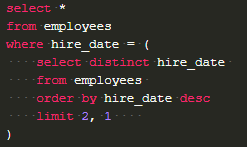
# Sql

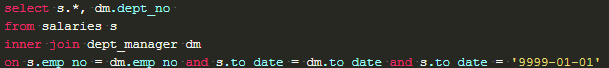
## 查找入职员工时间排名倒数第三的员工所有信息 CREATE TABLE `employees` ( `emp\_no` int(11) NOT NULL, `birth\_date` date NOT NULL, `first\_name` varchar(14) NOT NULL, `last\_name` varchar(16) NOT NULL, `gender` char(1) NOT NULL, `hire\_date` date NOT NULL, PRIMARY KEY (`emp\_no`));

思路：如果这里用limit，那么要注意，这里时间的话是会重复的，所以必须用distinct排除相同时间的人。



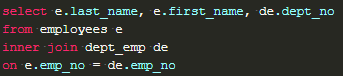
## 查找各个部门当前(to\_date='9999-01-01')领导当前薪水详情以及其对应部门编号dept\_no CREATE TABLE `dept\_manager` ( `dept\_no` char(4) NOT NULL, `emp\_no` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`)); **CREATE TABLE `salaries` (** `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

**思路：这里部门经理单独成一张表，包含了其部门编号，员工号，任期其实时间和结束时间；薪水单独成一张表，包含了员工的编号，薪水，起始时间和结束时间，工资表里面包含的就不止经理的工资了，还包含普通员工的工资，所以这里就是从薪水表里面找出经理的工资情况，并打印出他所在的部门号。**



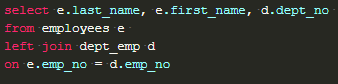
## 查找所有已经分配部门的员工的last\_name和first\_name以及dept\_no CREATE TABLE `dept\_emp` ( `emp\_no` int(11) NOT NULL, `dept\_no` char(4) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`)); CREATE TABLE `employees` ( `emp\_no` int(11) NOT NULL, `birth\_date` date NOT NULL, `first\_name` varchar(14) NOT NULL, `last\_name` varchar(16) NOT NULL, `gender` char(1) NOT NULL, `hire\_date` date NOT NULL, PRIMARY KEY (`emp\_no`));

思路：dept-emp是一张部门和员工的中间表，很明显是用来连接部门表和员工表的。此处要找出已经分配部门的员工的信息，只要将该两边连接即可。



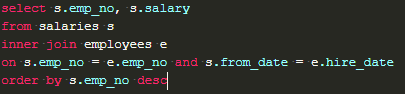
## 查找所有员工的last\_name和first\_name以及对应部门编号dept\_no，也包括展示没有分配具体部门的员工 CREATE TABLE `dept\_emp` ( `emp\_no` int(11) NOT NULL, `dept\_no` char(4) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`)); CREATE TABLE `employees` ( `emp\_no` int(11) NOT NULL, `birth\_date` date NOT NULL, `first\_name` varchar(14) NOT NULL, `last\_name` varchar(16) NOT NULL, `gender` char(1) NOT NULL, `hire\_date` date NOT NULL, PRIMARY KEY (`emp\_no`));

思路：相比于上题来说其实就是换成了左连接。



5、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  
CREATE TABLE `employees` (  
`emp\_no` int(11) NOT NULL,  
`birth\_date` date NOT NULL,  
`first\_name` varchar(14) NOT NULL,  
`last\_name` varchar(16) NOT NULL,  
`gender` char(1) NOT NULL,  
`hire\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`));  
CREATE TABLE `salaries` (  
`emp\_no` int(11) NOT NULL,  
`salary` int(11) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`,`from\_date`));

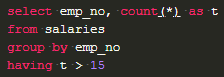
思路：其实就是查找薪水表中from\_date == hire\_date时的薪水。



## 查找薪水涨幅超过15次的员工号emp\_no以及其对应的涨幅次数t CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

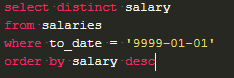
思路：值得注意的是这里涨幅可能为0，可能为负，就是记录薪水变动了多少次，那么只要看这个人在工资表中出现了多少次就好。

那么就要对emp\_no进行分组，然后count（\*）> 15就能得到。



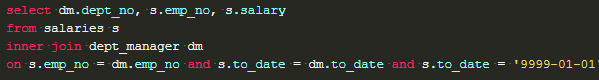
## 找出所有员工当前(to\_date='9999-01-01')具体的薪水salary情况，对于相同的薪水只显示一次,并按照逆序显示 CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

思路：这题目找出当前所有员工所有薪水的可能情况。



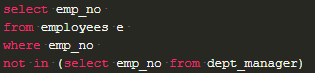
## 获取所有部门当前manager的当前薪水情况，给出dept\_no, emp\_no以及salary，当前表示to\_date='9999-01-01' CREATE TABLE `dept\_manager` ( `dept\_no` char(4) NOT NULL, `emp\_no` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`)); CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

思路：跟第2题是一样的思路



## 获取所有非manager的员工emp\_no CREATE TABLE `dept\_manager` ( `dept\_no` char(4) NOT NULL, `emp\_no` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`)); CREATE TABLE `employees` ( `emp\_no` int(11) NOT NULL, `birth\_date` date NOT NULL, `first\_name` varchar(14) NOT NULL, `last\_name` varchar(16) NOT NULL, `gender` char(1) NOT NULL, `hire\_date` date NOT NULL, PRIMARY KEY (`emp\_no`));

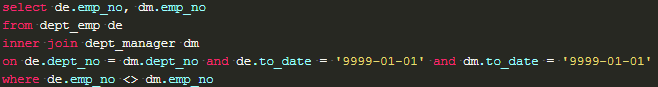
思路：就是找到员工表中不是经理的员工，用not in就好。



## 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'。 结果第一列给出当前员工的emp\_no,第二列给出其manager对应的manager\_no。 CREATE TABLE `dept\_emp` ( `emp\_no` int(11) NOT NULL, `dept\_no` char(4) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`)); CREATE TABLE `dept\_manager` ( `dept\_no` char(4) NOT NULL, `emp\_no` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`));

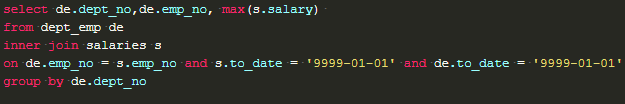
思路：部门有两种员工，经理和员工，经理和员工的工资存在一张表，他们员工号存在一张表。dept\_emp表对应着部门号和员工号之间的关系，其中包含经理，dept\_manager对应着部门号和经理之间的关系。

题目求的是所有员工对应的manager，那么就是将部门号相同的连接起来，就能得到，但是因为第一张表包含第二张表，要排除是自己的情况。



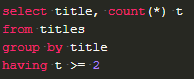
## 获取所有部门中当前员工薪水最高的相关信息，给出dept\_no, emp\_no以及其对应的salary CREATE TABLE `dept\_emp` ( `emp\_no` int(11) NOT NULL, `dept\_no` char(4) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`)); CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

思路：对部门进行分组，求每个部门中的最高薪水。



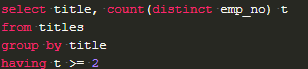
## 从titles表获取按照title进行分组，每组个数大于等于2，给出title以及对应的数目t。 CREATE TABLE IF NOT EXISTS "titles" ( `emp\_no` int(11) NOT NULL, `title` varchar(50) NOT NULL, `from\_date` date NOT NULL, `to\_date` date DEFAULT NULL);

思路：按title进行分组后，使用having和count（\*）。



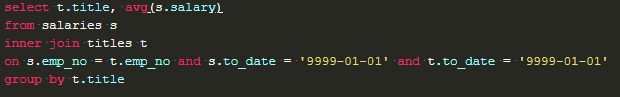
## 从titles表获取按照title进行分组，每组个数大于等于2，给出title以及对应的数目t。 注意对于重复的emp\_no进行忽略。 CREATE TABLE IF NOT EXISTS `titles` ( `emp\_no` int(11) NOT NULL, `title` varchar(50) NOT NULL, `from\_date` date NOT NULL, `to\_date` date DEFAULT NULL);

思路：那就不count（\*）了，改为count（distinct emp\_no）



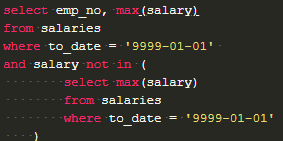
## 13统计出当前各个title类型对应的员工当前（to\_date='9999-01-01'）薪水对应的平均工资。结果给出title以及平均工资avg。 CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`)); CREATE TABLE IF NOT EXISTS "titles" ( `emp\_no` int(11) NOT NULL, `title` varchar(50) NOT NULL, `from\_date` date NOT NULL, `to\_date` date DEFAULT NULL);

思路：titles表包含了员工工号，员工title和称呼的任期时间，对title进行分组，去当前有效title，求出平均工资。也就是求出每个title的平均工资。



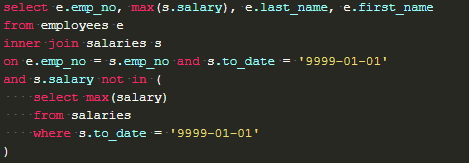
## 获取当前（to\_date='9999-01-01'）薪水第二多的员工的emp\_no以及其对应的薪水salary CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

思路：薪水第二多的员工可以不用limit，直接使用not in。



## 查找当前薪水(to\_date='9999-01-01')排名第二多的员工编号emp\_no、薪水salary、last\_name以及first\_name，不准使用order by CREATE TABLE `employees` ( `emp\_no` int(11) NOT NULL, `birth\_date` date NOT NULL, `first\_name` varchar(14) NOT NULL, `last\_name` varchar(16) NOT NULL, `gender` char(1) NOT NULL, `hire\_date` date NOT NULL, PRIMARY KEY (`emp\_no`)); CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

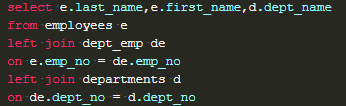
思路：上题的再应用，区别在于这里找出来薪资第二高的人的有关信息。



## 查找所有员工的last\_name和first\_name以及对应的dept\_name，也包括暂时没有分配部门的员工 CREATE TABLE `departments` ( `dept\_no` char(4) NOT NULL, `dept\_name` varchar(40) NOT NULL, PRIMARY KEY (`dept\_no`)); CREATE TABLE `dept\_emp` ( `emp\_no` int(11) NOT NULL, `dept\_no` char(4) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`)); CREATE TABLE `employees` ( `emp\_no` int(11) NOT NULL, `birth\_date` date NOT NULL, `first\_name` varchar(14) NOT NULL, `last\_name` varchar(16) NOT NULL, `gender` char(1) NOT NULL, `hire\_date` date NOT NULL, PRIMARY KEY (`emp\_no`));

思路：这里出现了部门表，里面包含了部门号和部门名，员工的姓名要从employee表取，部门名要从department表取，连接二者的表毫无疑问就是dept-emp表。

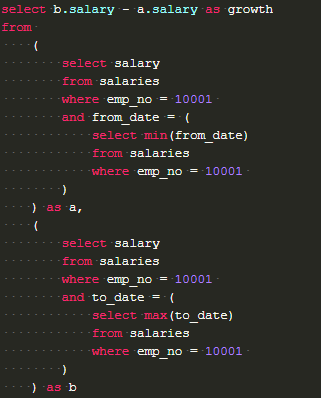
且这里还应包含没有分配部门的员工，那么应该employee表放左边，三表连接用left join



## 查找员工编号emp\_no为10001其自入职以来的薪水salary涨幅值growth CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

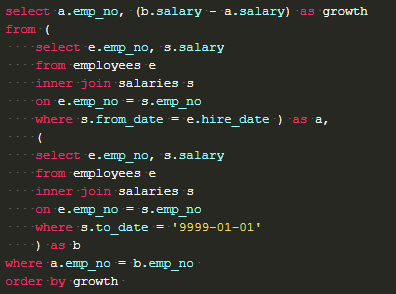
思路：其实就是求该员工的入职时间时的工资和当前时间的工资差。如果要求所有员工的涨幅，那么对emp-no分组，求出最小工资和最大工资的差

其实这种想法是错误的，因为员工可能会降薪，所以应该找最小任期时间和最晚任期时间



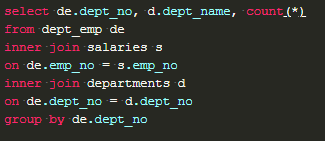
## 查找所有员工自入职以来的薪水涨幅情况，给出员工编号emp\_no以及其对应的薪水涨幅growth，并按照growth进行升序 CREATE TABLE `employees` ( `emp\_no` int(11) NOT NULL, `birth\_date` date NOT NULL, `first\_name` varchar(14) NOT NULL, `last\_name` varchar(16) NOT NULL, `gender` char(1) NOT NULL, `hire\_date` date NOT NULL, PRIMARY KEY (`emp\_no`)); CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

思路：相比于上一题，这题要求找出所有员工的涨幅，且需要打印出员工的id。我的想法是，找到所有员工的起始工资，再找到所有员工的当前工资，求出他们的差。



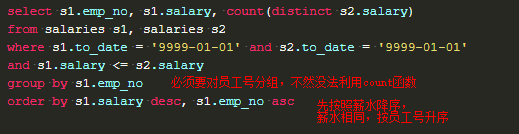
## 统计各个部门的工资记录数，给出部门编码dept\_no、部门名称dept\_name以及次数sum CREATE TABLE `departments` ( `dept\_no` char(4) NOT NULL, `dept\_name` varchar(40) NOT NULL, PRIMARY KEY (`dept\_no`)); CREATE TABLE `dept\_emp` ( `emp\_no` int(11) NOT NULL, `dept\_no` char(4) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`dept\_no`)); CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

思路：每个部门有多个员工，每个员工有多条工资记录，那么根据部门分组就可以进行计数，再连接部门表，就能找到部门名



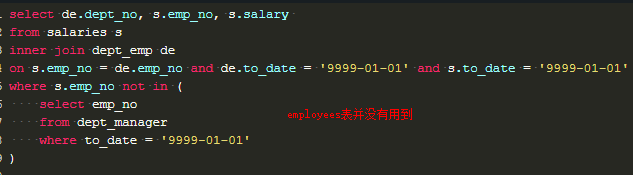
## 对所有员工的当前(to\_date='9999-01-01')薪水按照salary进行按照1-N的排名，相同salary并列且按照emp\_no升序排列 CREATE TABLE `salaries` ( `emp\_no` int(11) NOT NULL, `salary` int(11) NOT NULL, `from\_date` date NOT NULL, `to\_date` date NOT NULL, PRIMARY KEY (`emp\_no`,`from\_date`));

思路：首先将薪水表中的当前薪水取出，然后对所有员工的薪水进行排序，这是表1要进行的操作，计数主要要用到表2，看表2中大于表1中某条记录的记录有多少条



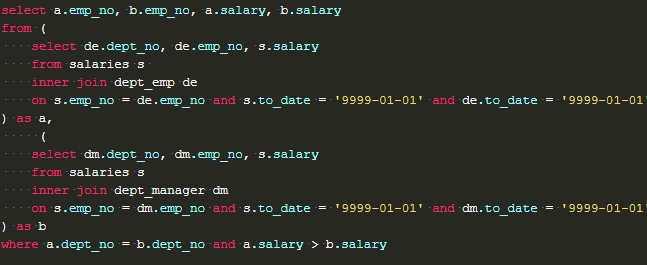
## 获取所有非manager员工当前的薪水情况，给出dept\_no、emp\_no以及salary ，当前表示to\_date='9999-01-01'

思路：就是在薪水表中找到所有非manager员工的情况，按理来说只需要salaries表和dept\_manager表，但是因为需要dept\_no,所以还需要dept\_emp表。  
CREATE TABLE `dept\_emp` (  
`emp\_no` int(11) NOT NULL,  
`dept\_no` char(4) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`,`dept\_no`));  
CREATE TABLE `dept\_manager` (  
`dept\_no` char(4) NOT NULL,  
`emp\_no` int(11) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`,`dept\_no`));  
CREATE TABLE `employees` (  
`emp\_no` int(11) NOT NULL,  
`birth\_date` date NOT NULL,  
`first\_name` varchar(14) NOT NULL,  
`last\_name` varchar(16) NOT NULL,  
`gender` char(1) NOT NULL,  
`hire\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`));  
CREATE TABLE `salaries` (  
`emp\_no` int(11) NOT NULL,  
`salary` int(11) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`,`from\_date`));



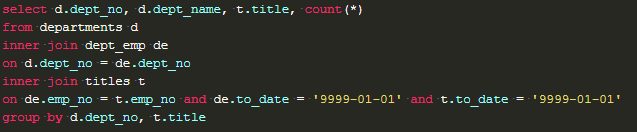
## 获取员工其当前的薪水比其manager当前薪水还高的相关信息，当前表示to\_date='9999-01-01', 结果第一列给出员工的emp\_no， 第二列给出其manager的manager\_no， 第三列给出该员工当前的薪水emp\_salary, 第四列给该员工对应的manager当前的薪水manager\_salary

思路：求出两个临时表，每个表包含员工和经理的员工号、部门号和薪水，其中部门号相等，且经理薪水大于员工薪水的就是符合题意的  
CREATE TABLE `dept\_emp` (  
`emp\_no` int(11) NOT NULL,  
`dept\_no` char(4) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`,`dept\_no`));  
CREATE TABLE `dept\_manager` (  
`dept\_no` char(4) NOT NULL,  
`emp\_no` int(11) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`,`dept\_no`));  
CREATE TABLE `salaries` (  
`emp\_no` int(11) NOT NULL,  
`salary` int(11) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`,`from\_date`));



## 汇总各个部门当前员工的title类型的分配数目，结果给出部门编号dept\_no、dept\_name、其当前员工所有的title以及该类型title对应的数目count

思路：对部门编号和title进行符合分组  
CREATE TABLE `departments` (  
`dept\_no` char(4) NOT NULL,  
`dept\_name` varchar(40) NOT NULL,  
PRIMARY KEY (`dept\_no`));  
CREATE TABLE `dept\_emp` (  
`emp\_no` int(11) NOT NULL,  
`dept\_no` char(4) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`,`dept\_no`));  
CREATE TABLE IF NOT EXISTS `titles` (  
`emp\_no` int(11) NOT NULL,  
`title` varchar(50) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date DEFAULT NULL);



## 23、给出每个员工每年薪水涨幅超过5000的员工编号emp\_no、薪水变更开始日期from\_date以及薪水涨幅值salary\_growth，并按照salary\_growth逆序排列。

提示：在sqlite中获取datetime时间对应的年份函数为strftime('%Y', to\_date)  
CREATE TABLE `salaries` (  
`emp\_no` int(11) NOT NULL,  
`salary` int(11) NOT NULL,  
`from\_date` date NOT NULL,  
`to\_date` date NOT NULL,  
PRIMARY KEY (`emp\_no`,`from\_date`));

思路：每年的涨幅怎么理解呢，复用工资表，两张表相差一年。薪水变更日期是涨工资的日期，所以要用s2

