**----------------------------------第一页--------------------------------------**

注意事项：  
（1）from\_date：自始至终不会变，每次薪酬变化，都是to\_date在变化！  
  
1 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`));  
  
 select emp\_no, count(emp\_no) as t  
 from salaries  
 group by (emp\_no)  
 having t>15  
  
2 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`));  
  
 select employees.emp\_no  
 from employees  
 where employees.emp\_no not in (  
 select dept\_manager.emp\_no  
 from dept\_manager  
 )  
  
3 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`));  
  
 select dept\_emp.dept\_no, dept\_emp.emp\_no, max(salaries.salary)  
 from dept\_emp, salaries  
 where dept\_emp.emp\_no = salaries.emp\_no  
 and dept\_emp.to\_date = '9999-01-01'  
 and salaries.to\_date = '9999-01-01'  
 group by dept\_no  
  
4 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);  
  
select title, count(distinct(emp\_no)) as t  
from titles  
group by title  
having t >= 2  
  
5 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`));  
  
 select emp\_no, salary  
 from salaries as s  
 where s.to\_date = '9999-01-01'  
 order by s.salary desc  
 limit 1,1  
  
 SELECT e.emp\_no, MAX(s.salary) AS salary, e.last\_name, e.first\_name  
 FROM employees AS e INNER JOIN salaries AS s  
 ON e.emp\_no = s.emp\_no  
 WHERE s.to\_date = '9999-01-01'  
 AND s.salary NOT IN (SELECT MAX(salary) FROM salaries WHERE to\_date = '9999-01-01')

**----------------------------------第二页--------------------------------------**

1 题目描述  
 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`));  
  
select a.emp\_no, (b.salary - c.salary) as growthg  
from  
 employees as a  
 inner join salaries as b on a.emp\_no = b.emp\_no and b.to\_date = '9999-01-01'  
 inner join salaries as c on a.emp\_no = c.emp\_no and a.hire\_date = c.from\_date  
order by growth asc  
  
2 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`));  
  
 本题的主要思想是复用salaries表进行比较排名，具体思路如下：  
 (1)、从两张相同的salaries表（分别为s1与s2）进行对比分析，先将两表限定条件设为to\_date = '9999-01-01'，挑选出当前所有员工的薪水情况。  
 (2)、本题的精髓在于 s1.salary <= s2.salary，意思是在输出s1.salary的情况下，有多少个s2.salary大于等于s1.salary，比如当s1.salary=94409时，有3个s2.salary（分别为94692,94409,94409）大于等于它，但由于94409重复，利用COUNT(DISTINCT s2.salary)去重可得工资为94409的rank等于2。其余排名以此类推。  
 (3)、千万不要忘了GROUP BY s1.emp\_no，否则输出的记录只有一条（可能是第一条或者最后一条，根据不同的数据库而定），因为用了合计函数COUNT()  
 (4)、最后先以 s1.salary 逆序排列，再以 s1.emp\_no 顺序排列输出结果  
  
 SELECT  
 s1.emp\_no,  
 s1.salary,  
 COUNT(DISTINCT s2.salary) AS rank  
 FROM  
 salaries AS s1,  
 salaries AS s2  
 WHERE  
 s1.to\_date = '9999-01-01'  
 AND s2.to\_date = '9999-01-01'  
 AND s1.salary <= s2.salary  
 GROUP BY  
 s1.emp\_no  
 ORDER BY  
 s1.salary DESC,  
 s1.emp\_no ASC  
  
3 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'  
 表说明：dept\_emp（员工-部门），dept\_manager（员工-经理）  
 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`));  
  
  
 select  
 dept\_emp.dept\_no,  
 employees.emp\_no,  
 salaries.salary  
 from  
 employees  
 inner join dept\_emp on dept\_emp.emp\_no = employees.emp\_no  
 inner join salaries on salaries.emp\_no = dept\_emp.emp\_no  
 where  
 employees.emp\_no not in (  
 select dept\_manager.emp\_no from dept\_manager where dept\_manager.to\_date='9999-01-01'  
 )  
 and salaries.to\_date='9999-01-01'  
 and dept\_emp.to\_date='9999-01-01'  
  
 4 查找描述信息中包括robot的电影对应的分类名称以及电影数目，而且还需要该分类对应电影数量>=5部  
 select category.name, count(film.film\_id)  
 from film, category, film\_category, (  
 select film\_category.category\_id from film\_category  
 group by film\_category.category\_id  
 having count(film\_category.category\_id) >= 5  
 )as A  
 where description like '%robot%'  
 and category.category\_id = film\_category.category\_id  
 and film.film\_id = film\_category.film\_id  
 and film\_category.category\_id = A.category\_id  
  
5 对于表actor批量插入如下数据  
 CREATE TABLE IF NOT EXISTS actor (  
 actor\_id smallint(5) NOT NULL PRIMARY KEY,  
 first\_name varchar(45) NOT NULL,  
 last\_name varchar(45) NOT NULL,  
 last\_update timestamp NOT NULL DEFAULT (datetime('now','localtime')))  
  
 insert into 表名 values(数据1),(数据2)  
  
6 对于表actor批量插入如下数据, 如果数据已经存在，请忽略，不使用replace操作  
 CREATE TABLE IF NOT EXISTS actor (  
 actor\_id smallint(5) NOT NULL PRIMARY KEY,  
 first\_name varchar(45) NOT NULL,  
 last\_name varchar(45) NOT NULL,  
 last\_update timestamp NOT NULL DEFAULT (datetime('now','localtime')))  
  
 INSERT OR IGNORE INTO actor  
 VALUES(3,'ED','CHASE','2006-02-15 12:34:33');  
  
7 创建一个表，数据源为另外一张表  
 drop table if exists actor\_name;  
 create table actor\_name (  
 first\_name varchar(45) not null,  
 last\_name varchar(45) not null  
 );  
 insert into actor\_name  
 select first\_name,last\_name from actor;  
  
8 针对如下表actor结构创建索引：  
 CREATE TABLE IF NOT EXISTS actor (  
 actor\_id smallint(5) NOT NULL PRIMARY KEY,  
 first\_name varchar(45) NOT NULL,  
 last\_name varchar(45) NOT NULL,  
 last\_update timestamp NOT NULL DEFAULT (datetime('now','localtime')))  
 对first\_name创建唯一索引uniq\_idx\_firstname，对last\_name创建普通索引idx\_lastname  
  
 CREATE UNIQUE INDEX uniq\_idx\_firstname ON actor(first\_name);  
 CREATE INDEX idx\_lastname ON actor(last\_name);  
  
9 创建视图  
 create view t\_v as (  
 select \* from sys\_role  
 )  
  
10 为已有表格新增列  
 ALTER TABLE actor ADD COLUMN create\_date datetime NOT NULL DEFAULT '0000-00-00 00:00:00';  
  
11 更改已有表格的列名  
 ALTER TABLE titles\_test RENAME TO titles\_2017  
  
12 两表求交集  
 SELECT \* FROM employees INTERSECT SELECT \* FROM emp\_v  
  
13 构造一个触发器audit\_log，在向employees\_test表中插入一条数据的时候，触发插入相关的数据到audit中。  
 CREATE TABLE employees\_test(  
 ID INT PRIMARY KEY NOT NULL,  
 NAME TEXT NOT NULL,  
 AGE INT NOT NULL,  
 ADDRESS CHAR(50),  
 SALARY REAL  
 );  
 CREATE TABLE audit(  
 EMP\_no INT NOT NULL,  
 NAME TEXT NOT NULL  
 );  
  
 可以使用 NEW与OLD 关键字访问触发后或触发前的employees\_test表单记录  
  
 CREATE TRIGGER audit\_log AFTER INSERT ON employees\_test  
 BEGIN  
 INSERT INTO audit VALUES (NEW.ID, NEW.NAME);  
 END;

**----------------------------------第三页--------------------------------------**

1 删除emp\_no重复的记录，只保留最小的id对应的记录。  
 CREATE TABLE IF NOT EXISTS titles\_test (  
 id int(11) not null primary key,  
 emp\_no int(11) NOT NULL,  
 title varchar(50) NOT NULL,  
 from\_date date NOT NULL,  
 to\_date date DEFAULT NULL);  
  
 DELETE FROM titles\_test WHERE id NOT IN  
 (SELECT MIN(id) FROM titles\_test GROUP BY emp\_no)  
  
2 将所有to\_date为9999-01-01的全部更新为NULL,且 from\_date更新为2001-01-01。  
 CREATE TABLE IF NOT EXISTS titles\_test (  
 id int(11) not null primary key,  
 emp\_no int(11) NOT NULL,  
 title varchar(50) NOT NULL,  
 from\_date date NOT NULL,  
 to\_date date DEFAULT NULL);  
  
 **update titles\_test  
 set to\_date = null, from\_date = '2001-01-01'  
 where to\_date = '9999-01-01'**  
  
3 将employees表中的所有员工的last\_name和first\_name通过(')连接起来。  
 select last\_name||"'"||first\_name  
 from employees  
  
 select concat(last\_name,"'",first\_name)  
 from employees  
 注意，一般都可以直接'符号'，但是题目的符号特殊，只能用双引号！  
  
4 查找字符串'10,A,B' 中逗号','出现的次数cnt。  
 select (  
 length('10,A,B')-length(  
 **replace**('10,A,B', ',', '')  
 )  
 )as cnt  
  
5 获取Employees中的first\_name，查询按照first\_name最后两个字母，按照升序进行排列  
 select first\_name from employees  
 order by **substr**(first\_name,length(first\_name)-1,2)  
  
6 按照dept\_no进行汇总，属于同一个部门的emp\_no按照逗号进行连接，结果给出dept\_no以及连接出的结果employees  
 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`));  
  
 SELECT dept\_no, **GROUP\_CONCAT**(emp\_no)  
 FROM dept\_emp  
 GROUP BY dept\_no;  
  
7 分页查询employees表，每5行一页，返回第2页的数据  
 select \* from employees  
 limit 5,5  
 第一个参数从0开始，代表从第几个记录开始查询  
  
8 使用含有关键字exists查找未分配具体部门的员工的所有信息。  
 select employees.\*  
 from employees  
 where not exists (  
 select dept\_emp.emp\_no  
 from dept\_emp  
 where dept\_emp.emp\_no = employees.emp\_no  
 )  
  
9 case用法  
 SELECT e.emp\_no, e.first\_name, e.last\_name, b.btype, s.salary,  
 (CASE b.btype  
 WHEN 1 THEN s.salary \* 0.1  
 WHEN 2 THEN s.salary \* 0.2  
 ELSE s.salary \* 0.3 END) AS bonus  
 FROM employees AS e INNER JOIN emp\_bonus AS b ON e.emp\_no = b.emp\_no  
 INNER JOIN salaries AS s ON e.emp\_no = s.emp\_no AND s.to\_date = '9999-01-01'  
  
10 求每个员工自身（包括自身）之前的员工工资总和  
 SELECT s2.emp\_no, s2.salary, SUM(s1.salary) AS running\_total  
 FROM salaries AS s1  
 INNER JOIN salaries AS s2  
 ON s1.emp\_no <= s2.emp\_no  
 WHERE s1.to\_date = "9999-01-01"  
 AND s2.to\_date = "9999-01-01"  
 group by s2.emp\_no  
  
11 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`));  
  
 SELECT s2.emp\_no, s2.from\_date, (s2.salary - s1.salary) AS salary\_growth  
 FROM salaries AS s1, salaries AS s2  
 WHERE s1.emp\_no = s2.emp\_no  
 AND s2.salary - s1.salary > 5000  
 AND strftime('%Y',s2.to\_date) - strftime('%Y',s1.to\_date) = 1  
 ORDER BY salary\_growth DESC;