8. 复合查询（重点）

前面我们讲解的mysql表的查询都是对一张表进行查询，在实际开发中这远远不够。

8.1 基本查询回顾

- **查询工资高于500或岗位为MANAGER的雇员，同时还要满足他们的姓名首字母为大写的J**

```
select * from EMP where (sal>500 or job='MANAGER') and ename like 'J%';
```

- **按照部门号升序而雇员的工资降序排序**

```
select * from EMP order by deptno, sal desc;
```

- **使用年薪进行降序排序**

```
select ename, sal*12+ifnull(comm,0) as '年薪' from EMP order by 年薪 desc;
```

- **显示工资最高的员工的名字和工作岗位**

```
select ename, job from EMP where sal = (select max(sal) from EMP);
```

- **显示工资高于平均工资的员工信息**

```
select ename, sal from EMP where sal>(select avg(sal) from EMP);
```

- **显示每个部门的平均工资和最高工资**

```
select deptno, format(avg(sal), 2) , max(sal) from EMP group by deptno;
```

- **显示平均工资低于2000的部门号和它的平均工资**

```
select deptno, avg(sal) as avg_sal from EMP group by deptno having
avg_sal<2000;
```

- **显示每种岗位的雇员总数，平均工资**

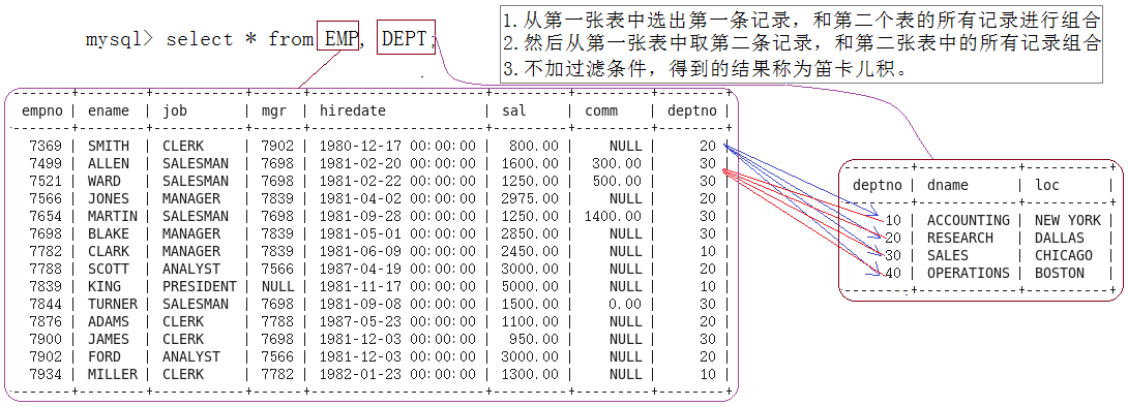
```
select job,count(*), format(avg(sal),2) from EMP group by job;
```

8.2 多表查询

实际开发中往往数据来自不同的表，所以需要多表查询。本节我们用一个简单的公司管理系统，有三张表EMP,DEPT,SALGRADE来演示如何进行多表查询。

案例：

- **显示雇员名、雇员工资以及所在部门的名字因为上面的数据来自EMP和DEPT表，因此要联合查询**



其实我们只要emp表中的deptno = dept表中的deptno字段的记录

```
select EMP.ename, EMP.sal, DEPT.dname from EMP, DEPT where EMP.deptno =
DEPT.deptno;
```

- **显示部门号为10的部门名，员工名和工资**

```
select ename, sal,dname from EMP, DEPT where EMP.deptno=DEPT.deptno and
DEPT.deptno = 10;
```

- **显示各个员工的姓名，工资，及工资级别**

```
select ename, sal, grade from EMP, SALGRADE where EMP.sal between losal and
hisal;
```

8.3 自连接

自连接是指在同一张表连接查询

案例：

显示员工FORD的上级领导的编号和姓名（mgr是员工领导的编号--empno）

- **使用的子查询：**

```
select empno,ename from emp where emp.empno=(select mgr from emp where
ename='FORD');
```

- **使用多表查询（自查询）**

```
-- 使用到表的别名
--from emp leader, emp worker，给自己的表起别名，因为要先做笛卡尔积，所以别名可以先识
别

select leader.empno,leader.ename from emp leader, emp worker where
leader.empno = worker.mgr and worker.ename='FORD';
```

8.4 子查询

子查询是指嵌入在其他sql语句中的select语句，也叫嵌套查询

8.4.1 单行子查询

返回一行记录的子查询

- **显示SMITH同一部门的员工**

```
select * from EMP WHERE deptno = (select deptno from EMP where
ename='smith');
```

8.4.2 多行子查询

返回多行记录的子查询

- **in**关键字；查询和10号部门的工作岗位相同的雇员的名字，岗位，工资，部门号，但是不包含10自己的

```
select ename,job,sal,deptno from emp where job in (select distinct job from
emp where deptno=10) and deptno<>10;
```

- **all**关键字；显示工资比部门30的所有员工的工资高的员工的姓名、工资和部门号

```
select ename, sal, deptno from EMP where sal > all(select sal from EMP where
deptno=30);
```

- **any**关键字；显示工资比部门30的任意员工的工资高的员工的姓名、工资和部门号（包含自己部门的员工）

```
select ename, sal, deptno from EMP where sal > any(select sal from EMP where
deptno=30);
```

8.4.3 多列子查询

单行子查询是指子查询只返回单列，单行数据；多行子查询是指返回单列多行数据，都是针对单列而言的，而多列子查询则是指查询返回多个列数据的子查询语句

案例：查询和SMITH的部门和岗位完全相同的所有雇员，不含SMITH本人

```
mysql> select ename from EMP where (deptno, job)=(select deptno, job from EMP
where ename='SMITH') and ename <> 'SMITH';
+-------+
| ename |
+-------+
| ADAMS |
+-------+
```

8.4.4 在from子句中使用子查询

子查询语句出现在from子句中。这里要用到数据查询的技巧，把一个子查询当做一个临时表使用。

案例：

- **显示每个高于自己部门平均工资的员工的姓名、部门、工资、平均工资**

```
//获取各个部门的平均工资，将其看作临时表
select ename, deptno, sal, format(asal,2) from EMP,
 (select avg(sal) asal, deptno dt from EMP group by deptno) tmp
where EMP.sal > tmp.asal and EMP.deptno=tmp.dt;
```

- **查找每个部门工资最高的人的姓名、工资、部门、最高工资**

```
select EMP.ename, EMP.sal, EMP.deptno, ms from EMP,
(select max(sal) ms, deptno from EMP group by deptno) tmp
 where EMP.deptno=tmp.deptno and EMP.sal=tmp.ms;
```

- **显示每个部门的信息（部门名，编号，地址）和人员数量**
  - 方法1：使用多表

```
select DEPT.dname, DEPT.deptno, DEPT.loc,count(*) '部门人数' from EMP,
DEPT
where EMP.deptno=DEPT.deptno
group by DEPT.deptno,DEPT.dname,DEPT.loc;
```

  - 方法2：使用子查询

```
-- 1. 对EMP表进行人员统计
select count(*), deptno from  EMP group by deptno;
-- 2. 将上面的表看作临时表
select DEPT.deptno, dname, mycnt, loc from DEPT,
(select count(*) mycnt, deptno from  EMP group by deptno) tmp
where DEPT.deptno=tmp.deptno;
```

8.4.5 合并查询

在实际应用中，为了合并多个select的执行结果，可以使用集合操作符 union，union all

8.4.5.1 union

该操作符用于取得两个结果集的并集。当使用该操作符时，会自动去掉结果集中的重复行。

案例：将工资大于2500或职位是MANAGER的人找出来

```
mysql> select ename, sal, job from EMP where sal>2500 union
    -> select ename, sal, job from EMP where job='MANAGER';--去掉了重复记录
+-------+---------+-----------+
| ename | sal     | job       |
+-------+---------+-----------+
| JONES | 2975.00 | MANAGER   |
| BLAKE | 2850.00 | MANAGER   |
| SCOTT | 3000.00 | ANALYST   |
| KING  | 5000.00 | PRESIDENT |
| FORD  | 3000.00 | ANALYST   |
| CLARK | 2450.00 | MANAGER   |
+-------+---------+-----------+
```

8.4.5.3 union all

该操作符用于取得两个结果集的并集。当使用该操作符时，不会去掉结果集中的重复行。

案例：将工资大于25000或职位是MANAGER的人找出来

```
mysql> select ename, sal, job from EMP where sal>2500 union all
    -> select ename, sal, job from EMP where job='MANAGER';
+-------+---------+-----------+
| ename | sal     | job       |
+-------+---------+-----------+
| JONES | 2975.00 | MANAGER   |
| BLAKE | 2850.00 | MANAGER   |
| SCOTT | 3000.00 | ANALYST   |
| KING  | 5000.00 | PRESIDENT |
| FORD  | 3000.00 | ANALYST   |
| JONES | 2975.00 | MANAGER   |
| BLAKE | 2850.00 | MANAGER   |
| CLARK | 2450.00 | MANAGER   |
+-------+---------+-----------+
```

8.5 实战OJ

- 牛客：查找所有员工入职时候的薪水情况，给出emp_no以及salary， 并按照emp_no进行逆序
- 牛客：针对库中的所有表生成select count(*) from tableName 对应的SQL语句
- 牛客：获取所有非manager的员工emp_no
- 牛客：获取所有员工当前的manager,获取所有员工当前的manager，如果当前的manager是自己的话结果不显示，当前表示to_date='9999-01-01'