Chapter 6

- 牛客：批量插入数据

```
-- 方法一:
insert into actor (actor_id, first_name, last_name, last_update)
values
(1, 'PENELOPE', 'GUINESS', '2006-02-15 12:34:33'),
(2, 'NICK', 'WAHLBERG','2006-02-15 12:34:33');
-- 方法二:
insert into actor values (1,'PENELOPE','GUINESS','2006-02-15   12:34:33'),
(2,'NICK','WAHLBERG','2006-02-15 12:34:33');
```

- 牛客：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

```
-- 方法一:
select salary from salaries where to_date = '9999-01-01' group by salary  order by
salary desc;
-- 方法二:
select distinct salary from salaries where to_date='9999-01-01' order by salary
desc;
```

- 牛客：查找最晚入职员工的所有信息

```
-- 方法一:
select emp_no, birth_date, first_name, last_name, gender, hire_date from employees
order by hire_date desc limit 1 offset 0;

-- 方法二:
select * from employees order by hire_date desc limit 1;

-- 方法三:
select * from employees order by hire_date desc limit 0,1;

-- 方法四:
select * from employees where hire_date in (select max(hire_date) from employees);

-- 方法五:
select * from employees where hire_date = (select max(hire_date) from employees);
```

- 牛客：查找入职员工时间排名倒数第三的员工所有信息

```
-- 方法一:
select emp_no, birth_date, first_name, last_name, gender , hire_date from employees
order by hire_date desc limit 1 offset 2;
-- 方法二:
select * from employees order by hire_date desc limit 1 offset 2;
-- 方法三:
select * from employees order by hire_date desc limit 2,1;
```

- 牛客：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(emp_no) as t from salaries group by emp_no having
count(emp_no) > 15; have in
```

- 牛客：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 dept_manager as d , salaries as s
where d.emp_no = s.emp_no and s.to_date='9999-01-01' and d.to_date='9999-01-01'
```

- 牛客：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(title) as t from titles group by title having  t >=2;
```

- leetcode: duplicate-emails

```
-- 方法一:
select Email from Person
where Id in (select Id from Person group by Email having count(Email)>1);
-- 方法二:
select Email from Person group by Email having count(Email) > 1;
--方法三:
select  distinct Email from person where Email='a@b.com';
```

- leetcode: big-countries

```
-- 方法一:
select name,population,area from World
where area > 3000000
union
select name,population,area from World
where population > 25000000;
-- 方法二:
select name,population,area from World where area>3000000  or population >25000000;
```

- leetcode: nth-highest-salary

```
CREATE FUNCTION getNthHighestSalary(N INT) RETURNS INT
BEGIN
  set N = N-1;
  RETURN (
      # Write your MySQL query statement below.
      select distinct salary  from Employee order by salary desc limit 1 offset N
  );
END
```

Chapter 7

- 牛客：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
```

Chapter 8

- 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

```
-- 方法一:
select e.emp_no, s.salary
from employees as e left join salaries as s
on e.emp_no = s.emp_no
and
e.hire_date = s.from_date
order by e.emp_no desc;

-- 方法二:
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;
```

- 牛客：针对库中的所有表生成select count(*) from tableName 对应的SQL语句

```
-- information_schema 数据库
select concat('select count(*) from ', s.TABLE_NAME ) as cnts from TABLES as s;

--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 || ";" AS cnts
FROM sqlite_master WHERE type = 'table';
```

- 牛客：获取所有非manager的员工emp_no

```
select emp_no from employees where emp_no not in (select emp_no from dept_manager);
```

- 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'

```
select
    distinct de.emp_no as emp_no, dm.emp_no as manager_no
from
    dept_emp as de , dept_manager as dm
where de.emp_no !=dm.emp_no
and de.dept_no = dm.dept_no
and de.to_date='9999-01-01'
and dm.to_date='9999-01-01';
```

Chapter 9

- leetcode: rank-scores

```
# Write your MySQL query statement below
select
    Scores.Score as Score, Rank
from Scores
left join
(select Score, @rank:=@rank + 1 as Rank from
    (select distinct Score from Scores order by Score desc) t,
    (select @rank:=0) r
)
as score_rank
on Scores.Score = score_rank.Score
order by Score desc
```

- leetcode: exchange-seats

```
-- least () : 返回最小参数  %: 取余
select least((id +(id%2)*2-1),(select count(*) from seat )) as id ,student from
seat order by id asc;
```

Chapter 10

Chapter 12

- 牛客：针对actor表创建视图actor_name_view

```
create view if not exists actor_name_view  as select first_name as first_name_v ,
last_name as last_name_v from actor;
```