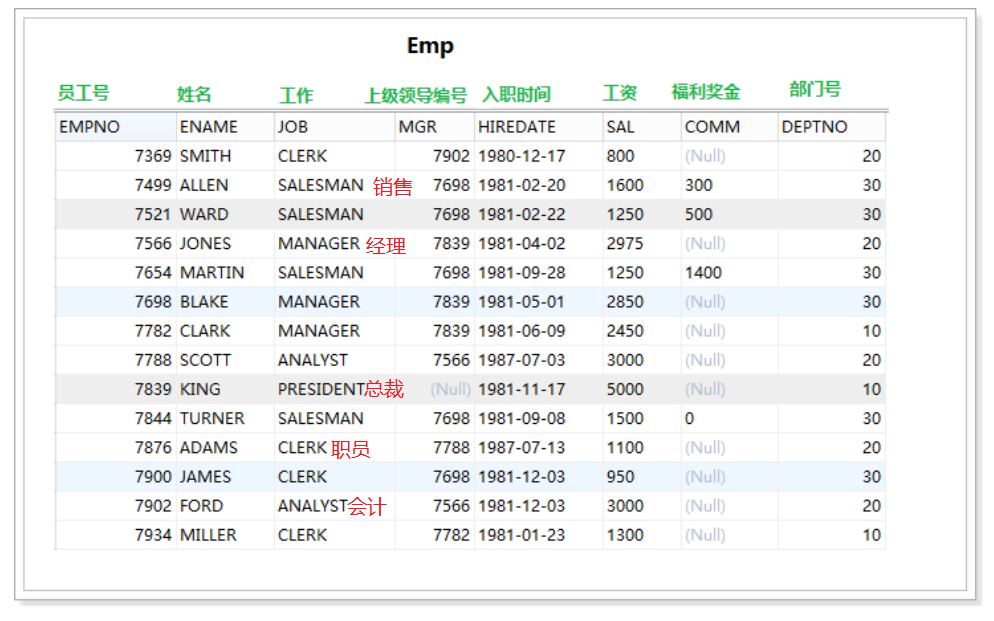
2019-3-28

SQL高级查询

讲师:BRUCELIU

# 数据表的结构

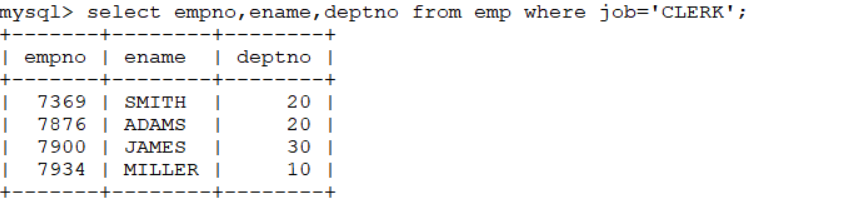


# 基础查询

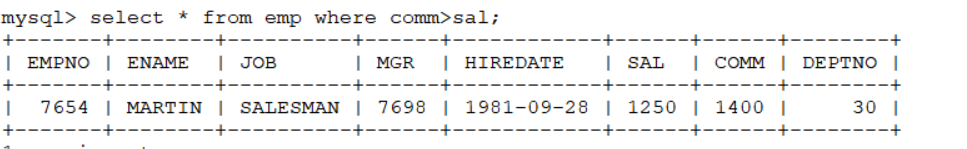
-- 1、查找部门30中员工的详细信息。

select \* from emp where deptno=30;

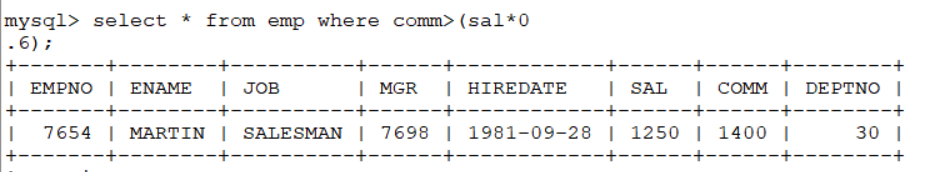
-- 2.找出从事clerk工作的员工的编号、姓名、部门号。



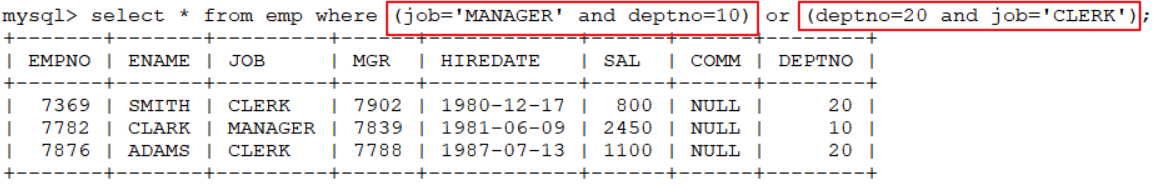
-- 3.检索出奖金多于基本工资的员工信息。



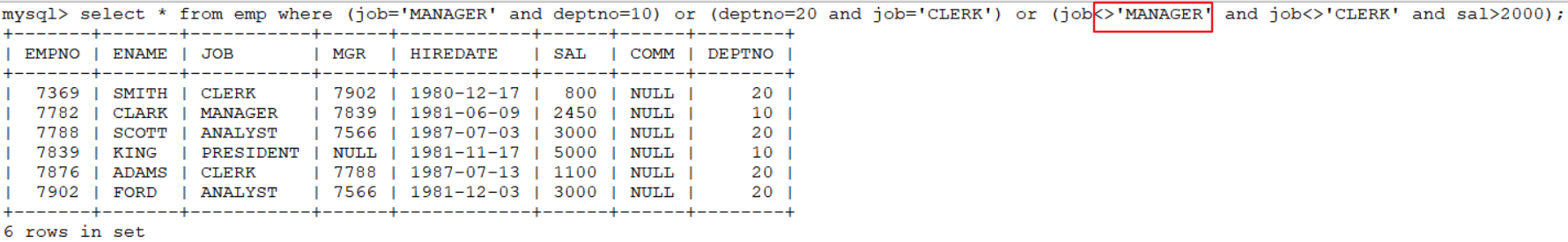
-- 4.检索出奖金多于基本工资的60%的员工信息。



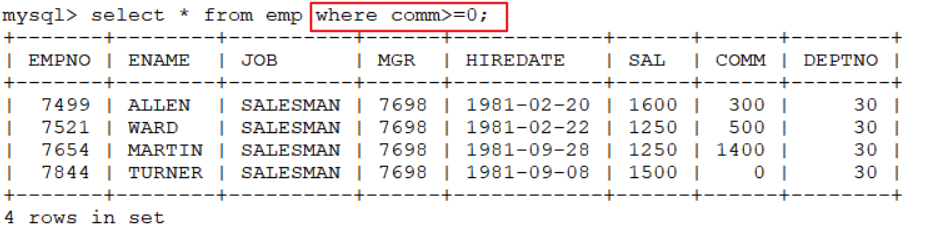
-- 5.找出10部门的经理、20部门的职员的员工信息。



-- 6.找出10部门的经理、20部门的职员 或者 既不是经理也不是职员但是工资高于2000元的员工信息。



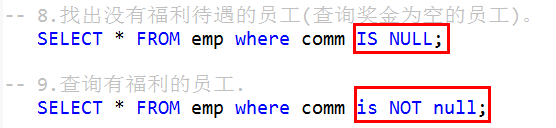
-- 7.找出获得奖金的员工。

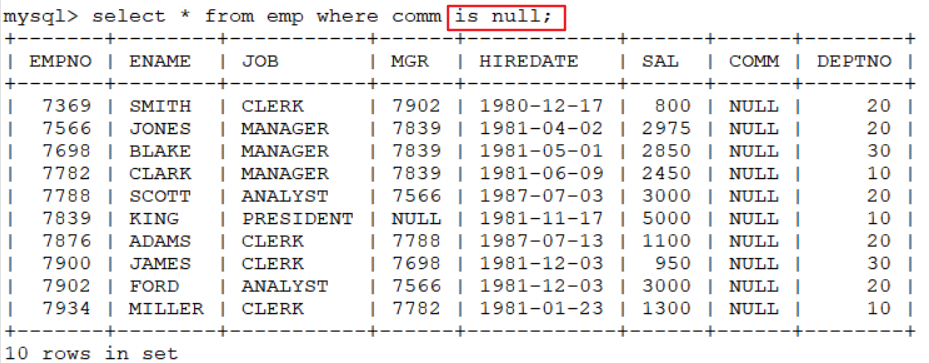


# 空或非空查询

在SQL中使用 **IS NULL** 判断一个字段是否为空

* **IS NULL： 为空**
* **IS NOT NULL : 非空**

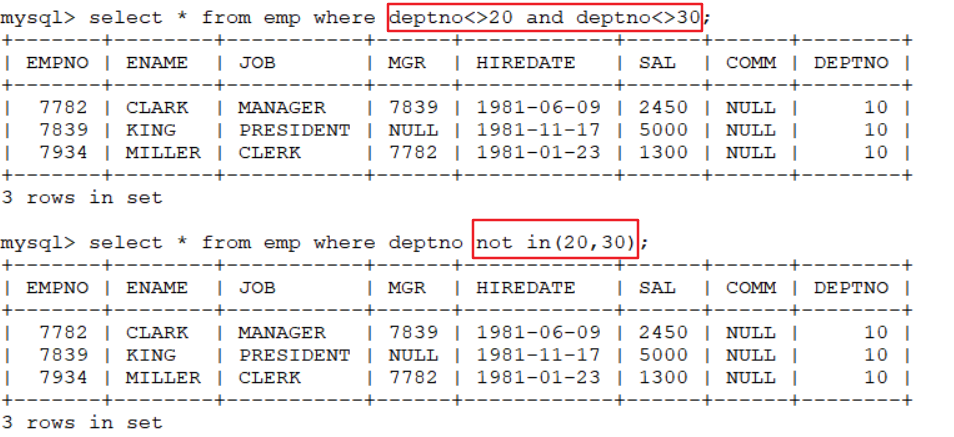




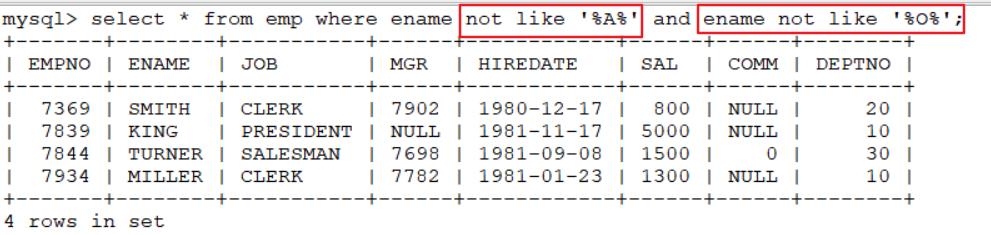
# NOT关键字：

作用：取反

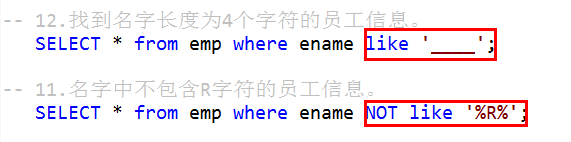
**NOT IN( )** 不在括号中的那些



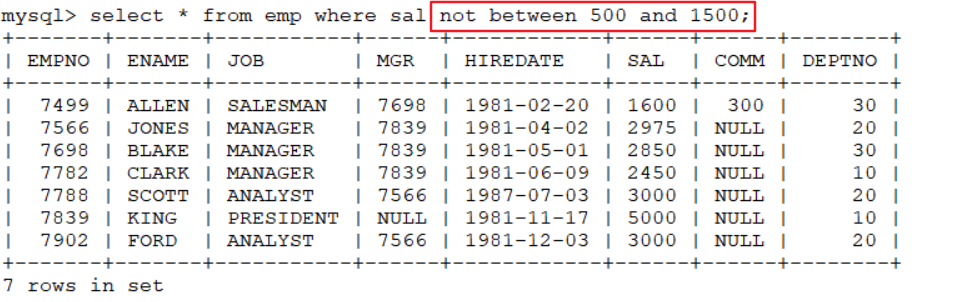
**NOT LIKE**  不包含的那些



**IS NOT NULL ( ) 非空**



Not Between XXX and XXX



# 分组查询(重要)

在SQL查询的过程中，有时候要按照**某种规则把数据**分为一组之后再**统计**，就要使用分组查询。

知道按照**什么**分组？

**GROUP BY 列名 分组**

**HAVING ：对分组之后的数据再次筛选，如果使用HAVING一定要接在GROUP BY之后使用，不能单独使用**

语法：

SELECT 列名, 聚合函数 FROM 表名

WHERE 条件1….

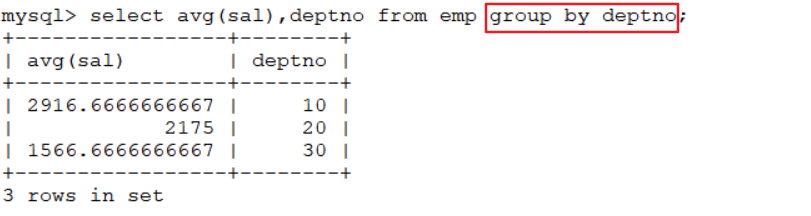
**GROUP BY 分组的列……**

**HAVING 条件**

**Order By 排序条件**

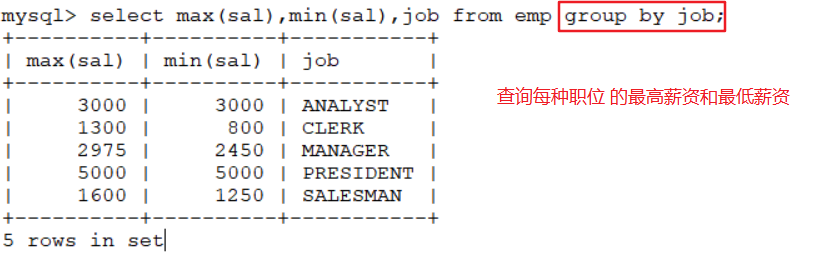
1. **查询**每个部门**的平均薪资？**

思考：按照部门来分组



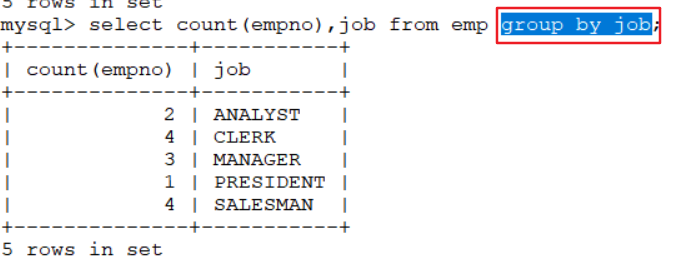
1. **查询每种职位的最高薪资和最低薪资？**

思考：按照职位来分组

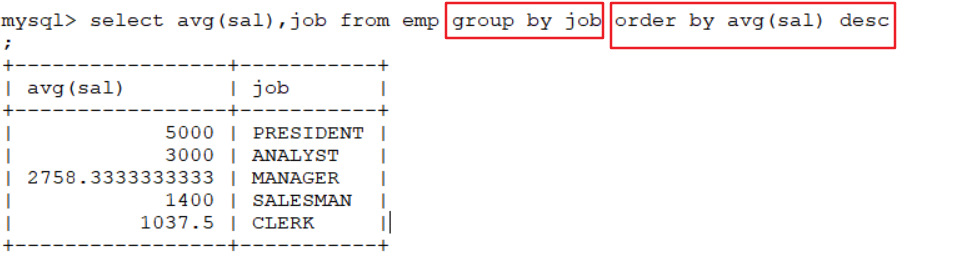


1. **查询每种职位有几人？**

思考：按照职位来分组



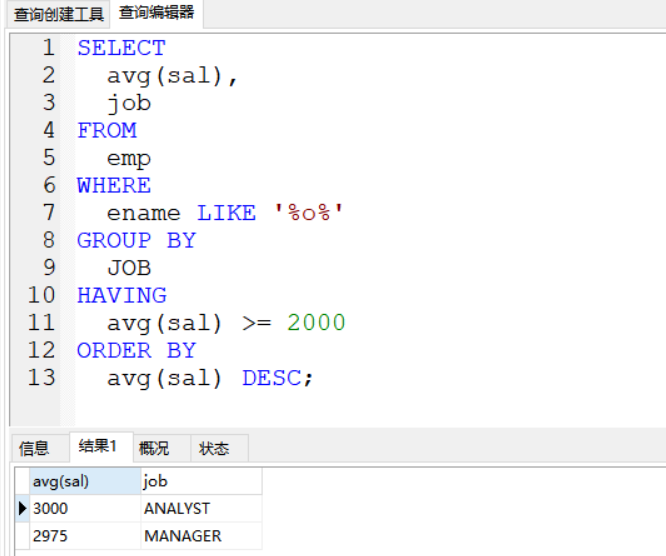
1. **查询每种职位的平均薪资，并按照平均薪资降序排列！！！**

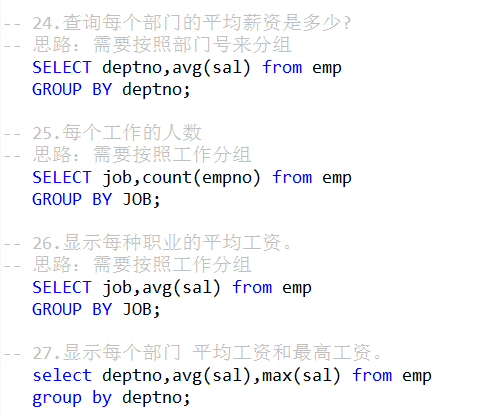


1. **查询每种职位平均薪资，且是高于2000的平均薪资，并按照平均薪资降序排列！！！**



**6.查询名字中包含o的员工，的每种职位平均薪资，且是高于2000的平均薪资，并按照平均薪资降序排列！！！**

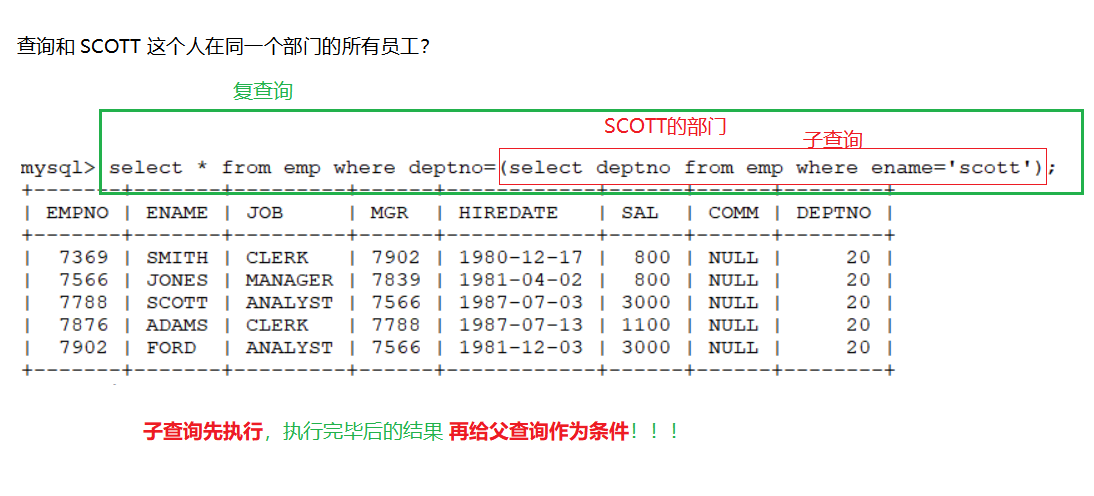




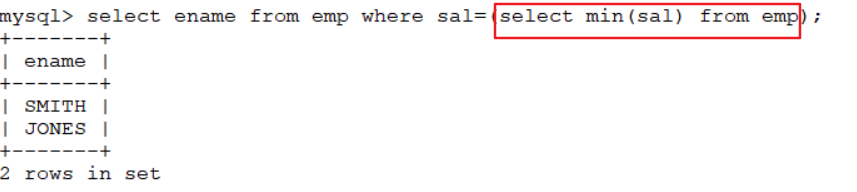


# 子查询

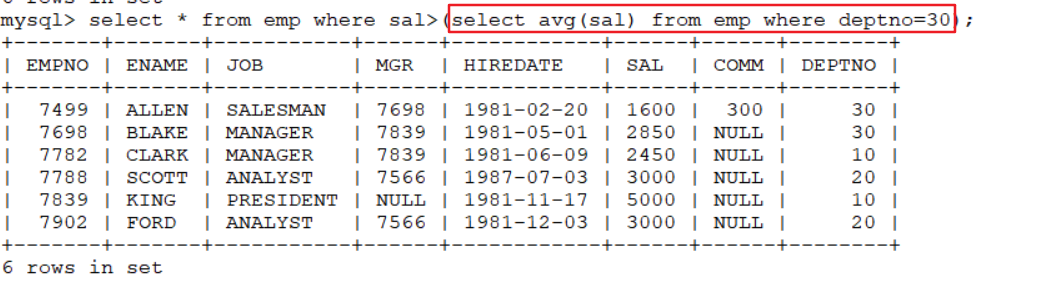
1. 查询和SCOTT这个人在同一个部门的所有员工？



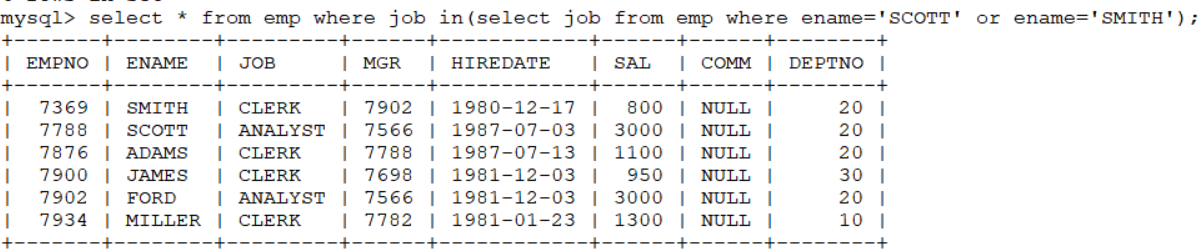
1. 薪资最低的人的姓名?



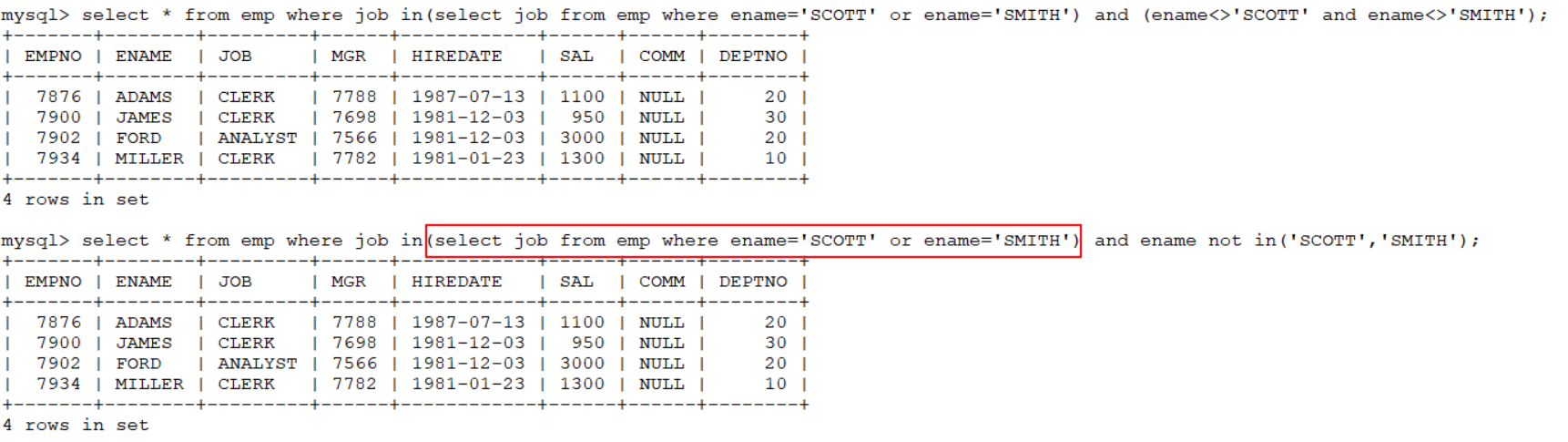
1. 查询薪资高于30号部门平均薪资的所有员工？



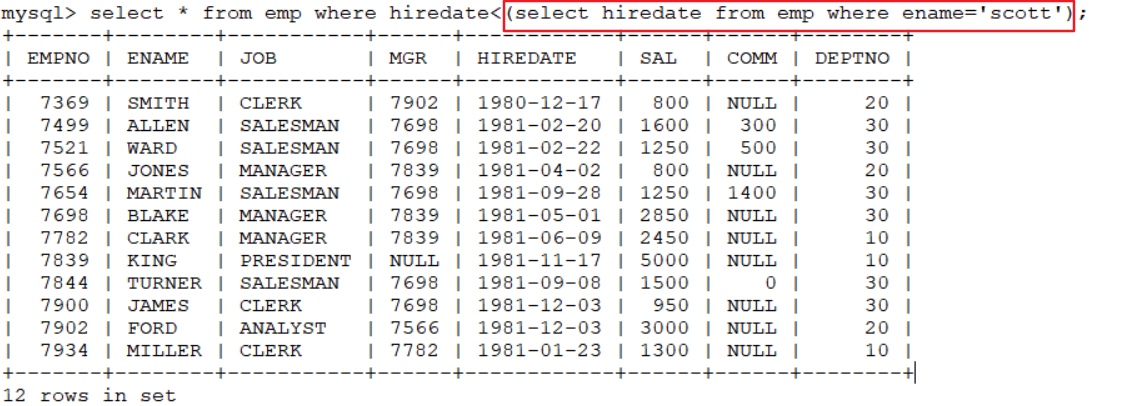
1. 查询和 SCOTT或者 SMITH 干同一种工作的所有员工？

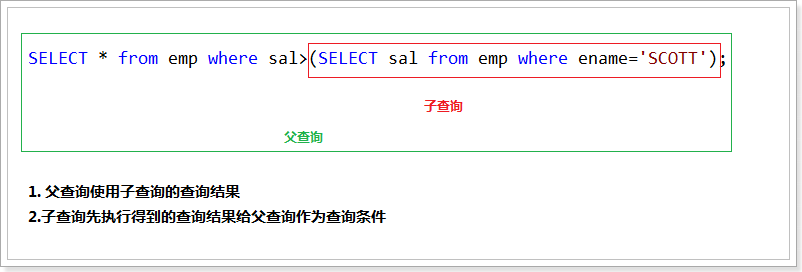


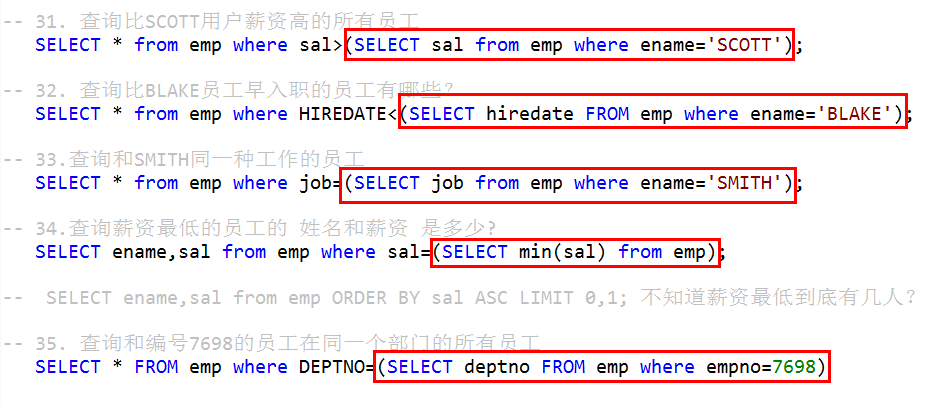
1. 查询和 SCOTT或者 SMITH 干同一种工作的**同事**(把自己去掉)？



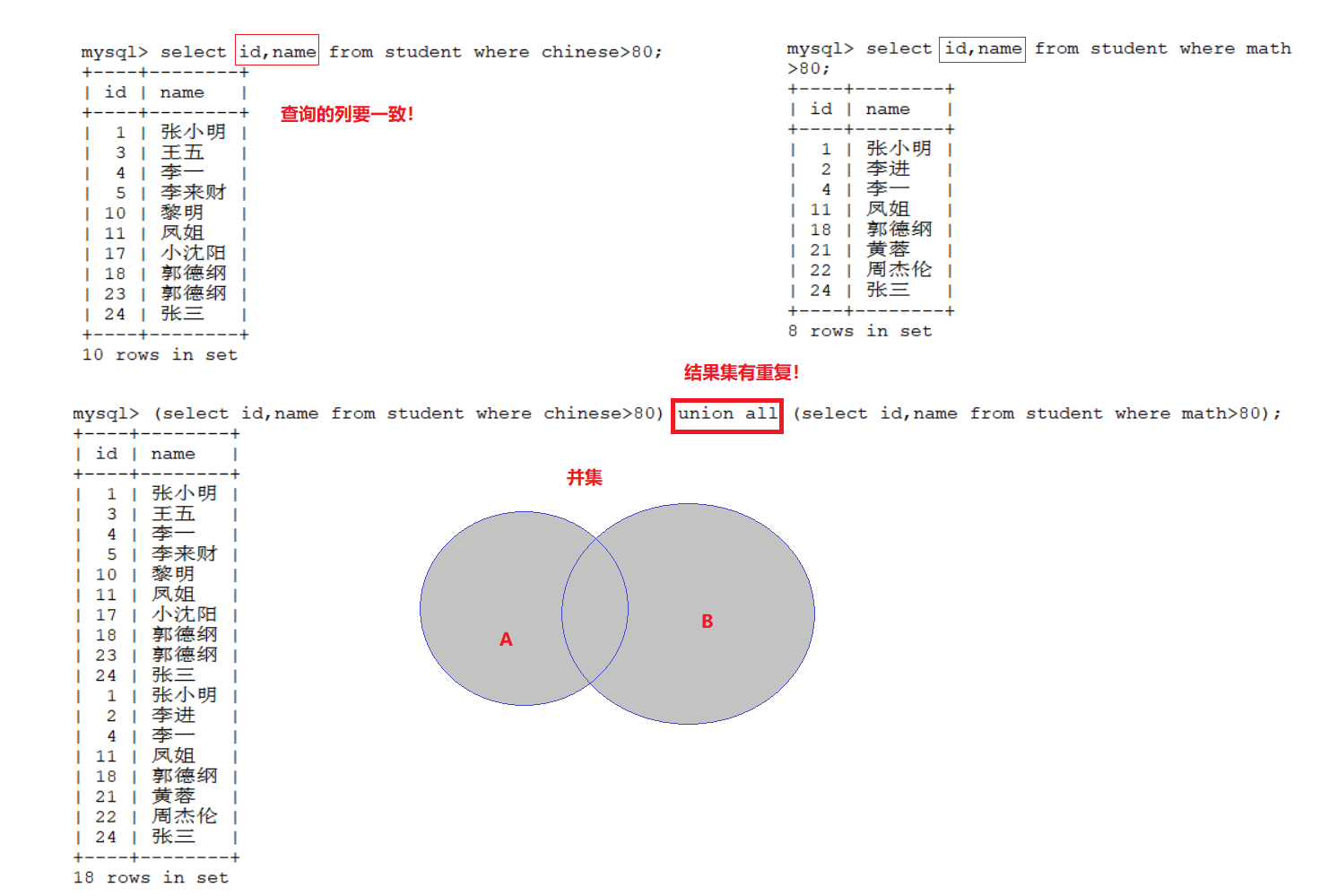
1. 查询比SCOTT 入职还要早的哪些员工的姓名？







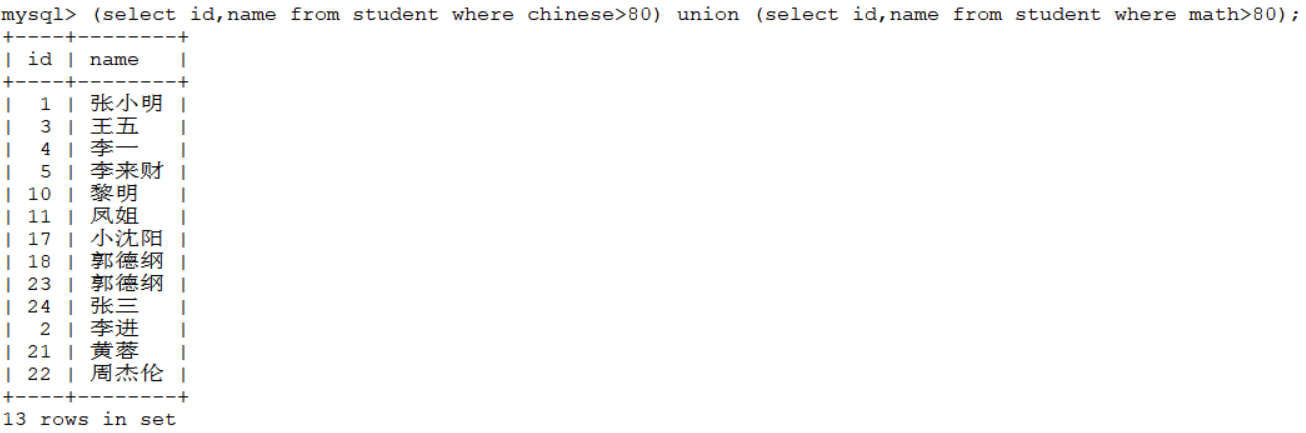
# 交集 并集 差集 （扩展）



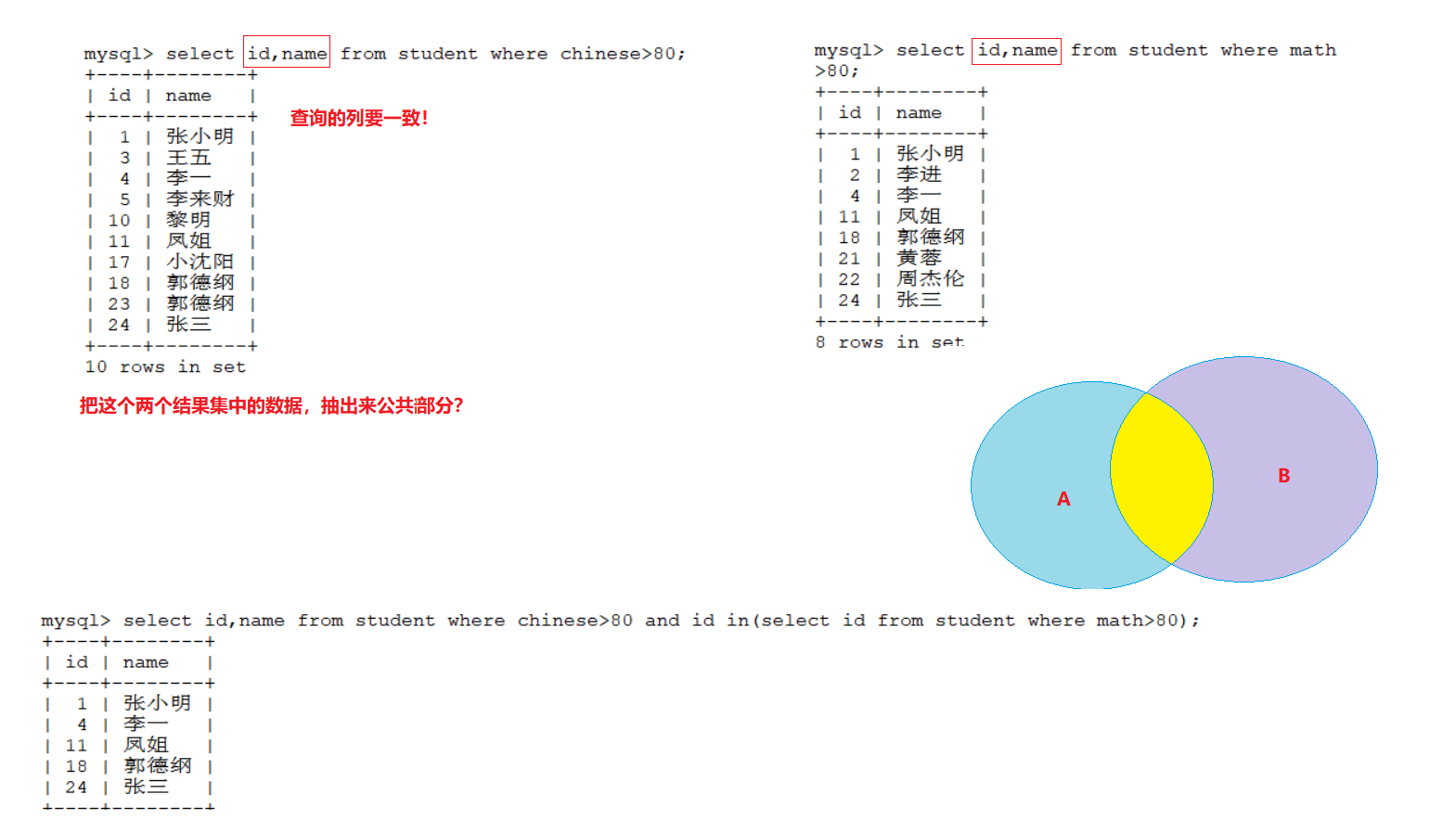
* **并集 UNION**

UNION  ALL 把两个结果集合并在一起，会有重复的！

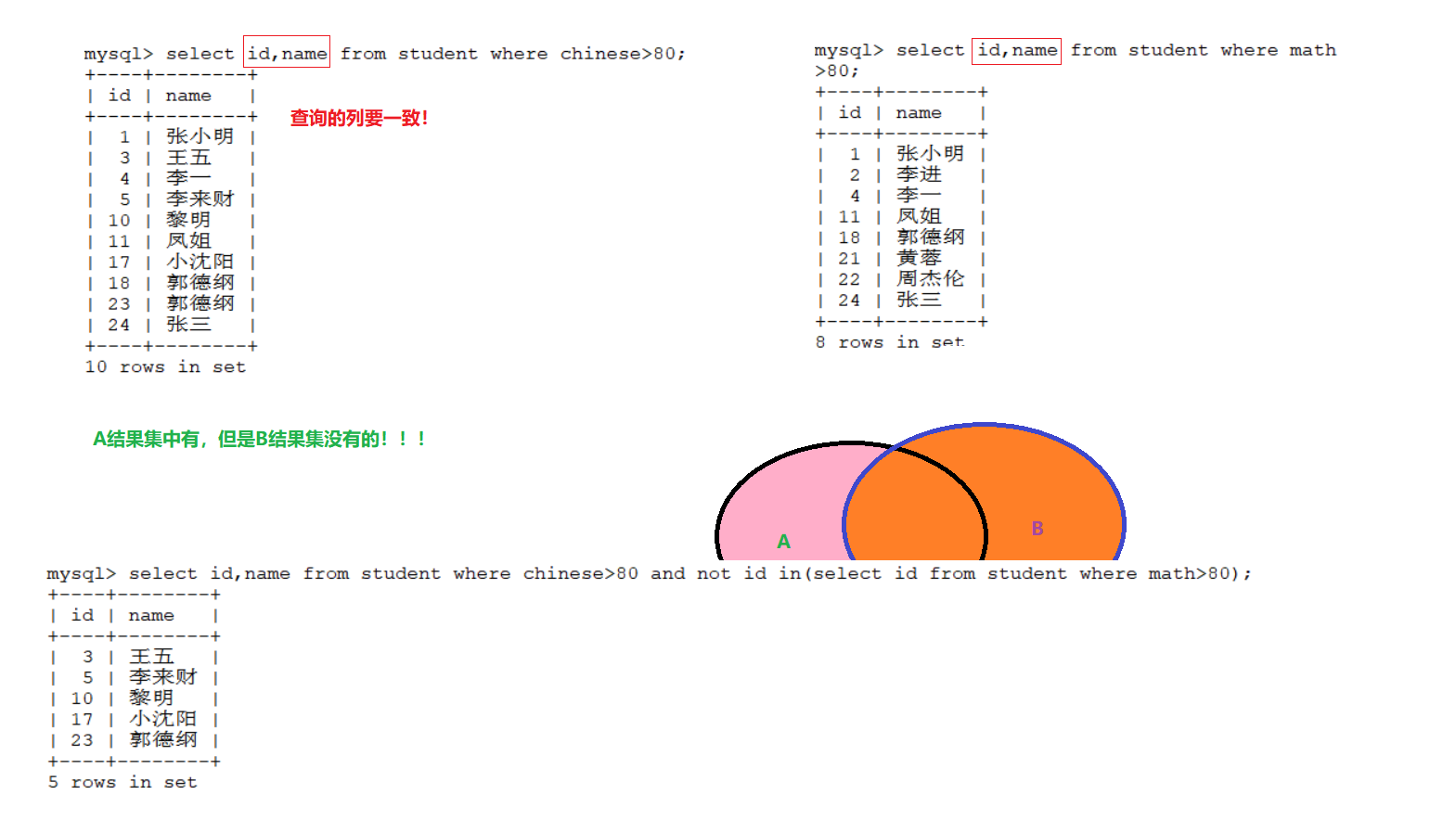
UNION 把结果集合并在一起，去重复！



* **交集**

****

* **差集**

****