1.sq|特点：

综合统一、高度非过程化、面向集合的操作方式，以同一种语法结构提供多种使用方式、语言简洁易学易用

2.DROPTABLE时 RESTRICT和CASCADE的区别

RESTRICT：删除有限制条件。想要删除的表不能被其他表的约束所引用，不能有视图，不能有触发器等CASCADE：删除没有限制条件。删除基本表的同时相关的依赖对象都会被删除。

3

(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 \* from S ,T where S.C=T.C;

(5) select \* from S ,T where S.A<T.E;

(6) select S.C,S.D,T.\* from S ,T ;

4.

#建立S表

CREATE TABLE S (SNO CHAR(2), SNAME VARCHAR(6), STATUS CHAR(2), CITY CHAR(4));

#建立P表

CREATE TABLE P (PNO CHAR(2), PNAME VARCHAR(6), COLOR CHAR(2), WEIGHT INT);

#建立J表

CREATE TABLE J (JNO CHAR(2), JNAME VARCHAR(8), CITY CHAR(4));

#建立SPJ表

CREATE TABLE SPJ (SNO CHAR(2), PNO CHAR(2), JNO CHAR(2), QTY INT);

#使用SQL语句完成查询

（1）求供应工程 J1 零件的供应商代码 SNO；

SELECT DISTINCT SNO

FROM SPJ

WHERE JNO='J1';

（2）求供应工程 J1 零件 P1 的供应商代码 SNO；

SELECT DISTINCT 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代码 SNO；

SELECT DISTINCT SNO

FROM SPJ,P

WHERE SPJ.PNO=P.PNO AND

JNO='J1' 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；

提交：

SELECT JNO

FROM J

WHERE NOT EXISTS

(SELECT \*

FROM SPJ,S,P

WHERE SPJ.JNO=J.JNO AND SPJ.SNO=S.SNO AND SPJ.PNO=P.PNO AND

CITY='天津' AND COLOR='红');

SELECT DISTINCT JNO

FROM SPJ

WHERE JNO NOT IN

(SELECT JNO

FROM SPJ,P,S

WHERE S.SNO=SPJ.SNO AND P.PNO=SPJ.PNO AND

CITY=‘天津’ AND COLOR=‘红’);

（5）求至少用了与供应商 S1 所供应的全部零件相同零件号的工程号 JNO。

SELECT DISTINCT JNO

FROM SPJ SPJA

WHERE NOT EXISTS

(SELECT \*

FROM SPJ SPJX

WHERE SPJX.SNO='S1' AND

NOT EXISTS

(SELECT \*

FROM SPJ SPJY

WHERE SPJA.JNO=SPJY.JNO AND

SPJY.PNO=SPJX.PNO));

5.

（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所供应零件的工程代码；

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 PNAME,QTY

FROM SPJ,P

WHERE SPJ.PNO=P.PNO AND

JNO='J2';

（5）找出上海厂商供应的所有零件代码；

SELECT DISTINCT PNO

FROM S,SPJ

WHERE SPJ.SNO=S.SNO AND

CITY='上海';

（6）找出使用上海产的零件的工程名称；

SELECT JNAME

FROM J,SPJ,S

WHERE J.JNO=SPJ.JNO AND SPJ.SNO=S.SNO AND

S.CITY='上海'

（7）找出没有使用天津产的零件的工程代码；

SELECT JNO

FROM J

WHERE NOT EXISTS

(SELECT \*

FROM SPJ,S,P

WHERE SPJ.JNO=J.JNO AND SPJ.SNO=S.SNO AND

CITY='天津');

（8）把全部红色零件的颜色改成蓝色；

UPDATE P

SET COLOR='蓝'

WHERE COLOR='红';

（9）把由S5供给J2的零件P6改为由S3供应；

UPDATE SPJ

SET SNO='S3'

WHERE SNO='S5' AND JNO='J2' AND PNO='P6';

（10）从供应商关系中删除S2的记录，并从供应关系中删除相应的记录；

DELETE FROM S WHERE SNO='S2';

DELETE FROM SPJ WHERE SNO='S2';

（11）请将(S2,J6,P4,200)插入供应情况关系

INSERT INTO SPJ

VALUES ('S2','P4','J6',200)

6.

基本表是本身独立存在的表，在关系数据库管理系统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。它本身不独立存储在数据库中。

区别：基本表存在数据库中。视图不。

联系：视图的数据还是存储在基本表里的。基本表数据发生变化，视图中查询出的数据也改变了。

7.

视图优点：

1.封装复杂的SQL查询语句为简单的视图访问，从而简化用户操作

2.利用视图可以限制用户对底层表中某些列的访问权限，对机密数据提供安全保护层。

3. 视图机制使用户能以不同的方式或多种角度来查看同一份数据，提供了更灵活的数据访问和展示方式

4.提高数据可读性

8.

行列子集视图一般是可更新的。比如把学生视图Student中学号为“201215122”的学生姓名改为“刘晨”。

有些视图的更新不能唯一有意义地转换成对相应基本表的更新，则不能更新。比如视图S\_G中的平均成绩是由表Student中成绩计算平均值得来的，改变一个学生的平均成绩，系统不能相应地修改各科成绩，所以不可更新。

9.

#建立视图

CREATE VIEW VIEW\_SPQ

AS

SELECT SNO,PNO,QTY

FROM SPJ

WHERE JNO=

(SELECT JNO

FROM J

WHERE JNAME='三建')

#完成查询

1. 找出三建工程项目使用的各种零件代码及数量；

SELECT PNO,QTY

FROM VIEW\_SPQ

2. 找出供应商S1供应三建工程情况。

SELECT SNO,PNO,QTY

FROM VIEW\_SPQ

WHERE SNO='S1'

10.

空值就是“不知道”“不存在”或“无意义”的值。比如缺考同学的成绩。

SQL语言用NULL来表示。

空值与另一个值算术计算的结果为空值，空值与另一个值比较运算的结果为UNKNOWN。