笔记本：	任务七		
创建时间：	2018/5/29 星期二 下午 5:42	**更新时间：**	2018/5/29 星期二 下午 5:56
作者：	18291893776@139.com		

--MySQL数据库基本操作：

```
--使用MySQL命令连接MySQL数据库
root@may-virtual-machine:/home/may# mysql -h localhost -u root -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.22-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

--省略-h，默认localhost连接MySQL数据库
may@may-virtual-machine:~$ sudo su
[sudo] may 的密码：
root@may-virtual-machine:/home/may# mysql -u root -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.7.22-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

数据库操作
--============================================================
-- 查看当前MySQL下的所有数据库

mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.18 sec)

-- 创建一个mydb2的数据库
mysql> create database mydb2;
Query OK, 1 row affected (0.08 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydb2              |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

-- 删除数据库mydb2
mysql> drop database mydb2
    -> ;
Query OK, 0 rows affected (0.68 sec)

mysql> shoe databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'shoe databases
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

-- 创建一个mydb数据库
mysql> create database mydb;
Query OK, 1 row affected (0.06 sec)

-- 再次创建mydb数据库会报错
mysql> create database mydb;
ERROR 1007 (HY000): Can't create database 'mydb'; database exists

-- 尝试创建mydb数据库（若已存在则会报一个警告，不会报Error错误）
mysql> create database if not exists mydb;
Query OK, 1 row affected, 1 warning (0.00 sec)

-- 查看mydb的建库语句
mysql> show create database mydb;
+----------+------------------------------------------------------------------+
| Database | Create Database                                                  |
+----------+------------------------------------------------------------------+
| mydb     | CREATE DATABASE `mydb` /*!40100 DEFAULT CHARACTER SET latin1 */ |
+----------+------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> show create database mydb\G
*************************** 1. row ***************************
       Database: mydb
Create Database: CREATE DATABASE `mydb` /*!40100 DEFAULT CHARACTER SET latin1 */
1 row in set (0.00 sec)

-- 查看当前所在数据库位置：NULL表示没有在任何数据库中
mysql> select database();
+------------+
| database() |
+------------+
| NULL       |
+------------+
1 row in set (0.00 sec)

-- 选择进入mydb数据库
mysql> use mydb;
Database changed

-- 查看当前所在数据库的位置
mysql> select database();
+------------+
| database() |
+------------+
| mydb       |
+------------+
1 row in set (0.00 sec)

-- 数据表操作
--============================================
-- 查看当前数据库中的所有表
mysql> show tables;
Empty set (0.22 sec)

-- 创建一个uu表，内有三个字段id，name和age
mysql> create table uu(id int, name varchar(16),age int);
Query OK, 0 rows affected (1.17 sec)


-- 创建一个tt表，内有三个字段id，name和age
mysql> create table tt(
    -> id int,
    -> name varchar(16),
    -> age int
    -> );
Query OK, 0 rows affected (0.18 sec)


-- 查看当前库中有两个表
mysql> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| tt             |
| uu             |
+----------------+
2 rows in set (0.06 sec)

-- 查看uu表的表结构
mysql> desc uu;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | YES  |     | NULL    |       |
| name  | varchar(16) | YES  |     | NULL    |       |
| age   | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (2.68 sec)

-- 查看uu表的建表语句
```

```
mysql> show create table uu\G
*************************** 1. row ***************************
       Table: uu
Create Table: CREATE TABLE `uu` (
  `id` int(11) DEFAULT NULL,
  `name` varchar(16) DEFAULT NULL,
  `age` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.03 sec)


-- 删除uu表
mysql> drop table uu;
Query OK, 0 rows affected (0.37 sec)


--查看库中的表
mysql> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| tt             |
+----------------+
1 row in set (0.06 sec)
```

更改表名称：

```
    ALTER TABLE 旧表名 RENAME AS 新表名
```

更改AUTO_INCREMENT初始值:

```
    ALTER TABLE 表名称 AUTO_INCREMENT=1
```

更改表类型：

```
   ALTER TABLE 表名称 ENGINE="InnoDB"
```

```
mysql> show create table stu;
+-------+------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------+
| Table | Create
Table
+-------+------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------+
| stu   | CREATE TABLE `stu` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(8) NOT NULL,
  `age` tinyint(3) unsigned DEFAULT NULL,
  `sex` enum('m','w') NOT NULL DEFAULT 'm',
  `classid` char(8) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`)
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=latin1 |
+-------+------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> alter table stu ENGINE='MyISAM';
Query OK, 12 rows affected (1.69 sec)
Records: 12  Duplicates: 0  Warnings: 0

mysql> show create table stu;
+-------+------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------+
| Table | Create
Table
+-------+------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------+
| stu   | CREATE TABLE `stu` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(8) NOT NULL,
  `age` tinyint(3) unsigned DEFAULT NULL,
  `sex` enum('m','w') NOT NULL DEFAULT 'm',
  `classid` char(8) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`)
) ENGINE=MyISAM AUTO_INCREMENT=13 DEFAULT CHARSET=latin1 |
+-------+------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------+
1 row in set (0.04 sec)

mysql>
```

MySQL数据库中的表类型一般常用两种：MyISAM和InnoDB

区别：MyISAM类型的数据文件有三个frm(结构)、MYD（数据）、MYI（索引）

MyISAM类型中的表数据增 删 改速度快，不支持事务，没有InnoDB安全。

InnoDB类型的数据文件只有一个 .frm

InnoDB类型的表数据增 删 改速度没有MyISAM的快，但支持事务，相对安全。

```
--数据操作
mysql> desc uu;
ERROR 1146 (42S02): Table 'mydb.uu' doesn't exist
mysql> create table uu(
    -> id int(11),
    -> name varchar(16),
    -> age int(11)
    -> );
Query OK, 0 rows affected (0.17 sec)

--数据操作
--添加数据
-- 格式：insert into 表名[(字段列表)] values(值列表)[,(值列表)...]
--==============================================================
mysql> desc uu;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | YES  |     | NULL    |       |
| name  | varchar(16) | YES  |     | NULL    |       |
| age   | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

-- 添加一个数据，给定所有字段，所有的值
mysql> insert into uu(id,name,age) values(1,'zhangsan',20)
    -> ;
Query OK, 1 row affected (0.15 sec)

mysql> insert into uu(id,name,age) values(2,'lisi',22);
Query OK, 1 row affected (0.10 sec)

-- 不指定字段，添加值，值按默认顺序写
mysql> insert into uu values(3,'wangwu',25);
Query OK, 1 row affected (0.04 sec)

-- 批量添加值
mysql> insert into uu values(4,'lisi',21),
    -> (5,'xiaoli',22),
    -> (6,'xiaozhang',19);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0


--查看数据
-- 查看uu表的所有数据
mysql> select * from uu;
+------+-----------+------+
| id   | name      | age  |
+------+-----------+------+
|    1 | zhangsan  |   20 |
|    2 | lisi      |   22 |
|    3 | wangwu    |   25 |
|    4 | lisi      |   21 |
|    5 | xiaoli    |   22 |
|    6 | xiaozhang |   19 |
+------+-----------+------+
6 rows in set (0.00 sec)


-- 查看uu表中的name和age字段的所有数据
mysql> select name,age from uu;
+-----------+------+
| name      | age  |
+-----------+------+
| zhangsan  |   20 |
| lisi      |   22 |
| wangwu    |   25 |
| lisi      |   21 |
| xiaoli    |   22 |
| xiaozhang |   19 |
+-----------+------+
6 rows in set (0.02 sec)


mysql> select * from stu;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
|  4 | li       |   28 | m   | python02 |
|  5 | zhang    |   27 | w   | python01 |
|  6 | zhao     |   27 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
|  8 | uu02     |   26 | m   | python02 |
|  9 | uu03     | NULL | m   | NULL     |
| 10 | uu04     |   26 | m   | python02 |
| 11 | uu06     | NULL | m   | NULL     |
| 12 | UU08     |   24 | m   | python02 |
+----+----------+------+-----+----------+
12 rows in set (0.01 sec)

mysql> select id,name,age from stu;
+----+----------+------+
| id | name     | age  |
+----+----------+------+
|  1 | zhangsan |   20 |
|  2 | lisi     |   22 |
|  3 | wangwu   |   25 |
|  4 | li       |   28 |
|  5 | zhang    |   27 |
|  6 | zhao     |   27 |
|  7 | uu01     |   18 |
|  8 | uu02     |   26 |
|  9 | uu03     | NULL |
| 10 | uu04     |   26 |
| 11 | uu06     | NULL |
| 12 | UU08     |   24 |
+----+----------+------+
12 rows in set (0.00 sec)

mysql> select id,name as username,age from stu;
+----+----------+------+
| id | username | age  |
+----+----------+------+
|  1 | zhangsan |   20 |
|  2 | lisi     |   22 |
|  3 | wangwu   |   25 |
|  4 | li       |   28 |
|  5 | zhang    |   27 |
|  6 | zhao     |   27 |
|  7 | uu01     |   18 |
|  8 | uu02     |   26 |
|  9 | uu03     | NULL |
| 10 | uu04     |   26 |
| 11 | uu06     | NULL |
| 12 | UU08     |   24 |
+----+----------+------+
12 rows in set (0.00 sec)

mysql> select *,age+5 age5 from stu;
+----+----------+------+-----+----------+------+
| id | name     | age  | sex | classid  | age5 |
+----+----------+------+-----+----------+------+
|  1 | zhangsan |   20 | m   | python03 |   25 |
|  2 | lisi     |   22 | m   | python02 |   27 |
|  3 | wangwu   |   25 | w   | python03 |   30 |
|  4 | li       |   28 | m   | python02 |   33 |
|  5 | zhang    |   27 | w   | python01 |   32 |
|  6 | zhao     |   27 | m   | python03 |   32 |
|  7 | uu01     |   18 | m   | python03 |   23 |
|  8 | uu02     |   26 | m   | python02 |   31 |
|  9 | uu03     | NULL | m   | NULL     | NULL |
| 10 | uu04     |   26 | m   | python02 |   31 |
| 11 | uu06     | NULL | m   | NULL     | NULL |
| 12 | UU08     |   24 | m   | python02 |   29 |
+----+----------+------+-----+----------+------+
12 rows in set (0.00 sec)

mysql> select *,"xi'an" as city from stu
    -> ;
+----+----------+------+-----+----------+-------+
| id | name     | age  | sex | classid  | city  |
+----+----------+------+-----+----------+-------+
|  1 | zhangsan |   20 | m   | python03 | xi'an |
|  2 | lisi     |   22 | m   | python02 | xi'an |
|  3 | wangwu   |   25 | w   | python03 | xi'an |
|  4 | li       |   28 | m   | python02 | xi'an |
|  5 | zhang    |   27 | w   | python01 | xi'an |
|  6 | zhao     |   27 | m   | python03 | xi'an |
|  7 | uu01     |   18 | m   | python03 | xi'an |
|  8 | uu02     |   26 | m   | python02 | xi'an |
|  9 | uu03     | NULL | m   | NULL     | xi'an |
| 10 | uu04     |   26 | m   | python02 | xi'an |
| 11 | uu06     | NULL | m   | NULL     | xi'an |
| 12 | UU08     |   24 | m   | python02 | xi'an |
+----+----------+------+-----+----------+-------+
12 rows in set (0.00 sec)

mysql> select *,"xi'an" city from stu;
+----+----------+------+-----+----------+-------+
| id | name     | age  | sex | classid  | city  |
+----+----------+------+-----+----------+-------+
|  1 | zhangsan |   20 | m   | python03 | xi'an |
|  2 | lisi     |   22 | m   | python02 | xi'an |
|  3 | wangwu   |   25 | w   | python03 | xi'an |
|  4 | li       |   28 | m   | python02 | xi'an |
|  5 | zhang    |   27 | w   | python01 | xi'an |
|  6 | zhao     |   27 | m   | python03 | xi'an |
|  7 | uu01     |   18 | m   | python03 | xi'an |
|  8 | uu02     |   26 | m   | python02 | xi'an |
|  9 | uu03     | NULL | m   | NULL     | xi'an |
| 10 | uu04     |   26 | m   | python02 | xi'an |
| 11 | uu06     | NULL | m   | NULL     | xi'an |
| 12 | UU08     |   24 | m   | python02 | xi'an |
+----+----------+------+-----+----------+-------+
12 rows in set (0.00 sec)
```

```
mysql> select concat(classid,":",name) from stu;
+--------------------------+
| concat(classid,":",name) |
+--------------------------+
| python03:zhangsan        |
| python02:lisi            |
| python03:wangwu          |
| python02:li              |
| python01:zhang           |
| python03:zhao            |
| python03:uu01            |
| python02:uu02            |
| NULL                     |
| python02:uu04            |
| NULL                     |
| python02:UU08            |
+--------------------------+
12 rows in set (0.06 sec)

mysql>

-- where条件查询
--1. 查询班级为python03期的所有学生信息
mysql> select * from stu where classid='python03';
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  3 | wangwu   |   25 | w   | python03 |
|  6 | zhao     |   27 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
+----+----------+------+-----+----------+
4 rows in set (0.00 sec)

--2. 查询班级为python03期，并且性别为m的所有学生信息
mysql> select * from stu where classid='python03'
and sex='m';
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  6 | zhao     |   27 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
+----+----------+------+-----+----------+
3 rows in set (0.00 sec)

--3. 查询年龄大于20，性别为w的所有信息
mysql> select * from stu where age>20 and sex='w';
+----+--------+------+-----+----------+
| id | name   | age  | sex | classid  |
+----+--------+------+-----+----------+
|  3 | wangwu |   25 | w   | python03 |
|  5 | zhang  |   27 | w   | python01 |
+----+--------+------+-----+----------+
2 rows in set (0.00 sec)

--4. 查询年龄是20~25的所有信息
mysql> select * from stu where age>20 and age<25;
+----+------+------+-----+----------+
| id | name | age  | sex | classid  |
+----+------+------+-----+----------+
|  2 | lisi |   22 | m   | python02 |
| 12 | UU08 |   24 | m   | python02 |
+----+------+------+-----+----------+
2 rows in set (0.00 sec)

--5. 查询年龄不在20~25的学生信息
mysql> select * from stu where age<20 or age>25;
+----+-------+------+-----+----------+
| id | name  | age  | sex | classid  |
+----+-------+------+-----+----------+
|  4 | li    |   28 | m   | python02 |
|  5 | zhang |   27 | w   | python01 |
|  6 | zhao  |   27 | m   | python03 |
|  7 | uu01  |   18 | m   | python03 |
|  8 | uu02  |   26 | m   | python02 |
| 10 | uu04  |   26 | m   | python02 |
+----+-------+------+-----+----------+
6 rows in set (0.00 sec)

--6. 查询id号为1,3,5,7,9的学生信息
mysql> select * from stu where id in(1,3,5,7,9);
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  3 | wangwu   |   25 | w   | python03 |
|  5 | zhang    |   27 | w   | python01 |
|  7 | uu01     |   18 | m   | python03 |
|  9 | uu03     | NULL | m   | NULL     |
+----+----------+------+-----+----------+
5 rows in set (0.00 sec)
```

```
--7. 查询classid不为null所有信息
mysql> select * from stu where classid is not null;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
|  4 | li       |   28 | m   | python02 |
|  5 | zhang    |   27 | w   | python01 |
|  6 | zhao     |   27 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
|  8 | uu02     |   26 | m   | python02 |
| 10 | uu04     |   26 | m   | python02 |
| 12 | UU08     |   24 | m   | python02 |
+----+----------+------+-----+----------+
10 rows in set (0.00 sec)

--8. 查询班级为python01和python02期所有男生（sex='m'）信息
mysql> select * from stu where (classid='python01' or classid='python02') and sex='m';
+----+------+------+-----+----------+
| id | name | age  | sex | classid  |
+----+------+------+-----+----------+
|  2 | lisi |   22 | m   | python02 |
|  4 | li   |   28 | m   | python02 |
|  8 | uu02 |   26 | m   | python02 |
| 10 | uu04 |   26 | m   | python02 |
| 12 | UU08 |   24 | m   | python02 |
+----+------+------+-----+----------+
5 rows in set (0.00 sec)

mysql> select * from stu where classid in('python01','python02') and sex='m';
+----+------+------+-----+----------+
| id | name | age  | sex | classid  |
+----+------+------+-----+----------+
|  2 | lisi |   22 | m   | python02 |
|  4 | li   |   28 | m   | python02 |
|  8 | uu02 |   26 | m   | python02 |
| 10 | uu04 |   26 | m   | python02 |
| 12 | UU08 |   24 | m   | python02 |
+----+------+------+-----+----------+
5 rows in set (0.04 sec)


--9. 查询姓名中含有an子串的所有信息
-- like 模糊查询，支持俩个特殊符号：'%'和'_'   %表示任意数量的任意字符， _表示任意一位字符
mysql> select * from stu where name regexp 'an';
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  3 | wangwu   |   25 | w   | python03 |
|  5 | zhang    |   27 | w   | python01 |
+----+----------+------+-----+----------+
3 rows in set (0.05 sec)
-- like 模糊查询，支持俩个特殊符号：'%'和'_'   %表示任意数量的任意字符， _表示任意一位字符
mysql> select * from stu where name like '%an%';
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  3 | wangwu   |   25 | w   | python03 |
|  5 | zhang    |   27 | w   | python01 |
+----+----------+------+-----+----------+
3 rows in set (0.00 sec)

--10. 查询姓名是有4位任意小写字符或数字构成的信息
mysql> select * from stu where name like '____';
+----+------+------+-----+----------+
| id | name | age  | sex | classid  |
+----+------+------+-----+----------+
|  2 | lisi |   22 | m   | python02 |
|  6 | zhao |   27 | m   | python03 |
|  7 | uu01 |   18 | m   | python03 |
|  8 | uu02 |   26 | m   | python02 |
|  9 | uu03 | NULL | m   | NULL     |
| 10 | uu04 |   26 | m   | python02 |
| 11 | uu06 | NULL | m   | NULL     |
| 12 | UU08 |   24 | m   | python02 |
+----+------+------+-----+----------+
8 rows in set (0.00 sec)
mysql> select * from stu where name regexp '^[a-z0-9]{4}$';
+----+------+------+-----+----------+
| id | name | age  | sex | classid  |
+----+------+------+-----+----------+
|  2 | lisi |   22 | m   | python02 |
|  6 | zhao |   27 | m   | python03 |
|  7 | uu01 |   18 | m   | python03 |
|  8 | uu02 |   26 | m   | python02 |
|  9 | uu03 | NULL | m   | NULL     |
| 10 | uu04 |   26 | m   | python02 |
| 11 | uu06 | NULL | m   | NULL     |
| 12 | UU08 |   24 | m   | python02 |
+----+------+------+-----+----------+
8 rows in set (0.00 sec)

-- 统计函数（聚合函数）max() min() sum() avg() count()
-- 获取最大年龄，最小年龄，年龄总和，平均年龄，总计条数
mysql> select max(age), min(age),sum(age),avg(age),count(id) from stu;
+----------+----------+----------+----------+-----------+
| max(age) | min(age) | sum(age) | avg(age) | count(id) |
+----------+----------+----------+----------+-----------+
|       28 |       18 |      243 |  24.3000 |        12 |
+----------+----------+----------+----------+-----------+
1 row in set (0.07 sec)

-- group by 字段名   分组
-- 按性别sex分组，并统计人数

mysql> select sex,count(*) from stu group by sex;
+-----+----------+
| sex | count(*) |
+-----+----------+
| m   |       10 |
| w   |        2 |
+-----+----------+
2 rows in set (0.00 sec)


-- 按班级分组统计每个班级的人数（排除班级信息为null的数据）
mysql> select classid,count(*) from stu where classid is not null group by classid;
+----------+----------+
| classid  | count(*) |
+----------+----------+
| python01 |        1 |
| python02 |        5 |
| python03 |        4 |
+----------+----------+
3 rows in set (0.00 sec)

-- 按班级分组，并统计每个班级的男生和女生人数（排除班级信息为null的数据
mysql> select classid,sex,count(*) from stu where classid is not null group by classid,sex;
+----------+-----+----------+
| classid  | sex | count(*) |
+----------+-----+----------+
| python01 | w   |        1 |
| python02 | m   |        5 |
| python03 | m   |        3 |
| python03 | w   |        1 |
+----------+-----+----------+
4 rows in set (0.00 sec)


-- 在上面的查询中，加入过滤条件(人数大于等于3的信息）
mysql> select classid,sex,count(*) num from stu where classid is not null group by classid,sex having num>=3;
+----------+-----+-----+
| classid  | sex | num |
+----------+-----+-----+
| python02 | m   |   5 |
| python03 | m   |   3 |
+----------+-----+-----+
2 rows in set (0.02 sec)

-- 排序：order by 字段名 asc(默认升序)|desc（降序）
-------------------------------------------------------------
mysql> select * from stu order by age;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  9 | uu03     | NULL | m   | NULL     |
| 11 | uu06     | NULL | m   | NULL     |
|  7 | uu01     |   18 | m   | python03 |
|  1 | zhangsan |   20 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
| 12 | UU08     |   24 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
|  8 | uu02     |   26 | m   | python02 |
| 10 | uu04     |   26 | m   | python02 |
|  5 | zhang    |   27 | w   | python01 |
|  6 | zhao     |   27 | m   | python03 |
|  4 | li       |   28 | m   | python02 |
+----+----------+------+-----+----------+
12 rows in set (0.06 sec)

mysql> select * from stu order by age asc;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  9 | uu03     | NULL | m   | NULL     |
| 11 | uu06     | NULL | m   | NULL     |
|  7 | uu01     |   18 | m   | python03 |
|  1 | zhangsan |   20 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
| 12 | UU08     |   24 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
|  8 | uu02     |   26 | m   | python02 |
| 10 | uu04     |   26 | m   | python02 |
|  5 | zhang    |   27 | w   | python01 |
|  6 | zhao     |   27 | m   | python03 |
|  4 | li       |   28 | m   | python02 |
+----+----------+------+-----+----------+
12 rows in set (0.00 sec)

mysql> select * from stu order by age desc;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  4 | li       |   28 | m   | python02 |
|  5 | zhang    |   27 | w   | python01 |
|  6 | zhao     |   27 | m   | python03 |
|  8 | uu02     |   26 | m   | python02 |
| 10 | uu04     |   26 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
| 12 | UU08     |   24 | m   | python02 |
|  2 | lisi     |   22 | m   | python02 |
|  1 | zhangsan |   20 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
|  9 | uu03     | NULL | m   | NULL     |
| 11 | uu06     | NULL | m   | NULL     |
+----+----------+------+-----+----------+
12 rows in set (0.00 sec)

mysql> select * from stu order by classid desc;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  3 | wangwu   |   25 | w   | python03 |
|  6 | zhao     |   27 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
|  4 | li       |   28 | m   | python02 |
|  8 | uu02     |   26 | m   | python02 |
| 10 | uu04     |   26 | m   | python02 |
| 12 | UU08     |   24 | m   | python02 |
|  5 | zhang    |   27 | w   | python01 |
|  9 | uu03     | NULL | m   | NULL     |
| 11 | uu06     | NULL | m   | NULL     |
+----+----------+------+-----+----------+
12 rows in set (0.00 sec)

mysql> select * from stu order by classid asc;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  9 | uu03     | NULL | m   | NULL     |
| 11 | uu06     | NULL | m   | NULL     |
|  5 | zhang    |   27 | w   | python01 |
|  2 | lisi     |   22 | m   | python02 |
|  4 | li       |   28 | m   | python02 |
|  8 | uu02     |   26 | m   | python02 |
| 10 | uu04     |   26 | m   | python02 |
| 12 | UU08     |   24 | m   | python02 |
|  1 | zhangsan |   20 | m   | python03 |
|  3 | wangwu   |   25 | w   | python03 |
|  6 | zhao     |   27 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
+----+----------+------+-----+----------+
12 rows in set (0.00 sec)

mysql> select * from stu order by classid asc, age desc;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  9 | uu03     | NULL | m   | NULL     |
| 11 | uu06     | NULL | m   | NULL     |
|  5 | zhang    |   27 | w   | python01 |
|  4 | li       |   28 | m   | python02 |
|  8 | uu02     |   26 | m   | python02 |
| 10 | uu04     |   26 | m   | python02 |
| 12 | UU08     |   24 | m   | python02 |
|  2 | lisi     |   22 | m   | python02 |
|  6 | zhao     |   27 | m   | python03 |
|  3 | wangwu   |   25 | w   | python03 |
|  1 | zhangsan |   20 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
+----+----------+------+-----+----------+
12 rows in set (0.00 sec)


--获取部分数据：limit
-- 分页公式： limit (页号-1)*页大小,页大小
-------------------------------------------
mysql> select * from stu limit 0,3;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
+----+----------+------+-----+----------+
3 rows in set (0.00 sec)

mysql> select * from stu limit 3,3;
+----+-------+------+-----+----------+
| id | name  | age  | sex | classid  |
+----+-------+------+-----+----------+
|  4 | li    |   28 | m   | python02 |
|  5 | zhang |   27 | w   | python01 |
|  6 | zhao  |   27 | m   | python03 |
+----+-------+------+-----+----------+
3 rows in set (0.00 sec)

mysql> select * from stu limit 6,3;
+----+------+------+-----+----------+
| id | name | age  | sex | classid  |
+----+------+------+-----+----------+
|  7 | uu01 |   18 | m   | python03 |
|  8 | uu02 |   26 | m   | python02 |
|  9 | uu03 | NULL | m   | NULL     |
+----+------+------+-----+----------+
3 rows in set (0.00 sec)

mysql> select * from stu limit 9,3;
+----+------+------+-----+----------+
| id | name | age  | sex | classid  |
+----+------+------+-----+----------+
| 10 | uu04 |   26 | m   | python02 |
| 11 | uu06 | NULL | m   | NULL     |
| 12 | UU08 |   24 | m   | python02 |
+----+------+------+-----+----------+
3 rows in set (0.00 sec)

-- 综合查询练习
--1. 查询python03期所有学员，按年龄降序排序
mysql> select * from stu where classid='python03' order by age desc;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  6 | zhao     |   27 | m   | python03 |
|  3 | wangwu   |   25 | w   | python03 |
|  1 | zhangsan |   20 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
+----+----------+------+-----+----------+
4 rows in set (0.00 sec)

--2. 查询python03期，年龄最大的3位学员信息
mysql> select * from stu where classid='python03' order by age desc limit 3;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  6 | zhao     |   27 | m   | python03 |
|  3 | wangwu   |   25 | w   | python03 |
|  1 | zhangsan |   20 | m   | python03 |
+----+----------+------+-----+----------+
3 rows in set (0.03 sec)


--3. 统计每个班级人数，并按人数降序排序（排除班级信息为null的数据）

mysql> select classid,count(*) from stu where cla
+----------+----------+
| classid  | count(*) |
+----------+----------+
| python02 |        5 |
| python03 |        4 |
| python01 |        1 |
+----------+----------+
3 rows in set (0.00 sec)

--4. 统计每个班级年龄在20~30的学员人数信息，并按人数降序排序（排除班级信息为null的数据）
mysql> select classid,count(*) from stu where classid is not null and age between 20 and 30
    -> group by classid order by m desc
+----------+----------+
| classid  | count(*) |
+----------+----------+
| python02 |        5 |
| python03 |        3 |
| python01 |        1 |
+----------+----------+
3 rows in set (0.02 sec)

--5. 统计每个班级男女生人数最多3条记录信息。（排除班级信息为null的数据）
mysql> select classid,sex,count(*) m from stu where classid is not null group by classid,sex order by m desc limit 3
    -> ;
+----------+-----+---+
| classid  | sex | m |
+----------+-----+---+
| python02 | m   | 5 |
| python03 | m   | 3 |
| python03 | w   | 1 |
+----------+-----+---+
3 rows in set (0.00 sec)
```

数据的DQL操作：数据查询
==

```
    格式：
        select [字段列表]|* from 表名
        [where 搜索条件]
        [group by 分组字段 [having 子条件]]
        [order by 排序 asc|desc]
        [limit 分页参数]
```

```
mysql> select classid,count(*) from stu group by classid order by count(*) desc;
+----------+----------+
| classid  | count(*) |
+----------+----------+
| python02 |        5 |
| python03 |        4 |
| NULL     |        2 |
| python01 |        1 |
+----------+----------+
4 rows in set (0.01 sec)

mysql>

mysql> select classid,count(*) from stu group by classid having count(*)>=3 order by count(*) desc;
```

注意：这种情况使用having语句主要是因为count(*) from stu group by classid 相当于一个整体，只有是group by classid 之后才会知道count(*),所以千万不能在count(*)后面写where c
count(*)>=3。

```
+----------+----------+
| classid  | count(*) |
+----------+----------+
| python02 |        5 |
| python03 |        4 |
+----------+----------+
2 rows in set (0.06 sec)

mysql> select ntid,count(*) from news where click>=100 group by ntid;
```

注意：这个语句和上个语句也是截然不同。

```
+------+----------+
| ntid | count(*) |
+------+----------+
|    4 |        2 |
+------+----------+
1 row in set (0.00 sec)

mysql>
```

-- 多表查询：
--==
-- 1. 嵌套查询（一个查询的结果是另外查询的条件）

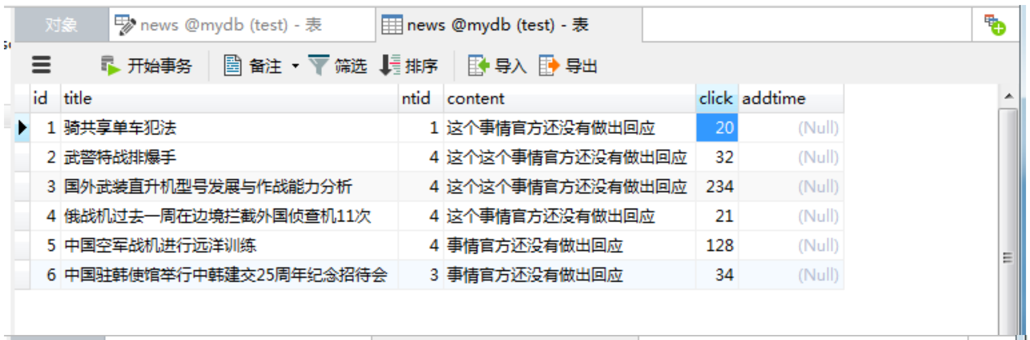
-- 2. where关联查询，相当于是join连接查询中的内联查询。都是要两个表中同时存在的才可以显示。

-- 3. join连接查询：内联inner join，左联 left join，右联right join

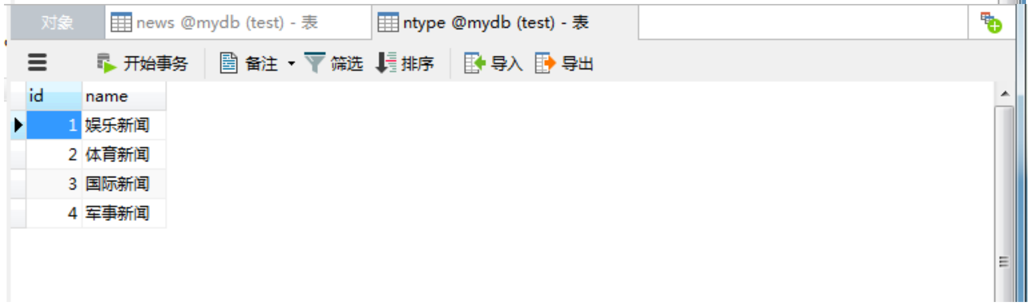
-- 1. 嵌套查询：

-- 查询年龄最大的学生信息

```
mysql> select * from stu where age=(select max(age) from stu);
+----+------+------+-----+----------+
| id | name | age  | sex | classid  |
+----+------+------+-----+----------+
|  4 | li   |   28 | m   | python02 |
+----+------+------+-----+----------+
1 row in set (0.06 sec)

mysql>

mysql> select ntid,count(*) from news group by ntid;
+------+----------+
| ntid | count(*) |
+------+----------+
|    1 |        1 |
|    3 |        1 |
|    4 |        4 |
+------+----------+
3 rows in set (0.14 sec)



mysql> select * from news;
+----+--------------------------------------------------------+------+---------------------------------------------+-------+---------+
| id | title                                                  | ntid | content                                     | click | addtime |
+----+--------------------------------------------------------+------+---------------------------------------------+-------+---------+
|  1 | 骑共享单车犯法                                         |    1 | 这个事情官方还没有做出回应                  |    20 |    NULL |
|  2 | 武警特战排爆手                                         |    4 | 这个这个事情官方还没有做出回应              |    32 |    NULL |
|  3 | 国外武装直升机型号发展与作战能力分析                   |    4 | 这个这个事情官方还没有做出回应              |   234 |    NULL |
|  4 | 俄战机过去一周在边境拦截外国侦查机11次                 |    4 | 这个事情官方还没有做出回应                  |    21 |    NULL |
|  5 | 中国空军战机进行远洋训练                               |    4 | 事情官方还没有做出回应                      |   128 |    NULL |
|  6 | 中国驻韩使馆举行中韩建交25周年纪念招待会               |    3 | 事情官方还没有做出回应                      |    34 |    NULL |
+----+--------------------------------------------------------+------+---------------------------------------------+-------+---------+
6 rows in set (0.00 sec)
```



id	title	ntid	content	click	addtime
1	骑共享单车犯法	1	这个事情官方还没有做出回应	20	(Null)
2	武警特战排爆手	4	这个这个事情官方还没有做出回应	32	(Null)
3	国外武装直升机型号发展与作战能力分析	4	这个这个事情官方还没有做出回应	234	(Null)
4	俄战机过去一周在边境拦截外国侦查机11次	4	这个事情官方还没有做出回应	21	(Null)
5	中国空军战机进行远洋训练	4	事情官方还没有做出回应	128	(Null)
6	中国驻韩使馆举行中韩建交25周年纪念招待会	3	事情官方还没有做出回应	34	(Null)



id	name
1	娱乐新闻
2	体育新闻
3	国际新闻
4	军事新闻

```
mysql> select ntid,count(*) from news group by ntid order by count(*) desc;
+------+----------+
| ntid | count(*) |
+------+----------+
|    4 |        4 |
|    1 |        1 |
|    3 |        1 |
+------+----------+
3 rows in set (0.06 sec)

mysql> select news.ntid,count(*),ntype.name from news,ntype where news.ntid=ntypp e.id group by news.ntid order by count(*) desc;

+------+----------+--------------+
| ntid | count(*) | name         |
+------+----------+--------------+
|    4 |        4 | 军事新闻     |
|    1 |        1 | 娱乐新闻     |
|    3 |        1 | 国际新闻     |
+------+----------+--------------+
3 rows in set (0.24 sec)

mysql>
```

2.where关联查询：

```
mysql> select n.id,n.title,t.name from news n,ntype t where n.ntid=t.id;
+----+-------------------------------------------------------------+--------------+
| id | title                                                       | name         |
+----+-------------------------------------------------------------+--------------+
|  1 | 骑共享单车犯法                                               | 娱乐新闻     |
|  2 | 武警特战排爆手                                               | 军事新闻     |
|  3 | 国外武装直升机型号发展与作战能力分析                         | 军事新闻     |
|  4 | 俄战机过去一周在边境拦截外国侦查机11次                       | 军事新闻     |
|  5 | 中国空军战机进行远洋训练                                     | 军事新闻     |
|  6 | 中国驻韩使馆举行中韩建交25周年纪念招待会                     | 国际新闻     |
+----+-------------------------------------------------------------+--------------+
6 rows in set (0.00 sec)
```

和内联的查询结果是一样的，实际上它们就是一样的，只是写法不同。

```
mysql> select n.id,n.title,t.name from news n inner join ntype t on n.ntid=t.id;
+----+-------------------------------------------------------------+--------------+
| id | title                                                       | name         |
+----+-------------------------------------------------------------+--------------+
|  1 | 骑共享单车犯法                                               | 娱乐新闻     |
|  2 | 武警特战排爆手                                               | 军事新闻     |
|  3 | 国外武装直升机型号发展与作战能力分析                         | 军事新闻     |
|  4 | 俄战机过去一周在边境拦截外国侦查机11次                       | 军事新闻     |
|  5 | 中国空军战机进行远洋训练                                     | 军事新闻     |
|  6 | 中国驻韩使馆举行中韩建交25周年纪念招待会                     | 国际新闻     |
+----+-------------------------------------------------------------+--------------+
6 rows in set (0.00 sec)

mysql> select * from type;
+----+-----------+------+
| id | name      | pid  |
+----+-----------+------+
|  1 | 服装      |    0 |
|  2 | 数码      |    0 |
|  3 | 男装      |    1 |
|  4 | 手机      |    2 |
|  5 | 相机      |    2 |
|  6 | 电脑      |    2 |
|  7 | 女装      |    1 |
|  8 | 童装      |    1 |
|  9 | 食品      |    0 |
| 10 | 零食      |    9 |
| 11 | 特产      |    9 |
| 12 | 休闲装    |    1 |
+----+-----------+------+
12 rows in set (0.00 sec)

mysql> desc type;
+-------+------------------+------+-----+---------+----------------+
| Field | Type             | Null | Key | Default | Extra          |
+-------+------------------+------+-----+---------+----------------+
| id    | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| name  | varchar(16)      | NO   |     | NULL    |                |
| pid   | int(10) unsigned | YES  |     | NULL    |                |
+-------+------------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

-- 查询二级类别信息，并关联出他们的父类别名称
mysql> select  t1.id,t1.name,t2.name  from type t1,type t2 where t1.pid!=0 and t1.pid=t2.id;
+----+-----------+--------+
| id | name      | name   |
+----+-----------+--------+
|  3 | 男装      | 服装   |
|  4 | 手机      | 数码   |
|  5 | 相机      | 数码   |
|  6 | 电脑      | 数码   |
|  7 | 女装      | 服装   |
|  8 | 童装      | 服装   |
| 10 | 零食      | 食品   |
| 11 | 特产      | 食品   |
| 12 | 休闲装    | 服装   |
+----+-----------+--------+
9 rows in set (0.01 sec)

--统计每个一级类别下都有多少个子类别。
mysql> select t1.id,t1.name,count(t2.id) from type t1,type t2 where t1.pid=0 and t1.id=t2.pid group by t1.id;
+----+--------+--------------+
| id | name   | count(t2.id) |
+----+--------+--------------+
|  1 | 服装   |            4 |
|  2 | 数码   |            3 |
|  9 | 食品   |            2 |
+----+--------+--------------+
3 rows in set (0.00 sec)
```

3.join查询---左联查询：

```
mysql> select n.id,n.title,t.name from news n left join ntype t on n.ntid=t.id;
+----+-----------------------------------------------------------------+--------------+
| id | title                                                           | name         |
+----+-----------------------------------------------------------------+--------------+
|  1 | 骑共享单车犯法                                                  | 娱乐新闻     |
|  6 | 中国驻韩使馆举行中韩建交25周年纪念招待会                        | 国际新闻     |
|  2 | 武警特战排爆手                                                  | 军事新闻     |
|  3 | 国外武装直升机型号发展与作战能力分析                            | 军事新闻     |
|  4 | 俄战机过去一周在边境拦截外国侦查机11次                          | 军事新闻     |
|  5 | 中国空军战机进行远洋训练                                        | 军事新闻     |
+----+-----------------------------------------------------------------+--------------+
6 rows in set (0.00 sec)

mysql>
```

把news的这张表乱加了一条数据。再进行左联查询：

```
mysql> select n.id,n.title,t.name from news n left join ntype t on n.ntid=t.id;
+----+-----------------------------------------------------------------+--------------+
| id | title                                                           | name         |
+----+-----------------------------------------------------------------+--------------+
|  1 | 骑共享单车犯法                                                  | 娱乐新闻     |
|  6 | 中国驻韩使馆举行中韩建交25周年纪念招待会                        | 国际新闻     |
|  2 | 武警特战排爆手                                                  | 军事新闻     |
|  3 | 国外武装直升机型号发展与作战能力分析                            | 军事新闻     |
|  4 | 俄战机过去一周在边境拦截外国侦查机11次                          | 军事新闻     |
|  5 | 中国空军战机进行远洋训练                                        | 军事新闻     |
|  7 | aaaaaaaaa                                                       | NULL         |
+----+-----------------------------------------------------------------+--------------+
7 rows in set (0.00 sec)
```

右联查询也是同样的道理，将left换成right就OK！

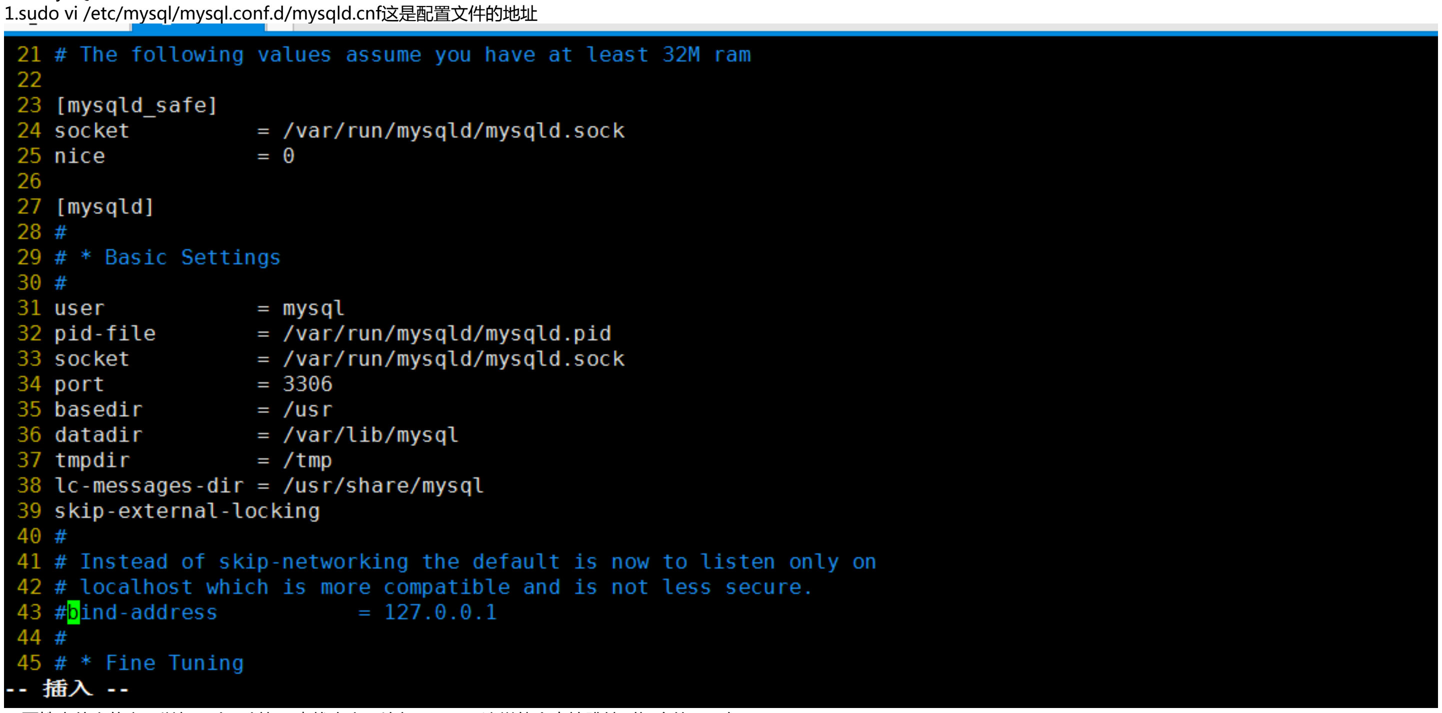
```
--3. 修改数据
-- 格式：update 表名 set 字段名1=值1[,字段名2=值2,...] [where 条件...]
--==================================================================
mysql> select * from uu;
+------+-----------+------+
| id   | name      | age  |
+------+-----------+------+
|    1 | zhangsan  |   20 |
|    2 | lisi      |   22 |
|    3 | wangwu    |   25 |
|    4 | lisi      |   21 |
|    5 | xiaoli    |   22 |
|    6 | xiaozhang |   19 |
+------+-----------+------+
6 rows in set (1.14 sec)

-- 将id值为4的信息age改为30（修改）
mysql> update uu set age=30 where id=4;
Query OK, 1 row affected (0.34 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from uu;
+------+-----------+------+
| id   | name      | age  |
+------+-----------+------+
|    1 | zhangsan  |   20 |
|    2 | lisi      |   22 |
|    3 | wangwu    |   25 |
|    4 | lisi      |   30 |
|    5 | xiaoli    |   22 |
|    6 | xiaozhang |   19 |
+------+-----------+------+
6 rows in set (0.00 sec)

-- 将id为8,10和12 的年龄age改为26，班级classid改为python02
mysql> select * from stu;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
|  4 | li       |   28 | m   | python02 |
|  5 | zhang    |   27 | w   | python01 |
|  6 | zhao     |   27 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
|  8 | uu02     | NULL | m   | NULL     |
|  9 | uu03     | NULL | m   | NULL     |
| 10 | uu04     | NULL | m   | NULL     |
| 11 | uu06     | NULL | m   | NULL     |
| 12 | UU08     | NULL | m   | NULL     |
+----+----------+------+-----+----------+
12 rows in set (0.03 sec)

mysql> update stu set age=26, classid='python02' where id in(8,10,12);
Query OK, 3 rows affected (0.03 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> select * from stu;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
|  4 | li       |   28 | m   | python02 |
|  5 | zhang    |   27 | w   | python01 |
|  6 | zhao     |   27 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
|  8 | uu02     |   26 | m   | python02 |
|  9 | uu03     | NULL | m   | NULL     |
| 10 | uu04     |   26 | m   | python02 |
| 11 | uu06     | NULL | m   | NULL     |
| 12 | UU08     |   26 | m   | python02 |
+----+----------+------+-----+----------+
12 rows in set (0.00 sec)

mysql>

--修改年龄的大小
mysql> update stu set age=age-2 where id=12;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from stu;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
|  4 | li       |   28 | m   | python02 |
|  5 | zhang    |   27 | w   | python01 |
|  6 | zhao     |   27 | m   | python03 |
|  7 | uu01     |   18 | m   | python03 |
|  8 | uu02     |   26 | m   | python02 |
|  9 | uu03     | NULL | m   | NULL     |
| 10 | uu04     |   26 | m   | python02 |
| 11 | uu06     | NULL | m   | NULL     |
| 12 | UU08     |   24 | m   | python02 |
+----+----------+------+-----+----------+
12 rows in set (0.00 sec)

mysql>
```

```
-- 2. 删除数据
-- 格式： delete from 表名 [where 条件 [分组、排序、limit]]
--==========================================================
-- 删除id为5的所有信息
mysql> delete from uu where id=5;
Query OK, 1 row affected (0.05 sec)

mysql> select * from uu;
+------+-----------+------+
| id   | name      | age  |
+------+-----------+------+
|    1 | zhangsan  |   20 |
|    2 | lisi      |   22 |
|    3 | wangwu    |   25 |
|    4 | lisi      |   30 |
|    6 | xiaozhang |   19 |
+------+-----------+------+
5 rows in set (0.00 sec)

mysql>
--查看stu表的所有数据
ysql> select * from stu;
+----+----------+------+-----+----------+
| id | name     | age  | sex | classid  |
+----+----------+------+-----+----------+
|  1 | zhangsan |   20 | m   | python03 |
|  2 | lisi     |   22 | m   | python02 |
|  3 | wangwu   |   25 | w   | python03 |
|  8 | xiaoli   |   28 | m   | python02 |
|  9 | zhang    |   21 | w   | python01 |
| 10 | zhao     |   27 | m   | python03 |
| 11 | uu01     |   18 | m   | python03 |
+----+----------+------+-----+----------+
7 rows in set (0.00 sec)

-- 删除id为22的信息
mysql> delete from stu where id=22
    -> ;
Query OK, 0 rows affected (0.06 sec)

-- 删除id大于100的所有信息
mysql> delete from stu where id > 100;
Query OK, 0 rows affected (0.00 sec)

-- 删除id是100~200的所有信息
mysql> delete from stu where id>=100 and id<=200;
Query OK, 0 rows affected (0.00 sec)

-- 删除性别为w，年龄大于25的所有信息
mysql> delete from stu where sex='w' and age>25;
Query OK, 0 rows affected (0.00 sec)

mysql>


-- 数据类型：数值类型
--====================================
mysql> create table m1(
    -> id int unsigned auto_increment primary key,
    -> age tinyint unsigned,
    -> num tinyint,
    -> n1 int(4
    -> ),
    -> n2 int(6) zerofill);
Query OK, 0 rows affected (0.56 sec)

mysql> desc m1;
+-------+-------------------------+------+-----+---------+----------------+
| Field | Type                    | Null | Key | Default | Extra          |
+-------+-------------------------+------+-----+---------+----------------+
| id    | int(10) unsigned        | NO   | PRI | NULL    | auto_increment |
| age   | tinyint(3) unsigned     | YES  |     | NULL    |                |
| num   | tinyint(4)              | YES  |     | NULL    |                |
| n1    | int(4)                  | YES  |     | NULL    |                |
| n2    | int(6) unsigned zerofill | YES  |     | NULL    |                |
+-------+-------------------------+------+-----+---------+----------------+
5 rows in set (0.14 sec)

mysql> insert into m1 values(1,88,-20,12345,5678);
Query OK, 1 row affected (0.07 sec)

mysql> select * from m1;
+----+------+------+-------+--------+
| id | age  | num  | n1    | n2     |
+----+------+------+-------+--------+
|  1 |   88 |  -20 | 12345 | 005678 |
+----+------+------+-------+--------+
1 row in set (0.00 sec)

mysql> insert into m1 values(2,300,128,32423,2345678);
ERROR 1264 (22003): Out of range value for column 'age' at row 1
```

```
mysql> select * from m1;
+----+------+------+-------+--------+
| id | age  | num  | n1    | n2     |
+----+------+------+-------+--------+
|  1 |   88 |  -20 | 12345 | 005678 |
+----+------+------+-------+--------+
1 row in set (0.00 sec)

mysql> insert into m1 values(1,23,34,100,34);
ERROR 1062 (23000): Duplicate entry '1' for key 'PRIMARY'
mysql> insert into m1 values(null,23,34,100,34);
Query OK, 1 row affected (0.05 sec)

mysql> select * from m1;
+----+------+------+-------+--------+
| id | age  | num  | n1    | n2     |
+----+------+------+-------+--------+
|  1 |   88 |  -20 | 12345 | 005678 |
|  2 |   23 |   34 |   100 | 000034 |
+----+------+------+-------+--------+
2 rows in set (0.00 sec)


-- 子串类型实例
--====================================
mysql> create table m2(
    -> c1 char(8),
    -> c2 varchar(8),
    -> c3 text,
    -> c4 enum('y','n') not null default 'y'
    -> );
Query OK, 0 rows affected (0.34 sec)

mysql> desc m2;
+-------+---------------+------+-----+---------+-------+
| Field | Type          | Null | Key | Default | Extra |
+-------+---------------+------+-----+---------+-------+
| c1    | char(8)       | YES  |     | NULL    |       |
| c2    | varchar(8)    | YES  |     | NULL    |       |
| c3    | text          | YES  |     | NULL    |       |
| c4    | enum('y','n') | NO   |     | y       |       |
+-------+---------------+------+-----+---------+-------+
4 rows in set (0.10 sec)

mysql> insert m2 values('qwertyuio','qwertyuio','we','n');
ERROR 1406 (22001): Data too long for column 'c1' at row 1
mysql> insert m2 values('qwertyui','qwertyui','we','n');
Query OK, 1 row affected (0.01 sec)

mysql> select * from m2;
+----------+----------+------+----+
| c1       | c2       | c3   | c4 |
+----------+----------+------+----+
| qwertyui | qwertyui | we   | n  |
+----------+----------+------+----+
1 row in set (0.00 sec)

mysql> create table m3(
    -> d1 date,
    -> d2 datetime,
    -> d3 timestamp);
Query OK, 0 rows affected (0.16 sec)


-- 时间日期类型
--=================================================
mysql> desc m3;
+-------+-----------+------+-----+-------------------+-----------------------------+
| Field | Type      | Null | Key | Default           | Extra                       |
+-------+-----------+------+-----+-------------------+-----------------------------+
| d1    | date      | YES  |     | NULL              |                             |
| d2    | datetime  | YES  |     | NULL              |                             |
| d3    | timestamp | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+-------+-----------+------+-----+-------------------+-----------------------------+
3 rows in set (0.00 sec)

mysql> insert into m3 values('2018-05-26',now(),current_timestamp());
Query OK, 1 row affected (0.06 sec)

mysql> select * from m3;
+------------+---------------------+---------------------+
| d1         | d2                  | d3                  |
+------------+---------------------+---------------------+
| 2018-05-26 | 2018-05-26 13:21:08 | 2018-05-26 13:21:08 |
+------------+---------------------+---------------------+
1 row in set (0.00 sec)

mysql>

-- 表操作实战
--=================================================================
mysql> create table stu(
    -> id int unsigned not null auto_increment primary key,
    -> name varchar(8) not null unique,
    -> age tinyint unsigned,
    -> sex enum('m','w') not null default 'm',
    -> classid char(8)
    -> );
Query OK, 0 rows affected (0.13 sec)

--表结构说明：
--id 整型int  无符号 非空 自增 主键约束
--name 可变子串8个长度 非空 唯一性约束
--age 非常小整型 无符号约束
--sex  性别枚举类型m或w值，非空，默认值m
--classid 定长子串8个长度

mysql> desc stu;
+---------+---------------------+------+-----+---------+----------------+
| Field   | Type                | Null | Key | Default | Extra          |
+---------+---------------------+------+-----+---------+----------------+
| id      | int(10) unsigned    | NO   | PRI | NULL    | auto_increment |
| name    | varchar(8)          | NO   | UNI | NULL    |                |
| age     | tinyint(3) unsigned | YES  |     | NULL    |                |
| sex     | enum('m','w')       | NO   |     | m       |                |
| classid | char(8)             | YES  |     | NULL    |                |
+---------+---------------------+------+-----+---------+----------------+
5 rows in set (0.06 sec)

mysql> show create table stu\G
*************************** 1. row ***************************
       Table: stu
Create Table: CREATE TABLE `stu` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(8) NOT NULL,
  `age` tinyint(3) unsigned DEFAULT NULL,
  `sex` enum('m','w') NOT NULL DEFAULT 'm',
  `classid` char(8) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)




-- 数据库备份与恢复

root@may-virtual-machine:/home/may# mysqldump -u root -p mydb > mydb.sql
Enter password: 
root@may-virtual-machine:/home/may# ls
examples.desktop  show    模板  图片  下载  桌面
mydb.sql          公共的  视频  文档  音乐
root@may-virtual-machine:/home/may# vim mydb.sql
root@may-virtual-machine:/home/may# mysqldump -u root -p mydb stu > mydb_stu.sql
Enter password: 
root@may-virtual-machine:/home/may# mysqldump -u root -p mydb m1 > m1.sql
Enter password: 
root@may-virtual-machine:/home/may# ls;
examples.desktop  mydb_stu.sql  模板  文档  桌面
m1.sql            show          视频  下载
mydb.sql          公共的        图片  音乐

root@may-virtual-machine:/home/may# mysql -u root -p mydb < mydb.sql
Enter password: 
root@may-virtual-machine:/home/may# 

mysql> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| m1             |
| m2             |
| m3             |
| stu            |
| tt             |
| uu             |
+----------------+
6 rows in set (0.00 sec)

mysql> select * from stu;
Empty set (0.00 sec)

mysql> select * from m1;
+----+------+------+-------+--------+
| id | age  | num  | n1    | n2     |
+----+------+------+-------+--------+
|  1 |   88 |  -20 | 12345 | 005678 |
|  2 |   23 |   34 |   100 | 000034 |
+----+------+------+-------+--------+
2 rows in set (0.00 sec)

mysql> drop table m1
    -> ;
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| m2             |
| m3             |
| stu            |
| tt             |
| uu             |
+----------------+
5 rows in set (0.00 sec)

mysql> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| m2             |
| m3             |
| stu            |
| tt             |
| uu             |
+----------------+
5 rows in set (0.00 sec)

mysql> create table m1;
ERROR 1113 (42000): A table must have at least 1 column
mysql> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| m1             |
| m2             |
| m3             |
| stu            |
| tt             |
| uu             |
+----------------+
6 rows in set (0.00 sec)

mysql> select * from m1;
+----+------+------+-------+--------+
| id | age  | num  | n1    | n2     |
+----+------+------+-------+--------+
|  1 |   88 |  -20 | 12345 | 005678 |
|  2 |   23 |   34 |   100 | 000034 |
+----+------+------+-------+--------+
2 rows in set (0.03 sec)

mysql>

-- 修改表结构实例
--===============================================
mysql> show create table tt\G
*************************** 1. row ***************************
       Table: tt
Create Table: CREATE TABLE `tt` (
  `id` int(11) DEFAULT NULL,
  `name` varchar(16) DEFAULT NULL,
  `age` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

mysql> desc tt;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | YES  |     | NULL    |       |
| name  | varchar(16) | YES  |     | NULL    |       |
| age   | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)


--1. 在tt表末尾添加一个phone字段，类型varchar(11)，无其他约束
mysql> alter table tt add phone varchar(11);
Query OK, 0 rows affected (0.29 sec)
Records: 0  Duplicates: 0  Warnings: 0


--2. 在tt表中age字段后添加一个address字段，类型varchar(100) ，无其他约束
mysql> alter table tt add address varchar(100) after age;
Query OK, 0 rows affected (0.15 sec)
Records: 0  Duplicates: 0  Warnings: 0


--3. 在tt表首位插入一个mm字段，类型int
mysql> alter table tt add mm int first;
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc tt;
+---------+-------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| mm      | int(11)      | YES  |     | NULL    |       |
| id      | int(11)      | YES  |     | NULL    |       |
| name    | varchar(16)  | YES  |     | NULL    |       |
| age     | int(11)      | YES  |     | NULL    |       |
| address | varchar(100) | YES  |     | NULL    |       |
| phone   | varchar(11)  | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>

--4. 删除tt表的mm字段
mysql> alter table tt drop mm;
Query OK, 0 rows affected (0.12 sec)
Records: 0  Duplicates: 0  Warnings: 0


--5. 修改字段：tt表age字段类型改为tinyint类型，unsigned not null default 20
mysql> alter table tt modify age tinyint unsigned not null default 20;
Query OK, 0 rows affected (0.17 sec)
Records: 0  Duplicates: 0  Warnings: 0


--6. 修改name字段名为username
mysql> alter table tt change name username varchar(16);
Query OK, 0 rows affected (0.31 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc tt;
+----------+---------------------+------+-----+---------+-------+
| Field    | Type                | Null | Key | Default | Extra |
+----------+---------------------+------+-----+---------+-------+
| id       | int(11)             | YES  |     | NULL    |       |
| username | varchar(16)         | YES  |     | NULL    |       |
| age      | tinyint(3) unsigned | NO   |     | 20      |       |
| address  | varchar(100)        | YES  |     | NULL    |       |
| phone    | varchar(11)         | YES  |     | NULL    |       |
+----------+---------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
mysql>
```

Windows操作数据库：
进入MySQL的配置文件，将bind-address注释掉
1.sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf这是配置文件的地址



```
# The following values assume you have at least 32M ram

[mysqld_safe]
socket          = /var/run/mysqld/mysqld.sock
nice            = 0

[mysqld]
#
# * Basic Settings
#
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port            = 3306
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address           = 127.0.0.1
#
# * Fine Tuning
-- 插入 --
```

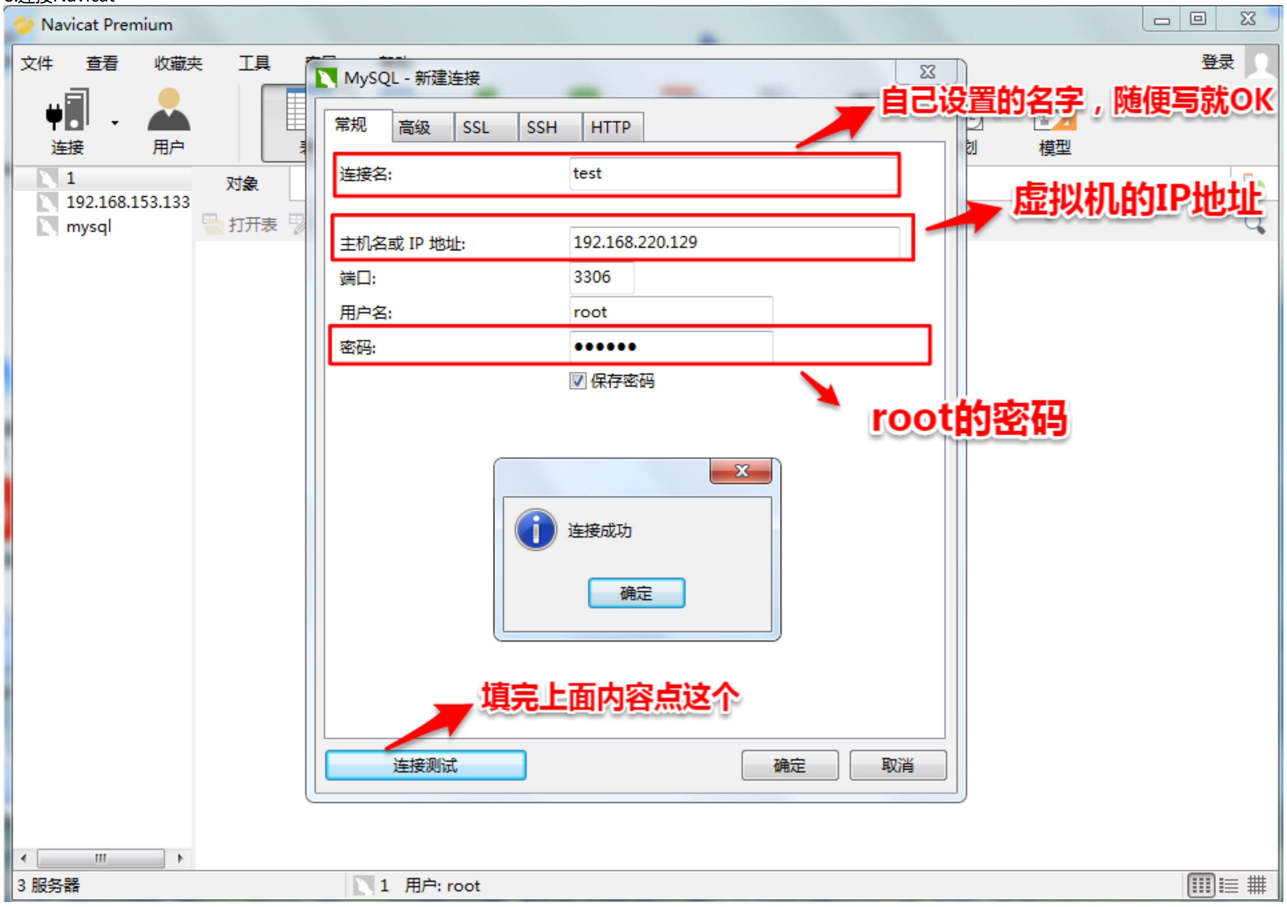
（要搜索什么信息可以加一个正斜杠+查找内容。比如：/127 这样就会直接跳转到相应的那一行）

```
2.进行授权
mysql> grant all on *.* to root@'%' identified by '123456' with grant option;
Query OK, 0 rows affected, 1 warning (1.34 sec)
3.进行刷新（如果不刷新的话就需要对数据库进行重启）
mysql> flush privileges;
Query OK, 0 rows affected (0.46 sec)
4.退出MySQL
mysql> exit;
Bye
5.重启MySQL（安全起见）
6.查看自己虚拟机的IP地址
root@may-virtual-machine:/home/may# ifconfig
ens33     Link encap:以太网  硬件地址 00:0c:29:f3:f7:8d
          inet 地址:192.168.220.129  广播:192.168.220.255  掩码:255.255.255.0
          inet6 地址: fe80::e61:809a:a6d1:cf4e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:20931 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:8368 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:27335354 (27.3 MB)  发送字节:650539 (650.5 KB)

lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  跃点数:1
          接收数据包:268 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:268 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:22294 (22.2 KB)  发送字节:22294 (22.2 KB)

root@may-virtual-machine:/home/may#
```

7.进入数据库

```
root@may-virtual-machine:/home/may# mysql -u root -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.22-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydb               |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.16 sec)

mysql>
```

8.连接Navicat



MySQL - 新建连接

常规 | 高级 | SSL | SSH | HTTP

连接名: test

主机名或 IP 地址: 192.168.220.129

端口: 3306

用户名: root

密码: ●●●●●●

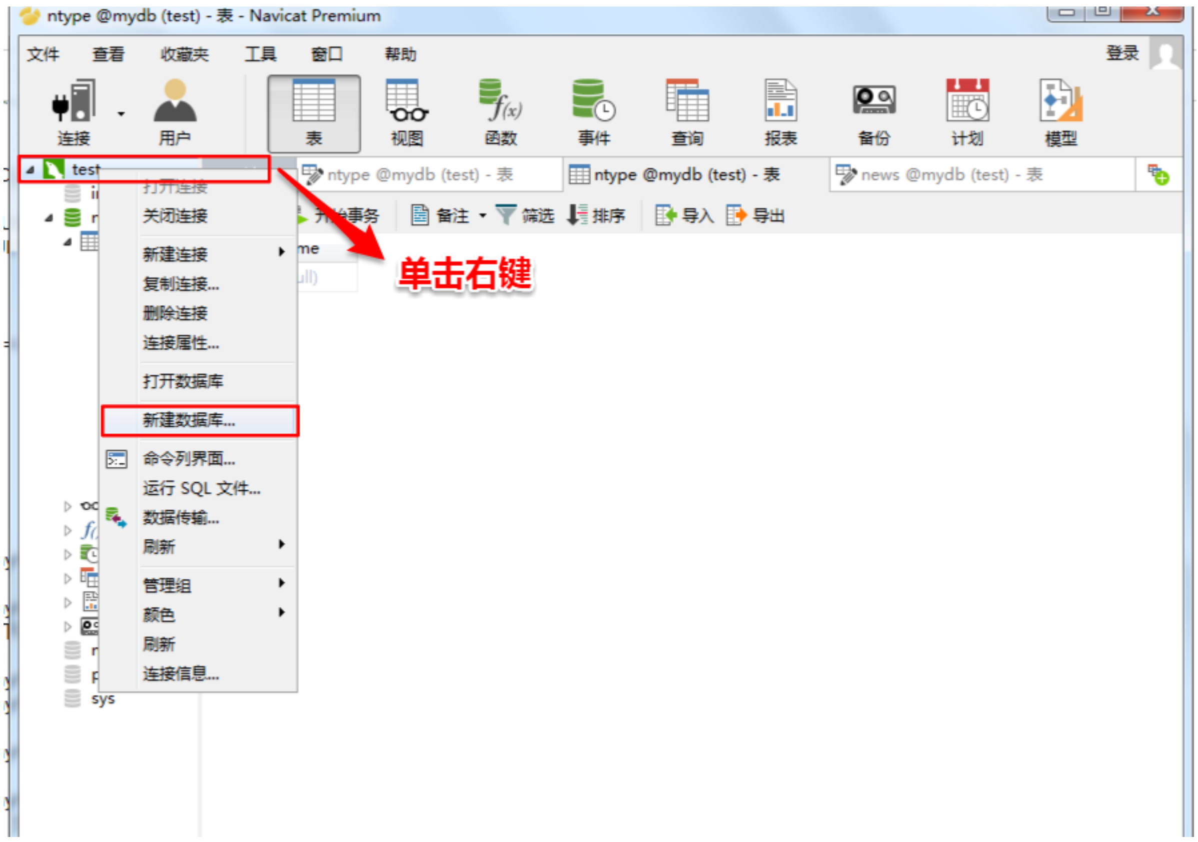
☑ 保存密码

连接成功

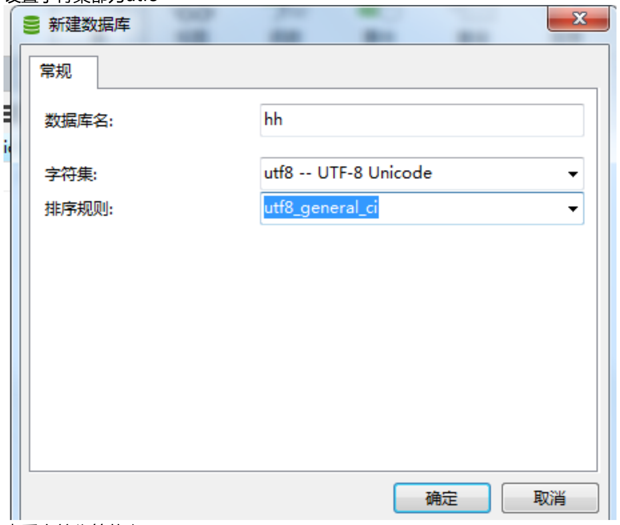
确定

连接测试 | 确定 | 取消

在navicat创建数据库等等操作



设置字符集都为utf8



数据库名: hh

字符集: utf8 -- UTF-8 Unicode

排序规则: utf8_general_ci

查看字符集等信息：

```
mysql> \s
--------------
mysql  Ver 14.14 Distrib 5.7.22, for Linux (x86_64) using  EditLine wrapper

Connection id:        3
Current database:    mydb
Current user:        root@localhost
SSL:            Not in use
Current pager:        stdout
Using outfile:        ''
Using delimiter:    ;
Server version:        5.7.22-0ubuntu0.16.04.1 (Ubuntu)
Protocol version:    10
Connection:        Localhost via UNIX socket
Server characterset:    latin1
Db     characterset:    latin1
Client characterset:    utf8
Conn.  characterset:    utf8
UNIX socket:        /var/run/mysqld/mysqld.sock
Uptime:            47 min 0 sec

Threads: 4  Questions: 89  Slow queries: 0  Opens: 121  Flush tables: 1  Open tables: 40  Queries per second avg: 0.031
--------------

mysql>
```

注意：字符集不统一是导致乱码的最主要的原因。乱码有两种情况，一种是 数据完全被破坏，无法恢复。另一种是当前显示乱码，但是可以恢复。
例如：

```
mysql> set names latin1;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from ntype;
+------------+-----------------+
| id         | name            |
+------------+-----------------+
| 0000000001 | yu le xin wen   |
| 0000000002 | ti yu xin wen   |
| 0000000003 | guo ji xin wen  |
| 0000000004 | jun shi xin wen |
| 0000000005 | ????            |
+------------+-----------------+
5 rows in set (0.00 sec)

mysql> set names utf8;
Query OK, 0 rows affected (0.05 sec)

mysql> select * from ntype;
+------------+-----------------+
| id         | name            |
+------------+-----------------+
| 0000000001 | yu le xin wen   |
| 0000000002 | ti yu xin wen   |
| 0000000003 | guo ji xin wen  |
| 0000000004 | jun shi xin wen |
| 0000000005 | 其他新闻        |
+------------+-----------------+
5 rows in set (0.00 sec)
```

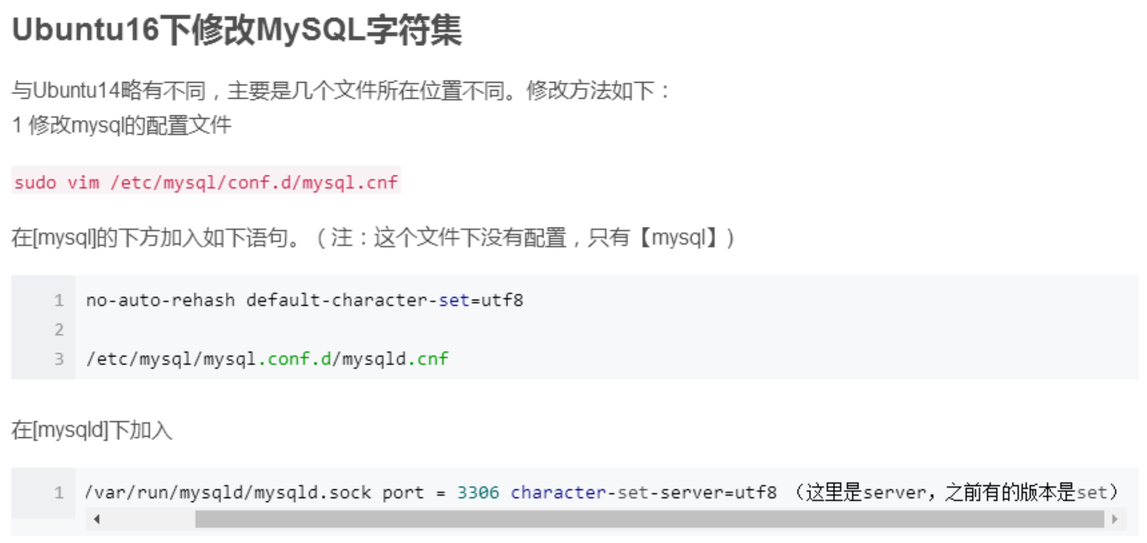
这就是属于可以恢复的。

```
mysql>
```

Server characterset: latin1 这个是MySQL服务器的字符集编码可以在MySQL的配置文件中国进行设置及修改。

1.首先进入MySQL的配置文件

```
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
```



Ubuntu16下修改MySQL字符集

与Ubuntu14略有不同，主要是几个文件所在位置不同。修改方法如下：

1 修改mysql的配置文件

```
sudo vim /etc/mysql/conf.d/mysql.cnf
```

在[mysql]的下方加入如下语句。（注：这个文件下没有配置，只有【mysql】）

```
no-auto-rehash default-character-set=utf8

/etc/mysql/mysql.conf.d/mysqld.cnf
```

在[mysqld]下加入

```
/var/run/mysqld/mysqld.sock port = 3306 character-set-server=utf8 （这里是server，之前有的版本是set）
```

在配置文件中进行上图的操作，在port=3306的下一行插入character-set-server=utf8

2.保存退出之后重启MySQL服务器：

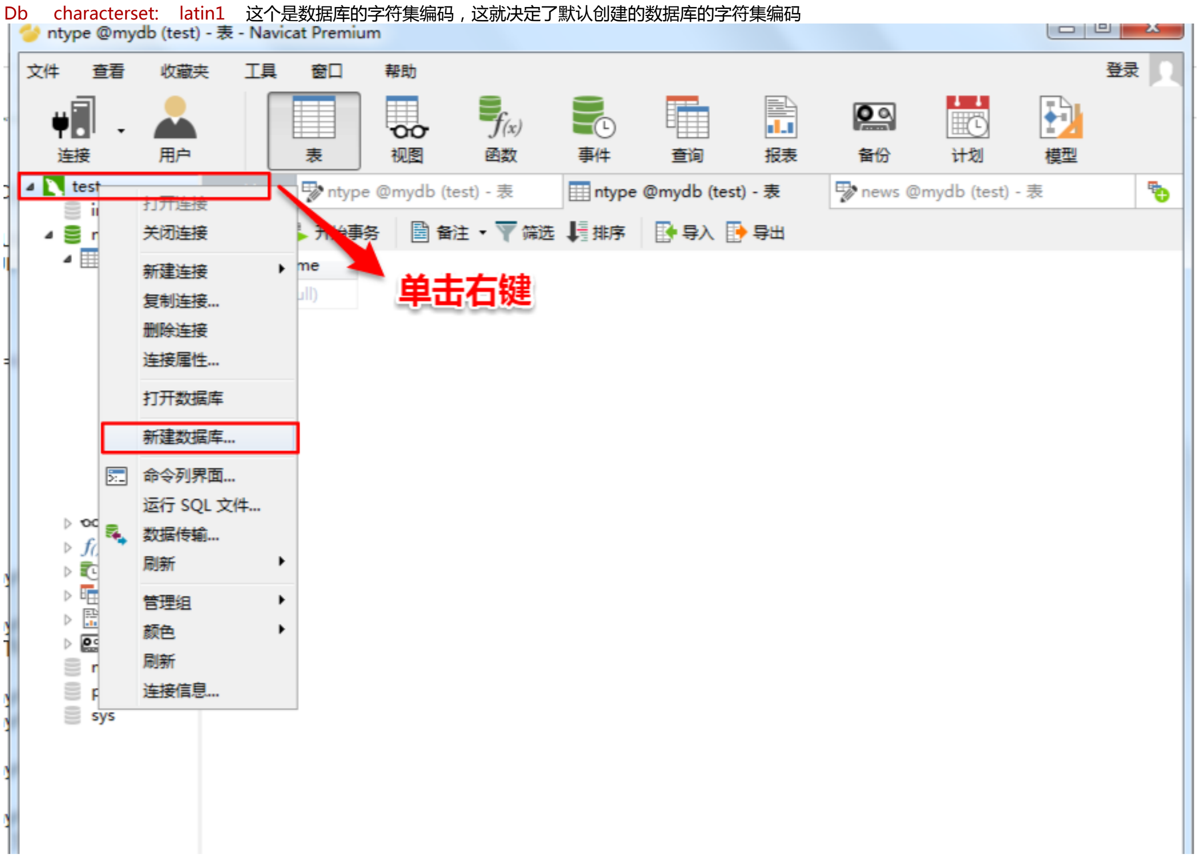
```
service mysql restart
```

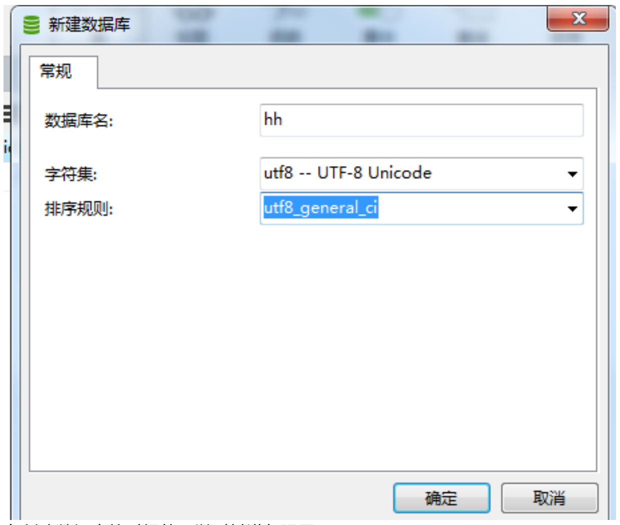
3.进入MySQL查看：

```
mysql> use mydb
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> \s
--------------
mysql  Ver 14.14 Distrib 5.7.22, for Linux (x86_64) using  EditLine wrapper

Connection id:          3
Current database:       mydb
Current user:           root@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.7.22-0ubuntu0.16.04.1 (Ubuntu)
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    utf8
Db     characterset:    latin1
Client characterset:    utf8
Conn.  characterset:    utf8
UNIX socket:            /var/run/mysqld/mysqld.sock
Uptime:                 1 min 5 sec

Threads: 1  Questions: 21  Slow queries: 0  Opens: 117  Flush tables: 1  Open tables: 36  Queries per second avg: 0.323
--------------
```

mysql>
修改成功。

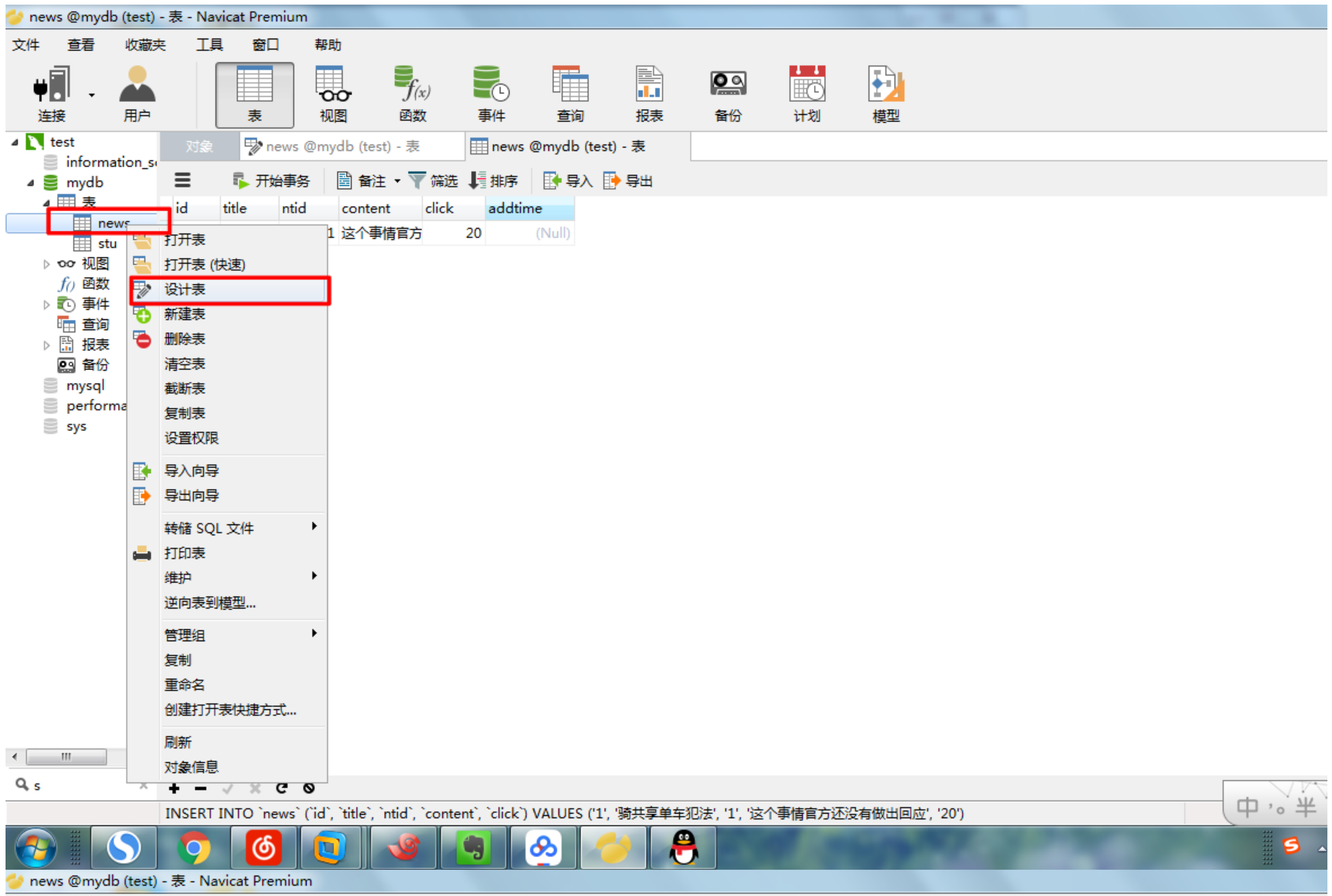
Db characterset: latin1 这个是数据库的字符集编码，这就决定了默认创建的数据库的字符集编码





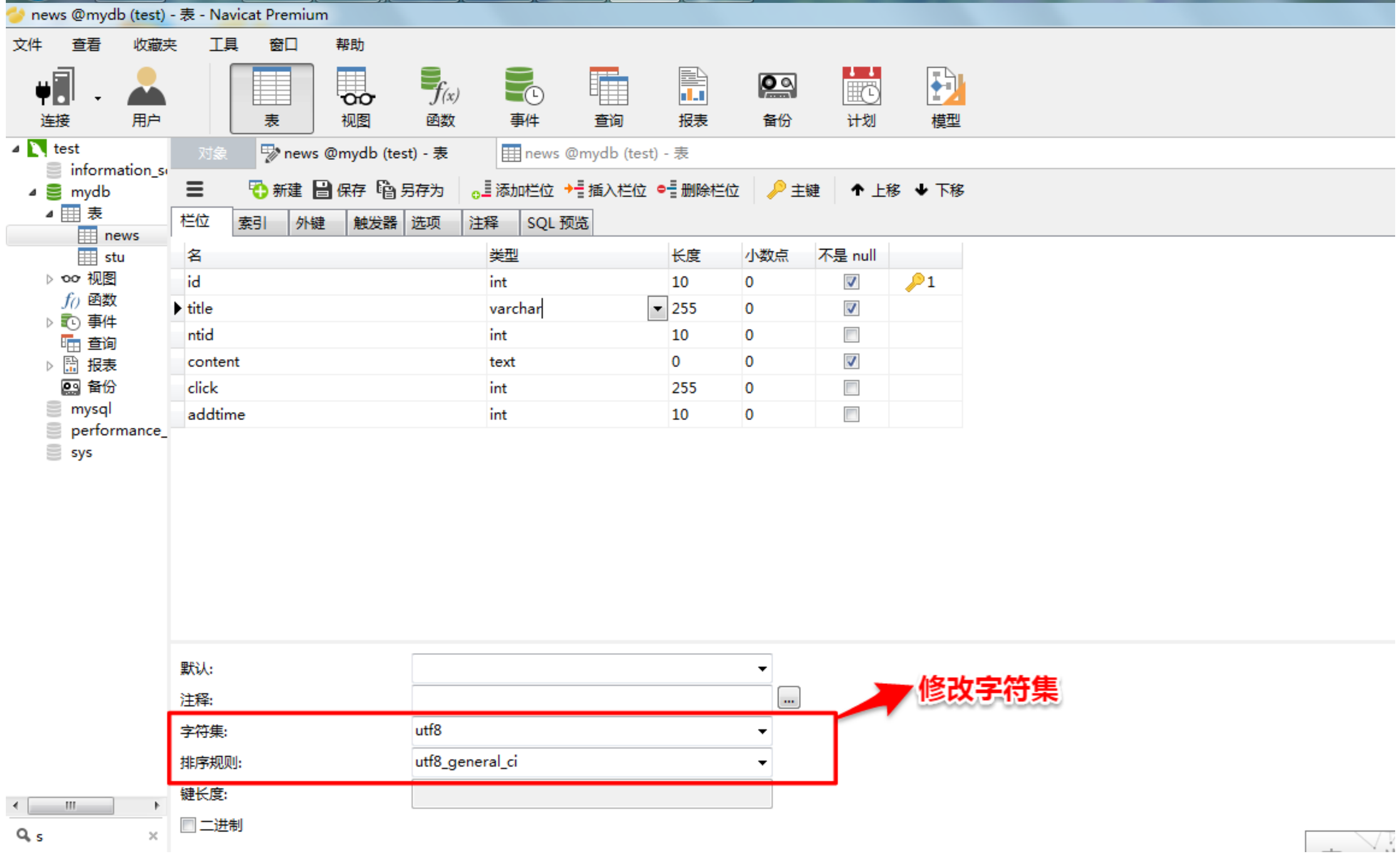
数据库名: hh
字符集: utf8 -- UTF-8 Unicode
排序规则: utf8_general_ci

在创建数据库的时候就可以对其进行设置。
如果已经存在的数据库字符集编码不是utf8，那么就要利用命令对其进行修改，修改后，就算MySQL重启也会生效。
mysql> set character_set_database=utf8;
如果这样修改之后再Navicat中填入数据仍然显示字符集不正确，那么需要再进行一步操作：


news @mydb (test) - 表 - Navicat Premium
文件 查看 收藏夹 工具 窗口 帮助
连接 用户 表 视图 函数 事件 查询 报表 备份 计划 模型
test information_s mydb 表 news stu 视图 函数 事件 查询 报表 备份 mysql performa sys
打开表
打开表 (快速)
设计表
新建表
删除表
清空表
截断表
复制表
设置权限
导入向导
导出向导
转储 SQL 文件
打印表
维护
逆向表到模型...
管理组
复制
重命名
创建打开表快捷方式...
刷新
对象信息
开始事务 备注 筛选 排序 导入 导出

id	title	ntid	content	click	addtime
		1	这个事情官方	20	(Null)

```
INSERT INTO `news` (`id`, `title`, `ntid`, `content`, `click`) VALUES ('1', '骑共享单车犯法', '1', '这个事情官方还没有做出回应', '20')
```


news @mydb (test) - 表 - Navicat Premium
文件 查看 收藏夹 工具 窗口 帮助
连接 用户 表 视图 函数 事件 查询 报表 备份 计划 模型
新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移 下移
栏位 索引 外键 触发器 选项 注释 SQL 预览
修改字符集

名	类型	长度	小数点	不是 null	
id	int	10	0	✓	1
title	varchar	255	0	✓	
ntid	int	10	0		
content	text	0	0	✓	
click	int	255	0		
addtime	int	10	0		

默认:	
注释:	
字符集:	utf8
排序规则:	utf8_general_ci
键长度:	
二进制	

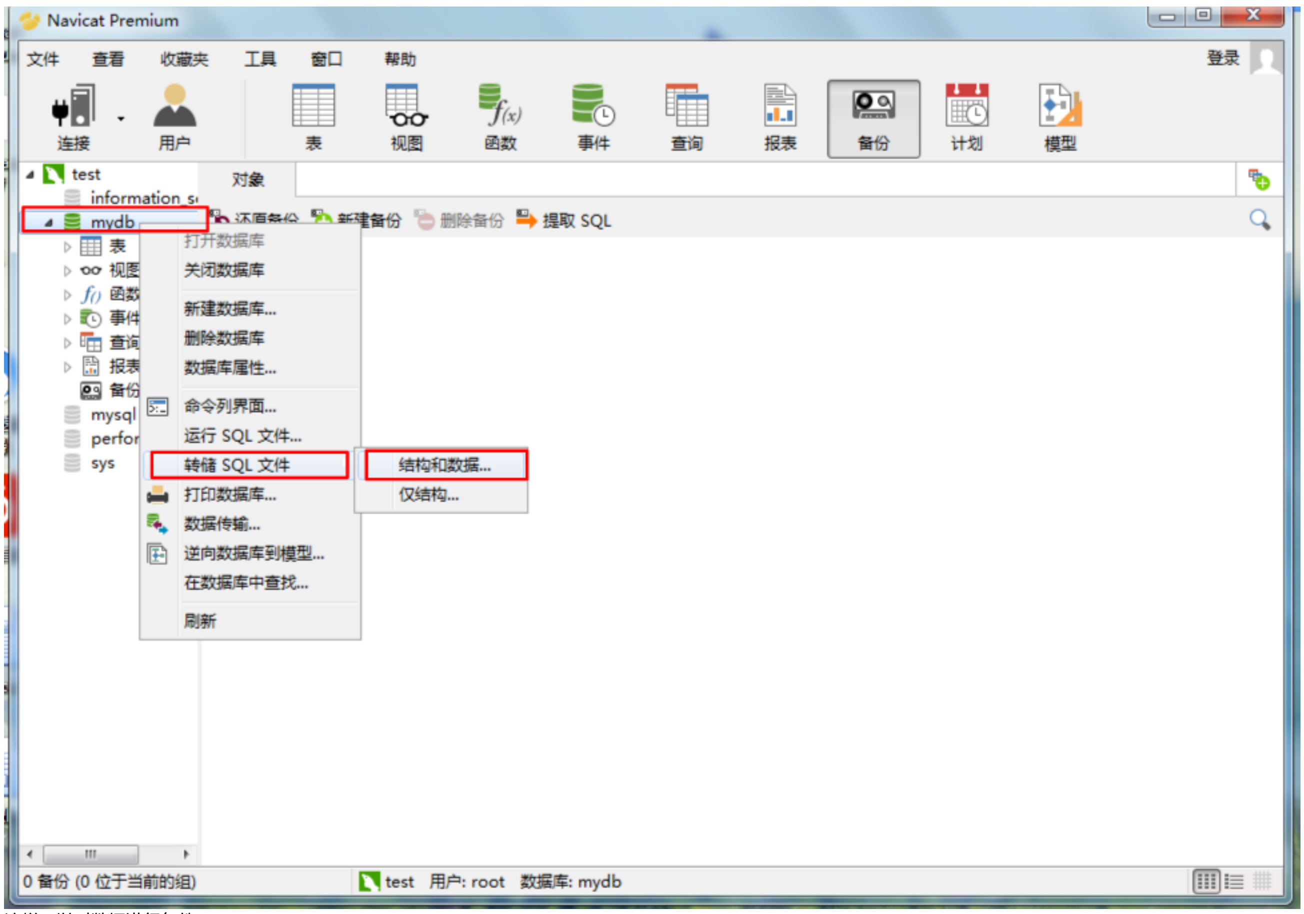
```
mysql> \s
--------------
mysql  Ver 14.14 Distrib 5.7.22, for Linux (x86_64) using  EditLine wrapper

Connection id:        3
Current database:    mydb
Current user:         root@localhost
SSL:            Not in use
Current pager:        stdout
Using outfile:        ''
Using delimiter:    ;
```

```
Server version:        5.7.22-0ubuntu0.16.04.1 (Ubuntu)
Protocol version:    10
Connection:        Localhost via UNIX socket
Server characterset:    utf8
Db     characterset:    utf8
Client characterset:    utf8
Conn.  characterset:    utf8
UNIX socket:        /var/run/mysqld/mysqld.sock
Uptime:            1 hour 1 min 9 sec

Threads: 3  Questions: 442  Slow queries: 0  Opens: 149  Flush tables: 1  Open tables: 62  Queries per second avg: 0.120
--------------
```

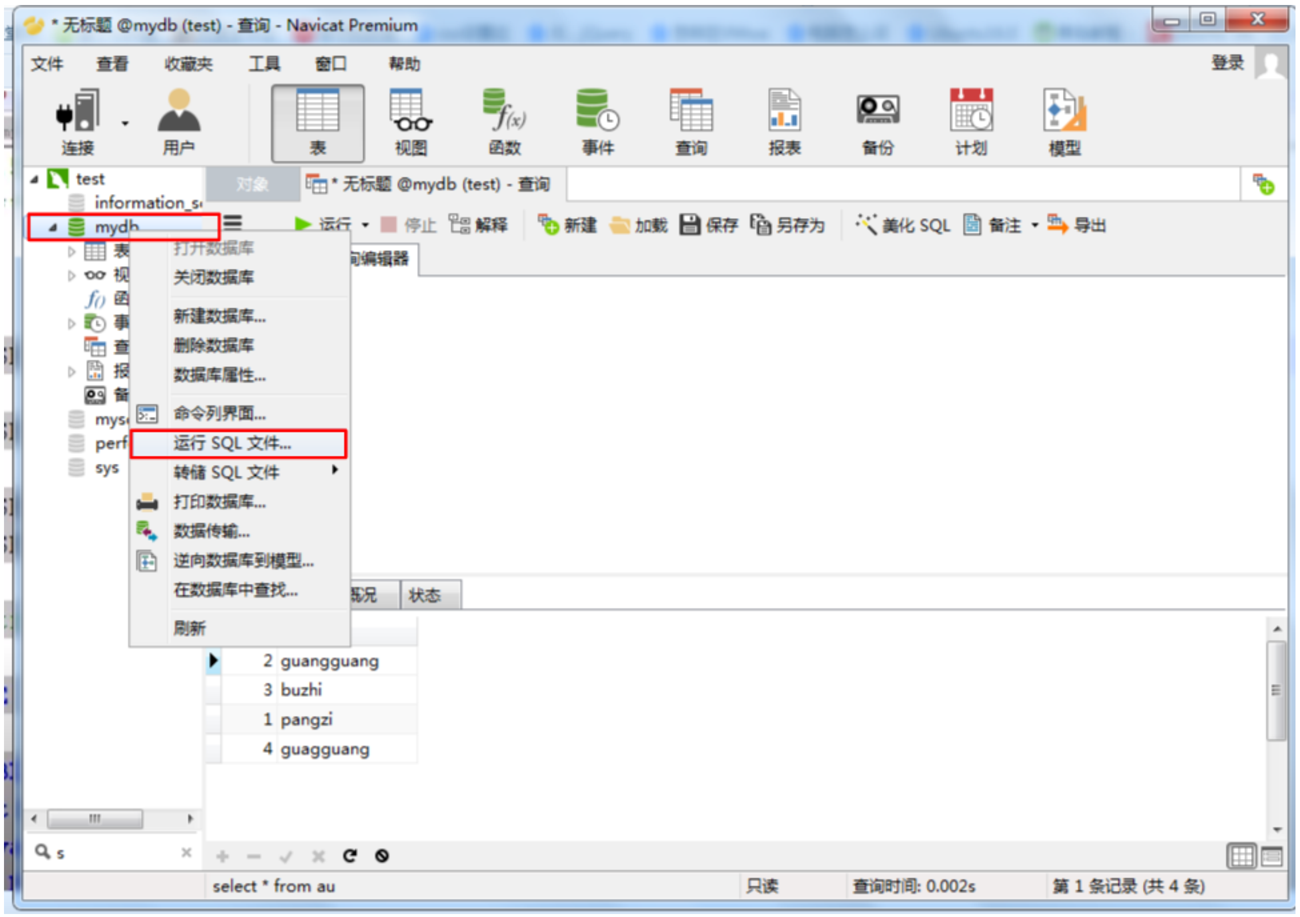
```
Client characterset:    utf8
Conn.  characterset:    utf8
```

这两个字符集编码可以通过命令对其进行改变

```
mysql> set names latin1;
Query OK, 0 rows affected (0.00 sec)

mysql> \s
--------------
mysql  Ver 14.14 Distrib 5.7.22, for Linux (x86_64) using  EditLine wrapper

Connection id:        3
Current database:    mydb
Current user:        root@localhost
SSL:            Not in use
Current pager:        stdout
Using outfile:        ''
Using delimiter:    ;
Server version:        5.7.22-0ubuntu0.16.04.1 (Ubuntu)
Protocol version:    10
Connection:        Localhost via UNIX socket
Server characterset:    latin1
Db     characterset:    latin1
Client characterset:    latin1
Conn.  characterset:    latin1
UNIX socket:        /var/run/mysqld/mysqld.sock
Uptime:            54 min 3 sec

Threads: 4  Questions: 93  Slow queries: 0  Opens: 121  Flush tables: 1  Open tables: 40  Queries per second avg: 0.028
--------------

mysql> set names utf8
    -> ;
Query OK, 0 rows affected (0.02 sec)

mysql> \s
--------------
mysql  Ver 14.14 Distrib 5.7.22, for Linux (x86_64) using  EditLine wrapper

Connection id:        3
Current database:    mydb
Current user:        root@localhost
SSL:            Not in use
Current pager:        stdout
Using outfile:        ''
Using delimiter:    ;
Server version:        5.7.22-0ubuntu0.16.04.1 (Ubuntu)
Protocol version:    10
Connection:        Localhost via UNIX socket
Server characterset:    latin1
Db     characterset:    latin1
Client characterset:    utf8
Conn.  characterset:    utf8
UNIX socket:        /var/run/mysqld/mysqld.sock
Uptime:            55 min 7 sec

Threads: 4  Questions: 97  Slow queries: 0  Opens: 121  Flush tables: 1  Open tables: 40  Queries per second avg: 0.029
--------------

mysql>
```



这样可以对数据进行备份。


* 无标题 @mydb (test) - 查询 - Navicat Premium
文件 查看 收藏夹 工具 窗口 帮助 登录
连接 用户 表 视图 函数 事件 查询 报表 备份 计划 模型
test
information_s
mydb
对象
* 无标题 @mydb (test) - 查询
运行 停止 解释 新建 加载 保存 另存为 美化 SQL 备注 导出
打开数据库
关闭数据库
新建数据库...
删除数据库
数据库属性...
命令列界面...
运行 SQL 文件...
转储 SQL 文件
打印数据库...
数据传输...
逆向数据库到模型...
在数据库中查找...
刷新
状态

```
2 guangguang
3 buzhi
1 pangzi
4 guagguang
```

select * from au | 只读 | 查询时间: 0.002s | 第 1 条记录 (共 4 条)

这样可以恢复数据。

```
mysql> select ntid,count(*) from news group by ntid;
+------+----------+
| ntid | count(*) |
+------+----------+
|    1 |        1 |
|    3 |        1 |
|    4 |        4 |
+------+----------+
3 rows in set (0.14 sec)

mysql> select * from news;
+----+------------------------------------------------------------+------+----------------------------------------------+-------+---------+
| id | title                                                      | ntid | content                                      | click | addtime |
+----+------------------------------------------------------------+------+----------------------------------------------+-------+---------+
|  1 | 骑共享单车犯法                                              |    1 | 这个事情官方还没有做出回应                      |    20 |    NULL |
|  2 | 武警特战排爆手                                              |    4 | 这个这个事情官方还没有做出回应                   |    32 |    NULL |
|  3 | 国外武装直升机型号发展与作战能力分析                          |    4 | 这个这个事情官方还没有做出回应                   |   234 |    NULL |
|  4 | 俄战机过去一周在边境拦截外国侦查机11次                        |    4 | 这个事情官方还没有做出回应                      |    21 |    NULL |
|  5 | 中国空军战机进行远洋训练                                     |    4 | 事情官方还没有做出回应                         |   128 |    NULL |
|  6 | 中国驻韩使馆举行中韩建交25周年纪念招待会                      |    3 | 事情官方还没有做出回应                         |    34 |    NULL |
+----+------------------------------------------------------------+------+----------------------------------------------+-------+---------+
6 rows in set (0.00 sec)

mysql> select ntid,count(*) from news group by ntid order by count(*) desc;
+------+----------+
| ntid | count(*) |
+------+----------+
|    4 |        4 |
|    1 |        1 |
|    3 |        1 |
+------+----------+
3 rows in set (0.06 sec)

mysql> select news.ntid,count(*),ntype.name from news,ntype where news.ntid=ntypp e.id group by news.ntid order by count(*) desc;
+------+----------+--------------+
| ntid | count(*) | name         |
+------+----------+--------------+
|    4 |        4 | 军事新闻     |
|    1 |        1 | 娱乐新闻     |
|    3 |        1 | 国际新闻     |
+------+----------+--------------+
3 rows in set (0.24 sec)

mysql>
```

MySQL的高级操作：
特殊查询及表的操作
sql内置函数，事务处理和触发器
1. MySQL的表复制
   复制表结构
   mysql> create table 目标表名 like 原表名;

   复制表数据
   mysql> insert into 目标表名 select * from 原表名;

2. 数据表的索引
   创建索引
   CREATE INDEX index_name ON table_name (column_list)
   CREATE UNIQUE INDEX index_name ON table_name (column_list)

   删除索引
   DROP INDEX index_name ON talbe_name

3. mysql视图
   创建视图:
   mysql> create view v_t1 as select * from t1 where id>4 and id<11;
   Query OK, 0 rows affected (0.00 sec)

   view视图的帮助信息:
   mysql> ? view
   ALTER VIEW
   CREATE VIEW
   DROP VIEW

   查看视图:
   mysql> show tables;

   删除视图v_t1:
   mysql> drop view v_t1;

4. MySQL的内置函数
   字符串处理函数
   ----
   *concat(s1,s2,…Sn) 连接s1,s2..Sn为一个字符串
   insert(str,x,y,instr)将字符串str从第xx位置开始，y字符串的子字符串替换为字符串str
   lower(str)将所有的字符串变为小写
   upper(str)将所有的字符串变为大写
   left(str,x)返回字符串中最左边的x个字符
   rigth(str,y)返回字符串中最右边的x个字符
   lpad(str,n,pad)用字符串pad对str最左边进行填充，直到长度为n个字符串长度

rpad(str,n,pad)用字符串pad对str最右边进行填充，直到长度为n个字符串长度
trim(str) 去掉左右两边的空格
ltrim(str) 去掉字符串str左侧的空格
rtrim(str) 去掉字符串str右侧的空格
repeat(str,x)   返回字符串str重复x次
replace(str,a,b)将字符串的的a替换成b
strcmp(s1,s2)   比较字符串s1和s2
substring(s,x,y)返回字符串指定的长度
*length(str) 返回值为字符串str 的长度

数值函数
----
*abs(x)   返回x的绝对值
ceil(x)   返回大于x的最小整数值
floor(x) 返回小于x的最大整数值
mod(x,y) 返回x/y的取余结果
rand()   返回0~1之间的随机数
*round(x,y)返回参数x的四舍五入的有y位小数的值
truncate(x,y) 返回x截断为y位小数的结果

日期和时间函数
----
curdate() 返回当前日期,按照’YYYY-MM-DD’格式
curtime() 返回当前时间,当前时间以'HH:MM:SS'
*now()   返回当前日期和时间,
*unix_timestamp(date) 返回date时间的unix时间戳
from_unixtime(unix_timestamp[,format])   返回unix时间的时间
week(date)   返回日期是一年中的第几周
year(date)   返回日期的年份
hour(time)   返回time的小时值
minute(time)   返回日time的分钟值
monthname(date) 返回date的月份
*date_fomat(date,fmt) 返回按字符串fmt格式化日期date值
date_add(date,INTERVAL,expr type) 返回一个日期或者时间值加上一个时间间隔的时间值
*datediff(expr,expr2)   返回起始时间和结束时间的间隔天数

//统计时间戳647583423距离当前时间相差天数（生日天数（不考虑年份））
mysql> select datediff(date_format(from_unixtime(647583423),"2017-%m-%d %h:%i:%s"),now());

其他常用函数
----
*database() 返回当前数据库名
version()   返回当前服务器版本
user()   返回当前登陆用户名
inet_aton   返回当前IP地址的数字表示 inet_aton("192.168.80.250");
inet_ntoa(num) 返回当前数字表示的ip   inet_ntoa(3232256250);
*password(str) 返回当前str的加密版本
*md5(str)   返回字符串str的md5值

5. MySQL的事务处理
关闭自动提交功能（开启手动事务）
mysql> set autocommit=0;
从表t1中删除了一条记录
mysql> delete from t1 where id=11;
此时做一个p1还原点:
mysql> savepoint p1;
再次从表t1中删除一条记录:
mysql> delete from t1 where id=10;
再次做一个p2还原点:
mysql> savepoint p2;
此时恢复到p1还原点，当然后面的p2这些还原点自动会失效:
mysql> rollback to p1;
退回到最原始的还原点:
mysql> rollback;
回滚

开启自动事务提交（关闭手动事务）
mysql> set autocommit=1;

6. MySQL的触发器
格式：1、触发器的定义：
```
CREATE TRIGGER trigger_name trigger_time trigger_event
  ON tbl_name FOR EACH ROW trigger_stmt
```

说明：

```
    # trigger_name：触发器名称
    # trigger_time:触发时间，可取值：BEFORE或AFTER
    # trigger_event：触发事件，可取值：INSERT、UPDATE或DELETE。
    # tb1_name：指定在哪个表上
    # trigger_stmt：触发处理SQL语句。
```

示例：

```
    mysql> delimiter $$
    mysql> create trigger del_stu before delete on stu for each row
        -> begin
        ->  insert into stu_bak values(old.id,old.name,old.sex,old.age,old.addtime);
        -> end;
        -> $$
    Query OK, 0 rows affected (0.05 sec)

    mysql> delimiter ;
```

7. mysql日志
日志：
数据库的日志必须开启着才可以恢复和找回数据。
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
这个是MySQL的配置文件位置。

查看是不会计入到日志的。
show status like "com_%"
show status

主数据库主要进行增删改操作，从数据库主要进行查看操作，主要是通过bin-log日志进行跟踪数据的。

cd bin
ls
有一个mysqlbinlog 可以运行日志，可以实现数据的完整性恢复
  开启日志： 在mysql配置文件中开启：log-bin=mysql-bin

  查看bin-log日志:
  mysql>show binary logs;

  查看最后一个bin-log日志:
  mysql>show master status;

  此时就会多一个最新的bin-log日志
  mysql>flush logs;

  查看最后一个bin日志.
  mysql>show master status;

  mysql>reset master;
  清空所有的bin-log日志
  执行查看bin-log日志

  备份数据:
  mysqldump -uroot -pwei test -l -F '/tmp/test.sql'
  其中：-F即flush logs，可以重新生成新的日志文件，当然包括log-bin日志

  // Linux关闭MySQL的命令
  $mysql_dir/bin/mysqladmin -uroot -p shutdown
  // linux启动MySQL的命令
  $mysql_dir/bin/mysqld_safe &

8、有关慢查询操作：
  开户和设置慢查询时间:
  vi /etc/my.cnf
  log_slow_queries=slow.log
  long_query_time=5
  查看设置后是否生效
  mysql> show variables like "%quer%";
  慢查询次数:
  mysql> show global status like "%quer%";

9 数据库的恢复

  1. 首先恢复最后一次的备份完整数据
  [root@localhost mnt]# mysql -u root -p mydemo<mydemo_2017-7-26.sql
  Enter password:

2. 查看bin-log日志
[root@localhost data]# mysqlbinlog --no-defaults mysql-bin.000009;
  查找到恢复的节点

3. 执行bin-log日志文件，恢复最后一块的增量数据。
[root@localhost data]# mysqlbinlog --no-defaults --stop-position="802" mysql-bin.000009|mysql -u root -p123456 mydemo;

例子：

```
mysql> select * from stu;
+----+-----------+------+-----+---------+
| id | name      | age  | sex | classid |
+----+-----------+------+-----+---------+
|  1 | zhangsan  |   20 | w   |       1 |
|  2 | lisi      |   25 | m   |       2 |
|  3 | wangwu    |   22 | w   |       5 |
|  4 | zhaoliu   |   21 | m   |       4 |
|  5 | uu01      |   27 | w   |       1 |
|  6 | uu02      |   25 | m   |       2 |
|  7 | uu03      |   28 | w   |       2 |
|  8 | uu05      |   22 | m   |       4 |
|  9 | xiaoli    |   29 | w   |       2 |
| 10 | xiaozhang |   19 | w   |       1 |
| 11 | xiaoyan   |   22 | m   |       2 |
| 12 | xiaoxin   |   28 | w   |       4 |
| 13 | wangwen   |   27 | w   |       2 |
| 14 | zhangle   |   29 | m   |       5 |
+----+-----------+------+-----+---------+
14 rows in set (0.00 sec)

mysql> create table stu2 like stu;
Query OK, 0 rows affected (0.10 sec)

mysql> desc stu;
+---------+--------------------+------+-----+---------+----------------+
| Field   | Type               | Null | Key | Default | Extra          |
+---------+--------------------+------+-----+---------+----------------+
| id      | int(10) unsigned   | NO   | PRI | NULL    | auto_increment |
| name    | varchar(16)        | NO   | UNI | NULL    |                |
| age     | tinyint(3) unsigned | YES |     | NULL    |                |
| sex     | enum('w','m')      | NO   |     | w       |                |
| classid | int(10) unsigned   | YES  |     | NULL    |                |
+---------+--------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> desc stu2;
+---------+--------------------+------+-----+---------+----------------+
| Field   | Type               | Null | Key | Default | Extra          |
+---------+--------------------+------+-----+---------+----------------+
| id      | int(10) unsigned   | NO   | PRI | NULL    | auto_increment |
| name    | varchar(16)        | NO   | UNI | NULL    |                |
| age     | tinyint(3) unsigned | YES |     | NULL    |                |
| sex     | enum('w','m')      | NO   |     | w       |                |
| classid | int(10) unsigned   | YES  |     | NULL    |                |
+---------+--------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> select * from stu2;
Empty set (0.01 sec)

mysql> insert into stu2 select * from stu limit 5;
Query OK, 5 rows affected (0.05 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from stu2;
+----+----------+------+-----+---------+
| id | name     | age  | sex | classid |
+----+----------+------+-----+---------+
|  1 | zhangsan |   20 | w   |       1 |
|  2 | lisi     |   25 | m   |       2 |
|  3 | wangwu   |   22 | w   |       5 |
|  4 | zhaoliu  |   21 | m   |       4 |
|  5 | uu01     |   27 | w   |       1 |
+----+----------+------+-----+---------+
5 rows in set (0.00 sec)

mysql> desc stu;
+---------+--------------------+------+-----+---------+----------------+
| Field   | Type               | Null | Key | Default | Extra          |
+---------+--------------------+------+-----+---------+----------------+
| id      | int(10) unsigned   | NO   | PRI | NULL    | auto_increment |
| name    | varchar(16)        | NO   | UNI | NULL    |                |
| age     | tinyint(3) unsigned | YES |     | NULL    |                |
| sex     | enum('w','m')      | NO   |     | w       |                |
| classid | int(10) unsigned   | YES  |     | NULL    |                |
+---------+--------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> create index index_age on stu(age);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> desc stu;
+---------+---------------------+------+-----+---------+----------------+
| Field   | Type                | Null | Key | Default | Extra          |
+---------+---------------------+------+-----+---------+----------------+
| id      | int(10) unsigned    | NO   | PRI | NULL    | auto_increment |
| name    | varchar(16)         | NO   | UNI | NULL    |                |
| age     | tinyint(3) unsigned | YES  | MUL | NULL    |                |
| sex     | enum('w','m')       | NO   |     | w       |                |
| classid | int(10) unsigned    | YES  |     | NULL    |                |
+---------+---------------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)

mysql> drop index index_age on stu;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc stu;
+---------+---------------------+------+-----+---------+----------------+
| Field   | Type                | Null | Key | Default | Extra          |
+---------+---------------------+------+-----+---------+----------------+
| id      | int(10) unsigned    | NO   | PRI | NULL    | auto_increment |
| name    | varchar(16)         | NO   | UNI | NULL    |                |
| age     | tinyint(3) unsigned | YES  |     | NULL    |                |
| sex     | enum('w','m')       | NO   |     | w       |                |
| classid | int(10) unsigned    | YES  |     | NULL    |                |
+---------+---------------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)

mysql> alter table stu add index index_age(age);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc stu;
+---------+---------------------+------+-----+---------+----------------+
| Field   | Type                | Null | Key | Default | Extra          |
+---------+---------------------+------+-----+---------+----------------+
| id      | int(10) unsigned    | NO   | PRI | NULL    | auto_increment |
| name    | varchar(16)         | NO   | UNI | NULL    |                |
| age     | tinyint(3) unsigned | YES  | MUL | NULL    |                |
| sex     | enum('w','m')       | NO   |     | w       |                |
| classid | int(10) unsigned    | YES  |     | NULL    |                |
+---------+---------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> drop index index_age on stu;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> select * from stu order by age desc limit 3;
+----+---------+------+-----+---------+
| id | name    | age  | sex | classid |
+----+---------+------+-----+---------+
| 14 | zhangle |   29 | m   |       5 |
|  9 | xiaoli  |   29 | w   |       2 |
| 12 | xiaoxin |   28 | w   |       4 |
+----+---------+------+-----+---------+
3 rows in set (0.00 sec)

mysql> create view vstu as select * from stu order by age desc limit 3;
Query OK, 0 rows affected (0.05 sec)



mysql> select * from vstu;
+----+---------+------+-----+---------+
| id | name    | age  | sex | classid |
+----+---------+------+-----+---------+
| 14 | zhangle |   29 | m   |       5 |
|  9 | xiaoli  |   29 | w   |       2 |
| 12 | xiaoxin |   28 | w   |       4 |
+----+---------+------+-----+---------+
3 rows in set (0.00 sec)

mysql> select * from vstu where sex="w";
+----+---------+------+-----+---------+
| id | name    | age  | sex | classid |
+----+---------+------+-----+---------+
|  9 | xiaoli  |   29 | w   |       2 |
| 12 | xiaoxin |   28 | w   |       4 |
+----+---------+------+-----+---------+
2 rows in set (0.00 sec)

mysql> select * from stu;
+----+----------+------+-----+---------+
| id | name     | age  | sex | classid |
+----+----------+------+-----+---------+
|  1 | zhangsan |   20 | w   |       1 |
|  2 | lisi     |   25 | m   |       2 |
|  3 | wangwu   |   22 | w   |       5 |
|  4 | zhaoliu  |   21 | m   |       4 |
|  5 | uu01     |   27 | w   |       1 |
|  6 | uu02      |   25 | m   |       2 |
|  7 | uu03      |   28 | w   |       2 |
|  8 | uu05      |   22 | m   |       4 |
|  9 | xiaoli    |   29 | w   |       2 |
| 10 | xiaozhang |   19 | w   |       1 |
| 11 | xiaoyan   |   22 | m   |       2 |
| 12 | xiaoxin   |   28 | w   |       4 |
| 13 | wangwen   |   27 | w   |       2 |
| 14 | zhangle   |   29 | m   |       5 |
+----+-----------+------+-----+---------+
14 rows in set (0.00 sec)

mysql> update stu set age=39 where id=10;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from vstu;
+----+-----------+------+-----+---------+
| id | name      | age  | sex | classid |
+----+-----------+------+-----+---------+
| 10 | xiaozhang |   39 | w   |       1 |
|  9 | xiaoli    |   29 | w   |       2 |
| 14 | zhangle   |   29 | m   |       5 |
+----+-----------+------+-----+---------+
3 rows in set (0.00 sec)

mysql> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| classes        |
| d1             |
| d2             |
| d3             |
| dd             |
| news           |
| ntype          |
| stu            |
| stu2           |
| type           |
| uu             |
| vstu           |
+----------------+
12 rows in set (0.00 sec)

mysql> drop table vstu;
ERROR 1051 (42S02): Unknown table 'vstu'
mysql> drop view vstu;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

事务：

```
mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from stu where classid=0;
Query OK, 1 row affected (0.00 sec)

mysql> delete from stu where id>10;
Query OK, 4 rows affected (0.07 sec)

mysql> select * from stu;
+----+----------+------+-----+---------+
| id | name     | age  | sex | classid |
+----+----------+------+-----+---------+
|  1 | zhangsan |   20 | w   |       1 |
|  2 | lisi     |   25 | m   |       2 |
|  3 | wangwu   |   22 | w   |       5 |
|  4 | zhaoliu  |   21 | m   |       4 |
|  5 | uu01     |   27 | w   |       1 |
|  6 | uu02     |   25 | m   |       2 |
|  7 | uu03     |   28 | w   |       2 |
|  8 | uu05     |   22 | m   |       4 |
|  9 | xiaoli   |   29 | w   |       2 |
+----+----------+------+-----+---------+
9 rows in set (0.00 sec)

mysql> update stu sex='m';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '='m'' at line 1
mysql> update stu set sex='m';
Query OK, 5 rows affected (0.00 sec)
Rows matched: 9  Changed: 5  Warnings: 0

mysql> select * from stu;
+----+----------+------+-----+---------+
| id | name     | age  | sex | classid |
+----+----------+------+-----+---------+
|  1 | zhangsan |   20 | m   |       1 |
|  2 | lisi     |   25 | m   |       2 |
|  3 | wangwu   |   22 | m   |       5 |
|  4 | zhaoliu  |   21 | m   |       4 |
|  5 | uu01     |   27 | m   |       1 |
|  6 | uu02     |   25 | m   |       2 |
|  7 | uu03     |   28 | m   |       2 |
|  8 | uu05     |   22 | m   |       4 |
|  9 | xiaoli   |   29 | m   |       2 |
+----+----------+------+-----+---------+
9 rows in set (0.00 sec)

mysql> rollback;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from stu;
+----+-----------+------+-----+---------+
| id | name      | age  | sex | classid |
+----+-----------+------+-----+---------+
|  1 | zhangsan  |   20 | w   |       1 |
|  2 | lisi      |   25 | m   |       2 |
|  3 | wangwu    |   22 | w   |       5 |
|  4 | zhaoliu   |   21 | m   |       4 |
|  5 | uu01      |   27 | w   |       1 |
|  6 | uu02      |   25 | m   |       2 |
|  7 | uu03      |   28 | w   |       2 |
|  8 | uu05      |   22 | m   |       4 |
|  9 | xiaoli    |   29 | w   |       2 |
| 10 | xiaozhang |   22 | w   |       0 |
| 11 | xiaoyan   |   22 | m   |       2 |
| 12 | xiaoxin   |   28 | w   |       4 |
| 13 | wangwen   |   27 | w   |       2 |
| 14 | zhangle   |   29 | m   |       5 |
+----+-----------+------+-----+---------+
14 rows in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> rollback;

--随机函数的应用
mysql> select * from type order by rand() desc limit 3;
+------+-----------+------+
| id   | name      | pid  |
+------+-----------+------+
|    9 | 食品      |    0 |
|    8 | 童装      |    1 |
|   12 | 休闲装    |    1 |
+------+-----------+------+
3 rows in set (0.52 sec)

mysql> select * from type order by rand() desc limit 3;
+------+--------+------+
| id   | name   | pid  |
+------+--------+------+
|    4 | 手机   |    2 |
|    7 | 女装   |    1 |
|    9 | 食品   |    0 |
+------+--------+------+
3 rows in set (0.00 sec)

mysql> select *,rand() from type order by rand() desc limit 3;
+------+--------+------+--------------------+
| id   | name   | pid  | rand()             |
+------+--------+------+--------------------+
|    5 | 相机   |    2 | 0.9624774539493746 |
|    2 | 数码   |    0 | 0.9208985110008144 |
|   11 | 特产   |    9 | 0.7281597077177462 |
+------+--------+------+--------------------+
3 rows in set (0.00 sec)
```

Python3 MySQL 数据库连接
===

1. 什么是 PyMySQL？
   PyMySQL 是在 Python3.x 版本中用于连接 MySQL 服务器的一个库，Python2中则使用mysqldb。
   PyMySQL 遵循 Python 数据库 API v2.0 规范，并包含了 pure-Python MySQL 客户端库。

2. PyMySQL安装

   PyMySQL下载地址：https://github.com/PyMySQL/PyMySQL。

   2.1 使用pip命令进行安装：

```
   $ pip install PyMySQL
```

   2.2 使用 git 命令下载安装包安装(你也可以手动下载)：

```
   $ git clone https://github.com/PyMySQL/PyMySQL
   $ cd PyMySQL/
   $ python3 setup.py install
```

3. 数据库连接

  通过如下代码测试数据库连接

```
   #!/usr/bin/python3

   import pymysql
```

```
    # 打开数据库连接
    db = pymysql.connect("localhost","root","123456","mydb" )

    # 使用 cursor() 方法创建一个游标对象 cursor
    cursor = db.cursor()

    # 使用 execute()  方法执行 SQL 查询
    cursor.execute("SELECT VERSION()")

    # 使用 fetchone() 方法获取单条数据.
    data = cursor.fetchone()

    print ("Database version : %s " % data)

    # 关闭数据库连接
    db.close()
```

4. 执行数据查询：

```
    #!/usr/bin/python3

    import pymysql

    # 打开数据库连接
    db = pymysql.connect("localhost","root","","mydemo" )

    # 使用 cursor() 方法创建一个游标对象 cursor
    cursor = db.cursor()

    # SQL 查询语句
    sql = "select * from stu limit %d" % (3)
    #sql = "select * from stu"

    try:
       # 执行SQL语句
       cursor.execute(sql)
       # 获取所有记录列表
       results = cursor.fetchall()
       for row in results:
          id = row[0]
          name = row[1]
          sex = row[2]
          age = row[3]
          classid = row[4]
           # 打印结果
          print ("id=%d,name=%s,sex=%s,age=%d,classid=%s" % (id,name,sex,age,classid))
    except:
       print ("Error: unable to fetch data")

    # 关闭数据库连接
    db.close()
```

5. 执行数据添加

```
#!/usr/bin/python3

import pymysql

# 打开数据库连接
db = pymysql.connect("localhost","root","","mydemo" )

# 使用 cursor() 方法创建一个游标对象 cursor
cursor = db.cursor()

# SQL 插入语句
sql = "INSERT INTO stu(name,sex,age,classid) values('%s','%c','%d','%s')" % ('uu142','m',22,'lamp180')

try:
   # 执行sql语句
   cursor.execute(sql)
   # 执行sql语句
db.commit()
   print("ok: %d " % (cursor.rowcount))
except:
   # 发生错误时回滚
   db.rollback()

# 关闭数据库连接
db.close()
```

6. 执行删除操作

```
    #!/usr/bin/python3

    import pymysql

    # 打开数据库连接
    db = pymysql.connect("localhost","root","","mydemo" )

    # 使用 cursor() 方法创建一个游标对象 cursor
    cursor = db.cursor()
    # SQL 删除语句
    sql = "delete from stu where id = '%d'" % (13)
    try:
       # 执行SQL语句
       cursor.execute(sql)
       # 提交修改
db.commit()
    except:
       # 发生错误时回滚
       db.rollback()

    # 关闭数据库连接
    db.close()
```

数据库查询操作：
    Python查询Mysql使用 fetchone() 方法获取单条数据, 使用fetchall() 方法获取多条数据。
        fetchone(): 该方法获取下一个查询结果集。结果集是一个对象
        fetchall(): 接收全部的返回结果行.
        rowcount: 这是一个只读属性，并返回执行execute()方法后影响的行数。

pip命令

列出已安装的包：

```
    $ pip list
    $ pip freeze     # 查看自己安装的
```

安装软件（安装特定版本的package，通过使用==, &gt;=, &lt;=, &gt;, &lt;来指定一个版本号）**

```
    $ pip install SomePackage
    $ pip install 'Markdown<2.0'
    $ pip install 'Markdown>2.0,<2.0.3'
```

卸载软件pip uninstall SomePackage

```
    $ pip uninstall SomePackage
```

下载所需的软件包：

```
    $ pip download SomePackage -d directory
    例如下载PyMySQL软件包
    $ pip download PyMySQL -d D:/pypackage
```

安装下载好的软件包文件

```
    $ pip install 目录/软件包文件名
    如安装PyMySQL软件包
    $ pip3.6 install D:/pypackage/PyMySQL-0.7.11-py2.py3-none-any.whl
```

MySQL服务器的用户权限管理
==

```
-- 授权一个用户（zhangsan）密码123，可以对所有的库，所有的表做所有操作。
mysql> grant all on *.* to zhangsan@'%' identified by '123';
Query OK, 0 rows affected (0.17 sec)

--刷新生效，否则就要重启MySQL服务才可以。
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

--浏览当前MySQL用户信息
mysql> select user,host,password from mysql.user;
+----------+-----------------+-------------------------------------------+
| user     | host            | password                                  |
+----------+-----------------+-------------------------------------------+
| root     | localhost       | *23AE809DDACAF96AF0FD78ED04B6A265E05AA257 |
| root     | 127.0.0.1       |                                           |
|          | localhost       |                                           |
| zhangsan | %               | *23AE809DDACAF96AF0FD78ED04B6A265E05AA257 |
| admin    | 192.168.112.132 | *23AE809DDACAF96AF0FD78ED04B6A265E05AA257 |
+----------+-----------------+-------------------------------------------+
5 rows in set (0.00 sec)

-- 移除一些权限
-- revoke:只删除了用户权限，但没有删除这个用户
mysql> revoke insert,delete on *.* from admin@192.168.112.132 identified by'123';

-- 查看指定用户的权限信息
mysql> show grants for xbb@localhost;
+-----------------------------------------------------------------------------------------------------------+
| Grants for xbb@localhost                                                                                  |
+-----------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'xbb'@'localhost' IDENTIFIED BY PASSWORD '*23AE809DDACAF96AF0FD78ED04B6A265E05AA257' |
+-----------------------------------------------------------------------------------------------------------+

--drop user:删除了整个用户及其权限（包括数据字典中的数据）
mysql> drop user 'xbb'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> select user,host from mysql.user;
+------------------+-----------+
| user             | host      |
+------------------+-----------+
| root             | 127.0.0.1 |
| debian-sys-maint | localhost |
| root             | localhost |
| root             | wangxg    |
+------------------+-----------+
4 rows in set (0.00 sec)
```

1个字节代表8个位
int(3)指的是三位数
int(4)指的是四位数
当我们在选择使用int的类型的时候，不论是int(3)还是int(11)，它在数据库里面存储的都是4个字节的长度，在使用int(3)的时候如果你输入的是10，会
就是说这个3代表的是默认的一个长度，当你不足3位时，会帮你补全，当你超过3位时，就没有任何的影响。
（ ）是宽度显示，不是储存范围

double(6,2)表示的是总共6位数字，其中2位是小数位

tinyint(4)表示从-128到127
tinyint(3） unsigned
tinyint 型的字段如果设置为UNSIGNED类型，只能存储从0到255的整数,不能用来储存负数。
tinyint 型的字段如果不设置UNSIGNED类型,存储-128到127的整数。
1个tinyint型数据只占用一个字节;一个INT型数据占用四个字节。

电脑中，最基础的单位是位（bit），只能为0或1，所有的数据由多个位的二进制组成。8位=1比特（*Byte*）=1字节，1个字节的数据存量是2的8次方，*4个*
42亿多的*数字*。更大的就需要更多的存量。C语言中常用的数值类型其实有很多，比如最常用的int就是*4个字节*，-2147483648~2147483647；但是也有其
就用不到*4个字节*。

很多东西发生过，永远被保留！

用二进制类型储存视频、声音、图片
微博是采用varchar进存储　帖子什么的用text进行存储

数字比字符串快
一般自增都带着主键　主键可以没有自增

事务是共同进退，表示的是张三把钱转给李四，需要两个sql语句去完成，如果一个sql语句成功了，另一个语句没有完成，这样是不行的，必须两者都成功
功一个失败了就不行，这样的情况就是事务。
事务是一个操作单元，必须同时进行。一般采用innodb格式的表单。

查询是不会计入日志的

编码不统一是导致乱码的最主要原因。
MySQL服务器有编码，建库建表有编码，连接库和表的时候也有编码，编码最好统一
设置编码格式：

计算机集群，每个服务器贡献2个G，那加在一起就很多了，这样就可以将数据缓存进去。在需要数据的时候就现在缓存里面找，缓存里面没有就在数据库
存里面缓存一份，这样速度就会加快了，提高性能。

数据特殊的对应关系：
比如微博上的互相关注的人，粉丝和楼主，全部用户都是在同一张表中，因此会出现另一张关联表。另一种特殊情况是比如课程和学生之间，一个学生可
被很多学生选择。这个是多对多的关系，但凡存在多对多的关系时都需要在中间添加一张关联表。

利用索引查找数据是非常方便的，比如要查10000条数据，先从第5000条进行查看判断，再折半再折半后就会找到大致的位置。索引常常出现在有where
索引相当于是在原表中添加了一列索引，提高了查询的速度，但是相应的，增删改的速度也会下降。

视图是用在数据需要频繁更新的情况下，比如最新的新闻，销量最大的商品等等情况，一旦表发生改变，视图就会相应地发生变化。

触发器：
事件有修改，添加和删除。事件在修改前后都可以，一般添加是有之后的时间。事件是在删除之前。

这个案例就是监听，当把stu的数据删的时候就挪到stu2表里面。

日志：
数据库的日志必须开启着才可以恢复和找回数据。
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
这个是MySQL的配置文件位置。

查看是不会计入到日志的。
show status like "com_%"
show status

主数据库主要进行增删改操作，从数据库主要进行查看操作，主要是通过bin-log日志进行跟踪数据的。

cd bin

ls
有一个mysqlbinlog 可以运行日志，可以实现数据的完整性恢复

执行bin-log日志文件，恢复最后一块的增量数据。

```
/var/lib/mysql# mysqlbinlog --no-defaults --stop-datetime='2018-05-12 14:11:05' binlog.000002|mysql -u root -p mydb
```

权限的设置：

```
mysql> grant all on *.* to zhangsan@'%' identified by '123';
mysql> \s;
mysql> grant all on mydb.* to lisi@'%' identified by '123';
mysql>flush privileges;
drop user zhangsan@'%';
```

先关掉mysql-server然后再开启。

pymysql

```
root@may-virtual-machine:~# python -V
Python 2.7.12
```

python -V 查看Python的版本
pip -V 查看pip的版本
pip list 查看用pip下载的软件有哪些
pip freeze 查看用pip下载的软件有哪些

使用pip命令进行安装：

```
   $ pip install PyMySQL
```

```
root@may-virtual-machine:/var/lib/python/lesson03# ls
1.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 1.py
root@may-virtual-machine:/var/lib/python/lesson03# python3 1.py
Traceback (most recent call last):
  File "1.py", line 1, in <module>
    import pymysql
ImportError: No module named 'pymysql'
root@may-virtual-machine:/var/lib/python/lesson03# python2 1.py
root@may-virtual-machine:/var/lib/python/lesson03# python -V
Python 2.7.12
root@may-virtual-machine:/var/lib/python/lesson03# python3 -V
Python 3.5.2
root@may-virtual-machine:/var/lib/python/lesson03# python3 1.py
Traceback (most recent call last):
  File "1.py", line 1, in <module>
    import pymysql
ImportError: No module named 'pymysql'
root@may-virtual-machine:/var/lib/python/lesson03# python2 1.py
root@may-virtual-machine:/var/lib/python/lesson03# vi 1.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 1.py
Database info:5.7.22-0ubuntu0.16.04.1-log
root@may-virtual-machine:/var/lib/python/lesson03# vi 1.py
root@may-virtual-machine:/var/lib/python/lesson03# cp 1.py 2.py
root@may-virtual-machine:/var/lib/python/lesson03# ls
1.py  2.py
root@may-virtual-machine:/var/lib/python/lesson03# vi 2.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 2.py
  File "2.py", line 1
SyntaxError: Non-ASCII character '\xe6' in file 2.py on line 1, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details
root@may-virtual-machine:/var/lib/python/lesson03# vi 2.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 2.py
  File "2.py", line 1
SyntaxError: Non-ASCII character '\xe6' in file 2.py on line 1, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details
root@may-virtual-machine:/var/lib/python/lesson03# ls
1.py  2.py
root@may-virtual-machine:/var/lib/python/lesson03# vi 2.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 2.py
  File "2.py", line 1
SyntaxError: Non-ASCII character '\xe6' in file 2.py on line 1, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details
root@may-virtual-machine:/var/lib/python/lesson03# vi 2.py
root@may-virtual-machine:/var/lib/python/lesson03# vi 2.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 2.py
  File "2.py", line 1
SyntaxError: Non-ASCII character '\xe6' in file 2.py on line 1, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details
root@may-virtual-machine:/var/lib/python/lesson03# ls
1.py  2.py
root@may-virtual-machine:/var/lib/python/lesson03# rm -rf 2.py
root@may-virtual-machine:/var/lib/python/lesson03# ls
1.py
root@may-virtual-machine:/var/lib/python/lesson03# cp 1.py 2.py
root@may-virtual-machine:/var/lib/python/lesson03# ls
1.py  2.py
root@may-virtual-machine:/var/lib/python/lesson03# vi 2.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 2.py
  File "2.py", line 1
SyntaxError: Non-ASCII character '\xe6' in file 2.py on line 1, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details
root@may-virtual-machine:/var/lib/python/lesson03# python3 2.py
Traceback (most recent call last):
  File "2.py", line 2, in <module>
    import pymysql
ImportError: No module named 'pymysql'
```

```
root@may-virtual-machine:/var/lib/python/lesson03# vi 2.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 2.py
Traceback (most recent call last):
  File "2.py", line 21, in <module>
    print("id=%d, name=%s, age=%d, sex=%c, classid=%d" % (id,name,age,sex,classid))
TypeError: %d format: a number is required, not str
root@may-virtual-machine:/var/lib/python/lesson03# vi 2.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 2.py
id=1, name=zsan, age=20, sex=m, classid=python03
id=2, name=lisi, age=22, sex=m, classid=python02
id=3, name=wang, age=25, sex=w, classid=python03
id=5, name=huai, age=21, sex=w, classid=python01
id=6, name=uu07, age=21, sex=w, classid=python03
id=7, name=uu01, age=18, sex=m, classid=python03
id=8, name=uu02, age=19, sex=m, classid=python02
id=9, name=uu03, age=23, sex=m, classid=python02
id=10, name=uu04, age=19, sex=m, classid=python03
id=12, name=uu06, age=25, sex=m, classid=python03
id=13, name=qq01, age=21, sex=m, classid=4
Traceback (most recent call last):
  File "2.py", line 22, in <module>
    print("Database info:%s" % data)
NameError: name 'data' is not defined
root@may-virtual-machine:/var/lib/python/lesson03# cp 2.py 3.py
root@may-virtual-machine:/var/lib/python/lesson03# ls
1.py  2.py  3.py
root@may-virtual-machine:/var/lib/python/lesson03# vi 3.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 3.py
Error
root@may-virtual-machine:/var/lib/python/lesson03# vi 3.py
root@may-virtual-machine:/var/lib/python/lesson03# python2 3.py
ok:1
```

1.py里面的代码

2.py里面的代码

```
#data analysis
import pymysql

db = pymysql.connect("localhost","root","123456","mydb")

cursor = db.cursor()

sql = "select * from stu"

cursor.execute(sql)


result = cursor.fetchall()

for row in result:
        id = row[0]
        name = row[1]
        age = row[2]
        sex = row[3]
        classid = row[4]
        print("id=%d, name=%s, age=%d, sex=%c, classid=%s" % (id,name,age,sex,classid))
"2.py" 25L, 410C
```

```
#data analysis
import pymysql

db = pymysql.connect("localhost","root","123456","mydb")

cursor = db.cursor()

sql = "insert into stu(name,age,sex,classid) values('%s', '%d', '%c', '%s')" % ('qq10',22,'w','2')

try:
        cursor.execute(sql)
        print("ok:%d" % (cursor.rowcount))

except:
        print("Error")


db.close()


~
"3.py" 20L, 315C
```

安装Samba

ubuntu16.04下安装samba

2017年07月05日 11:12:22

1、安装 samba软件包

```
sudo apt-get install samba

sudo apt-get install smbclient
```

2、启动或停止samba服务

```
sudo /etc/init.d/samba start

sudo /etc/init.d/samba stop
```

Ubuntu 16.04安装配置Samba服务

Samba是开源软件，用来让Linux系统与Windows系统的SMB/CIFS网络协定做连结，实现Windows主机与Linux服务器之间的资源共享。Samba服务为两种不同的操作系统架起了一座桥梁
统之间能够实现互相通信，为广泛的Linux爱好者提供了极大方便。

安装Samba

使用apt-get安装：

```
$ sudo apt-get install samba samba-common
```

如果你开启了防火墙，关闭：

```
$ sudo systemctl
```

配置Samba

编辑配置文件：

```
$ sudo vim /etc/samba/smb.conf
```

添加Samba共享目录：

```
[homes
   browseable
```

添加一个用户：

```
$ sudo smbpasswd
```

我这里输入的是root用户，也可以输入其他的存在用户名。

重启samba服务生效：

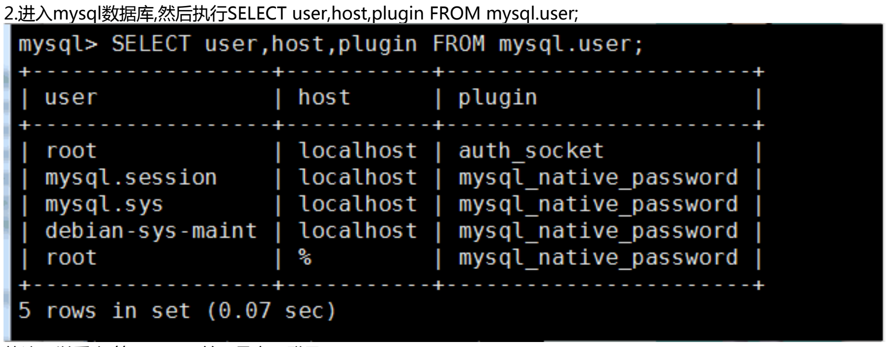
```
$ sudo systemctl restart
```

测试：在Windows下运行窗口输入\加上IP，例如：\\192.168.1.199\root。在弹出的窗口，输入刚刚添加的用户名和密码，就可以访问Linux的文件目录了。

更多信息：https://www.samba.org/

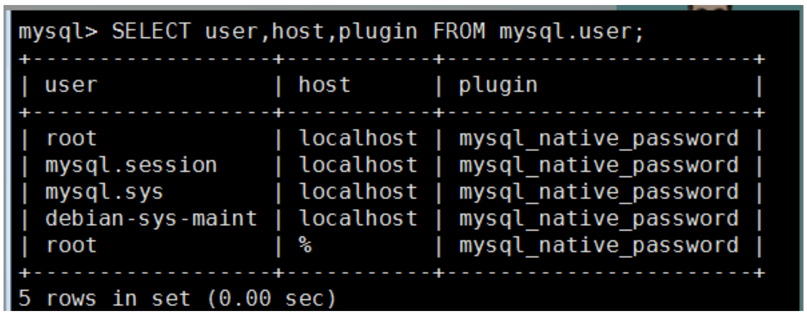
注意：

在利用pymysql连接Python和MySQL数据库的时候，也许会连不上
1.首先看Windows下的Navicat能连接上不，如果连接不上，应该对配置文件等作出修改
2.进入mysql数据库,然后执行SELECT user,host,plugin FROM mysql.user;



```
mysql> SELECT user,host,plugin FROM mysql.user;
+------------------+-----------+-----------------------+
| user             | host      | plugin                |
+------------------+-----------+-----------------------+
| root             | localhost | auth_socket           |
| mysql.session    | localhost | mysql_native_password |
| mysql.sys        | localhost | mysql_native_password |
| debian-sys-maint | localhost | mysql_native_password |
| root             | %         | mysql_native_password |
+------------------+-----------+-----------------------+
5 rows in set (0.07 sec)
```

从这可以看出 第一项root并不是密码登录
3.UPDATE mysql.user SET authentication_string=PASSWORD('Avalon'), plugin='mysql_native_password' WHERE user='root';
4.然后FLUSH PRIVILEGES;
然后重启mysql
5.进入mysql执行 SELECT user,host,plugin FROM mysql.user;


```
mysql> SELECT user,host,plugin FROM mysql.user;
+------------------+-----------+-----------------------+
| user             | host      | plugin                |
+------------------+-----------+-----------------------+
| root             | localhost | mysql_native_password |
| mysql.session    | localhost | mysql_native_password |
| mysql.sys        | localhost | mysql_native_password |
| debian-sys-maint | localhost | mysql_native_password |
| root             | %         | mysql_native_password |
+------------------+-----------+-----------------------+
5 rows in set (0.00 sec)
```

OK这样就可以了，写好Python文本 执行Python 1.py就可以了（说多了都是泪 ）