部门平均薪水的等级：

部门平均的薪水等级：

那些人是经理

平均薪水最高的部门编号与名称

--求平均薪水的等级最低的部门的部门名称 *

比普通员工的最高薪水还要高的经理人名称：*

SQL面试题

有三个表S,C,SC

S(SNO, SNAME)(学号, 姓名)

C(CNO, CNAME, CTEACHER)(课号, 课名, 教师)

SC(SNO, CNO, SCGRADE)(学号, 课号, 成绩)

1.找出所有没选过"黎明"老师的所有学生的名字

2.列出2门以上(包括2门)不及格学生姓名及平均成绩

3.既学过1号课程又学过2号课程的学生的姓名

==========

部门平均薪水的等级：

```
SELECT s.grade,e.deptno,e.avg_sal from
(SELECT deptno,avg(sal) as avg_sal  from emp group by deptno) e
LEFT JOIN salgrade s
on e.avg_sal between s.losal and s.hisal
```

部门平均的薪水等级：

```
select avg(grade), deptno from
 (SELECT e.sal,s.grade grade,e.deptno deptno
from emp e
 left join salgrade s
on e.sal BETWEEN s.losal and s.hisal
) t
GROUP BY t.deptno
```

那些人是经理

```
解一：select ename from emp where empno in(
      select DISTINCT mgr from emp where mgr is not null
)

解二：SELECT DISTINCT e2.ename from emp e1,emp e2 where e1.mgr = e2.empno
```

不用组函数求最高薪水：*
解一：SELECT sal from emp order by sal DESC limit 1
解二（下面有详解）：select distinct sal from emp where sal not in
(select distinct e1.sal from emp e1 join emp e2 on (e1.sal < e2.sal));

SELECT * from emp JOIN (SELECT sal from emp order by sal DESC limit 1 ) t on t.sal = emp.sal

平均薪水最高的部门编号与名称

解一： SELECT t1.deptno,dname from
(select avg(sal) as avg_sal,deptno from emp GROUP BY deptno) t1
LEFT JOIN dept on dept.deptno = t1.deptno
where t1.avg_sal =
(
select MAX(t2.avg_sal) as max_sal from (
select avg(sal) as avg_sal,deptno from emp GROUP BY deptno
) t2
)

解二：
select dname from dept where deptno =
( select deptno from
(select avg(sal) avg_sal, deptno from emp group by deptno) t1
where avg_sal =
(select max(avg_sal) from
(select avg(sal) avg_sal, deptno from emp group by deptno) t2
)
);

解三：视图
create view v1 as select avg(sal) avg_sal, deptno from emp group by deptno
select dname from dept where deptno =
( select deptno from
v1
where avg_sal =
(select max(avg_sal) from v1)
);

解四：-->改进版本（MySQL5.5不支持）
select deptno, avg_sal from
(select avg(sal) avg_sal, deptno from emp group by deptno)
where avg_sal =
(select max(avg(sal)) from emp group by deptno);
组函数嵌套

组函数可以互相嵌套，但是最多只能嵌套两层

第一个组函数可能会产生多行，第二个组函数产生的一定是一行
只有一个值就无法再次嵌套组函数了

## MySQL的视图不支持select语句嵌套

--求平均薪水的等级最低的部门的部门名称 *

解一：SELECT dept.dname from (
SELECT deptno from
(
SELECT s.grade t1_grade,t.deptno deptno from (select avg(sal)as avg_sal,deptno from emp group by deptno) t
LEFT JOIN
salgrade s on t.avg_sal BETWEEN s.losal and s.hisal
) t1
JOIN (
SELECT s.grade g_grade from (select avg(sal)as avg_sal,deptno from emp group by deptno) t
LEFT JOIN salgrade s on t.avg_sal BETWEEN s.losal and s.hisal ORDER BY s.grade asc LIMIT 1
) g
on t1.t1_grade = g.g_grade

) t2
LEFT JOIN dept on t2.deptno = dept.deptno

解二：
```
select dname, t1.deptno, grade, avg_sal from
(
select deptno, grade, avg_sal from
(select deptno, avg(sal) avg_sal from emp group by deptno) t
join salgrade s on (t.avg_sal between s.losal and s.hisal)
) t1
join dept on (t1.deptno = dept.deptno)
where t1.grade =
(
select min(grade) from
(
select deptno, grade, avg_sal from
(select deptno, avg(sal) avg_sal from emp group by deptno) t
join salgrade s on (t.avg_sal between s.losal and s.hisal)
) t2
```

```
        )
```
解三:
```
create view v$dept_avg_sal_info as
        (
        select deptno, grade, avg_sal from
            (select deptno, avg(sal) avg_sal from emp group by deptno) t
        join salgrade s on (t.avg_sal between s.losal and s.hisal)
        )
```
---
改进后的代码
```
        select dname, t1.deptno, grade, avg_sal from
            v$dept_avg_sal_info t1
        join dept on (t1.deptno = dept.deptno)
        where t1.grade =
        (
            select min(grade) from v$dept_avg_sal_info
        )
```

比普通员工的最高薪水还要高的经理人名称：*

```
SELECT e.ename,e.sal from emp e
where
empno in (SELECT DISTINCT mgr from emp where mgr is not null)
and sal > ( select max(sal) from emp where empno not in (select DISTINCT mgr
from emp where mgr is not null) )
```

==========

SQL面试题

有三个表S,C,SC

S(SNO, SNAME)(学号, 姓名)

C(CNO, CNAME, CTEACHER)(课号, 课名, 教师)

SC(SNO, CNO, SCGRADE)(学号, 课号, 成绩)

1.找出所有没选过"黎明"老师的所有学生的名字

2.列出2门以上(包括2门)不及格学生姓名及平均成绩

3.既学过1号课程又学过2号课程的学生的姓名

```
1.  select SNAME from S where SNO not in (
        select distanct SC.SNO from SC LEFT JOIN C on SC.CNO=C.CNO where
C.CTEACHER='黎明'
)
```

解二

```
SELECT
	sno,
	sname
FROM
	s
WHERE
	sno NOT IN (
	SELECT DISTINCT
		ss.sno
	FROM
		s ss
		JOIN sc ON ( ss.sno = sc.sno )
		JOIN c ON ( c.cno = sc.cno )
	WHERE
	c.cteacher = 'liming'
	)
```

```
2.select t.avg_scgrage,S.SNAME from(
	select avg(SCGRADE) as avg_scgrage,SNO from SC where SNO in (
		select SNO
			from (
				select count(CNO) as count_sno,SNO from SC
				where SCGRADE < 60 group by SNO
			)
			where  count_sno >=2
	) group by SNO
)t LEFT JOIN S on t.SNO = S.SNO
```

解二:

```
select sname, avg_grade from s join
 (
 select sno, avg(scgrade) avg_grade from sc
          where sno in
          (SELECT sno FROM sc
        WHERE scgrade < 60 GROUP BY sno HAVING count(*) >= 2 )
               group by sno
 ) t

 on (s.sno = t.sno)
```

```
3. 错误select S.SNAME from (
	SELECT  DISTINCT SC.SNO from SC
	left join C on C.CNO=SC.CNO
	where C.CNAME='1号课程' and C.CNAME='2号课程'
)t left join S on S.SNO=t.SNO
```

```
SELECT
	sname
FROM
	s
WHERE
	sno IN
( SELECT sno FROM sc WHERE cno = 1 AND sno IN
	( SELECT sno FROM sc WHERE cno = 2 ) );
```

==========

部门平均薪水的等级：

把部门的平均薪水当成一张表，与另外一张表做连接

```
	select deptno, avg_sal, grade from
	(select deptno, avg(sal) avg_sal from emp group by deptno) t
	join salgrade s on (t.avg_sal between s.losal and s.hisal);
```

部门平均的薪水等级：

先求出每个人的薪水等级，然后再平均

-->求每个人的薪水等级

```
	select deptno, ename, grade from emp join salgrade s on (emp.sal between
s.losal and s.hisal);
```

求出每个人的部门编号，名字，薪水等级

-->求部门的平均薪水等级

```
	select deptno, avg(grade) from
	(select deptno, ename, grade from emp join salgrade s on (emp.sal
between s.losal and s.hisal))
	group by deptno;
```

那些人是经理

```
select ename from emp where empno in(select mgr from emp);
```

编号出现在mgr的都是经理，要根据编号求出这个人的名字

```
select ename from emp where empno in(select distinct mgr from emp);
```

//更有效的方式

## 不用组函数求最高薪水：

--->这是一道面试题

不准用组函数，求薪水的最高值

-->解决方法：使用自连接，用非等值

两张表相同，左边的表的薪水值小于右边的表，这时，左边表会有多条记录连接不上(薪水的最大值)，

薪水最大的在右边找不到比他更大的值。连接不上的就是最大值

```
-->select e1.sal from emp e1 join emp e2 on (e1.sal < e2.sal);
```

这里我们会得到多条记录，因为这两张表的每一个值都会进行比较，凡是小于的全部做连接

-->使用distinct去掉重复的

```
select distinct e1.sal from emp e1 join emp e2 on (e1.sal < e2.sal);
```

-->根据改进的语句我们可以得到了所有小于的数值，剩下的数值就是最大的薪资，我们怎么取这个剩下的数值？

```
select distinct sal from emp where sal not in
(select distinct e1.sal from emp e1 join emp e2 on (e1.sal < e2.sal));
```

## 平均薪水最高的部门编号与名称

首先，求每个部门的平均薪水

```
1.select avg(sal), deptno from emp gruop by deptno;      //根据部门编号分组查出平均薪水和部门编号
```

2.求平均薪水最高的值

```
select max(avg_sal) from
(select avg(sal) avg_sal, deptno from emp gruop by deptno);
```

3.求平均薪水最高的值所在的部门编号

```
select deptno, avg_sal from
```

```
        (select avg(sal) avg_sal, deptno from emp gruop by deptno)
        where avg_sal =
        (select max(avg_sal) from (select avg(sal) avg_sal, deptno from emp
gruop by deptno));
        //把where之前的语句看成是一张表，把where之后判断的语句看为一个值，
                这样就可以查出平均薪水最高的部门编号。
```

---

查平均薪水最高的部门的名称

我们已经查询出了部门的编号，我们可以根据这个编号查询出部门的名称

```
    select dname from dept where deptno =
        (       select deptno from
                (select avg(sal) avg_sal, deptno from emp group by deptno) t1
                where avg_sal =
                (select max(avg_sal) from
                        (select avg(sal) avg_sal, deptno from emp group by
deptno) t2
                )
        );
```

---

求平均薪水最高的部门的部门编号

-->改进版本（MySQL5.5不支持）

```
        select deptno, avg_sal from
        (select avg(sal) avg_sal, deptno from emp group by deptno)
        where avg_sal =
        (select max(avg(sal)) from emp group by deptno);
```

组函数嵌套

组函数可以互相嵌套，但是最多只能嵌套两层

第一个组函数可能会产生多行，第二个组函数产生的一定是一行

只有一个值就无法再次嵌套组函数了

--求平均薪水的等级最低的部门的部门名称

1，求出平均薪水

```
select avg(sal) from emp group by deptno;
```

2.求平均薪水的等级

```
select deptno, grade, avg_sal from
        (select avg(sal) avg_sal from emp group by deptno) t
join salgrade s on (t.avg_sal between s.losal and s.hisal);
```

3.求最低的平均薪水等级

```
select min(grade) from
(
        select deptno, grade, avg_sal from
        (select avg(sal) avg_sal from emp group by deptno) t
        join salgrade s on (t.avg_sal between s.losal and
s.hisal)
);
```

4.求最低的等级对应的部门编号

```
select dname, t1.deptno, grade, avg_sal from
        (
        select deptno, grade, avg_sal from
                (select deptno, avg(sal) avg_sal from emp group
by deptno) t
        join salgrade s on (t.avg_sal between s.losal and
s.hisal)
        ) t1
join dept on (t1.deptno = dept.deptno)
where t1.grade =
(
        select min(grade) from
        (
                select deptno, grade, avg_sal from
                        (select deptno, avg(sal) avg_sal from
emp group by deptno) t
                join salgrade s on (t.avg_sal between s.losal
and s.hisal)
```

```
                ) t2
        )
```

对　　求平均薪水的等级最低的部门的部门名称　　进行优化

简化的办法，创建一个视图 ，视图就是一张虚表，一个子查询

MySQL的视图不支持select语句嵌套

create view创建视图　通常视图的名字以v$开头

```
create view v$dept_avg_sal_info as
        (
        select deptno, grade, avg_sal from
                (select deptno, avg(sal) avg_sal from emp group by deptno) t
        join salgrade s on (t.avg_sal between s.losal and s.hisal)
        )
```

---

改进后的代码

```
        select dname, t1.deptno, grade, avg_sal from
                v$dept_avg_sal_info t1
        join dept on (t1.deptno = dept.deptno)
        where t1.grade =
        (
                select min(grade) from v$dept_avg_sal_info
        )
```

首先建立一张虚表，在虚表的基础之上，再取数据，这是视图的作用

比普通员工的最高薪水还要高的经理人名称：

思路：

1.求出普通员工的最高薪水

```
select max(sal) from emp
where empno not in (select distinct mgr from emp where mgr is not null);
```

因为emp表中KING是没有经理的，所以要把他排除一下：is not null

2.查找sal>上面语句的结果

不仅sal要大，而且这个结果的本身必须是经理人

```
select ename from emp
where empno in (select distinct mgr from emp where mgr is not null)
and sal>
(
        select max(sal) from emp
        where empno not in (select distinct mgr from emp where mgr is
not null)
);
```

SQL面试题

有三个表S,C,SC

S(SNO, SNAME)(学号, 姓名)

C(CNO, CNAME, CTEACHER)(课号, 课名, 教师)

SC(SNO, CNO, SCGRADE)(学号, 课号, 成绩)

建表：

```
create table S
(
SNO int primary key,
SNAME VARCHAR(10)
)

create table C
(
CNO int primary key,
CNAME varchar(10),
CTEACHER varchar(10)
)

create table SC
(
SNO int,
```

```
CNO int,
SCGRADE int
)

insert into s values (1, 'zhangsan');
insert into s values (2, 'lisi');
insert into s values (3, 'wangwu');

insert into c values (1, '数学', 'liming');
insert into c values (2, '语文', 'liming');
insert into c values (3, '历史', 'xueyou');
insert into c values (4, '物理', 'guorong');
insert into c values (5, '化学', 'liming');

insert into sc values (1, 1, 59);
insert into sc values (1, 2, 70);
insert into sc values (2, 1, 30);
insert into sc values (1, 3, 16);
insert into sc values (2, 3, 61);
insert into sc values (3, 1, 17);
insert into sc values (3, 2, 100);
insert into sc values (3, 5, 25);
insert into sc values (1, 4, 99);



#最简单的多对多关系三表连接
select sname, cname, cteacher, scgrade
from s join sc
on (s.sno = sc.sno)
join c
on (sc.cno = c.cno)
```

1. 找出所有没选过"黎明"老师的所有学生的名字

```
SELECT
	sno,
	sname
FROM
	s
WHERE
	sno NOT IN (
	SELECT DISTINCT
		ss.sno
	FROM
		s ss
		JOIN sc ON ( ss.sno = sc.sno )
```

```
        JOIN c ON ( c.cno = sc.cno )
    WHERE
    c.cteacher = 'liming'
    )
```

//联表查询，学生在sc中的位置，老师在sc中的位置，以及老师不是黎明的记录，然后，根据老师不是黎明的学生编号

查找到学生的姓名

2.列出2门以上(包括2门)不及格学生姓名及平均成绩

1.找出不及格的学生的学生编号 select sno from sc where scgrade < 60;

2.select sno, count(*) from sc where scgrade <60 group by sno having count(*) >= 2;

查询不及格在两门及两门以上的学生的编号

```
    SELECT
    sname
from
    s
WHERE
    sno IN
 ( SELECT sno FROM sc WHERE scgrade < 60
    GROUP BY sno HAVING count(*) >= 2 );

SELECT
    sname,
    sc_grade
FROM
    s
    JOIN ( SELECT sno, avg( scgrade ) sc_grade FROM sc
   WHERE scgrade < 60 GROUP BY sno HAVING count(*) >= 2 ) t
    ON (
    s.sno = t.sno)



select sname, avg_grade from s join
 (
 select sno, avg(scgrade) avg_grade from sc
        where sno in
        (SELECT sno FROM sc
      WHERE scgrade < 60 GROUP BY sno HAVING count(*) >= 2 )
            group by sno
 ) t
```

on (s.sno = t.sno)

3.既学过1号课程又学过2号课程的学生的姓名

```
SELECT
	sname
FROM
	s
WHERE
	sno IN
( SELECT sno FROM sc WHERE cno = 1 AND sno IN
	( SELECT sno FROM sc WHERE cno = 2 ) );
```

```
create table S
(
SNO int primary key,
SNAME VARCHAR(10)
)

create table C
(
CNO int primary key,
CNAME varchar(10),
CTEACHER varchar(10)
)

create table SC
(
SNO int,
CNO int,
SCGRADE int
)

insert into s values (1, 'zhangsan');
```

```
insert into s values (2, 'lisi');
insert into s values (3, 'wangwu');

insert into c values (1, '数学', 'liming');
insert into c values (2, '语文', 'liming');
insert into c values (3, '历史', 'xueyou');
insert into c values (4, '物理', 'guorong');
insert into c values (5, '化学', 'liming');

insert into sc values (1, 1, 59);
insert into sc values (1, 2, 70);
insert into sc values (2, 1, 30);
insert into sc values (1, 3, 16);
insert into sc values (2, 3, 61);
insert into sc values (3, 1, 17);
insert into sc values (3, 2, 100);
insert into sc values (3, 5, 25);
insert into sc values (1, 4, 99);
```

| SNO | CNO | SCGRADE |
|---|---|---|
| 1 | 1 | 59 |
| 1 | 2 | 70 |
| 2 | 1 | 30 |
| 1 | 3 | 16 |
| 2 | 3 | 61 |
| 3 | 1 | 17 |
| 3 | 2 | 100 |
| 3 | 5 | 25 |
| 1 | 4 | 99 |

| SNO | SNAME |
|---|---|
| 1 | zhangsan |
| 2 | lisi |
| 3 | wangwu |

| CNO | CNAME | CTEACHER |
| --- | --- | --- |
| 1 | 数学 | liming |
| 2 | 语文 | liming |
| 3 | 历史 | xueyou |
| 4 | 物理 | guorong |
| 5 | 化学 | liming |