```
create table S (
        sno char(6) primary key,
        sname char(10),
        status smallint,
        city char(10)
);

create table J (
        jno char(6) primary key,
        jname char(10),
        city char(10)
);
```

```
create table P (
        pno char(6) primary key,
        pname char(10),
        color char(2),
        weight smallint
);
create table SPJ (
        sno char(6),
        pno char(6),
        jno char(6),
        qty smallint,
        primary key (sno, pno, jno),
        foreign key (sno) references S(sno),
        foreign key (pno) references P(pno),
        foreign key (jno) references J(jno)
);
```

求供应商J1零件的供应商号码SNO

```
select distinct sno
from SPJ
where jno = 'J1';
```

求供应商J1零件P1的供应商号码SNO。

```
select sno
from SPJ
where jno = 'J1' and pno = 'P1';
```

求供应商J1零件为红色的供应商号码SNO。

```
select sno
from SPJ, P
where SPJ.pno = P.pno and color = '红' and jno = 'J1';
```

求没有使用天津供应商生产的红色零件的工程号JNO。

```
select distinct jno
from SPJ
where jno not in (
        select jno
        from SPJ, S, P
        where city = '天津' and color = '红' and SPJ.pno =
P.pno and SPJ.sno = S.sno
);
```

求至少用了供应商S1所供应的全部零件的工程号JNO。

SQL语句：

```
select distinct jno
from SPJ SPJ1
where not exists (
        select *
        from SPJ SPJ2
        where SPJ2.sno = 'S1' and
                not exists (
                        select *
                        from SPJ SPJ3
where SPJ1.jno = SPJ3.jno and SPJ2.pno = SPJ3.pno
and SPJ3.sno = 'S1'
                ))
```

找出所有供应商的姓名和所在城市

```
select distinct sname, city
from S;
```

找出所有零件的名称，颜色，重量。

```
select pname, color, weight
from P;
```

找出使用供应商S1所供应零件的工程号。

```
select distinct jno
from SPJ
where sno = 'S1';
```

找出工程项目J2使用的各种零件的名称和数量。

```
select pname, qty
from SPJ, P
where SPJ.pno = P.pno and SPJ.jno = 'j2';
```

找出上海厂商供应的所有零件号码。

```
select pno
from SPJ, S
where SPJ.sno = S.sno and city = '上海';
```

找出使用上海产的零件工程名称。

```
select distinct SPJ.jno
from SPJ, S, J
where SPJ.sno = S.sno and SPJ.jno = J.jno and S.city
= '上海';
```

找出没有使用天津产的零件的工程号码。

```
select distinct jno
from SPJ, S
where SPJ.sno not in (
        select sno
        from S
        where S.city = '天津'
);
```

把全部红色零件的颜色改为蓝色。

```
update P
set color = '蓝' where color = '红';
```

有S5供给J4的零件P6改为由S3供应，请做必要的修改。

```
update SPJ
set sno = 'S3'
where sno = 'S5' and jno = 'J4' and pno = 'P6';
```

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';
```

请将(S2,J6,P4,200)插入供应情况关系。

```
insert into SPJ
values ('S2', 'P4', 'J6', 200);
```

请为三建工程项目监理一个供应情况的视图，包括供应商代码（SNO），零件代码（PNO），供应商数量（QTY）。针对视图完成下列查询：

```
create view view1 (sno, pno, qty) as
        select SPJ.sno, pno, qty
        from SPJ, J
        where SPJ.jno = J.jno and jname = '三建';
```

请为三建工程项目监理一个供应情况的视图，包括供应商代码（SNO），零件代码（PNO），供应商数量（QTY）。针对视图完成下列查询：

```
create view view1 (sno, pno, qty) as
        select SPJ.sno, pno, qty
        from SPJ, J
        where SPJ.jno = J.jno and jname = '三建';
```

找出三建工程项目使用的各种零件代码及其数量。

```
select distinct pno, qty
from view1;
```

找出供应商S1的供应情况。

```
select *
from view1
where sno = 'S1';
```

用户王明对两个表有SELECT权限。

```
grant select
on 职工
to 王明;

grant select
on 部门
to 王明;
```

用户李勇对两个表有SELECT和DELETE权限。

```
grant insert, delete
on 部门
to 李勇;

grant insert, delete
on 职工
to 李勇;
```

每个职工只对自己的记录有SELECT权限。

```
CREATEVIEW SELFINFO
AS
SELECT*FROM职工
WHEREUSER_NAME()=姓名

GRANT SELECT
ON SELFINFO
TO PUBLIC;
```

用户刘星对职工表有SELECT权限，对工资字段有更新权限。

```
grant select, update(工资)
on 职工
to 刘星;
```

用户张新具有修改这两张表的权限。

```
grant alter
on 职工
to 张新;

grant alter
on 部门
to 张新;
```

用户周平具有对两个表的所有权限，并且可给其他用户授权的权限。

```
grant all privileges
on 部门
to 周平
With grant option;

grant all privileges
on 职工
to 周平
With grant option;
```

用户杨兰具有从每个部门职工中SELECT最高工资，最低工资，平均工资的权限，他不能查看每个人的工资。

```
CREATE VIEW 部门工资
AS
SELECT 部门.名称,MAX(工资)最高工资,MIN(工资)最低工资,AVG(工资)平均工资
FROM职工,部门
WHERE职工.部门号=部门.部门号
GROUPBY部门.名称;

grant select
On 部门工资
to 杨兰;
```

第五章

6. Create table dept
   (deptno number(2),
     deptname var(10),
     manager var(10),
     phonenumber char(12),
     constraint pk primarty ke (deptno));

第五章

6.

```
create table emp
    ( empno number(4) primary key,
      ename var(10),
      age int,
     constraint  c1  check(age<=60),
     job var(10),
    sal number(2),
     constraint  fk foreign key (deptno) references
dept (deptno) );
```

第五章

8. 创建关系 female和male 略

```
create assertion party
check ( (select count(*) from male) +  (select count(*)
    from female)<=50);
```