Lab4-2 PostgreSQL进行SQL操作(多表、聚集查询和分组聚集)

本节实验是在对PostgreSQL数据库进行简单的DDL与DQL操作的基础上，进行SQL的多表查询、聚集查询和分组聚集等相关知识。

1. 使用psycopg2连接并操作postgresql，掌握基础语法

In [51]:
```
# 导入python与postgresql操作库
import psycopg2
```

连接数据库，host和port不用修改，user="ecnu学号", password="ECNU学号", database="ecnu学号"
获取SQL执行器cursor，后续会使用cursor执行SQL并获取返回结果。

In [52]:
```
conn = psycopg2.connect(host="pgm-uf6t8021ru5tac71.rwlb.rds.aliyuncs.com", port=
autocommit = psycopg2.extensions.ISOLATION_LEVEL_AUTOCOMMIT
conn.set_isolation_level(autocommit)
cur = conn.cursor()
cur.execute("SET search_path TO ecnu10222140402;")
```

2. 多表查询、聚集查询和分组聚集练习

考虑一个零件销售供应商数据库，里面包含四张数据表：S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息），每张数据表的初始化属性与约束如下：

S: SNO (CHAR(2)), SNAME (CHAR(6)), STATUS (CHAR(2)), CITY (CHAR(4)) SNO为主键

P: PNO (CHAR(2)), PNAME (CHAR(6)), COLOR (CHAR(2)), WEIGHT (INT) PNO为主键

J: JNO (CHAR(2)), JNAME (CHAR(8)), CITY (CHAR(4)) JNO为主键

SPJ: SNO char(2), PNO char(2), JNO char(2) (SNO,PNO,JNO)为主键

Prepare 创建表与约束

在创建表前，最基础的事情是确定属性及属性的类型，除了前面的讲解中出现的与下方列出的常用数据类型外，PostgreSQL提供了非常丰富的数据类型：
http://www.postgres.cn/docs/14/datatype.html

类型	描述
TEXT	字符类型，变长，无长度限制

类型	描述
CHAR(NUM)	字符类型，定长，字符长度为NUM
VARCHAR(NUM)	字符类型，变长， 字符最长限制为NUM
BOOL	布尔类型
REAL	浮点类型（单精度）
float8	浮点类型（双精度）
INT	整型

除了前面的讲解中出现的主键约束、非空约束与唯一约束外，数据库还包含外键约束和检查约束，简介如下：

约束	描述
主键约束	NOT NULL 和 UNIQUE 的结合。确保某列（或两个列多个列的结合）有唯一标识，有助于更容易更快速地找到表中的一个特定的记录。
唯一约束	在一列中或者一组列中保存的数据在表中所有行间是唯一的
非空约束	指定一个列中不会有空值
外键约束	保证一个表中的数据匹配另一个表中的值的参照完整性
检查约束	指定一个某个列的值必须要满足一个布尔表达式

在创建表的同时添加约束的语法细节可以参考文档：
http://www.postgres.cn/docs/14/ddl-constraints.html

请同学们完成下述创建表的练习：

准备工作：根据上述表的属性和约束信息，创建表：

In [53]:
```
# 初始化表结构
# 之前Lab用到这个表结构 如果有冲突，可以修改下名称保留原表 或者删除原表
sql1 = "create table S (Sno char(4) , Sname char(8), Status char(2), City char(4
sql2 = "create table P (Pno char(4) , Pname char(8), color char(2), weight int,
sql3 = "create table J (Jno char(4) , Jname char(8), CITY char(4), primary key(J
sql4 = "create table SPJ (Sno char(4), Pno char(4), Jno char(4), QTY int, primar

cur.execute(sql1)
cur.execute(sql2)
cur.execute(sql3)
cur.execute(sql4)
```

In [54]:
```
# 添加外键约束
sql1 = "ALTER TABLE SPJ ADD FOREIGN KEY (SNO) REFERENCES S;"
sql2 = "ALTER TABLE SPJ ADD FOREIGN KEY (PNO) REFERENCES P;"
sql3 = "ALTER TABLE SPJ ADD FOREIGN KEY (JNO) REFERENCES J;"
cur.execute(sql1)
cur.execute(sql2)
cur.execute(sql3)
```

In [55]:
```
# 初始化数据 数据部分大家可以自己制造很多 `python 程序产生`，以下案例中只给出了少
sql1 = "insert into S(Sno,Sname,Status,City) values " + \
       "('S1','精益','20','天津'), ('S2','盛锡','10','北京'), " + \
       "('S3','东方红','30','北京'), ('S4','丰泰盛','20','天津'), " + \
       "('S5','为民','30','上海'), ('S6','盛锡-2','10','上海'), " + \
       "('S7','三角-2','30','北京'), ('S8','精益-2','20','广州'), " + \
       "('S9','三角','10','深圳'), ('S10','顺丰','20','广州');"

sql2 = "insert into P(Pno,Pname,color,weight) values " + \
       "('P1','螺母','红',12), ('P2','螺栓','绿',17), " + \
       "('P3','螺丝刀','蓝',14), ('P4','螺丝刀','红',14), " + \
       "('P5','凸轮','蓝',40), ('P6','齿轮','红',30)," + \
       "('P7','轮胎','黑',100), ('P8','钢筋','灰',100);"

sql3 = "insert into J(Jno,Jname,CITY) values " + \
       "('J1','三建','北京'), ('J2','一汽','长春'), " + \
       "('J3','弹簧厂','天津'), ('J4','造船厂','天津'), " + \
       "('J5','机车厂','唐山'), ('J6','无线电厂','常州'), " + \
       "('J7','半导体厂','南京'), ('J8','上汽','上海'), ('J9','重汽','杭州');"

sql4 = "insert into SPJ(Sno,Pno,Jno,QTY) values " + \
       "('S1','P1','J1',200), ('S1','P1','J3',100), ('S1','P1','J4',700), " + \
       "('S1','P2','J2',100), ('S2','P3','J1',400), ('S2','P3','J2',200), " + \
       "('S2','P3','J4',500), ('S2','P3','J5',400), ('S2','P5','J1',400), " + \
       "('S2','P5','J2',100), ('S3','P1','J1',200), ('S3','P3','J1',200), " + \
       "('S4','P5','J1',100), ('S4','P6','J3',300), ('S4','P6','J4',200), " + \
       "('S5','P2','J4',100), ('S5','P3','J1',200), ('S5','P6','J2',200), " + \
       "('S5','P6','J4',500), ('S6','P8','J3',600), ('S6','P7','J8',900), " + \
       "('S6','P8','J4',100), ('S6','P7','J9',500), ('S6','P7','J2',500), " + \
       "('S6','P5','J7',50), ('S7','P8','J1',100), ('S7','P7','J2',200), " + \
       "('S7','P7','J9',500),('S8','P5','J5',300), ('S8','P6','J1',100), " + \
       "('S8','P7','J9',450), ('S9','P5','J1',100), ('S9','P6','J3',300), " + \
       "('S9','P6','J4',200), ('S10','P7','J9',300), ('S10','P7','J8',300), " + \
       "('S10','P1','J5',100);"

cur.execute(sql1)
cur.execute(sql2)
cur.execute(sql3)
cur.execute(sql4)
```

2.1 多表查询实现

In [56]:
```
# 查询其订购的供应商零件信息
sql = "select J.Jno, J.Jname, J.CITY, SPJ.Sno, SPJ.Pno, SPJ.Jno, SPJ.QTY from J,
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('J1  ', '三建      ', '北京  ', 'S1  ', 'P1  ', 'J1  ', 200)
('J3  ', '弹簧厂    ', '天津  ', 'S1  ', 'P1  ', 'J3  ', 100)
('J4  ', '造船厂    ', '天津  ', 'S1  ', 'P1  ', 'J4  ', 700)
('J2  ', '一汽      ', '长春  ', 'S1  ', 'P2  ', 'J2  ', 100)
('J1  ', '三建      ', '北京  ', 'S2  ', 'P3  ', 'J1  ', 400)
('J2  ', '一汽      ', '长春  ', 'S2  ', 'P3  ', 'J2  ', 200)
('J4  ', '造船厂    ', '天津  ', 'S2  ', 'P3  ', 'J4  ', 500)
('J5  ', '机车厂    ', '唐山  ', 'S2  ', 'P3  ', 'J5  ', 400)
('J1  ', '三建      ', '北京  ', 'S2  ', 'P5  ', 'J1  ', 400)
('J2  ', '一汽      ', '长春  ', 'S2  ', 'P5  ', 'J2  ', 100)
('J1  ', '三建      ', '北京  ', 'S3  ', 'P1  ', 'J1  ', 200)
('J1  ', '三建      ', '北京  ', 'S3  ', 'P3  ', 'J1  ', 200)
('J1  ', '三建      ', '北京  ', 'S4  ', 'P5  ', 'J1  ', 100)
('J3  ', '弹簧厂    ', '天津  ', 'S4  ', 'P6  ', 'J3  ', 300)
('J4  ', '造船厂    ', '天津  ', 'S4  ', 'P6  ', 'J4  ', 200)
('J4  ', '造船厂    ', '天津  ', 'S5  ', 'P2  ', 'J4  ', 100)
('J1  ', '三建      ', '北京  ', 'S5  ', 'P3  ', 'J1  ', 200)
('J2  ', '一汽      ', '长春  ', 'S5  ', 'P6  ', 'J2  ', 200)
('J4  ', '造船厂    ', '天津  ', 'S5  ', 'P6  ', 'J4  ', 500)
('J3  ', '弹簧厂    ', '天津  ', 'S6  ', 'P8  ', 'J3  ', 600)
('J8  ', '上汽      ', '上海  ', 'S6  ', 'P7  ', 'J8  ', 900)
('J4  ', '造船厂    ', '天津  ', 'S6  ', 'P8  ', 'J4  ', 100)
('J9  ', '重汽      ', '杭州  ', 'S6  ', 'P7  ', 'J9  ', 500)
('J2  ', '一汽      ', '长春  ', 'S6  ', 'P7  ', 'J2  ', 500)
('J7  ', '半导体厂  ', '南京  ', 'S6  ', 'P5  ', 'J7  ', 50)
('J1  ', '三建      ', '北京  ', 'S7  ', 'P8  ', 'J1  ', 100)
('J2  ', '一汽      ', '长春  ', 'S7  ', 'P7  ', 'J2  ', 200)
('J9  ', '重汽      ', '杭州  ', 'S7  ', 'P7  ', 'J9  ', 500)
('J5  ', '机车厂    ', '唐山  ', 'S8  ', 'P5  ', 'J5  ', 300)
('J1  ', '三建      ', '北京  ', 'S8  ', 'P6  ', 'J1  ', 100)
('J9  ', '重汽      ', '杭州  ', 'S8  ', 'P7  ', 'J9  ', 450)
('J1  ', '三建      ', '北京  ', 'S9  ', 'P5  ', 'J1  ', 100)
('J3  ', '弹簧厂    ', '天津  ', 'S9  ', 'P6  ', 'J3  ', 300)
('J4  ', '造船厂    ', '天津  ', 'S9  ', 'P6  ', 'J4  ', 200)
('J9  ', '重汽      ', '杭州  ', 'S10 ', 'P7  ', 'J9  ', 300)
('J8  ', '上汽      ', '上海  ', 'S10 ', 'P7  ', 'J8  ', 300)
('J5  ', '机车厂    ', '唐山  ', 'S10 ', 'P1  ', 'J5  ', 100)
```

In [57]:

```
# 查询零件的供应商信息
sql = "select P.Pno, P.Pname, P.color, P.weight, SPJ.Sno, SPJ.Pno, SPJ.Jno, SPJ.
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('P1  ', '螺母      ', '红 ', 12, 'S1  ', 'P1  ', 'J1  ', 200)
('P1  ', '螺母      ', '红 ', 12, 'S1  ', 'P1  ', 'J3  ', 100)
('P1  ', '螺母      ', '红 ', 12, 'S1  ', 'P1  ', 'J4  ', 700)
('P2  ', '螺栓      ', '绿 ', 17, 'S1  ', 'P2  ', 'J2  ', 100)
('P3  ', '螺丝刀     ', '蓝 ', 14, 'S2  ', 'P3  ', 'J1  ', 400)
('P3  ', '螺丝刀     ', '蓝 ', 14, 'S2  ', 'P3  ', 'J2  ', 200)
('P3  ', '螺丝刀     ', '蓝 ', 14, 'S2  ', 'P3  ', 'J4  ', 500)
('P3  ', '螺丝刀     ', '蓝 ', 14, 'S2  ', 'P3  ', 'J5  ', 400)
('P5  ', '凸轮      ', '蓝 ', 40, 'S2  ', 'P5  ', 'J1  ', 400)
('P5  ', '凸轮      ', '蓝 ', 40, 'S2  ', 'P5  ', 'J2  ', 100)
('P1  ', '螺母      ', '红 ', 12, 'S3  ', 'P1  ', 'J1  ', 200)
('P3  ', '螺丝刀     ', '蓝 ', 14, 'S3  ', 'P3  ', 'J1  ', 200)
('P5  ', '凸轮      ', '蓝 ', 40, 'S4  ', 'P5  ', 'J1  ', 100)
('P6  ', '齿轮      ', '红 ', 30, 'S4  ', 'P6  ', 'J3  ', 300)
('P6  ', '齿轮      ', '红 ', 30, 'S4  ', 'P6  ', 'J4  ', 200)
('P2  ', '螺栓      ', '绿 ', 17, 'S5  ', 'P2  ', 'J4  ', 100)
('P3  ', '螺丝刀     ', '蓝 ', 14, 'S5  ', 'P3  ', 'J1  ', 200)
('P6  ', '齿轮      ', '红 ', 30, 'S5  ', 'P6  ', 'J2  ', 200)
('P6  ', '齿轮      ', '红 ', 30, 'S5  ', 'P6  ', 'J4  ', 500)
('P8  ', '钢筋      ', '灰 ', 100, 'S6  ', 'P8  ', 'J3  ', 600)
('P7  ', '轮胎      ', '黑 ', 100, 'S6  ', 'P7  ', 'J8  ', 900)
('P8  ', '钢筋      ', '灰 ', 100, 'S6  ', 'P8  ', 'J4  ', 100)
('P7  ', '轮胎      ', '黑 ', 100, 'S6  ', 'P7  ', 'J9  ', 500)
('P7  ', '轮胎      ', '黑 ', 100, 'S6  ', 'P7  ', 'J2  ', 500)
('P5  ', '凸轮      ', '蓝 ', 40, 'S6  ', 'P5  ', 'J7  ', 50)
('P8  ', '钢筋      ', '灰 ', 100, 'S7  ', 'P8  ', 'J1  ', 100)
('P7  ', '轮胎      ', '黑 ', 100, 'S7  ', 'P7  ', 'J2  ', 200)
('P7  ', '轮胎      ', '黑 ', 100, 'S7  ', 'P7  ', 'J9  ', 500)
('P5  ', '凸轮      ', '蓝 ', 40, 'S8  ', 'P5  ', 'J5  ', 300)
('P6  ', '齿轮      ', '红 ', 30, 'S8  ', 'P6  ', 'J1  ', 100)
('P7  ', '轮胎      ', '黑 ', 100, 'S8  ', 'P7  ', 'J9  ', 450)
('P5  ', '凸轮      ', '蓝 ', 40, 'S9  ', 'P5  ', 'J1  ', 100)
('P6  ', '齿轮      ', '红 ', 30, 'S9  ', 'P6  ', 'J3  ', 300)
('P6  ', '齿轮      ', '红 ', 30, 'S9  ', 'P6  ', 'J4  ', 200)
('P7  ', '轮胎      ', '黑 ', 100, 'S10 ', 'P7  ', 'J9  ', 300)
('P7  ', '轮胎      ', '黑 ', 100, 'S10 ', 'P7  ', 'J8  ', 300)
('P1  ', '螺母      ', '红 ', 12, 'S10 ', 'P1  ', 'J5  ', 100)
```

In [58]:
```
# 查询 QTY 大于 500 的零件的供应商信息
sql = "select P.Pno, P.Pname, P.color, P.weight, SPJ.Sno, SPJ.Pno, SPJ.Jno, SPJ.
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('P1  ', '螺母      ', '红 ', 12, 'S1  ', 'P1  ', 'J4  ', 700)
('P8  ', '钢筋      ', '灰 ', 100, 'S6  ', 'P8  ', 'J3  ', 600)
('P7  ', '轮胎      ', '黑 ', 100, 'S6  ', 'P7  ', 'J8  ', 900)
```

练习一：多表查询练习

In [59]:
```
# 填写代码，可在多个cell中完成

# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 练习2.1.1 查询工厂订购供应商零件的信息（带有工厂、零件和供应商各自的属性信息）；
sql = "select J.Jno, J.Jname, J.CITY, P.Pno,P.Pname,P.color,P.weight,S.Sno,S.Sna
cur.execute(sql)
```

```
for tuple in cur.fetchall():
    print(tuple)
```

```
('J1  ', '三建      ', '北京  ', 'P1  ', '螺母      ', '红 ', 12, 'S1  ', '精益
', '20', '天津  ', 200)
('J3  ', '弹簧厂     ', '天津  ', 'P1  ', '螺母      ', '红 ', 12, 'S1  ', '精益
', '20', '天津  ', 100)
('J4  ', '造船厂     ', '天津  ', 'P1  ', '螺母      ', '红 ', 12, 'S1  ', '精益
', '20', '天津  ', 700)
('J2  ', '一汽       ', '长春  ', 'P2  ', '螺栓      ', '绿 ', 17, 'S1  ', '精益
', '20', '天津  ', 100)
('J1  ', '三建      ', '北京  ', 'P3  ', '螺丝刀     ', '蓝 ', 14, 'S2  ', '盛锡
', '10', '北京  ', 400)
('J2  ', '一汽       ', '长春  ', 'P3  ', '螺丝刀     ', '蓝 ', 14, 'S2  ', '盛锡
', '10', '北京  ', 200)
('J4  ', '造船厂     ', '天津  ', 'P3  ', '螺丝刀     ', '蓝 ', 14, 'S2  ', '盛锡
', '10', '北京  ', 500)
('J5  ', '机车厂     ', '唐山  ', 'P3  ', '螺丝刀     ', '蓝 ', 14, 'S2  ', '盛锡
', '10', '北京  ', 400)
('J1  ', '三建      ', '北京  ', 'P5  ', '凸轮      ', '蓝 ', 40, 'S2  ', '盛锡
', '10', '北京  ', 400)
('J2  ', '一汽       ', '长春  ', 'P5  ', '凸轮      ', '蓝 ', 40, 'S2  ', '盛锡
', '10', '北京  ', 100)
('J1  ', '三建      ', '北京  ', 'P1  ', '螺母      ', '红 ', 12, 'S3  ', '东方红
', '30', '北京  ', 200)
('J1  ', '三建      ', '北京  ', 'P3  ', '螺丝刀     ', '蓝 ', 14, 'S3  ', '东方
红     ', '30', '北京  ', 200)
('J1  ', '三建      ', '北京  ', 'P5  ', '凸轮      ', '蓝 ', 40, 'S4  ', '丰泰盛
', '20', '天津  ', 100)
('J3  ', '弹簧厂     ', '天津  ', 'P6  ', '齿轮      ', '红 ', 30, 'S4  ', '丰泰
盛     ', '20', '天津  ', 300)
('J4  ', '造船厂     ', '天津  ', 'P6  ', '齿轮      ', '红 ', 30, 'S4  ', '丰泰
盛     ', '20', '天津  ', 200)
('J4  ', '造船厂     ', '天津  ', 'P2  ', '螺栓      ', '绿 ', 17, 'S5  ', '为民
', '30', '上海  ', 100)
('J1  ', '三建      ', '北京  ', 'P3  ', '螺丝刀     ', '蓝 ', 14, 'S5  ', '为民
', '30', '上海  ', 200)
('J2  ', '一汽       ', '长春  ', 'P6  ', '齿轮      ', '红 ', 30, 'S5  ', '为民
', '30', '上海  ', 200)
('J4  ', '造船厂     ', '天津  ', 'P6  ', '齿轮      ', '红 ', 30, 'S5  ', '为民
', '30', '上海  ', 500)
('J3  ', '弹簧厂     ', '天津  ', 'P8  ', '钢筋      ', '灰 ', 100, 'S6  ', '盛
锡-2    ', '10', '上海  ', 600)
('J8  ', '上汽       ', '上海  ', 'P7  ', '轮胎      ', '黑 ', 100, 'S6  ', '盛锡-
2    ', '10', '上海  ', 900)
('J4  ', '造船厂     ', '天津  ', 'P8  ', '钢筋      ', '灰 ', 100, 'S6  ', '盛
锡-2    ', '10', '上海  ', 100)
('J9  ', '重汽       ', '杭州  ', 'P7  ', '轮胎      ', '黑 ', 100, 'S6  ', '盛锡-
2    ', '10', '上海  ', 500)
('J2  ', '一汽       ', '长春  ', 'P7  ', '轮胎      ', '黑 ', 100, 'S6  ', '盛锡-
2    ', '10', '上海  ', 500)
('J7  ', '半导体厂    ', '南京  ', 'P5  ', '凸轮      ', '蓝 ', 40, 'S6  ', '盛
锡-2    ', '10', '上海  ', 50)
('J1  ', '三建      ', '北京  ', 'P8  ', '钢筋      ', '灰 ', 100, 'S7  ', '三角-
2    ', '30', '北京  ', 100)
('J2  ', '一汽       ', '长春  ', 'P7  ', '轮胎      ', '黑 ', 100, 'S7  ', '三角-
2    ', '30', '北京  ', 200)
('J9  ', '重汽       ', '杭州  ', 'P7  ', '轮胎      ', '黑 ', 100, 'S7  ', '三角-
2    ', '30', '北京  ', 500)
('J5  ', '机车厂     ', '唐山  ', 'P5  ', '凸轮      ', '蓝 ', 40, 'S8  ', '精益-
2    ', '20', '广州  ', 300)
('J1  ', '三建      ', '北京  ', 'P6  ', '齿轮      ', '红 ', 30, 'S8  ', '精益-2
', '20', '广州  ', 100)
```

```
('J9  ', '重汽      ', '杭州  ', 'P7  ', '轮胎      ', '黑 ', 100, 'S8  ', '精益-
2    ', '20', '广州  ', 450)
('J1  ', '三建      ', '北京  ', 'P5  ', '凸轮      ', '蓝 ', 40, 'S9  ', '三角
', '10', '深圳  ', 100)
('J3  ', '弹簧厂     ', '天津  ', 'P6  ', '齿轮      ', '红 ', 30, 'S9  ', '三角
', '10', '深圳  ', 300)
('J4  ', '造船厂     ', '天津  ', 'P6  ', '齿轮      ', '红 ', 30, 'S9  ', '三角
', '10', '深圳  ', 200)
('J9  ', '重汽      ', '杭州  ', 'P7  ', '轮胎      ', '黑 ', 100, 'S10 ', '顺丰
', '20', '广州  ', 300)
('J8  ', '上汽      ', '上海  ', 'P7  ', '轮胎      ', '黑 ', 100, 'S10 ', '顺丰
', '20', '广州  ', 300)
('J5  ', '机车厂     ', '唐山  ', 'P1  ', '螺母      ', '红 ', 12, 'S10 ', '顺丰
', '20', '广州  ', 100)
```

In [60]:
```python
# 填写代码，可在多个cell中完成

# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# Pname：螺母  螺栓
# Jname：弹簧厂  机车厂

# 练习2.1.3 查询订购了以下零件【Pname 零件名称 `螺母` or `螺栓`】，且下列工厂【Jn
sql = "select J.Jno,J.Jname,J.CITY,P.Pno, P.Pname, P.color, P.weight, S.Sno,S.Sn
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('J3  ', '弹簧厂     ', '天津  ', 'P1  ', '螺母      ', '红 ', 12, 'S1  ', '精益
', '天津  ', 100)
('J5  ', '机车厂     ', '唐山  ', 'P1  ', '螺母      ', '红 ', 12, 'S10 ', '顺丰
', '广州  ', 100)
```

2.2 分组查询实现

In [61]:
```python
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 统计订购零件的订单的个数    # 假定工厂订购供应商零件信息表SPJ 中 一条数据为一个
sql = "select count(*) from SPJ;"
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(37,)
```

In [62]:
```python
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 统计订购零件的订单中 QTY 大于等于150 的订单的个数    # 假定工厂订购供应商零件信
sql = "select count(*) from SPJ where QTY >= 150;"
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(26,)
```

In [63]:
```python
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）
```

```
# 统计工厂所在地在【`北京`,`上海`,`天津`】的工厂个数
sql = "select count(*) from J where CITY in ('北京', '上海', '天津');"
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(4,)
```

In [64]:
```
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 统计工厂分布的城市的数量（去重）
sql = "select count(distinct CITY) as city_num from J;"
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(8,)
```

In [65]:
```
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 统计订购零件的订单的 QTY 的统计信息     # 假定工厂订购供应商零件信息表SPJ 中 一条
sql = "select MIN(QTY), MAX(QTY), SUM(QTY), COUNT(*), COUNT(1), AVG(QTY) from SP
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(50, 900, 10700, 37, 37, Decimal('289.1891891891891892'))
```

练习二：分组查询练习

In [66]:
```
# 填写代码，可在多个cell中完成
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 练习2.2.1 统计有订购零件（有订单） 的工厂的个数
sql = "select COUNT(DISTINCT Jno) AS Factory_Count from SPJ;"
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(8,)
```

In [67]:
```
# 填写代码，可在多个cell中完成
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 练习2.2.2 统计被订购零件（有订单） 的供应商的个数
sql = "select COUNT(DISTINCT Sno) AS Supplier_Count from SPJ;"
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(10,)
```

In [68]:
```
# 填写代码，可在多个cell中完成
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 练习2.2.3-1 分别计算出 北京 上海 天津 三个地区各自的经销商的个数    # 可使用多条
```

```
sql = "select COUNT(DISTINCT Sno) AS Beijing_Supplier_Count from S where CITY =
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(3,)
```

In [69]:
```
sql = "select COUNT(DISTINCT Sno) AS Shaihai_Supplier_Count from S where CITY =
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(2,)
```

In [70]:
```
sql = "select COUNT(DISTINCT Sno) AS Tianjin_Supplier_Count from S where CITY =
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
(2,)
```

In [71]:
```
# 练习2.2.3-2 一条SQL来解决上述问题
sql = "select CITY, COUNT(DISTINCT Sno) AS Supplier_Count from S where CITY IN (
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('上海  ', 2)
('北京  ', 3)
('天津  ', 2)
```

2.3 分组聚合查询实现

In [72]:
```
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 查询订购【北京 上海 天津】 三个地区的经销商订单 的工厂城市的分布 和 订单数
sql = "select J.CITY, COUNT(distinct SPJ.Jno) as city_num, COUNT(SPJ.Jno) as ord
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('上海  ', 1, 2)
('北京  ', 1, 10)
('南京  ', 1, 1)
('唐山  ', 1, 3)
('天津  ', 2, 11)
('杭州  ', 1, 4)
('长春  ', 1, 6)
```

In [73]:
```
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 查询订购的订单 的零件的颜色分布 和 订单数 和 总计的 QTY
sql = "select P.COLOR, COUNT(distinct P.Pno) as color_num, COUNT(SPJ.Pno) as ord
cur.execute(sql)
```

```
for tuple in cur.fetchall():
    print(tuple)
```

```
('灰 ', 1, 3, 3)
('红 ', 2, 8, 8)
('绿 ', 1, 2, 2)
('蓝 ', 2, 10, 10)
('黑 ', 1, 5, 5)
```

In [74]:
```
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 统计 每个经销商订购的每个工厂的订单个数 和 零件的总数量-QTY
sql = "select Sno, Jno, count(*) as order_num, sum(QTY) as SUM_QTY from SPJ grou
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('S7  ', 'J9  ', 1, 500)
('S9  ', 'J4  ', 1, 200)
('S7  ', 'J2  ', 1, 200)
('S6  ', 'J7  ', 1, 50)
('S2  ', 'J5  ', 1, 400)
('S1  ', 'J1  ', 1, 200)
('S1  ', 'J2  ', 1, 100)
('S6  ', 'J2  ', 1, 500)
('S5  ', 'J1  ', 1, 200)
('S4  ', 'J1  ', 1, 100)
('S5  ', 'J4  ', 2, 600)
('S10 ', 'J9  ', 1, 300)
('S8  ', 'J1  ', 1, 100)
('S4  ', 'J4  ', 1, 200)
('S10 ', 'J8  ', 1, 300)
('S9  ', 'J1  ', 1, 100)
('S5  ', 'J2  ', 1, 200)
('S9  ', 'J3  ', 1, 300)
('S6  ', 'J9  ', 1, 500)
('S2  ', 'J4  ', 1, 500)
('S1  ', 'J3  ', 1, 100)
('S3  ', 'J1  ', 2, 400)
('S6  ', 'J8  ', 1, 900)
('S2  ', 'J1  ', 2, 800)
('S6  ', 'J4  ', 1, 100)
('S10 ', 'J5  ', 1, 100)
('S8  ', 'J5  ', 1, 300)
('S1  ', 'J4  ', 1, 700)
('S4  ', 'J3  ', 1, 300)
('S8  ', 'J9  ', 1, 450)
('S2  ', 'J2  ', 2, 300)
('S7  ', 'J1  ', 1, 100)
('S6  ', 'J3  ', 1, 600)
```

练习三：分组聚合查询练习

In [85]:
```
# 填写代码，可在多个cell中完成
# 练习2.3.1 分别计算出 【北京 上海 天津】 三个地区各自的经销商的个数  单条 SQL
sql = "select city,Count(DISTINCT Sno) as Supplier_Count from S where city in('北
cur.execute(sql)
```

```
for tuple in cur.fetchall():
    print(tuple)
```

```
('上海  ', 2)
('北京  ', 3)
('天津  ', 2)
```

In [86]:
```
# 填写代码，可在多个cell中完成
# 练习2.3.2 查询订购的订单的 经销商城市的分布 和 订单数 和 总计的 QTY
sql = "select S.City, count(*) as order_num, sum(SPJ.QTY) as SUM_QTY from S join
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('广州  ', 6, 1550)
('北京  ', 11, 3200)
('上海  ', 10, 3650)
('深圳  ', 3, 600)
('天津  ', 7, 1700)
```

In [87]:
```
# 填写代码，可在多个cell中完成
# 练习2.3.3 查询订订单购经销商为【北京 上海 天津】三个地区，其订购的工厂的城市分布
# QTY 分层信息指的是【按照一百分层（QTY [0,100) => 100, [100,200) => 200... ） 70

# 提示 group by city, case when 【QTY】
sql = """
select
    Trim(J.City) as Factory_City,
    Count(*) as Order_Count,
    SUM(SPJ.QTY) as Total_QTY,
    case
        when SUM(SPJ.QTY) < 100 then '0-100'
        when SUM(SPJ.QTY) < 200 then '100-200'
        when SUM(SPJ.QTY) < 300 then '200-300'
        when SUM(SPJ.QTY) < 400 then '300-400'
        when SUM(SPJ.QTY) < 500 then '400-500'
        when SUM(SPJ.QTY) < 600 then '500-600'
        when SUM(SPJ.QTY) < 700 then '600-700'
        Else '700及以上'
    end as QTY_Range
from
    S
join
    SPJ on S.Sno = SPJ.Sno
join
    J on SPJ.Jno = J.Jno
where
    Trim(S.City) in ('北京', '上海', '天津')
    and Trim(J.City) in ('北京','上海','天津')
group by
    Trim(J.City);
"""
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('上海', 1, 900, '700及以上')
('北京', 8, 1800, '700及以上')
('天津', 9, 3100, '700及以上')
```

In [88]:
```
# 填写代码，可在多个cell中完成
# 练习2.3.4 查询订购的订单 的零件的颜色分布 和 订单数 和 总计的 QTY
sql = """
select
    P.Color as Part_Color,
    Count(*) as Order_Count,
    SUM(SPJ.QTY) as Total_QTY
from
    SPJ
join
    P on SPJ.Pno = P.Pno
group by
    P.Color
order by
    Order_Count DESC;
"""
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('红 ', 12, 3100)
('蓝 ', 12, 2950)
('黑 ', 8, 3650)
('灰 ', 3, 800)
('绿 ', 2, 200)
```

2.4 子查询实现

In [90]:
```
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 获取 经销商名称和订单数以及总的 QTY
sql = "select T.Sno, T.Sname, COUNT(*), sum(QTY) from SPJ, (select Sno, Sname fr
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('S1  ', '精益      ', 4, 1100)
('S2  ', '盛锡      ', 6, 2000)
('S3  ', '东方红     ', 2, 400)
('S4  ', '丰泰盛     ', 3, 600)
('S5  ', '为民      ', 4, 1000)
('S6  ', '盛锡-2    ', 6, 2650)
('S7  ', '三角-2    ', 3, 800)
```

In [92]:
```
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 获取 经销商名称和订单数以及总的 QTY
sql = "select T.Sno, T.Sname, COUNT(*), sum(QTY) from SPJ, (select Sno, Sname fr
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('S1  ', '精益      ', 4, 1100)
('S2  ', '盛锡      ', 6, 2000)
('S3  ', '东方红     ', 2, 400)
('S4  ', '丰泰盛     ', 3, 600)
('S5  ', '为民      ', 4, 1000)
('S6  ', '盛锡-2    ', 6, 2650)
('S7  ', '三角-2    ', 3, 800)
```

In [93]:
```
# S（供应商）、P（零件）、J（工厂）、SPJ（工厂订购供应商零件的信息）

# 查询 订购了 北京和天津 地区工厂生产的零件的 经销商信息及其订单信息
# 第一步 sql1 => as temporary table tmp 将sql内容的结果作为临时表 命名为 tmp
sql1 = "select Jno, Jname from J where CITY in ('北京', '天津') ;"

# 第二步
sql2 = "select S.Sno, Sname, SPJ.Jno, tmp.Jname, SPJ.QTY from SPJ, S, tmp where

# 合并 => 将 sql1 子查询 合并到 sql2 中, 得到单条 sql
sql = "select S.Sno, Sname, SPJ.Jno, tmp.Jname, SPJ.QTY, Pno from SPJ, S, (selec

cur.execute(sql)
for tuple in cur.fetchall():
    print(tuple)
```

```
('S1  ', '精益      ', 'J1  ', '三建      ', 200, 'P1  ')
('S1  ', '精益      ', 'J3  ', '弹簧厂     ', 100, 'P1  ')
('S1  ', '精益      ', 'J4  ', '造船厂     ', 700, 'P1  ')
('S2  ', '盛锡      ', 'J1  ', '三建      ', 400, 'P3  ')
('S2  ', '盛锡      ', 'J4  ', '造船厂     ', 500, 'P3  ')
('S2  ', '盛锡      ', 'J1  ', '三建      ', 400, 'P5  ')
('S3  ', '东方红     ', 'J1  ', '三建      ', 200, 'P1  ')
('S3  ', '东方红     ', 'J1  ', '三建      ', 200, 'P3  ')
('S4  ', '丰泰盛     ', 'J1  ', '三建      ', 100, 'P5  ')
('S4  ', '丰泰盛     ', 'J3  ', '弹簧厂     ', 300, 'P6  ')
('S4  ', '丰泰盛     ', 'J4  ', '造船厂     ', 200, 'P6  ')
('S5  ', '为民      ', 'J4  ', '造船厂     ', 100, 'P2  ')
('S5  ', '为民      ', 'J1  ', '三建      ', 200, 'P3  ')
('S5  ', '为民      ', 'J4  ', '造船厂     ', 500, 'P6  ')
('S6  ', '盛锡-2    ', 'J3  ', '弹簧厂     ', 600, 'P8  ')
('S6  ', '盛锡-2    ', 'J4  ', '造船厂     ', 100, 'P8  ')
('S7  ', '三角-2    ', 'J1  ', '三建      ', 100, 'P8  ')
('S8  ', '精益-2    ', 'J1  ', '三建      ', 100, 'P6  ')
('S9  ', '三角      ', 'J1  ', '三建      ', 100, 'P5  ')
('S9  ', '三角      ', 'J3  ', '弹簧厂     ', 300, 'P6  ')
('S9  ', '三角      ', 'J4  ', '造船厂     ', 200, 'P6  ')
```

练习四：子查询练习

In [94]:
```
# 填写代码，可在多个cell中完成
# 练习2.4.1 以经销商ID为粒度，获取经销商的名称和 其订单数、总的 QTY
# 思考该查询使用多表连接如何实现

sql = """
select
    S.Sno as Distributor_ID,
    S.Sname as Distributor_Name,
    Count(SPJ.Pno) as Order_Count,
    SUM(SPJ.QTY) as Total_QTY
from
    S
```

```
join
    SPJ on S.Sno = SPJ.Sno
group by
    S.Sno, S.Sname
order by
    Order_Count DESC;
"""
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('S6  ', '盛锡-2    ', 6, 2650)
('S2  ', '盛锡      ', 6, 2000)
('S1  ', '精益      ', 4, 1100)
('S5  ', '为民      ', 4, 1000)
('S7  ', '三角-2    ', 3, 800)
('S8  ', '精益-2    ', 3, 850)
('S9  ', '三角      ', 3, 600)
('S10 ', '顺丰      ', 3, 700)
('S4  ', '丰泰盛     ', 3, 600)
('S3  ', '东方红     ', 2, 400)
```

In [95]:

```
# 填写代码，可在多个cell中完成
# 练习2.4.2 单条 SQL: 获取有订单的工厂信息 & 有订单的供应商信息 & 有订单的零件信息
sql = """
select
    J.Jno as Factory_ID,
    J.Jname as Factory_Name,
    J.City as Factory_City,
    S.Sno as Supplier_ID,
    S.Sname as Supplier_Name,
    S.City as Supplier_City,
    P.Pno as Part_ID,
    P.Pname as Part_Name,
    SPJ.QTY as Quantity
from
    SPJ
join
    J on SPJ.Jno = J.Jno
join
    S on SPJ.Sno = S.Sno
join
    P on SPJ.Pno = P.Pno
where
    SPJ.QTY > 0;
"""
cur.execute(sql)

for tuple in cur.fetchall():
    print(tuple)
```

```
('J1  ', '三建      ', '北京  ', 'S1  ', '精益      ', '天津  ', 'P1  ', '螺母
', 200)
('J3  ', '弹簧厂     ', '天津  ', 'S1  ', '精益      ', '天津  ', 'P1  ', '螺母
', 100)
('J4  ', '造船厂     ', '天津  ', 'S1  ', '精益      ', '天津  ', 'P1  ', '螺母
', 700)
('J2  ', '一汽       ', '长春  ', 'S1  ', '精益      ', '天津  ', 'P2  ', '螺栓
', 100)
('J1  ', '三建      ', '北京  ', 'S2  ', '盛锡      ', '北京  ', 'P3  ', '螺丝刀
', 400)
('J2  ', '一汽       ', '长春  ', 'S2  ', '盛锡      ', '北京  ', 'P3  ', '螺丝刀
', 200)
('J4  ', '造船厂     ', '天津  ', 'S2  ', '盛锡      ', '北京  ', 'P3  ', '螺丝刀
', 500)
('J5  ', '机车厂     ', '唐山  ', 'S2  ', '盛锡      ', '北京  ', 'P3  ', '螺丝刀
', 400)
('J1  ', '三建      ', '北京  ', 'S2  ', '盛锡      ', '北京  ', 'P5  ', '凸轮
', 400)
('J2  ', '一汽       ', '长春  ', 'S2  ', '盛锡      ', '北京  ', 'P5  ', '凸轮
', 100)
('J1  ', '三建      ', '北京  ', 'S3  ', '东方红     ', '北京  ', 'P1  ', '螺母
', 200)
('J1  ', '三建      ', '北京  ', 'S3  ', '东方红     ', '北京  ', 'P3  ', '螺丝刀
', 200)
('J1  ', '三建      ', '北京  ', 'S4  ', '丰泰盛     ', '天津  ', 'P5  ', '凸轮
', 100)
('J3  ', '弹簧厂     ', '天津  ', 'S4  ', '丰泰盛     ', '天津  ', 'P6  ', '齿轮
', 300)
('J4  ', '造船厂     ', '天津  ', 'S4  ', '丰泰盛     ', '天津  ', 'P6  ', '齿轮
', 200)
('J4  ', '造船厂     ', '天津  ', 'S5  ', '为民      ', '上海  ', 'P2  ', '螺栓
', 100)
('J1  ', '三建      ', '北京  ', 'S5  ', '为民      ', '上海  ', 'P3  ', '螺丝刀
', 200)
('J2  ', '一汽       ', '长春  ', 'S5  ', '为民      ', '上海  ', 'P6  ', '齿轮
', 200)
('J4  ', '造船厂     ', '天津  ', 'S5  ', '为民      ', '上海  ', 'P6  ', '齿轮
', 500)
('J3  ', '弹簧厂     ', '天津  ', 'S6  ', '盛锡-2    ', '上海  ', 'P8  ', '钢筋
', 600)
('J8  ', '上汽       ', '上海  ', 'S6  ', '盛锡-2    ', '上海  ', 'P7  ', '轮胎
', 900)
('J4  ', '造船厂     ', '天津  ', 'S6  ', '盛锡-2    ', '上海  ', 'P8  ', '钢筋
', 100)
('J9  ', '重汽       ', '杭州  ', 'S6  ', '盛锡-2    ', '上海  ', 'P7  ', '轮胎
', 500)
('J2  ', '一汽       ', '长春  ', 'S6  ', '盛锡-2    ', '上海  ', 'P7  ', '轮胎
', 500)
('J7  ', '半导体厂    ', '南京  ', 'S6  ', '盛锡-2    ', '上海  ', 'P5  ', '凸轮
', 50)
('J1  ', '三建      ', '北京  ', 'S7  ', '三角-2    ', '北京  ', 'P8  ', '钢筋
', 100)
('J2  ', '一汽       ', '长春  ', 'S7  ', '三角-2    ', '北京  ', 'P7  ', '轮胎
', 200)
('J9  ', '重汽       ', '杭州  ', 'S7  ', '三角-2    ', '北京  ', 'P7  ', '轮胎
', 500)
('J5  ', '机车厂     ', '唐山  ', 'S8  ', '精益-2    ', '广州  ', 'P5  ', '凸轮
', 300)
('J1  ', '三建      ', '北京  ', 'S8  ', '精益-2    ', '广州  ', 'P6  ', '齿轮
', 100)
```

```
('J9  ', '重汽      ', '杭州  ', 'S8  ', '精益-2    ', '广州  ', 'P7  ', '轮胎
', 450)
('J1  ', '三建      ', '北京  ', 'S9  ', '三角      ', '深圳  ', 'P5  ', '凸轮
', 100)
('J3  ', '弹簧厂     ', '天津  ', 'S9  ', '三角      ', '深圳  ', 'P6  ', '齿轮
', 300)
('J4  ', '造船厂     ', '天津  ', 'S9  ', '三角      ', '深圳  ', 'P6  ', '齿轮
', 200)
('J9  ', '重汽      ', '杭州  ', 'S10 ', '顺丰      ', '广州  ', 'P7  ', '轮胎
', 300)
('J8  ', '上汽      ', '上海  ', 'S10 ', '顺丰      ', '广州  ', 'P7  ', '轮胎
', 300)
('J5  ', '机车厂     ', '唐山  ', 'S10 ', '顺丰      ', '广州  ', 'P1  ', '螺母
', 100)
```

In [ ]: