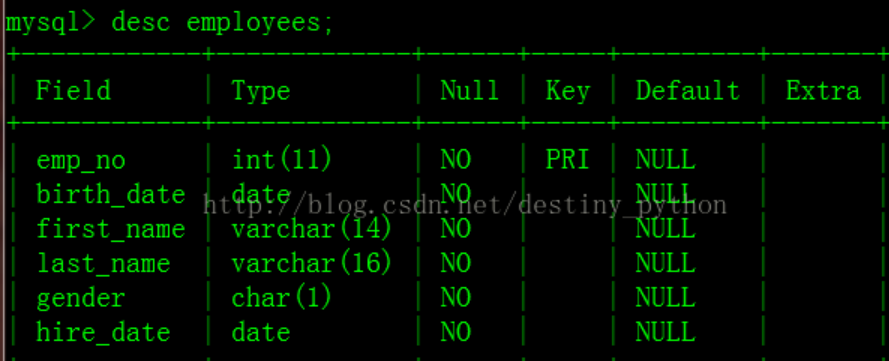
# 牛客网：数据库sql实战



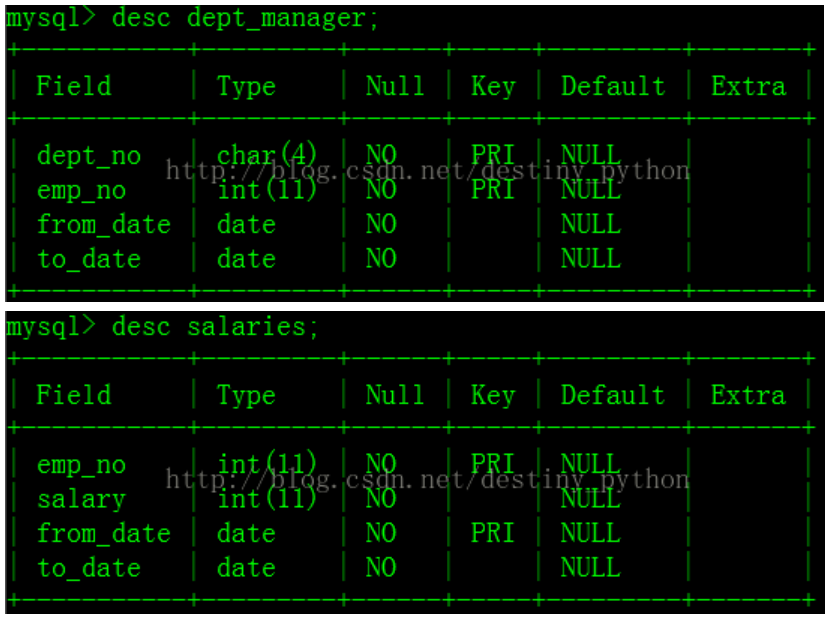
1、查找最晚入职员工的所有信息

select \* from employees where hire\_date = (select max(hire\_date) from employees);

2、查找入职员工时间排名倒数第三的员工所有

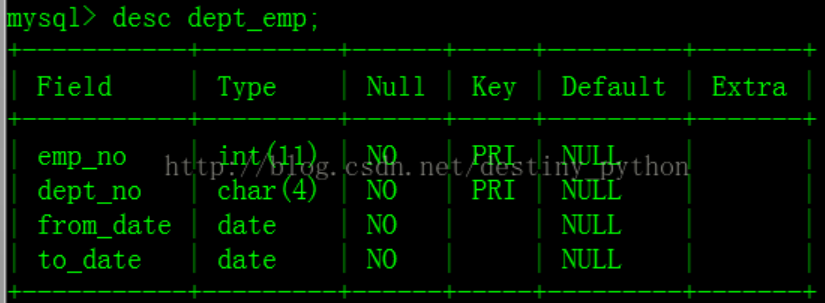
select \* from employees where hire\_date = (select hire\_date from employees order by hire\_date desc limit 2, 1);

limit m, n 表示从第m+1条开始，取n条数据

3、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

select s.\*, d.dept\_no from salaries s, dept\_manager d

where s.emp\_no = d.emp\_no and s.to\_date = '9999-01-01' and d.to\_date = '9999-01-01';

4、查找所有已经分配部门的员工的last\_name和first\_name

内连接：

select e.last\_name, e.first\_name, d.dept\_no from employees e inner join dept\_emp d on e.emp\_no = d.emp\_no;

自然连接：

select last\_name, first\_name, dept\_no from dept\_emp natural join employees;

5、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

select e.last\_name, e.first\_name, d.dept\_no from employees e left join dept\_emp d on d.emp\_no = e.emp\_no;

6、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

select e.emp\_no, s.salary from employees e, salaries s

where e.emp\_no = s.emp\_no and e.hire\_date = s.from\_date order by e.emp\_no desc;

7、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

select emp\_no, count(\*) from salaries group by emp\_no having count(\*) > 15;

8、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

select distinct salary from salaries where to\_date = '9999-01-01' order by salary desc;

9、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'

select d.dept\_no, d.emp\_no, s.salary from salaries s, dept\_manager d

where d.emp\_no = s.emp\_no and d.to\_date = '9999-01-01' and s.to\_date = '9999-01-01';

10、获取所有非manager的员工emp\_no

select e.emp\_no from employees e where e.emp\_no not in (select emp\_no from dept\_manager);

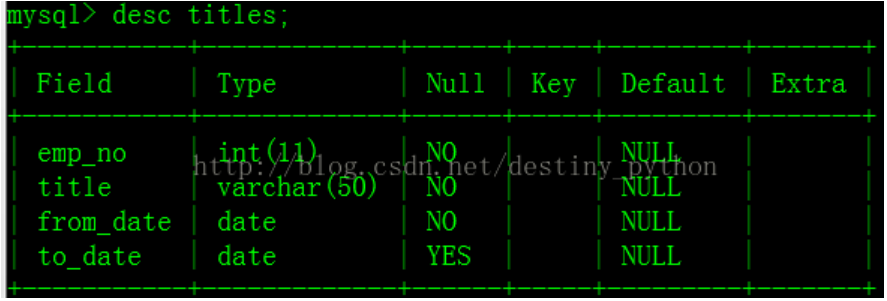
11、获取所有员工当前的manager，如果当前的manager是自己的话结果不显示，当前表示to\_date='9999-01-01'。结果第一列给出当前员工的emp\_no,第二列给出其manager对应的manager\_no。

select d.emp\_no, m.emp\_no from dept\_emp d, dept\_manager m where d.dept\_no = m.dept\_no and d.emp\_no <> m.emp\_no and d.to\_date = '9999-01-01' and m.to\_date = '9999-01-01';

12、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

select d.dept\_no, s.emp\_no, max(s.salary) as salary from dept\_emp d, salaries s

where s.emp\_no = d.emp\_no and d.to\_date = '9999-01-01' and s.to\_date = '9999-01-01' group by d.dept\_no;

13、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。

select title, count(\*) from titles group by title having count(\*) >= 2;

14、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。

select distinct title, count(distinct emp\_no) t from titles group by title having t >= 2;

15、查找employees表所有emp\_no为奇数，且last\_name不为Mary的员工信息，并按照hire\_date逆序排列

select \* from employees where emp\_no%2 = 1 and last\_name <> 'Mary' order by hire\_date desc;

16、统计出当前各个title类型对应的员工当前薪水对应的平均工资。结果给出title以及平均工资avg。

select t.title, avg(s.salary) from titles t, salaries s

where t.emp\_no = s.emp\_no and t.to\_date = '9999-01-01' and s.to\_date = '9999-01-01' group by t.title;

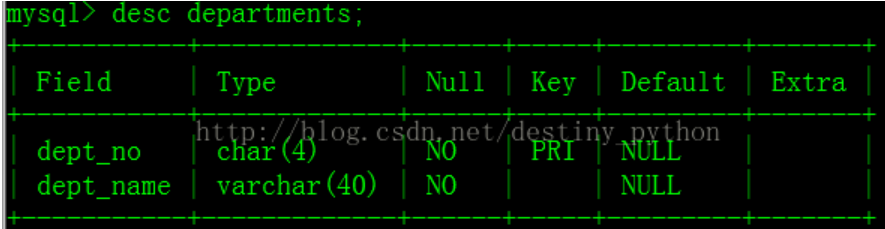
17、获取当前（to\_date='9999-01-01'）薪水第二多的员工的emp\_no以及其对应的薪水salary

select emp\_no, salary from salaries where to\_date = '9999-01-01' order by salary desc limit 1,1;

18、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

select e.emp\_no, max(s.salary) as salary, e.last\_name, e.first\_name from employees e, salaries s

where e.emp\_no = s.emp\_no and s.to\_date = '9999-01-01' and s.salary != (select max(salary) from salaries where to\_date = '9999-01-01');

19、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

select e.last\_name, e.first\_name, dm.dept\_name from employees e left join dept\_emp d on e.emp\_no = d.emp\_no left join departments dm on d.dept\_no = dm.dept\_no;

20、查找员工编号emp\_now为10001其自入职以来的薪水salary涨幅值growth

select (select salary from salaries where emp\_no = 10001 order by to\_date desc limit 1) -

(select salary from salaries where emp\_no = 10001 order by to\_date limit 1) growth;

21、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\_noy以及其对应的薪水涨幅growth，并按照growth进行升序

select sta.emp\_no,(cur.salary-sta.salary) growth from

(select e.emp\_no, s.salary from employees e, salaries s where s.emp\_no = e.emp\_no and s.to\_date = '9999-01-01') cur, #查找目前的工资

(select e.emp\_no, s.salary from employees e, salaries s where s.from\_date = e.hire\_date and s.emp\_no = e.emp\_no) sta #查找入职时候的工资

where cur.emp\_no = sta.emp\_no order by growth;

22、统计各个部门对应员工涨幅的次数总和，给出部门编码dept\_no、部门名称dept\_name以及次数sum

select dm.dept\_no, dm.dept\_name, count(s.salary) sum from

salaries s inner join dept\_emp d on s.emp\_no = d.emp\_no

inner join departments dm on d.dept\_no = dm.dept\_no

group by d.dept\_no;

23、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

select s1.emp\_no, s1.salary, count(distinct s2.salary) rank

from salaries s1, salaries 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;

24、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'

select d.dept\_no, e.emp\_no, s.salary

from employees e inner join salaries s on e.emp\_no = s.emp\_no and s.to\_date = '9999-01-01'

inner join dept\_emp d on s.emp\_no = d.emp\_no

where d.emp\_no not in (select emp\_no from dept\_manager);

25、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

select s1.emp\_no emp\_no, s2.emp\_no manager\_no, s1.salary emp\_salary, s2.salary manager\_salary

from (select d.emp\_no, d.dept\_no, s.salary from dept\_emp d, salaries s

where d.emp\_no = s.emp\_no and s.to\_date = '9999-01-01' and

d.emp\_no not in (select emp\_no from dept\_manager)) s1,

(select dm.emp\_no, dm.dept\_no, s.salary from dept\_manager dm, salaries s

where dm.emp\_no = s.emp\_no and s.to\_date = '9999-01-01') s2

where s1.salary > s2.salary and s1.dept\_no=s2.dept\_no;

26、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

select dm.dept\_no, dm.dept\_name, t.title, count(t.title) count

from dept\_emp d inner join titles t on d.emp\_no = t.emp\_no and d.to\_date = '9999-01-01' and t.to\_date = '9999-01-01'

inner join departments dm on d.dept\_no = dm.dept\_no

group by d.dept\_no, t.title;

27、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)

select s2.emp\_no, s1.from\_date, (s1.salary - s2.salary) salary\_growth

from salaries s2, salaries s1

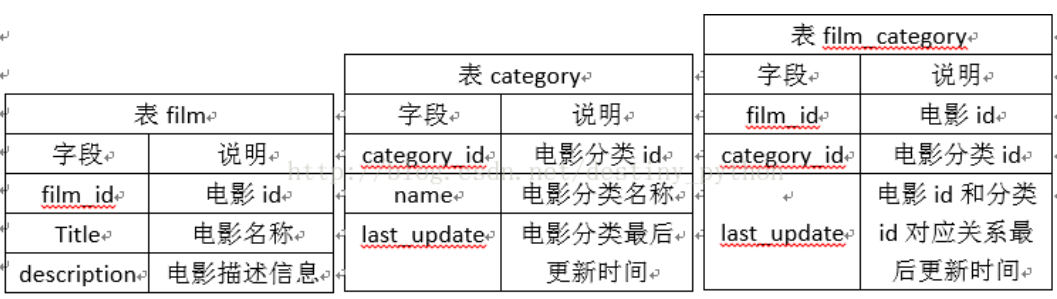
where s2.emp\_no = s1.emp\_no

and salary\_growth > 5000

and ((strftime("%Y", s1.to\_date) -strftime("%Y", s2.to\_date)) = 1

or (strftime("%y", s1.from\_date) -strftime("%y", s2.from\_date)) = 1)

order by salary\_growth desc;

28、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>=2部

select c.name, count(f.film\_id) count

from (film f inner join film\_category fc on f.film\_id = fc.film\_id)

inner join category c on c.category\_id = fc.category\_id

where f.description like '%robot%'

group by c.name

having count>=2;

29、使用join查询方式找出没有分类的电影id以及名称

select f.film\_id, f.title from

film f left join film\_category fc on f.film\_id = fc.film\_id

where fc.category\_id is null;

30、使用子查询的方式找出属于Action分类的所有电影对应的title,description

select f.title, f.description from

(select fc.film\_id from film\_category fc, category c

where fc.category\_id = c.category\_id and c.name = 'Action') a,

film f where f.film\_id = a.film\_id;

31、获取select \* from employees对应的执行计划

explain select \* from employees;

32、将employees表的所有员工的last\_name和first\_name拼接起来作为Name，中间以一个空格区分

select last\_name||' '||first\_name name from employees;

33、创建一个actor表，包含如下列信息

create table if not exists actor

(actor\_id smallint(5) not null,

first\_name varchar(45) not null,

last\_name varchar(45) not null,

last\_update timestamp not null default (datetime('now','localtime'))

PRIMARY KEY(actor\_id));

34、对于表actor批量插入如下数据

insert into actor values(1,'PENELOPE','GUINESS','2006-02-15 12:34:33'),

(2,'NICK','WAHLBERG','2006-02-15 12:34:33');

35、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

insert OR ignore into actor values(3,'ED','CHASE','2006-02-15 12:34:33');

#备注：mysql中需要删除or

36、创建一个actor\_name表，将actor表中的所有first\_name以及last\_name导入改表。

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;

37、针对表actor中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);

38、针对actor表创建视图actor\_name\_view，只包含first\_name以及last\_name两列，并对这两列重新命名，first\_name为first\_name\_v，last\_name修改为last\_name\_v：

create view actor\_name\_view as select first\_name first\_name\_v, last\_name last\_name\_v from actor;

39、针对salaries表emp\_no字段创建索引idx\_emp\_no，查询emp\_no为10005, 使用强制索引。

select \* from salaries indexed by idx\_emp\_no where emp\_no = 10005;

#在mysql中用force index select \* from salaries force index idx\_emp\_no where emp\_no = 10005;

40、针对actor表，现在在last\_update后面新增加一列名字为create\_date,类型为datetime, NOT NULL，默认值为'0000-00-00 00:00:00'

alter table actor add create\_date datetime not null default '0000-00-00 00:00:00';

41、构造一个触发器audit\_log，在向employees表中插入一条数据的时候，触发插入相关的数据到audit中。

create trigger audit\_log after

insert on employees\_test

begin

insert into audit values(new.id,new.name);

end;

42、删除emp\_no重复的记录，只保留最小的id对应的记录。

思路：第一步：按emp\_no分组，选出每组最小的id；

第二步：删除数据，id不在上述id内。

delete from titles\_test where id not in

(select min(id) from titles\_test group by emp\_no);

43、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。

update titles\_test set to\_date = null,from\_date = '2001-01-01';

44、将id=5以及emp\_no=10001的行数据替换成id=5以及emp\_no=10005,其他数据保持不变，使用replace实现。

update titles\_test set emp\_no=replace(emp\_no,10001,10005) where id=5;

45、将titles\_test表名修改为titles\_2017。

alter table titles\_test rename to titles\_2017;

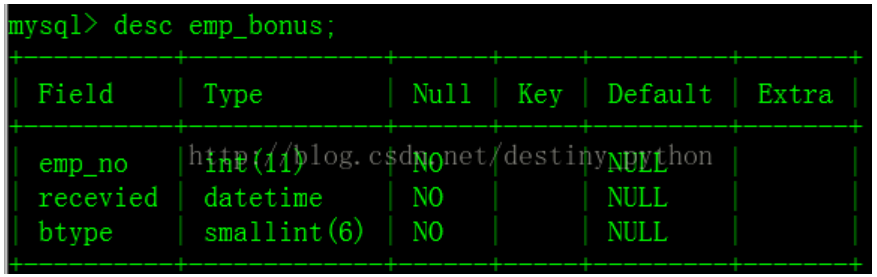
46、在audit表上创建外键约束，其emp\_no对应employees\_test表的主键id。

ALTER TABLE audit ADD FOREIGN KEY (emp\_no) REFERENCES employees\_test (id);

47、create view emp\_v asselect \* from employees where emp\_no >10005;

如何获取emp\_v和employees有相同的数据？

select e.\* from employees e, emp\_v ev where e.emp\_no = ev.emp\_no;

48、将所有获取奖金的员工当前的薪水增加10%。

update salaries set salary = salary\*1.1 where emp\_no in

(select s.emp\_no from salaries s, emp\_bonus e

where s.emp\_no=e.emp\_no and s.to\_date='9999-01-01');

49、针对库中的所有表生成selectcount(\*)对应的SQL语句

关键点：在 SQLite 系统表 sqlite\_master 中可以获得所有表的索引，其中字段 name 是所有表的名字，而且对于自己创建的表而言，字段 type 永远是 'table'

SELECT "select count(\*) from " || name || ";" cnts

FROM sqlite\_master WHERE type = 'table';

50、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;

51、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',',',''))) cnt;

52、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,-2);

53、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

本题要用到SQLite的聚合函数group\_concat(X,Y)，其中X是要连接的字段，Y是连接时用的符号，可省略，默认为逗号。

select dept\_no, group\_concat(emp\_no) employees from dept\_emp group by dept\_no;

54、查找排除当前最大、最小salary之后的员工的平均工资avg\_salary。

SELECT AVG(salary) avg\_salary FROM salaries

WHERE to\_date = '9999-01-01'

AND salary NOT IN (SELECT MAX(salary) FROM salaries WHERE to\_date = '9999-01-01')

AND salary NOT IN (SELECT MIN(salary) FROM salaries WHERE to\_date = '9999-01-01');

55、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;

56、获取所有员工的emp\_no、部门编号dept\_no以及对应的bonus类型btype和recevied，没有分配具体的员工不显示

select d.emp\_no, d.dept\_no, eb.btype, eb.recevied from

employees e inner join dept\_emp d on e.emp\_no = d.emp\_no

left join emp\_bonus eb on e.emp\_no=eb.emp\_no;

57、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 \* FROM employees WHERE NOT EXISTS

(SELECT emp\_no FROM dept\_emp WHERE emp\_no = employees.emp\_no);

58、存在如下的视图：

create view emp\_v as select \* from employees where emp\_no >10005;

获取employees中的行数据，且这些行也存在于emp\_v中。注意不能使用intersect关键字。

select \* from employees where emp\_no>10005;

59、给出emp\_no、first\_name、last\_name、奖金类型btype、对应的当前薪水情况salary以及奖金金额bonus。 bonus类型btype为1其奖金为薪水salary的10%，btype为2其奖金为薪水的20%，其他类型均为薪水的30%。 当前薪水表示to\_date='9999-01-01'

select e.emp\_no, e.first\_name, e.last\_name, eb.btype, s.salary,

(case eb.btype

when 1 then s.salary\*0.1

when 2 then s.salary\*0.2

else s.salary\*0.3 end) bonus

from employees e inner join salaries s on e.emp\_no = s.emp\_no

inner join emp\_bonus eb on e.emp\_no = eb.emp\_no

and s.to\_date = '9999-01-01';

60、按照salary的累计和running\_total，其中running\_total为前面员工的salary累计和，其他以此类推。 具体结果如下Demo展示

select s1.emp\_no, s1.salary,

(select sum(s2.salary) from salaries s2

where s2.emp\_no <= s1.emp\_no and s2.to\_date='9999-01-01') running\_total

from salaries s1 where s1.to\_date='9999-01-01';

61、对于employees表中，给出按first\_name升序排列的奇数行的first\_name

select e1.first\_name from employees e1

where (select count(\*) from employees e2

where e1.first\_name>=e2.first\_name)%2=1;