In [1]:
```
%load_ext sql

import pymysql
pymysql.install_as_MySQLdb()
%sql mysql+pymysql://root:my-secret-pw@localhost
%sql create database if not exists spj
%sql mysql+pymysql://root:my-secret-pw@localhost/spj
```

```
 * mysql+pymysql://root:***@localhost
1 rows affected.
```

In [2]:
```
%%sql

set @@foreign_key_checks=0;
drop table if exists S;
drop table if exists P;
drop table if exists J;
drop table if exists SPJ;
create table S(sno varchar(8), sname varchar(20), status integer, city varchar(10), primary key(sno));
create table P(pno varchar(8), pname varchar(20), color varchar(10), weight integer, city varchar(10), primary k
create table J(jno varchar(8), jname varchar(20), city varchar(10), primary key(jno));
create table SPJ(sno varchar(8), pno varchar(8), jno varchar(8), qty integer, price integer, primary key(sno, p
set @@foreign_key_checks=1;
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[2]: []

In [3]:
```
import os
conn = pymysql.connect(host='127.0.0.1', port=3306, user='root',password='my-secret-pw', db='spj')
c=conn.cursor()
```

In [4]:
```
import csv
fpath = '/home/zhangsan/data'
files = os.listdir(fpath)
csvs = map(lambda f: os.path.join(fpath, f),
           filter(lambda f: f.endswith('.csv'), files))
csvs = list(csvs)
print(csvs)
c.execute('SET FOREIGN_KEY_CHECKS = 0')
for fpath in csvs:
    print(fpath)
    table = os.path.splitext(os.path.basename(fpath))[0]
    cr = csv.reader(open(fpath), delimiter=',')
    header = next(cr)
    header = ['`{}`'.format(h) for h in header]
    sqltemp = 'insert into {}({}) value({})'.format(
        table, ','.join(header), ('%s,'*len(header))[:-1])
    for ri, row in enumerate(cr):
        print(ri, row)
        try:
            row = [None if r == '' else r for r in row]
            c.execute(sqltemp, row)
        except Exception as e:
            print('fpath:{} row:{}'.format(fpath, ri))
            print(row)
            raise e
    print(fpath, 'loaded into mysql')
c.execute('SET FOREIGN_KEY_CHECKS = 1')
conn.commit()
```

```
['/home/zhangsan/data/SPJ.csv', '/home/zhangsan/data/J.csv', '/home/zhangsan/data/S.csv', '/home/zhangsan/data/
P.csv']
/home/zhangsan/data/SPJ.csv
0 ['S1', 'P1', 'J1', '200', '100']
1 ['S1', 'P1', 'J4', '700', '100']
2 ['S2', 'P3', 'J1', '400', '10']
3 ['S2', 'P3', 'J2', '200', '10']
4 ['S2', 'P3', 'J3', '200', '12']
5 ['S2', 'P3', 'J4', '500', '10']
6 ['S2', 'P3', 'J5', '600', '20']
7 ['S2', 'P3', 'J6', '400', '10']
8 ['S2', 'P3', 'J7', '800', '8']
9 ['S2', 'P5', 'J2', '100', '10']
10 ['S3', 'P3', 'J1', '200', '20']
11 ['S3', 'P4', 'J2', '500', '18']
12 ['S4', 'P6', 'J3', '300', '30']
13 ['S4', 'P6', 'J7', '300', '38']
14 ['S5', 'P2', 'J2', '200', '40']
15 ['S5', 'P2', 'J4', '100', '45']
16 ['S5', 'P5', 'J5', '500', '30']
17 ['S5', 'P5', 'J7', '100', '30']
18 ['S5', 'P6', 'J2', '200', '30']
19 ['S5', 'P1', 'J4', '100', '30']
20 ['S5', 'P3', 'J4', '200', '30']
21 ['S5', 'P4', 'J4', '800', '28']
22 ['S5', 'P5', 'J4', '400', '40']
23 ['S5', 'P6', 'J4', '500', '29']
/home/zhangsan/data/SPJ.csv loaded into mysql
/home/zhangsan/data/J.csv
0 ['J1', 'Sorter', 'Paris']
1 ['J2', 'Punch', 'Rome']
2 ['J3', 'Reader', 'Athens']
3 ['J4', 'Console', 'Athens']
4 ['J5', 'Collator', 'London']
5 ['J6', 'Terminal', 'Oslo']
6 ['J7', 'Tape', 'London']
/home/zhangsan/data/J.csv loaded into mysql
/home/zhangsan/data/S.csv
0 ['S1', 'Smith', '20', 'London']
1 ['S2', 'Jones', '10', 'Paris']
2 ['S3', 'Blake', '30', 'Paris']
3 ['S4', 'Clark', '20', 'London']
4 ['S5', 'Adams', '30', 'Athens']
/home/zhangsan/data/S.csv loaded into mysql
/home/zhangsan/data/P.csv
0 ['P1', 'Nut', '红色', '12', 'London']
1 ['P2', 'Bolt', '绿色', '17', 'Paris']
2 ['P3', 'Screw', '蓝色', '17', 'Rome']
3 ['P4', 'Screw', '红色', '14', 'London']
4 ['P5', 'Cam', '蓝色', '12', 'Paris']
5 ['P6', 'Cog', '红色', '19', 'London']
/home/zhangsan/data/P.csv loaded into mysql
```

供应商数据库SPJ中有三张表：

- S(SNO, SNAME, STATUS, CITY)
- P(PNO, PNAME, COLOR, WEIGHT, CITY)
- J(JNO, JNAME,CITY)
- SPJ(SNO, PNO, JNO, QTY, PRICE)

其中，

- S表示供应商，各属性依次为供应商号，供应商名，供应商状态值，供应商所在城市；
- P表示零件，各属性依次为零件号，零件名，零件颜色，零件重量，零件存放的城市；
- J表示工程，各属性依次为工程号，工程名，工程所在城市；
- SPJ表示供货关系，各属性依次为供应商号，零件号，工程号，供货数量，单价。

上面已经导入了表，下面完成几项查询

1. 求没有供应零件号为P1和P2两种零件的供应商姓名

In [5]:
```
%%sql
select sname from S
where sno not in (
    select sno from SPJ where pno = 'P1' or 'P2'
);
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
3 rows affected.
```

Out[5]:

sname
Jones
Blake
Clark

2. 列出所有供应商的信息，包括供应商姓名、所供应的零件名（没有供应零件的供应商也要列出，最后结果中不要出现重复元组

In [6]:
```
%%sql
select distinct S.sname, P.pname from (S left join SPJ on S.sno = SPJ.sno), P
where S.sno = SPJ.sno and SPJ.pno = P.pno
order by S.sname;
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
10 rows affected.
```

Out[6]:

sname	pname
Adams	Nut
Adams	Bolt
Adams	Screw
Adams	Cam
Adams	Cog
Blake	Screw
Clark	Cog
Jones	Screw
Jones	Cam
Smith	Nut

3. 求只向与自己位于不同城市的工程供应零件的供应商姓名。

In [7]:
```
%%sql
select sname from S
where sname not in (
    select distinct sname from S, J, SPJ
    where S.sno = SPJ.sno and J.jno = SPJ.jno and S.city = J.city
);
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
1 rows affected.
```

Out[7]:

sname
Smith

4. 求只向与自己位于相同城市的工程供应零件的供应商姓名。

In [8]:
```
%%sql
select sname from S
where sname not in (
    select distinct sname from S, J, SPJ
    where S.sno = SPJ.sno and J.jno = SPJ.jno and S.city <> J.city
);
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
0 rows affected.
```

Out[8]:

sname

5. 求供应了所有零件的供应商姓名

In [9]:
```
%%sql
select sname from S
where not exists (
    select pno from P
    where not exists (
        select sno from SPJ
        where S.sno = SPJ.sno and SPJ.pno = P.pno
    )
);
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
1 rows affected.
```

Out[9]:

sname
Adams

6. 求供应了所有红色零件的供应商姓名。

In [10]:
```
%%sql
select sname from S
where not exists (
    select pno from P
    where P.color = '红色' and not exists (
        select sno from SPJ
        where S.sno = SPJ.sno and SPJ.pno = P.pno
    )
);
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
1 rows affected.
```

Out[10]:

sname
Adams

7. 列出每个城市的工程所使用的零件总的数量

In [11]:
```
%%sql
select city, sum(qty) from J left join SPJ on J.jno = SPJ.jno
group by city;
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
5 rows affected.
```

Out[11]:

city	sum(qty)
Paris	800
Rome	1200
Athens	3800
London	2300
Oslo	400

8. 按零件数量总和的降序列出每项工程所使用的每种红色零件的总的金额（工程可以向不同的供应商购买同一零件，总金额＝单价*供货数量）。输出工程号、零件号、总金额。

In [12]:
```
%%sql
select J.jno, P.pno, sum(qty*price) from J, SPJ, P
where J.jno = SPJ.jno and SPJ.pno = P.pno and P.color = '红色'
group by J.jno, P.pno
order by sum(qty) desc;
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
8 rows affected.
```

Out[12]:

jno	pno	sum(qty*price)
J4	P1	73000
J4	P4	22400
J2	P4	9000
J4	P6	14500
J3	P6	9000
J7	P6	11400
J1	P1	20000
J2	P6	6000

9. 求供应零件数量最多的供应商姓名。

In [13]:
```
%%sql
select S.sname from S, SPJ, P
where S.sno = SPJ.sno and P.pno = SPJ.pno
group by SPJ.sno having sum(SPJ.qty) >= all(
    select sum(SPJ.qty) from S, SPJ, P
    where S.sno = SPJ.sno and P.pno = SPJ.pno
    group by SPJ.sno
);
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
1 rows affected.
```

Out[13]:

sname
Jones

10. 求每个城市中供应零件数量最多的供应商姓名。

In [14]:
```
%%sql
select S1.city, S1.sname from S as S1, SPJ, P
where S1.sno = SPJ.sno and P.pno = SPJ.pno
group by S1.sno having sum(SPJ.qty) >= all(
    select sum(SPJ.qty) from S as S2, SPJ, P
    where S2.sno = SPJ.sno and P.pno = SPJ.pno and S1.city = S2.city
    group by SPJ.sno
);
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
3 rows affected.
```

Out[14]:

city	sname
London	Smith
Paris	Jones
Athens	Adams

11. 列出恰好供应了相同零件的供应商对。

注意：输出的时候，供应商对小号在前，相等不输出，要去重，都没有供应零件也算供应了相同零件。输出形式：(SNO,SNO)

In [15]:
```
%%sql
select distinct SP1.sno, SP2.sno from (
    select S.sno, group_concat(distinct P.pno order by P.pno separator ' ') as total_pno from S, SPJ, P
    where S.sno = SPJ.sno and SPJ.pno = P.pno
    group by S.sno
) as SP1, (
    select S.sno, group_concat(distinct P.pno order by P.pno separator ' ') as total_pno from S, SPJ, P
    where S.sno = SPJ.sno and SPJ.pno = P.pno
    group by S.sno
) as SP2
where SP1.total_pno = SP2.total_pno and SP1.sno < SP2.sno;
```

```
   mysql+pymysql://root:***@localhost
 * mysql+pymysql://root:***@localhost/spj
0 rows affected.
```

Out[15]:

sno	sno_1

In [ ]: