Database Mysql summary

一、有关 table 的操作

1.导入已有的 sql 类型的文件

- Create database hr
- Use hr
- Source directory of hr

2.查看当前 DB 的版本，状态，tables

- Show tables
- Select version（）
- Status

3. 创建当前数据库不存在的 table

```
Create table countries（
    Country_id char(2),
    Country_name varchar(40),
    Region_id decimal(10,0)
);
```

4. 创建当前数据库中已经存在的 table

```
Create table if not exists countries（
    Country_id char(2),
    Country_name varchar(40),
    Region_id decimal(10,0)
);
```

5. 创建一个与已经有的 table 类型相似的 table，但是里面的列没有复制

```
Create table if not exists copy_countries
Like countries;
```

6. 创建一个复制已有 table 并且复制所有列的 table，相当于除了表名，其他都是一样。

```
Create table if not exists copy_countries
As select * from countries;
```

7. 创建一个 table,check table 中某列值的取值范围

```
Create table if not exists jobs（
    Job_id char(10) not null,
    Job_title char(35) not null,
    Min_salary decimal(6,0),
    Max_salary decimal(6,0),
    Check(max_salary <= 250000)
```

```
);
```

8. 创建一个 table,check table 中某列值取几个离散的值

```
Create table if not exists countries（
    Country_id char(2),
    Country_name varchar(40),
    Check（country_name in（‘Italy’，‘India’，‘china’））），
    Region_id decimal(10,0)
);
```

9. 创建一个 table,check table 中某列值的格式按规定

```
Create table if not exists job_history(
    Employee_id decimal(6,0) not null,
    Start_date date not null,
    End_date date not null,
    Check(end_date like ‘--/--/--’),
    Department_id decimal(4,0) not null
);
```

10. 创建一个 table,确保某类没有重复数据

```
Create table if not exists countries（
    Country_id char(2),
    Country_name varchar(40),
    Region_id decimal(10,0)，
    Unique（country_id）
);
```

11. 创建一个 table,设置主键值

```
Create table if not exists countries（
    Country_id char(2) not null unique primary key，
    Country_name varchar(40),
    Region_id decimal(10,0)
);
```

12. 创建一个 table，设置额外的说明和主键

```
Create table if not exists countries（
    Country_id char(2) not null unique auto_increment primary key,
    Country_name varchar(40) not null,
    Region_id decimal(10,0) not null
);
```

13. 创建一个 table，引用外部 table 的列，并禁止 delete 和 update 列

```
Create table if not exists employees（
    Employee_id decimal(6,0) not null primary key,
```

```
    First_name char(20) default null,
    Foreign key(job_id) reference jobs(job_id),
    On delete no action,
    On update no action
) engine = InnoDB;
```

MySQL 外键约束 On Delete 和 On Update 的使用

On Delete 和 On Update 都有 Restrict，No Action, Cascade,Set Null 属性。

ON DELETE

restrict(约束):当在父表（即外键的来源表）中删除对应记录时，首先检查该记录是否有对应外键，如果有则不允许删除。

no action:意思同 restrict.即如果存在从数据，不允许删除主数据。

cascade(级联):当在父表（即外键的来源表）中删除对应记录时，首先检查该记录是否有对应外键，如果有则也删除外键在子表（即包含外键的表）中的记录。

set null:当在父表（即外键的来源表）中删除对应记录时，首先检查该记录是否有对应外键，如果有则设置子表中该外键值为 null（不过这就要求该外键允许取 null）

ON UPDATE

restrict(约束):当在父表（即外键的来源表）中更新对应记录时，首先检查该记录是否有对应外键，如果有则不允许更新。

no action:意思同 restrict.

cascade(级联):当在父表（即外键的来源表）中更新对应记录时，首先检查该记录是否有对应外键，如果有则也更新外键在子表（即包含外键的表）中的记录。

set null:当在父表（即外键的来源表）中更新对应记录时，首先检查该记录是否有对应外键，如果有则设置子表中该外键值为 null（不过这就要求该外键允许取 null）。

注：NO ACTION 和 RESTRICT 的区别：只有在及个别的情况下会导致区别，前者是在其他约束的动作之后执行，后者具有最高的优先权执行。

二、有关表数据项的操作

1.插入一个数据元素到 table 中

Insert into countries value（ ‘ ’， ‘ ’， ‘ ’）；

之后查看可以 select * from countries；

2.对数据元素的部分列进行添加

Insert into countries（country_id, country_name）values(‘c1’,’India’);

3.一次增加多个数据元素

Insert into countries value（‘c1 ’， ‘India ’，1001），（‘c02’， ‘usa’，1002），（‘c03’， ‘uk’，1003）；

4.将一个 table 中的数据元素添加到另外一个类似的 table 中

Insert into countries

Select * from country_new;

5.改变 table 中某个数据项的值

Update employees set email=‘N/A’；

SELECT * FROM EMPLOYEES LIMIT 2; //只打印出两行

6.改变 table 中满足某条件的数据元素对应的某键的值

Update employees set email = ‘N/A’,

Where department_id = 110;

7.改变 table 中某数据项的值，当另一数据项满足另一个 table 中的 sql 条件时

Update employees set email=‘N/A’

Where department_id=(select department_id from departments

Where department_name = ‘accounting’);

8.依据条件改变相应键的值

Update employees set salary = case department_id

When 40 then salary+(salary*.25)

When 90 then salary+(salary*.15)

When 110 then salary+(salary*.10)

Else salary

End

Where department_id in (40,50,60,70,80,90,110);

9.修改 table 的名字

Alter table countries rename country_new;

10.为 table 增加一个数据项

Alter table countries

Add region_id int;

11.将一个数据项添加到 table 中的首位置

Alter table locations

Add id int first；

12.修改 table 中某一元素项的类型 type

Alter table locations

Modify country_id int;

13.drop 丢掉 table 中某一数据项

Alter table locations

Drop city；

14.修改数据项的名称，不改变其位置

Alter table locations

drop state_province

Add state char(50)

After city;

三、有关表数据元素的一些增删改查操作

1.使用别名

Select first_name ‘First_name’, last_name ‘Last_name’ from employees;

2.查询时去掉重复的值

Select distinct department_id from employees;

3.查询时，按照某一列的升/降排列

Select * from employees order by first_name desc;(默认升序 asc）

4.查询表中数据元素的和，最大，最小，平均值，长度

Select sum(salary),max(salary),min(salary),avg(salary),length(first_

name) from employees;

5.查询计算表中有多少列

- select count(*) from employees;

6.将表中某列改为大写，或者小写

Select upper （ first_name ） ,lower(last_name) from employees;

7.查询表中列时，获取某列的部分字符串

Select substring（first_name,1,3） from employees;

8.用 select 语句计算表达式的值

Select 171*23+23 result；

9.查询表时，将多列作为一个组合列输出

Select concat(first_name,’’,last_name) ‘employee_name’ from employees;

10.查询表时，去掉某列前后的空白部分再输出

Select trim(first_name) from employees;

11.查询表时，check 某列是否满足正则表达式

Select * from employees where first_name regexp ‘[0-9]’;

12.查询表时，指定输出前 10 行内容

Select * from employees limit 10;

13.查询表时，读取列精确到小数点后 n 位

Select first_name,last_name,round(salary/12,n) ‘monthly salary’ from employees;

14.查询表时，判断不在某一取值区间

Select first_name,last_name,salary from employees where salary not between 10000 and 15000;

15.查询表时，数据项取离散的值

Select * from employees where department_id in(30,100) order by department_id asc;

16.查询表时，输出指定的年份的信息。

Select * from employees where year（hire_date） like '%87';

17.查询某数据项包含某一指定的字符

Select first_name from employees where fist_name like '%b%b' and first_name like '%c%';

18.查询时，指定字符串的长度为 n

Select * from employees where last_name like '------';

19.组合查询计数

Select job_id,count(*) from employees group by job_id;

20.查询时，列的值不包含某一字符串

Select job_id,avg(salary) from employees where job_id <> 'it_prog' group by job_id;

21.输出超过一定数量的数据项

Select * from employees group by department_id having count(*)>10;

Where 和 having 的区别：

Where 中不能使用聚组函数，必须位于 group by 之前;

Having 中可以使用聚组函数，必须位于 group by 之后。