

数据库实验

实验一

1、PostgreSQL 安装 [Windows](#) [Linux](#)

2、创建数据库和表

使用SQL语句，完成创建一个数据库，创建关系。

1) 创建数据库scDB;

2) 按要求创建四个表:

Student(Sno,Sname,Ssex,Sage,Sdept)

Course(Cno,Cname,Cpno,Ccredits)

SC(Sno,Cno,Grade)

3) 为属性选择合适的域、合适的主码和外键约束;

3、运行脚本SPJ.sql

```
1 | \i your_Path
```

4、简单单表查询

在SPJ的基础上，完成教材第三章习题5。

实验二

掌握SELECT/SELECT DISTINCT/WHERE/AND/OR/ORDER BY/INSERT INTO/UPDATE/DELETE等操作

5、查询操作

1) 在零件表的视图找出weight < 20 的零件名字(PNAME)

```
1 | select pname from p where weight < 20;
2 |      pname
3 | -----
4 |      螺母
5 |      螺栓
6 |      螺丝刀
7 |      螺丝刀
8 | (4 rows)
```

2) 查询供应商表中城市为北京的供应商姓名(SNAME)

```

1 select sname from s where city = '北京';
2     sname
3 -----
4     盛锡
5     东方红
6 (2 rows)

```

3) 在零件表中查询平均重量在15以上的零件名字和零件代码 (PNO)

```

1 select pname, pno from p group by (pname, pno) having avg(weight) > 15;
2     pname    | pno
3 -----+-----
4     凸轮      | P5
5     螺栓      | P2
6     齿轮      | P6
7 (3 rows)

```

but no aggregation causing by pno(pno must be added into group by's args because:)

```

1 select pname, pno from p group by (pname) having avg(weight) > 15;
2 ERROR: column "p.pno" must appear in the GROUP BY clause or be used in an aggregate
   function
3 LINE 1: select pname, pno from p group by (pname) having avg(weight)...
4                   ^

```

so if we need to output pname and pno, based on the aggregation by pname, use `correlation subquery`:

```

1 select pname, pno from p A where (select avg(B.weight) from p B where B.pname =
   A.pname) > 15;
2     pname    | pno
3 -----+-----
4     螺栓      | P2
5     凸轮      | P5
6     齿轮      | P6
7 (3 rows)

```

let's test:

```

1  --update the tabel p
2  update p set weight = 30 where pno = 'P4';
3  select * from p;
4  pno |  pname  | color | weight
5  -----+-----+-----
6  P1  |  螺母   |  红   |    12
7  P2  |  螺栓   |  绿   |    17
8  P3  |  螺丝刀 |  蓝   |    14
9  P5  |  凸轮   |  蓝   |    40
10 P6  |  齿轮   |  红   |    30
11 P4  |  螺丝刀 |  红   |    30
12 (6 rows)

```

```

1  select pname, pno from p A where (select avg(B.weight) from p B where B.pname =
2  A.pname) > 15;
3  pname | pno
4  -----+-----
5  螺栓   | P2
6  螺丝刀 | P3
7  凸轮   | P5
8  齿轮   | P6
9  螺丝刀 | P4
10 (5 rows)
11 -- the output is right: 2 螺丝刀 should be printed;

```

```

1  select pname, pno from p group by (pname, pno) having avg(weight) >15;
2  pname | pno
3  -----+-----
4  凸轮   | P5
5  螺栓   | P2
6  螺丝刀 | P4
7  齿轮   | P6
8  (4 rows)
9  -- output is wrong for no aggregation is applied, only 1 螺丝刀 is printed;

```

4) 查询全体供应商的姓名 (SNAME) 和状态(STATUS)

```

1  select sname, status from s;
2  sname | status
3  -----+-----
4  精益   | 20
5  盛锡   | 10
6  东方红 | 30
7  丰泰盛 | 20
8  为民   | 30
9  (5 rows)

```

5) 查询所有weight在13到20岁 (含13和20) 的零件代码 (PNO)、零件名 (PNAME) 和颜色(COLOR)

```

1 select pno, pname, color from p where weight between 13 and 20;
2   pno |  pname   | color
3 -----+-----+-----
4   P2  |  螺栓    |  绿
5   P3  |  螺丝刀   |  蓝
6 (2 rows)

```

6) 查询所有“螺”开头的零件代码 (PNO) 和零件名 (PNAME)

```

1 select pno, pname from p where pname like '螺%';
2   pno |  pname
3 -----+-----
4   P1  |  螺母
5   P2  |  螺栓
6   P3  |  螺丝刀
7   P4  |  螺丝刀
8 (4 rows)

```

7) 查询所有零件的平均重量

```

1 select avg(weight) from p;
2           avg
3 -----
4  21.166666666666667
5 (1 row)

```

8) 查询同在“天津”的工程项目名 (JNAME)

```

1 select jname from j where city = '天津';
2   jname
3 -----
4   弹簧厂
5   造船厂
6 (2 rows)

```

9) 查询在“精益”供应商下的零件，且质量小于15的零件详细信息

```

1 select distinct * from spj, s, p where spj.pno = p.pno
2 and s.sno = spj.sno
3 and s.sname = '精益'
4 and p.weight<15;
5  sno | pno | jno | qty | sno |  sname  | status | city | pno |  pname  | color |
6  weight
6  -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
7  S1   | P1   | J1   | 200 | S1   | 精益    | 20      | 天津 | P1   | 螺母    | 红     |
8  12
8  S1   | P1   | J3   | 100 | S1   | 精益    | 20      | 天津 | P1   | 螺母    | 红     |
9  12
9  S1   | P1   | J4   | 700 | S1   | 精益    | 20      | 天津 | P1   | 螺母    | 红     |
10  12
10 (3 rows)

```

实验三

练习带连接的子查询操作

6、复杂子查询操作

1) 在零件表中找出weight排名前三的零件名字(PNAME)，按降序输出

```

1 select p.pname from p order by weight desc limit 3 offset 0;
2  pname
3  -----
4  凸轮
5  齿轮
6  螺栓
7  (3 rows)

```

2) 查询至少使用了供应商S1所供应的全部零件的城是(CITY)

```

1 select j.city
2 from j, spj A
3 where j.jno = A.jno
4 and not exists(
5     select *
6     from spj B
7     where B.sno = 'S1'
8     and not exists (
9         select * from spj
10        where A.pno = B.pno)
11 );
12
13 city
14 -----

```

15 (0 rows)

?? not solved:

天津是对应2个工程的，if这两个工程分别用了p1和p2，用这种方法是没法输出天津的

let's test:

```
1 update spj set pno='P2' where jno='J4' and sno='S1';
2 UPDATE 1
3 spj=# select * from spj;
4   sno | pno | jno | qty
5 -----+-----+-----+-----
6   S1  | P1  | J1  | 200
7   S1  | P1  | J3  | 100
8   S1  | P2  | J2  | 100
9   S2  | P3  | J1  | 400
10  S2  | P3  | J2  | 200
11  S2  | P3  | J4  | 500
12  S2  | P3  | J5  | 400
13  S2  | P5  | J1  | 400
14  S2  | P5  | J2  | 100
15  S3  | P1  | J1  | 200
16  S3  | P3  | J1  | 200
17  S4  | P5  | J1  | 100
18  S4  | P6  | J3  | 300
19  S4  | P6  | J4  | 200
20  S5  | P2  | J4  | 100
21  S5  | P3  | J1  | 200
22  S5  | P6  | J2  | 200
23  S5  | P6  | J4  | 500
24  S1  | P2  | J4  | 700
25 (19 rows)
26 select * from j;
27   jno |   jname   | city
28 -----+-----+-----
29   J1  |  三建    | 北京
30   J2  |  一汽     | 长春
31   J3  |  弹簧厂   | 天津
32   J4  |  造船厂   | 天津
33   J5  |  机车厂   | 唐山
34   J6  |  无线电厂 | 常州
35   J7  |  半导体厂 | 南京
36 (7 rows)
37 -- right answer: 天津(J3, J4) should be printed;
```

```
1 --right
2 create view spj_sub as (select pno, jno from spj where sno='S1');
3 CREATE VIEW
4 spj=# select * from spj_sub;
5   pno | jno
6 -----+-----
7   P1  | J1
```

```

8  P1 | J3
9  P2 | J2
10 P2 | J4
11 (4 rows)
12
13 select jc from j jc
14 where jc.city in (select distinct city from j)
15 and (select count(*) from (
16     select spj_sub.pno from j j1, spj_sub
17     where j1.jno in (select j2.jno from j j2 where j2.city = jc.city)
18     and spj_sub.jno=j1.jno
19 )
20 z)
21 =
22 (select count(*) from (select distinct pno from spj_sub) z2);
23
24 --change to:(add a distinct before spj.pno)
25 select jc from j jc
26 where jc.city in (select distinct city from j)
27 and (select count(*) from (
28     select distinct spj_sub.pno from j j1, spj_sub where j1.jno in (
29     select j2.jno from j j2 where j2.city = jc.city)
30     and spj_sub.jno=j1.jno)
31 z)
32 =
33 (select count(*) from (select distinct pno from spj_sub) z2);
34 jc
35 -----
36 (J3,"弹簧厂", "天津 ")
37 (J4,"造船厂", "天津 ")
38 (2 rows)
39
40 --just change the jno into a set: (select j3.jno from j j3 where j3.city=j1.city)
41
42 --wrong1:
43 select j.city
44 from j, spj A
45 where j.jno = A.jno
46 and not exists(
47     select *
48     from spj B
49     where B.sno = 'S1'
50     and not exists (
51         select * from spj
52         where A.pno = B.pno)
53 );
54 city
55 -----
56 (0 rows)
57 --wrong2:
58 select distinct j1 from j j1, spj A where not exists(
59     select B.pno from spj B where B.sno='S1' and not exists(
60         select * from spj C, j j2 where j2.jno in (

```

```

61         select j3.jno from j j3 where j3.city = j1.city)
62         and C.jno = j2.jno and B.pno=C.pno));
63

```

3) 查询出供应商代码 (SNO) 为S1的, 生产零件的全部颜色 (COLOR)

```

1  select p.color from p,spj where spj.sno = 'S1' and spj.pno = p.pno;
2  color
3  -----
4  红
5  红
6  红
7  绿
8  (4 rows)

```

4) 查询所有WEIGHT > 20的零件名字(PNAME),零件代码(PNO),供应商代码(SNO), 供应商姓名(SNAME)

```

1  select distinct spj.pno, spj.sno, s.sname from spj, s, p where p.weight > 20 and
   spj.pno = p.pno and spj.sno = s.sno;
2  pno | sno | sname
3  -----+-----
4  P5  | S4  | 丰泰盛
5  P6  | S5  | 为民
6  P5  | S2  | 盛锡
7  P6  | S4  | 丰泰盛
8  (4 rows)

```

5) 查询供应工程J1零件为红色的供应商号码(SNO)

```

1  select distinct sno from spj, p where p.color = '红' and spj.pno = p.pno and spj.jno =
   'J1';
2  sno
3  -----
4  S1
5  S3
6  (2 rows)

```

Wrong: if join s, will be:

```

1  select distinct s.sno from s, spj, p where p.color = '红' and spj.pno = p.pno and
   spj.jno = 'J1';
2  sno
3  -----
4  S1
5  S2
6  S3
7  S4
8  S5
9  (5 rows)

```



```

10  -- change to :
11  select distinct s.sno from s, spj, p where p.color = '红' and spj.pno = p.pno and
    spj.jno = 'J1' and s.sno = spj.sno;
12      sno
13  -----
14      S1
15      S3
16  (2 rows)

```

mistake: need to `s.sno =spj.sno`

实验四

练习带分组聚集的查询

7、带分组聚集的查询操作

1) 查询大于平均WEIGHT的零件，列出他们的供应商代码（SNO），零件代码（PNO），工程代码（JNO），供应数量（QTY）

Way1:

```

1  select pno, sno, jno, qty from spj where pno in (select pno from p where weight >
2      (select avg(weight) from p));
3
4  pno | sno | jno
5  ----+-----+----
6  P5  | S2  | J1
7  P5  | S2  | J2
8  P5  | S4  | J1
9  P6  | S4  | J3
10 P6  | S4  | J4
11 P6  | S5  | J2
12 P6  | S5  | J4
13
14 select spj.pno, spj.sno, spj.jno, spj.qty from spj, p where p.weight > (select
15     avg(weight) from spj, p where spj.pno = p.pno) and spj.pno = p.pno;
16
17 pno | sno | jno | qty
18 ----+-----+----+----
19 P5  | S2  | J1  | 400
20 P5  | S2  | J2  | 100
21 P5  | S4  | J1  | 100
22 P6  | S4  | J3  | 300
23 P6  | S4  | J4  | 200
24 P6  | S5  | J2  | 200
25 P6  | S5  | J4  | 500
26 (7 rows)

```

Way 2:

```

1  select pno into a from p where p.weight > (select avg(weight) from p);
2  select spj.pno, sno, jno, qty from spj, a where a.pno = spj.pno;

```

2) 查询小于平均供应数量QTY的零件，列出他们的零件代码（PNO），零件名（PNAME），颜色（COLOR）

```
1 select p.pno, pname, color,qty from p, spj where p.pno = spj.pno and spj.qty <
  (select avg(qty) from spj);
2  pno |  pname  | color | qty
3  ----+-----+-----+----
4  P1  |  螺母   |  红   | 200
5  P1  |  螺母   |  红   | 100
6  P2  |  螺栓   |  绿   | 100
7  P3  |  螺丝刀 |  蓝   | 200
8  P5  |  凸轮   |  蓝   | 100
9  P1  |  螺母   |  红   | 200
10 P3  |  螺丝刀 |  蓝   | 200
11 P5  |  凸轮   |  蓝   | 100
12 P6  |  齿轮   |  红   | 200
13 P2  |  螺栓   |  绿   | 100
14 P3  |  螺丝刀 |  蓝   | 200
15 P6  |  齿轮   |  红   | 200
16 (12 rows)
```

3) 查询供应数量QTY不在99-301之间的零件，列出他们的零件代码（PNO），零件名（PNAME），颜色（COLOR）

```
1 select p.pno, pname, color from spj, p where p.pno = spj.pno and spj.qty not between
  99 and 301 group by p.pno;
2  pno |  pname  | color
3  ----+-----+-----
4  P5  |  凸轮   |  蓝
5  P6  |  齿轮   |  红
6  P3  |  螺丝刀 |  蓝
7  P1  |  螺母   |  红
8  (4 rows)
```

4) 查询WEIGHT大于15，且供应数量QTY必须在250以上的零件，列出他们的零件代码（PNO），零件名（PNAME），供应数量（QTY）

```
1 select p.pno, p.pname, p.color from p, spj where spj.pno = p.pno and p.weight > 15
  and spj.qty >250 group by p.pno;
2  pno |  pname  | color
3  ----+-----+-----
4  P6  |  齿轮   |  红
5  P5  |  凸轮   |  蓝
6  (2 rows)
```

5) 查询工程项目代码（JNO）为“J1”的项目，列出所有使用的零件代码（PNO），零件名（PNAME），颜色（COLOR）

```

1  select p.pno, p.pname, spj.qty from p, spj where spj.jno = 'J1' and p.pno = spj.pno
   group by p.pno, spj.qty;
2  pno |  pname  | qty
3  -----+-----+-----
4  P1  |  螺母   | 200
5  P3  |  螺丝刀 | 200
6  P3  |  螺丝刀 | 400
7  P5  |  凸轮   | 100
8  P5  |  凸轮   | 400
9  (5 rows)
10
11 select p.pno, p.pname, spj.qty, sno from p, spj where spj.jno = 'J1' and p.pno =
   spj.pno group by p.pno, spj.qty, sno;
12  pno |  pname  | qty | sno
13  -----+-----+-----+-----
14  P1  |  螺母   | 200 | S1
15  P1  |  螺母   | 200 | S3
16  P3  |  螺丝刀 | 200 | S3
17  P3  |  螺丝刀 | 200 | S5
18  P3  |  螺丝刀 | 400 | S2
19  P5  |  凸轮   | 100 | S4
20  P5  |  凸轮   | 400 | S2

```

?? when groupby

6) 查询供应商代码 (SNO) , 零件代码 (PNO) , 重量 (WEIGHT) , 通过零件代码 (PNO) , 重量 (WEIGHT) 排序

```

1  select distinct spj.sno, spj.pno, p.weight from spj, p where spj.pno = p.pno order
   by spj.pno, p.weight;
2  sno | pno | weight
3  -----+-----+-----
4  S1  | P1  | 12
5  S3  | P1  | 12
6  S1  | P2  | 17
7  S5  | P2  | 17
8  S2  | P3  | 14
9  S3  | P3  | 14
10 S5  | P3  | 14
11 S2  | P5  | 40
12 S4  | P5  | 40
13 S4  | P6  | 30
14 S5  | P6  | 30
15 (11 rows)

```

Tips:

\password 设置密码。

q 退出。

h 查看SQL命令的解释，比如\h select

l 列出所有数据库。

c [database_name] 连接其他数据库。

d 列出当前数据库的所有表格。

d [table_name] 列出某一张表格的结构。

du 列出所有用户。

e 打开文本编辑器。

conninfo 列出当前数据库和连接的信息。

\? 查看psql命令列表