SELECT A,AVG(C) FROM TEST GROUP BY A;
```

(1) 基本表的行列子集视图是可以更新的，如 V\_A

```
INSERT INTO V_A VALUE( 'A1' ,1);
```

将会转变为 对 TEST 表的更新

(2) 对于视图的属性来自集函数、表达式的视图不可更新，如 V\_B

```
INSERT INTO V_B VALUE( 'A2' ,1);
```

将出错

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)，并针对视图完成下列查询

```
CREATE VIEW V_SPJ AS
```

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

```
FROM SPJ
```

```
WHERE JNO=
```

```
        (SELECT JNO
```

```
        FROM J
```

```
        WHERE JNAME=' 三建' )
```

```
SELECT DISTINCT PNO,QTY
```

```
FROM V_SPJ
```

```
SELECT SUM(QTY)
```

```
FROM V_SPJ
```

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