

子查询

什么是子查询？简单理解，子查询就是嵌套在查询中的查询。

为什么要使用子查询？很多时候，我们对数据的查询处理不是一个查询语句就可以完成的；不能完成的原因，可能是查询条件比较复杂，也可能是受困于SQL自身语法的限制；这时候，子查询作为查询条件的重要组成部分，用在WHERE子句以及HAVING子句中，可以帮助我们快速灵活的完成查询操作.....

单行子查询

单行子查询是指返回一行数据的子查询语句。

当where子句引用单行子查询时，可以使用 比较运算符（=、>、<等）。

```
-- 查询工资大于平均工资的员工信息
select avg(sal) from emp; --单行单列
select * from emp where sal>(select avg(sal) from emp);

-- 查询部门平均工资大于整个公司平均工资的各部门平均工资
select deptno,avg(sal) from emp group by deptno having avg(sal)>(select
avg(sal) from emp)

-- 子查询可用于select中
select ename,sal,(select avg(sal) from emp) from emp

-- 子查询可用于from中，其中起的别名可用于外层语句
select pjf from (select avg(sal) pjf from emp)
```

多行子查询

多行子查询是指返回多行数据的子查询语句。当where子句引用多行子查询时，必须使用 多行比较符（in、any、all、exist等）

```
-- 查询部门编号>10的员工信息
select deptno from dept where deptno>10; --多行单列
select * from emp where deptno in (select deptno from dept where deptno>10)

-- 多条件查询，大多数可以使用开窗代替
-- 例子：查看每个部门工资最高的员工姓名和他的工资
-- 开窗的做法
select * from (
    select ename,sal,deptno,rank()over(partition by deptno order by sal desc)
```

```

r from emp
) where r=1;

-- 多条件多行操作
select ename,sal,deptno from emp where (deptno,sal) in (
    select deptno,max(sal) from emp group by deptno
)

-- 多行多列 单行多列 一个表
select ename,sal,rank()over(order by sal desc) from emp
select * from (
    select ename,sal,rank()over(order by sal desc) paiming from emp
) where paiming<=3

```

联合查询

SQL联合查询主要是涉及两个表或者多个表（自身表多次）的查询。主要分为**连接查询**和**集合查询**两大类，其中：

连接查询分为：

- 内连接（inner Join 或 Join）
- 左外连接（left outer Join 或 left Join）
- 右外连接（right outer Join 或 right Join）
- 全外连接（full outer Join 或 full Join）

集合查询分为：

- minus（差集）
- intersect（交集）
- union、union all（并集）

连接查询

交叉连接（笛卡尔连接） cross join

把笛卡尔积所有的结果给显示出来了

```

A cross join B
select * from emp cross join dept;
select * from emp,dept;
-- 应用举例，两个球队之间交叉比赛，循环赛

```

自然连接 natural join

一种特殊的内连接 没有链接条件on，如果两个表里面有一列相等， 系统默认把这一列作为链接条件，并且把这两列合成一列放到表的最前面, 适用于知道表里面有这样一列，并且要用这一列的时候

```
select * from emp natural join dept
```

等于内连接结果集，一般不用

using

跟自然连接作用相同，不同在于当有多个列相同时，可以指定用哪一列来做链接

```
select * from emp join dept using(deptno)
```

内连接 inner join

inner可不写，只写join默认相当于inner join

```
-- 等值连接：只返回两个表中连接字段相等的行。
-- 连接emp表和dept表，匹配deptno
select * from emp,dept where emp.deptno=dept.deptno
select * from emp inner join dept on emp.deptno=dept.deptno

-- 不等值连接：过滤条件的符号不是等号
-- 查询员工的工资级别：连接emp表和salgrade表，根据emp表的sal和salgrade表的losal和
hisal匹配其他内容
select ename,sal,grade from emp inner join salgrade on emp.sal between losal
and hisal
```

自连接

```
-- 查询出每个员工的上级领导(查询内容：员工编号、员工姓名、领导编号、领导姓名)
select yg.empno,yg.ename,ld.empno,ld.ename from emp yg inner join emp ld on
yg.mgr=ld.empno
```

多表连接

```
select yg.empno,yg.ename,ld.empno,ld.ename,dname from emp yg inner join emp ld
on yg.mgr=ld.empno inner join dept on ld.deptno=dept.deptno
```

外连接 outer join

left join（左连接，左外连接）：返回包括左表中的所有记录和右表中连接字段相等的记录，如果没有匹配上，以`null`值代表右边表的列。

-- 左表为主表，全显示，右表为子表，显示满足on匹配条件的数据

```
select * from emp left outer join dept on emp.deptno=dept.deptno
select * from dept left outer join emp on emp.deptno=dept.deptno
```

right join（右连接，右外连接）：返回包括右表中的所有记录和左表中连接字段相等的记录，如果没有匹配，以`null`值代表左边表的列

-- 右表为主表，全显示，左表为子表，显示满足on匹配条件的数据

```
select * from dept right outer join emp on emp.deptno=dept.deptno
select * from emp right outer join dept on emp.deptno=dept.deptno
```

full join（全外连接）：返回左右表中所有的记录和左右表中连接字段相等的记录，没有匹配上，以`null`值代表左右边表的列。

-- 左右表都显示，满足条件的显示，不满足条件的留空

```
select * from emp full outer join dept on emp.deptno=dept.deptno
```

-- 筛选出来emp表中不满足条件的数据

```
select * from emp e left join dept d on e.deptno=d.deptno where d.deptno is
null
```

-- 筛选出dept表中不满足条件数据

```
select * from emp e right join dept d on e.deptno=d.deptno where e.deptno is
null
```

-- 筛选出emp和dept表中不满足条件的数据

```
select * from emp e full join dept d on e.deptno=d.deptno where d.deptno is
null or e.deptno is null
```

集合查询

union 并集（去重且排序，效率低）

```
select deptno from emp
union
select deptno from dept
```

union all 并集（不去重不排序，效率高）

```
select deptno from emp
union all
select deptno from dept
```

--union all 并集全部显示（列名可以不一样，类型一样就可以了）

```
select deptno from emp
```

```
union all
select sal from emp
```

intersect 交集

```
select deptno from emp
intersect
select deptno from dept
```

minus 差集

union/union all/intersect 没有上下关系，minus有（永远是上减下）

```
select deptno from dept
minus
select deptno from emp
```

with 表名 as(子查询)

好用，多用

exists

用于检查子查询是否返回任何行

它是一个非常重要的高级查询工具，特别适合处理存在性检查的复杂查询场景

```
select 列 from 表 where exists(子查询);

-- 查询有下属的领导
select * from emp ld where exists(
    select 1 from emp yg where yg.mgr = ld.empno
);

-- 查询emp表中有2个以上直接下属的领导人员信息
select * from emp ld where exists(
    select 1 from emp where mgr=ld.empno group by mgr having count(*)=2
)
```

1. 执行过程：

- 对外部查询的每一行，执行一次子查询

- 如果子查询返回至少一行，返回 TRUE
- 如果子查询返回空结果，返回 FALSE

2. 关键特点：

- 不关心子查询返回的具体数据，只关心是否有数据返回
- 子查询通常使用 `select 1` 或 `select *`，因为返回的内容不重要
- 当找到第一个匹配项时就会停止搜索，因此性能通常较好

在 `in` 和 `exists` 中 子查询结果集大优先考虑用 `exists`

在 `not in` 和 `not exists` 中无脑选 `not exists`

作业练习

-- 上机练习10

--1. 查询工资比20号部门工资都大的员工信息

```
select * from emp where sal > (select max(sal) from emp where deptno=20)
```

--2. 查询工资比20号部门工资任意一个大的员工信息

```
select * from emp where sal > (select min(sal) from emp where deptno=20)
```

--3. 求最高工资员工的姓名

```
select ename from emp where sal = (select max(sal) from emp)
```

--4. 查询员工工资和工作都和20号部门同时一样的员工信息

```
select * from emp where (sal, job) in (select sal, job from emp where deptno=20)
```

--5. 查询部门平均工资大于整个公司平均工资的部门

```
select deptno from emp group by deptno having avg(sal) > (select avg(sal) from emp)
```

--6. 求工资前五名的员工信息

```
select * from (
    select emp.*, dense_rank() over (order by sal desc) rnk from emp
) where rnk <= 5
```

--7. 查询emp表中所有数据的第2条到第5条

```
select * from (
    select emp.*, row_number() over (order by rownum desc) rnk from emp
) where rnk between 2 and 5
```

--8. 求每个部门工资最高的员工信息

```
select * from emp where (deptno, sal) in (
    select deptno, max(sal) from emp group by deptno
)
```

-- 上机练习11

--1. 查询所有员工的姓名和部门名称

```
select ename,dname from emp,dept where emp.deptno=dept.deptno
select ename,dname from emp inner join dept on emp.deptno=dept.deptno
```

--2. 查询所有部门名称及部门的所有员工列表

```
select d.deptno,wm_concat(ename)
from dept d left join emp e on d.deptno = e.deptno
group by d.deptno,d.dname
order by d.deptno
```

--3. 求所有员工姓名,工资和其领导的姓名,工资

```
select yg.ename,yg.sal,ld.ename,ld.sal
from emp yg inner join emp ld on yg.mgr=ld.empno
```

--4. 求所有员工工资大于其领导工资的员工姓名和其上级领导姓名,上级领导工资及员工工资

```
select yg.ename,yg.sal,ld.ename,ld.sal
from emp yg inner join emp ld on yg.mgr=ld.empno
where yg.sal>ld.sal
```

--5. 在上题的基础上查询员工所对应的部门

```
select yg.deptno,dname,yg.ename,yg.sal,ld.ename,ld.sal
from emp yg inner join emp ld on yg.mgr=ld.empno
inner join dept on yg.deptno=dept.deptno
where yg.sal>ld.sal
```

--6. 查询出不是领导的员工

```
select * from emp where empno not in (select mgr from emp where mgr is not null)
```

--7. 查询出每个员工编号,姓名,部门名称,工资等级和他的上级领导的姓名,工资等级

```
select yg.empno,yg.ename,dname,yg_sg.grade,ld.ename,ld_sg.grade
from emp yg
inner join dept d on yg.deptno=d.deptno
left join emp ld on yg.mgr=ld.empno
left join salgrade yg_sg on yg.sal between yg_sg.losal and yg_sg.hisal
left join salgrade ld_sg on ld.sal between ld_sg.losal and ld_sg.hisal
```

-- 上机练习12

--1. 10号部门的部门名称和7369的员工姓名进行行拼显示,使用union all实现

```
select dname from dept where deptno=10
union all
select ename from emp where empno=7369
```

--2.查看不是领导的员工编号和员工姓名，使用minus和子查询实现

```
select empno,ename from emp
minus
select empno,ename from emp where empno in (select mgr from emp where mgr is
not null)
```

--上机练习13

--1.查询比三个部门平均工资都高的员工编号，员工姓名，工种和收入

```
select empno,ename,job,sal from emp where sal>all(
    select avg(sal) from emp group by deptno
)
```

--2.显示部门名称为RESEARCH的员工姓名，工资

```
select ename,sal from emp e
join dept d on e.deptno=d.deptno
where dname='RESEARCH'
```

--3.如果部门名称中含有字母T，则查询该部门所有员工信息，两种方法实现

--join

```
select * from emp e
join dept d on e.deptno=d.deptno
where dname like '%T%'
```

--子查询

```
select * from emp where deptno in(
    select deptno from dept where dname like '%T%'
)
```

--4.如果有平均工资不小于1500的部门信息则查询相应的部门信息

```
select * from dept where deptno in(
    select deptno from emp group by deptno having avg(sal)>=1500
)
```

--5.查询出Emp表中比所有销售员(“SALESMAN”)工资低的员工姓名、工作、工资

```
select ename,job,sal from emp where sal<(
    select min(sal) from emp where job='SALESMAN'
)
```

--6.显示比工资最高的员工参加工作时间晚的员工姓名，参加工作时间

```
select ename,hiredate from emp where hiredate>all(
    select hiredate from emp where sal=(
        select max(sal) from emp
    )
)
```


--7. 查询emp表中有2个以上直接下属的领导人员信息

```
select * from emp e where exists(  
    select 1 from emp where mgr=e.empno group by mgr having count(*)=2  
)
```

--8. 使用开窗函数求出emp表中每一个部门工资第一高的人员姓名ename和人员工资sal，以及他所在部门的最高工资

```
with rnk_emp as(  
    select ename,sal,deptno,  
        row_number()over(partition by deptno order by sal desc) row_num,  
        max(sal)over(partition by deptno) dept_max_sal  
    from emp  
)  
select * from rnk_emp where row_num=1
```

--9. emp表中每个部门员工入职日期最早的员工信息，多条件子查询

```
select e.* from emp e join(  
    select deptno,min(hiredate) min_date from emp group by deptno  
) d on e.deptno=d.deptno and e.hiredate = d.min_date
```

--10. 实现下表功能

--小计

```
select e.deptno "部门", d.dname "部门名称", e.job "工作", count(*) "小计", null  
"合计"  
from emp e join dept d on e.deptno = d.deptno  
group by e.deptno, d.dname, e.job  
union all
```

--合计

```
select e.deptno "部门", d.dname "部门名称", null "工作", null "小计", count(*)  
"合计"  
from emp e join dept d on e.deptno = d.deptno  
group by e.deptno, d.dname  
order by "部门", "工作" nulls first
```