

# 第三次理论作业

## 第一题

SQL 的特点有：

- 综合统一，集数据定义语言、数据操纵语言和数据控制语言的功能于一体
- 高度非过程化，使用 SQL 语言时只要提出“做什么”，无须指明怎么做，因此无须了解存储路径。存储路径的选择以及 SQL 语言的操作过程由数据库系统自动完成
- 面向集合的操纵方式，SQL 语言采用集合操作方式，操作对象、查找结果是元组的集合，一次插入、删除、更新操作的对象也可以是元组的集合
- 即是自含式语言，也是嵌入式语言

## 第三题

1.  $\sigma_{A=10}(S)$



```
SELECT * FROM S WHERE A = 10
```

2.  $\pi_{A,B}(S)$



```
SELECT DISTINCT A, B FROM S
```

3.  $S \bowtie T$



```
SELECT A, B, S.C, S.D, E, F  
FROM S, T  
WHERE S.C=T.C AND S.D=T.D
```

$$4. S \bowtie_{S.C=T.C} T$$



```
SELECT A,B,S.C,S.D,T.C,T.D,E,F
FROM S,T
WHERE S.C=T.C
```

$$5. S \bowtie_{A<E} T$$



```
SELECT A,B,S.C,S.D,T.C,T.D,E,F
FROM S,T
WHERE A<E
```

$$6. \pi_{C,D}(S) \times T$$



```
SELECT S1.C,S1.D,T.C,T.D,E,F
FROM T, (
    SELECT DISTINCT C,D FROM S
) AS S1
```

## 第四题

### 建表



```
CREATE TABLE S (
    SNO CHAR(3),
    SNAME CHAR(10),
    STATUS CHAR(2),
    CITY CHAR(10)
);
CREATE TABLE P (
    PNO CHAR(3),
    PNAME CHAR(10),
    COLOR CHAR(4),
    WEIGHT INT
);
CREATE TABLE J (
    JNO CHAR(3),
```

```
JNAME CHAR(10),
CITY CHAR(10)
);
CREATE TABLE SPJ (
SNO CHAR(3),
PNO CHAR(3),
JNO CHAR(3),
QTY INT
);
```

## 插入数据

```
INSERT INTO S VALUES('S1','精益',20,'天津');
INSERT INTO S VALUES('S2','盛锡',10,'北京');
INSERT INTO S VALUES('S3','东方红',30,'北京');
INSERT INTO S VALUES('S4','丰泰盛',20,'天津');
INSERT INTO S VALUES('S5','为民',30,'上海');

INSERT INTO P VALUES('P1','螺母','红',12);
INSERT INTO P VALUES('P2','螺栓','绿',17);
INSERT INTO P VALUES('P3','螺丝刀','蓝',14);
INSERT INTO P VALUES('P4','螺丝刀','红',14);
INSERT INTO P VALUES('P5','凸轮','蓝',40);
INSERT INTO P VALUES('P6','齿轮','红',30);

INSERT INTO J VALUES('J1','三建','北京');
INSERT INTO J VALUES('J2','一汽','长春');
INSERT INTO J VALUES('J3','弹簧厂','天津');
INSERT INTO J VALUES('J4','造船厂','天津');
INSERT INTO J VALUES('J5','机车厂','唐山');
INSERT INTO J VALUES('J6','无线电厂','常州');
INSERT INTO J VALUES('J7','半导体厂','南京');

INSERT INTO SPJ VALUES('S1','P1','J1',200);
INSERT INTO SPJ VALUES('S1','P1','J3',100);
INSERT INTO SPJ VALUES('S1','P1','J4',700);
INSERT INTO SPJ VALUES('S1','P2','J2',100);
INSERT INTO SPJ VALUES('S2','P3','J1',400);
INSERT INTO SPJ VALUES('S2','P3','J2',200);
INSERT INTO SPJ VALUES('S2','P3','J4',500);
INSERT INTO SPJ VALUES('S2','P3','J5',400);
INSERT INTO SPJ VALUES('S2','P5','J1',400);
INSERT INTO SPJ VALUES('S2','P5','J2',100);
INSERT INTO SPJ VALUES('S3','P1','J1',200);
INSERT INTO SPJ VALUES('S3','P3','J1',200);
INSERT INTO SPJ VALUES('S4','P5','J1',100);
INSERT INTO SPJ VALUES('S4','P6','J3',300);
INSERT INTO SPJ VALUES('S4','P6','J4',200);
INSERT INTO SPJ VALUES('S5','P2','J4',100);
INSERT INTO SPJ VALUES('S5','P3','J1',200);
INSERT INTO SPJ VALUES('S5','P6','J2',200);
```

```
INSERT INTO SPJ VALUES ('S5', 'P6', 'J4', 50);
```

## 结果

The screenshot shows a database client interface with a sidebar on the left displaying the database structure: postgres@localhost > postgres > public > 表 4 > j. The main pane shows the 'j' table with columns jno, jname, and city. The table contains 7 rows of data.

	jno	jname	city
1	J1	三建	北京
2	J2	一汽	长春
3	J3	弹簧厂	天津
4	J4	造船厂	天津
5	J5	机车厂	唐山
6	J6	无线电厂	常州
7	J7	半导体厂	南京

The screenshot shows a database client interface with a sidebar on the left displaying the database structure: postgres@localhost > postgres > public > 表 4 > p. The main pane shows the 'p' table with columns pno, pname, color, and weight. The table contains 6 rows of data.

	pno	pname	color	weight
1	P1	螺母	红	12
2	P2	螺栓	绿	17
3	P3	螺丝刀	蓝	14
4	P4	螺丝刀	红	14
5	P5	凸轮	蓝	40
6	P6	齿轮	红	30

The screenshot shows a database client interface with a sidebar on the left displaying the database structure: postgres@localhost > postgres > public > 表 4 > s. The main pane shows the 's' table with columns sno, sname, status, and city. The table contains 5 rows of data.

	sno	sname	status	city
1	S1	精益	20	天津
2	S4	丰泰盛	20	天津
3	S2	盛锡	10	北京
4	S3	东方红	30	北京
5	S5	为民	30	上海

	sno	pno	jno	qty
1	S1	P1	J1	200
2	S1	P1	J3	100
3	S1	P1	J4	700
4	S1	P2	J2	100
5	S2	P3	J1	400
6	S2	P3	J2	200
7	S2	P3	J4	500
8	S2	P3	J5	400
9	S2	P5	J1	400
10	S2	P5	J2	100
11	S3	P1	J1	200
12	S3	P3	J1	200
13	S4	P5	J1	100
14	S4	P6	J3	300
15	S4	P6	J4	200
16	S5	P2	J4	100
17	S5	P3	J1	200
18	S5	P6	J2	200
19	S5	P6	J4	50

## 回答

- 供应工程 J1 零件的供应商号码 SNO

```
SELECT DISTINCT SNO FROM SPJ WHERE JNO='J1';
```

sno
1 S1
2 S2
3 S3
4 S4
5 S5

- 供应工程 J1 零件 P1 的供应商号码 SNO

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

服务

Database

- postgres@localhost
  - console 46 ms
    - console 46 ms

输出 postgres.public.spj x

	sno
1	S1
2	S3

- 供应工程 J1 零件为红色的供应商号码 SNO

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

服务

Database

- postgres@localhost
  - console 28 ms
    - console 28 ms

输出 postgres.public.spj x

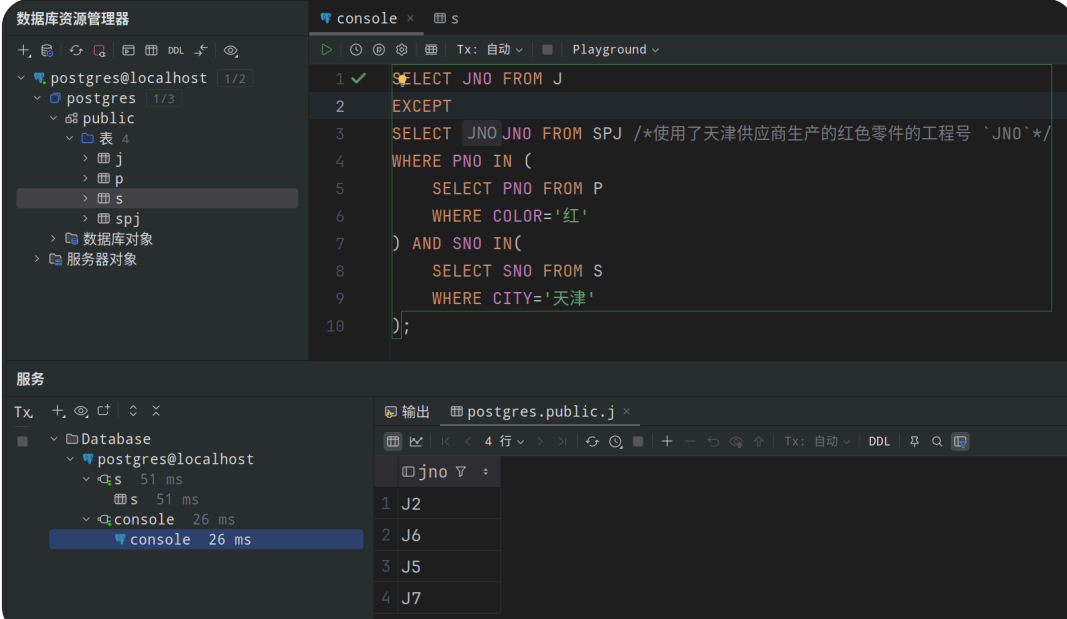
	sno
1	S1
2	S3

- 没有使用天津供应商生产的红色零件的工程号 JNO

- ```

SELECT JNO FROM J
MINUS /*mysql*/
SELECT JNO FROM SPJ /*使用了天津供应商生产的红色零件的工程师号 `JNO`*/
WHERE PNO IN (
    SELECT PNO FROM P
    WHERE COLOR='红'
) AND SNO IN (
    SELECT SNO FROM S
    WHERE CITY='天津'
);

```

- 

The screenshot shows a database management interface with a tree view on the left, a SQL editor in the center, and a results pane on the right. The tree view shows a database named 'postgres' with a public schema containing tables 'j', 'p', 's', and 'spj'. The SQL editor contains the following query:

```

1 SELECT JNO FROM J
2 EXCEPT
3 SELECT JNO JNO FROM SPJ /*使用了天津供应商生产的红色零件的工程师号 `JNO`*/
4 WHERE PNO IN (
5     SELECT PNO FROM P
6     WHERE COLOR='红'
7 ) AND SNO IN(
8     SELECT SNO FROM S
9     WHERE CITY='天津'
10 );

```

The results pane shows the output of the query, which is a list of JNO values:

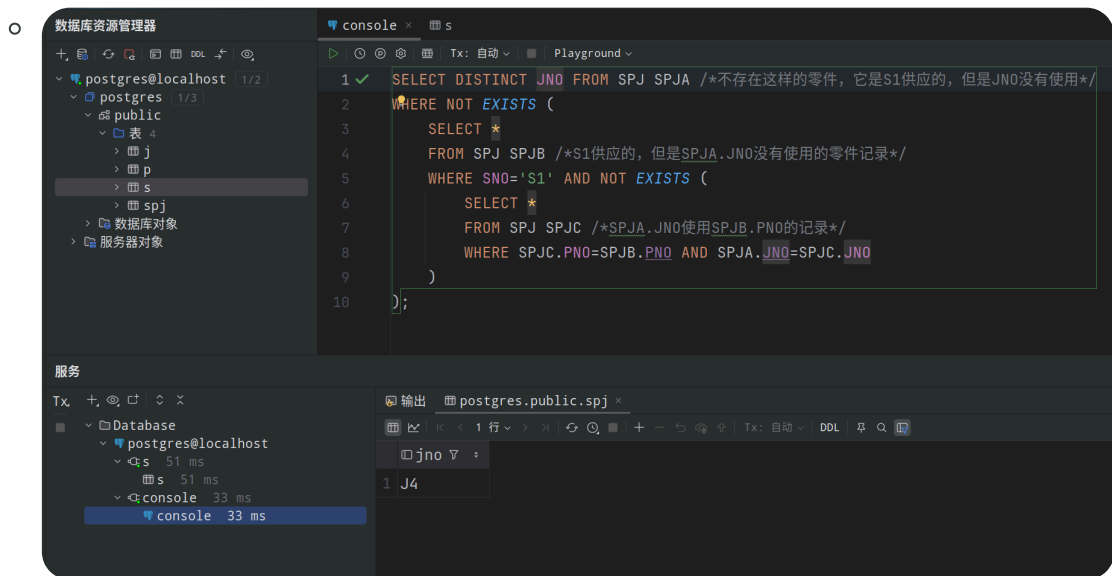
| jno |
|-----|
| J2  |
| J6  |
| J5  |
| J7  |

- 至少使用了供应商 `s1` 所供应的全部零件的工程师号 `JNO`

- ```

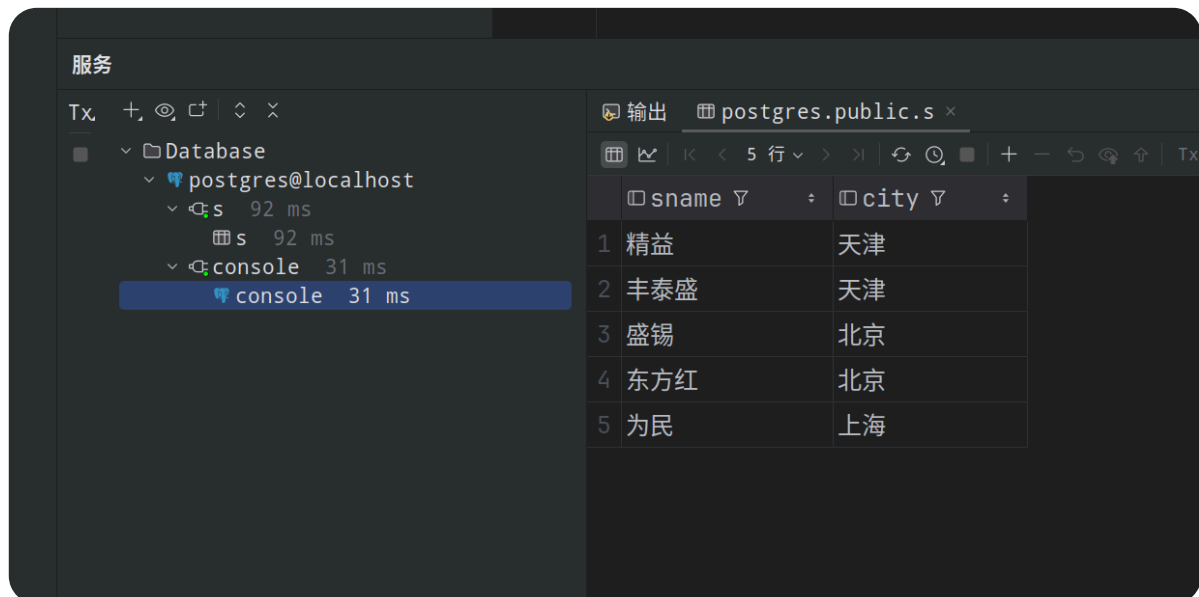
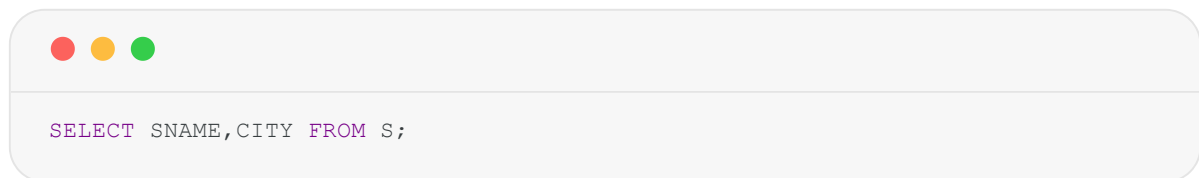
SELECT DISTINCT JNO FROM SPJ SPJA /*不存在这样的零件，它是s1供应的，但是JNO没有使用*/
WHERE NOT EXISTS (
    SELECT *
    FROM SPJ SPJB /*s1供应的，但是SPJA.JNO没有使用的零件记录*/
    WHERE SNO='S1' AND NOT EXISTS (
        SELECT *
        FROM SPJ SPJC /*SPJA.JNO使用SPJB.PNO的记录*/
        WHERE SPJC.PNO=SPJB.PNO AND SPJA.JNO=SPJC.JNO
    )
);

```



## 第五题

找出所有供应商的姓名和所在城市



找出所有零件的名称、颜色、重量





服务

输出 postgres.public.p ×

	pname ▼	color ▼	weight ▼
1	螺母	红	12
2	螺栓	绿	17
3	螺丝刀	蓝	14
4	螺丝刀	红	14
5	凸轮	蓝	40
6	齿轮	红	30

找出使用供应商 S1 所供应零件的工程号码

```
SELECT JNO FROM SPJ WHERE SNO='S1';
```

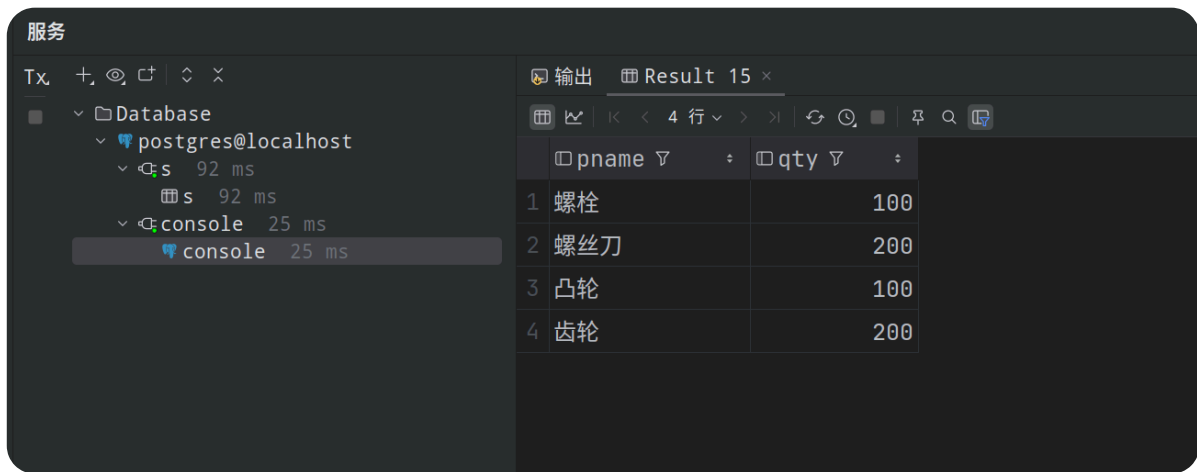
服务

输出 postgres.public.spj ×

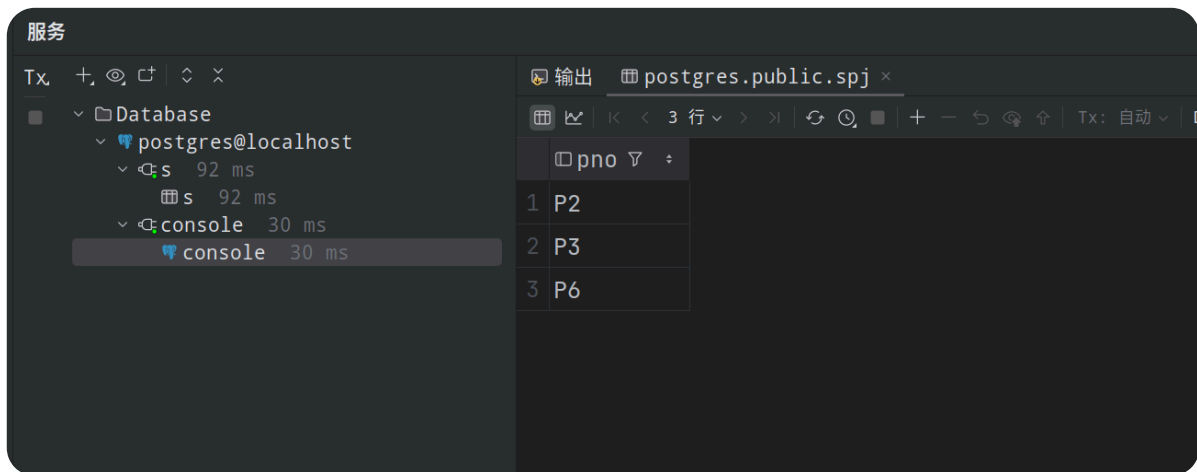
	jno ▼
1	J1
2	J3
3	J4
4	J2

找出工程项目 J2 使用的各种零件的名称及其数量

```
SELECT P.PNAME, SPJ.QTY
FROM P, SPJ
WHERE SPJ.JNO='J2' AND SPJ.PNO=P.PNO;
```

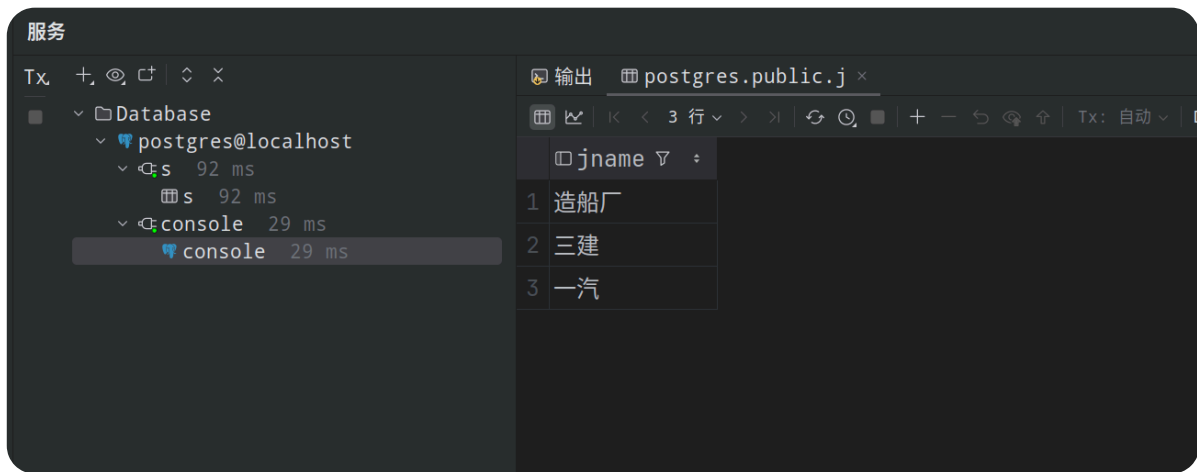


找出上海厂商供应的所有零件号码

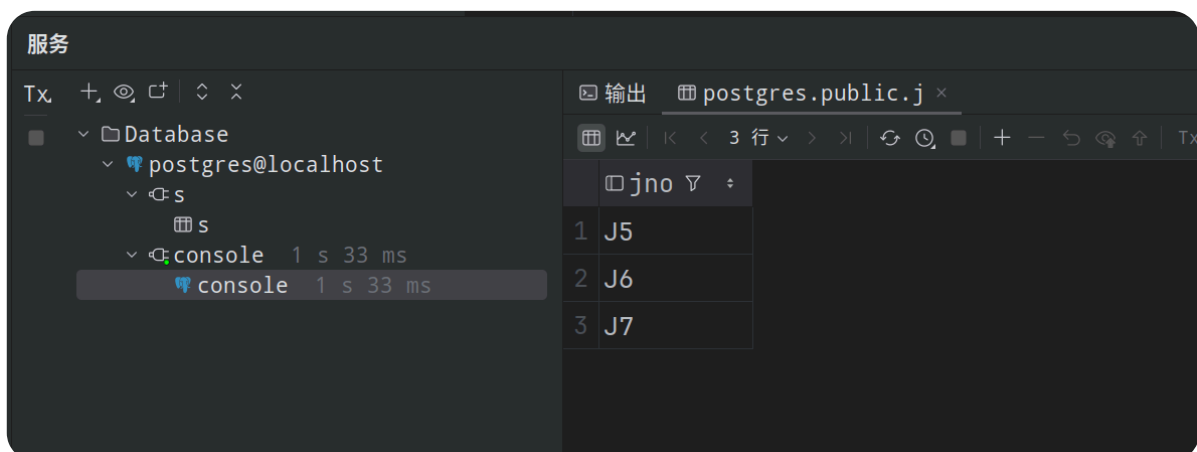


找出使用上海产的零件的工程名称

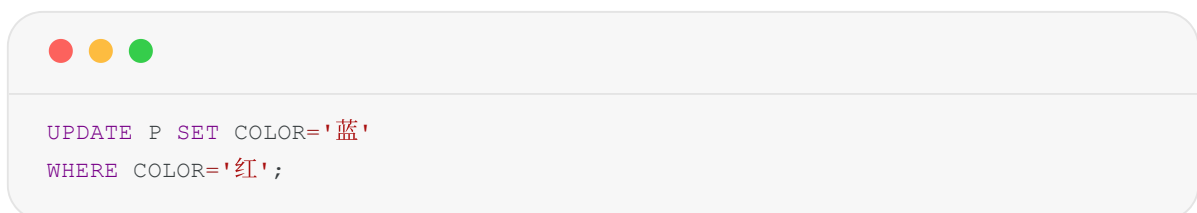




找出没有使用天津产的零件的工程号码



把全部红色零件的颜色改为蓝色



数据库资源管理器

postgres@localhost

1/2

postgres

1/3

public

表 4

j

p

s

spj

数据库对象

服务器对象

console

p

s

6 行

WHERE

ORDER BY

	pno	pname	color	weight
1	P2	螺栓	绿	17
2	P3	螺丝刀	蓝	14
3	P5	凸轮	蓝	40
4	P1	螺母	蓝	12
5	P4	螺丝刀	蓝	14
6	P6	齿轮	蓝	30

由 S5 供给 J4 的零件 P6 改为由 S3 供应

```
UPDATE SPJ SET SNO='S3'
WHERE SNO='S5' AND JNO='J4' AND PNO='P6';
```

数据库资源管理器

postgres@localhost 1/2

postgres 1/3

public

表 4

j

p

s

spj

数据库对象

服务器对象

console

spj

p

s

19 行

WHERE

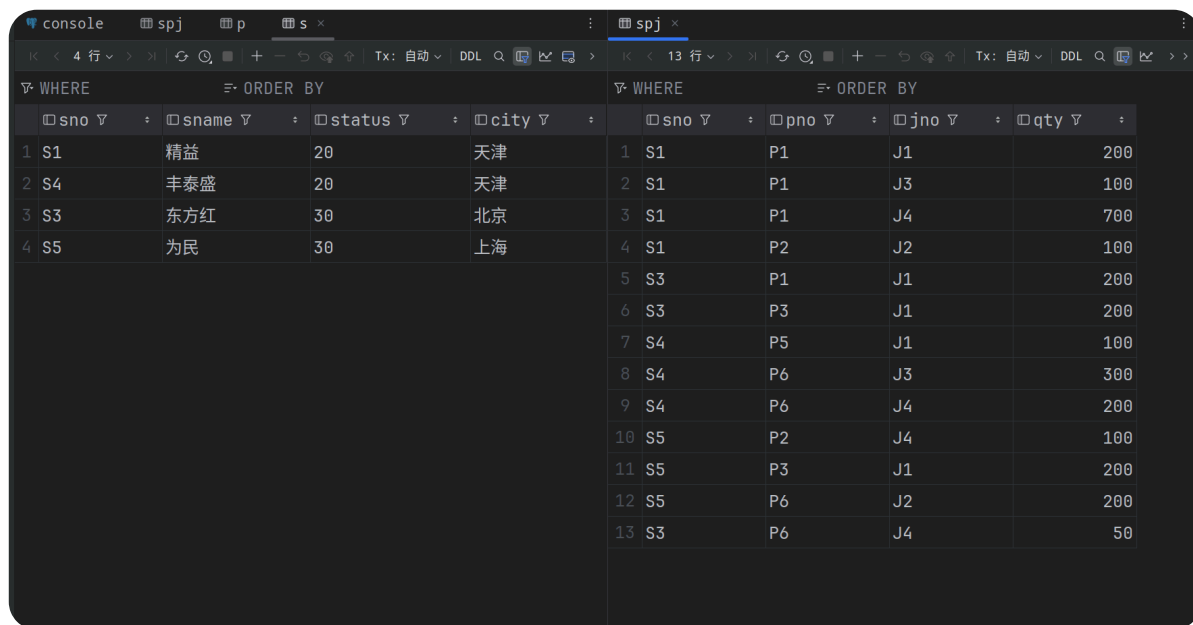
ORDER BY

	sno	pno	jno	qty
1	S1	P1	J1	200
2	S1	P1	J3	100
3	S1	P1	J4	700
4	S1	P2	J2	100
5	S2	P3	J1	400
6	S2	P3	J2	200
7	S2	P3	J4	500
8	S2	P3	J5	400
9	S2	P5	J1	400
10	S2	P5	J2	100
11	S3	P1	J1	200
12	S3	P3	J1	200
13	S4	P5	J1	100
14	S4	P6	J3	300
15	S4	P6	J4	200
16	S5	P2	J4	100
17	S5	P3	J1	200
18	S5	P6	J2	200
19	S3	P6	J4	50

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';
```



	sno	sname	status	city
1	S1	精益	20	天津
2	S4	丰泰盛	20	天津
3	S3	东方红	30	北京
4	S5	为民	30	上海

	sno	pno	jno	qty
1	S1	P1	J1	200
2	S1	P1	J3	100
3	S1	P1	J4	700
4	S1	P2	J2	100
5	S3	P1	J1	200
6	S3	P3	J1	200
7	S4	P5	J1	100
8	S4	P6	J3	300
9	S4	P6	J4	200
10	S5	P2	J4	100
11	S5	P3	J1	200
12	S5	P6	J2	200
13	S3	P6	J4	50

将 (S2, J6, P4, 200) 插入供应情况关系

```
INSERT INTO SPJ
VALUES ('S2', 'J6', 'P4', 200);
```

数据库资源管理器

console spj x p s

14 行

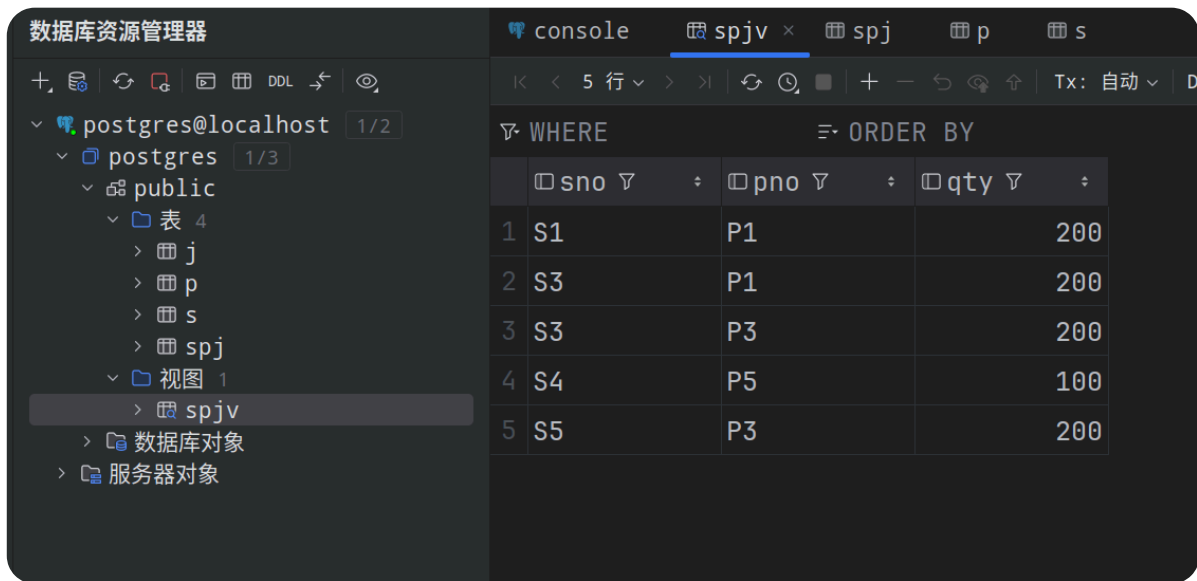
WHERE ORDER BY

	sno	pno	jno	qty
1	S1	P1	J1	200
2	S1	P1	J3	100
3	S1	P1	J4	700
4	S1	P2	J2	100
5	S3	P1	J1	200
6	S3	P3	J1	200
7	S4	P5	J1	100
8	S4	P6	J3	300
9	S4	P6	J4	200
10	S5	P2	J4	100
11	S5	P3	J1	200
12	S5	P6	J2	200
13	S3	P6	J4	50
14	S2	J6	P4	200

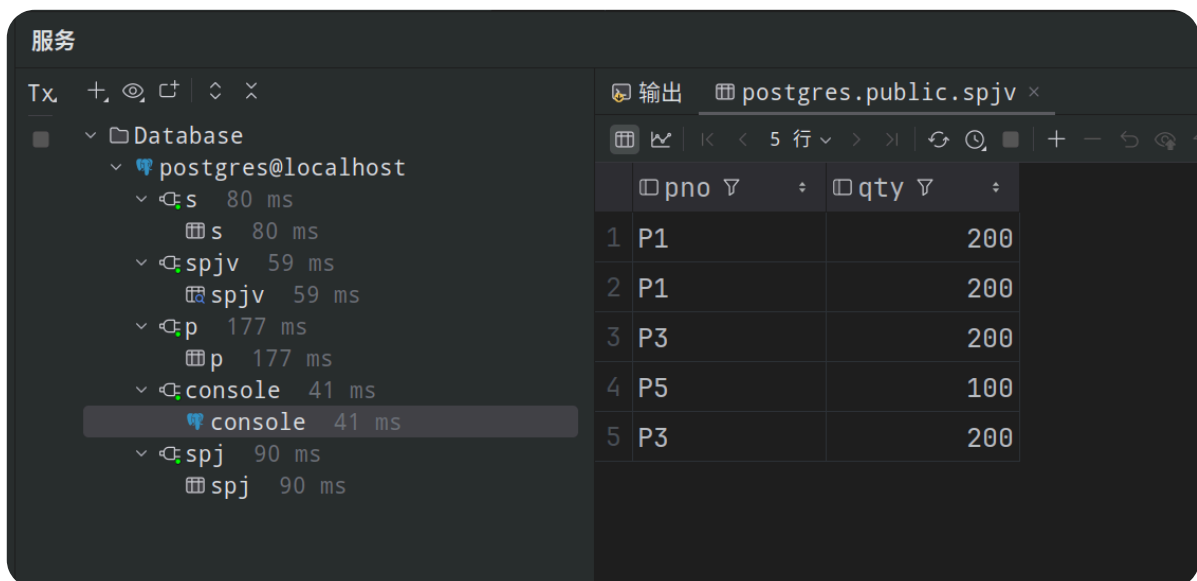
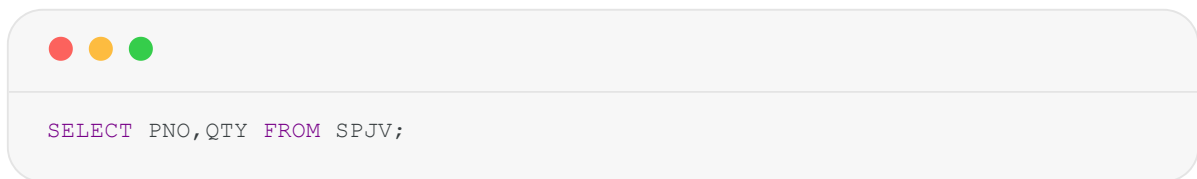
## 第九题

### 创建视图

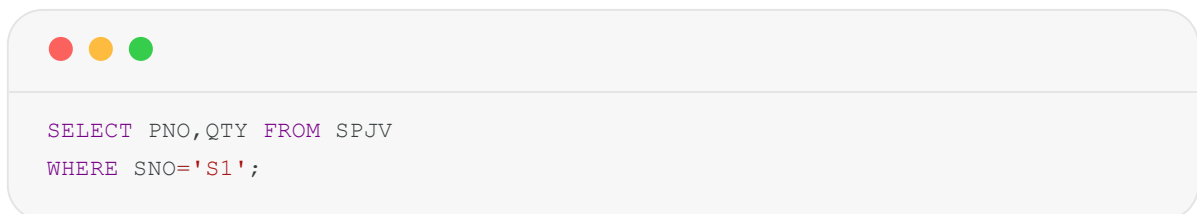
```
CREATE VIEW SPJV
AS
SELECT SNO,PNO,QTY
FROM SPJ
WHERE JNO=(
    SELECT JNO FROM J
    WHERE JNAME='三建'
);
```



找出三建工程项目使用的各种零件代码及其数量



找出供应商 S1 供应三建工程的情况



## 服务

Tx +, -, +, -, +, -

- Database
  - postgres@localhost
    - s 80 ms
      - s 80 ms
    - spjv 59 ms
      - spjv 59 ms
    - p 177 ms
      - p 177 ms
    - console 33 ms
      - console 33 ms
    - spj 90 ms
      - spj 90 ms

输出 postgres.public.spjv x

	pno	qty
1	P1	200