1、什么是子查询

select语句当中嵌套select语句，被嵌套的select语句就是子查询

子查询可出现的位置：

```
select
    ..(select)
from
    ..(select)
where
    ..(select)
```

2、where子句中使用子查询

```
--找出高于平均薪资的员工信息
由于分组函数不能直接做条件，所以第一种方法是先查询出员工的平均薪资，然后再用查询出来的具体薪资数据做为过滤条件
select avg(sal) from emp;
select * from emp where sal>上面的平均薪资;

--使用子查询，即将上面两个查询语句合并
select * from emp where sal>(select avg(sal) from emp);
+-------+-------+-----------+------+------------+---------+------+--------+
| EMPNO | ENAME | JOB       | MGR  | HIREDATE   | SAL     | COMM | DEPTNO |
+-------+-------+-----------+------+------------+---------+------+--------+
|  7566 | JONES | MANAGER   | 7839 | 1981-04-02 | 2975.00 | NULL |     20 |
|  7698 | BLAKE | MANAGER   | 7839 | 1981-05-01 | 2850.00 | NULL |     30 |
|  7782 | CLARK | MANAGER   | 7839 | 1981-06-09 | 2450.00 | NULL |     10 |
|  7788 | SCOTT | ANALYST   | 7566 | 1987-04-19 | 3000.00 | NULL |     20 |
|  7839 | KING  | PRESIDENT | NULL | 1981-11-17 | 5000.00 | NULL |     10 |
|  7902 | FORD  | ANALYST   | 7566 | 1981-12-03 | 3000.00 | NULL |     20 |
+-------+-------+-----------+------+------------+---------+------+--------+
6 rows in set (0.00 sec)
```

3、from后面嵌套子查询

```
--找出每个部门平均薪资的薪资等级
--第一步，找出每个部门的平均薪资
mysql> select deptno,avg(sal) as avgsal from emp group by deptno;
+--------+-------------+
| deptno | avgsal      |
+--------+-------------+
|     10 | 2916.666667 |
|     20 | 2175.000000 |
|     30 | 1566.666667 |
+--------+-------------+
3 rows in set (0.00 sec)
--第二步，将上述结果当作临时表t，让t表和薪资等级表salgrade s连接
select
    t.*,s.grade
from
    (select deptno,avg(sal) as avgsal from emp group by deptno) t
```

```
join
    salgrade s
on
    t.avgsal between s.losal and s.hisal;
+--------+-------------+-------+
| deptno | avgsal      | grade |
+--------+-------------+-------+
|     10 | 2916.666667 |     4 |
|     20 | 2175.000000 |     4 |
|     30 | 1566.666667 |     3 |
+--------+-------------+-------+
3 rows in set (0.00 sec)
```

```
--找出每个部门的平均薪资等级
--第一步，先找出每个部门员工的薪资等级
select
    e.deptno,e.ename,e.sal,s.grade
from
    emp e
join
    salgrade s
on
    e.sal between s.losal and s.hisal;
+--------+--------+---------+-------+
| deptno | ename  | sal     | grade |
+--------+--------+---------+-------+
|     20 | SMITH  |  800.00 |     1 |
|     30 | ALLEN  | 1600.00 |     3 |
|     30 | WARD   | 1250.00 |     2 |
|     20 | JONES  | 2975.00 |     4 |
|     30 | MARTIN | 1250.00 |     2 |
|     30 | BLAKE  | 2850.00 |     4 |
|     10 | CLARK  | 2450.00 |     4 |
|     20 | SCOTT  | 3000.00 |     4 |
|     10 | KING   | 5000.00 |     5 |
|     30 | TURNER | 1500.00 |     3 |
|     20 | ADAMS  | 1100.00 |     1 |
|     30 | JAMES  |  950.00 |     1 |
|     20 | FORD   | 3000.00 |     4 |
|     10 | MILLER | 1300.00 |     2 |
+--------+--------+---------+-------+
14 rows in set (0.00 sec)
--第二步，将上面的查询结果作为临时表t，以表t中的deptno字段进行分组，查询表t中的grade字段的平均值
select
    t.deptno,avg(t.grade) as avggrade
from
(select
    e.deptno,e.ename,e.sal,s.grade
from
    emp e
join
    salgrade s
on
    e.sal between s.losal and s.hisal) t
group by
```

```
    t.deptno;
+--------+----------+
| deptno | avggrade |
+--------+----------+
|     10 |   3.6667 |
|     20 |   2.8000 |
|     30 |   2.5000 |
+--------+----------+
3 rows in set (0.00 sec)
--上面这种虽然可以查询到我们想要的结果，但是没有必要这样嵌套子查询，这样做会降低效率
--高效的做法是在第一步中看到这样的查询结果，考虑结果表中的字段能否再用里面的字段在其sql语句添加东西对结果表再过滤，这里就是在第一步的sql语句中添加按e.deptno字段进行分组，然后查询e.name和s.grade，如下
select
    e.deptno,avg(s.grade)
from
    emp e
join
    salgrade s
on
    e.sal between s.losal and s.hisal
group by
    e.deptno;
+--------+--------------+
| deptno | avg(s.grade) |
+--------+--------------+
|     10 |       3.6667 |
|     20 |       2.8000 |
|     30 |       2.5000 |
+--------+--------------+
3 rows in set (0.00 sec)
```

4、select后面嵌套子查询

```
--查询员工的名字和其部门的名称
--这里不用连接查询语句，用子查询语句实现(不常用)
select
    e.ename,(select d.dname from dept d where e.deptno=d.deptno) as dname
from
    emp e;
+--------+------------+
| ename  | dname      |
+--------+------------+
| SMITH  | RESEARCH   |
| ALLEN  | SALES      |
| WARD   | SALES      |
| JONES  | RESEARCH   |
| MARTIN | SALES      |
| BLAKE  | SALES      |
| CLARK  | ACCOUNTING |
| SCOTT  | RESEARCH   |
| KING   | ACCOUNTING |
| TURNER | SALES      |
| ADAMS  | RESEARCH   |
| JAMES  | SALES      |
| FORD   | RESEARCH   |
| MILLER | ACCOUNTING |
```

```
+--------+------------+
14 rows in set (0.00 sec)
```