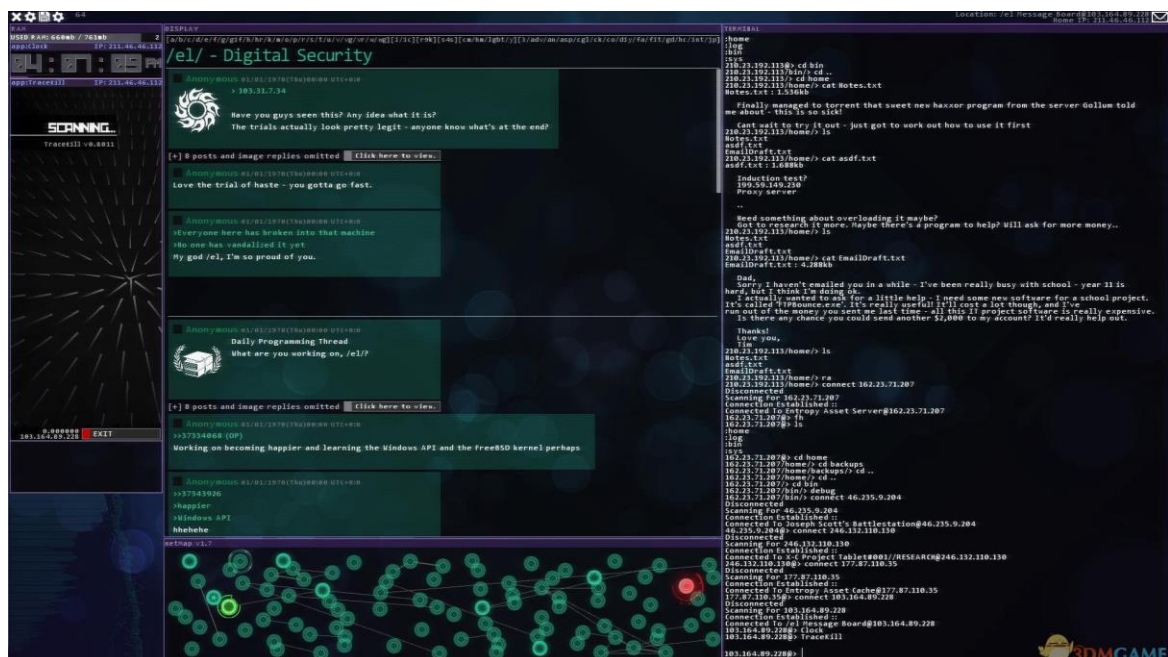


MySQL (三)

全球著名的勒索病毒:



攻击指令：这些指令用来操控一台或多台计算机

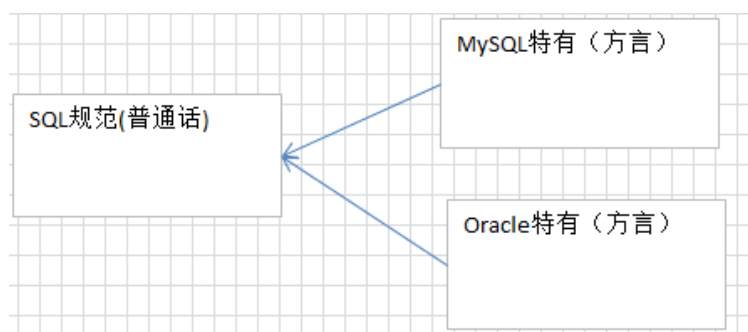


1. SQL 概述

1.1 SQL 语句介绍

数据库是不认识 PHP 语言的，但是我们同样要与数据库交互，这时需要使用到数据库认识的语言 SQL 语句，它是数据库的代码。

结构化查询语言(Structured Query Language)简称 SQL，是关系型数据库管理系统都需要遵循的规范。不同的数据库生产厂商都支持 SQL 语句，但都有特有内容。



1.2 SQL 语句分类

1.2.1 数据定义语言

简称 DDL(Data Definition Language)，用来定义数据库对象：数据库 database，表 table，列 column 等。关键字：创建 create，修改 alter，删除 drop 等（结构）

1.2.2 数据操作语言

简称 DML(Data Manipulation Language)，用来对数据库中表的记录进行更新。关键字：插入 insert，删除 delete，更新 update 等（数据）

1.2.3 数据查询语言

简称 DQL(Data Query Language)，用来查询数据库中表的记录。关键字：select，from，where 等

1.2.4 数据控制语言

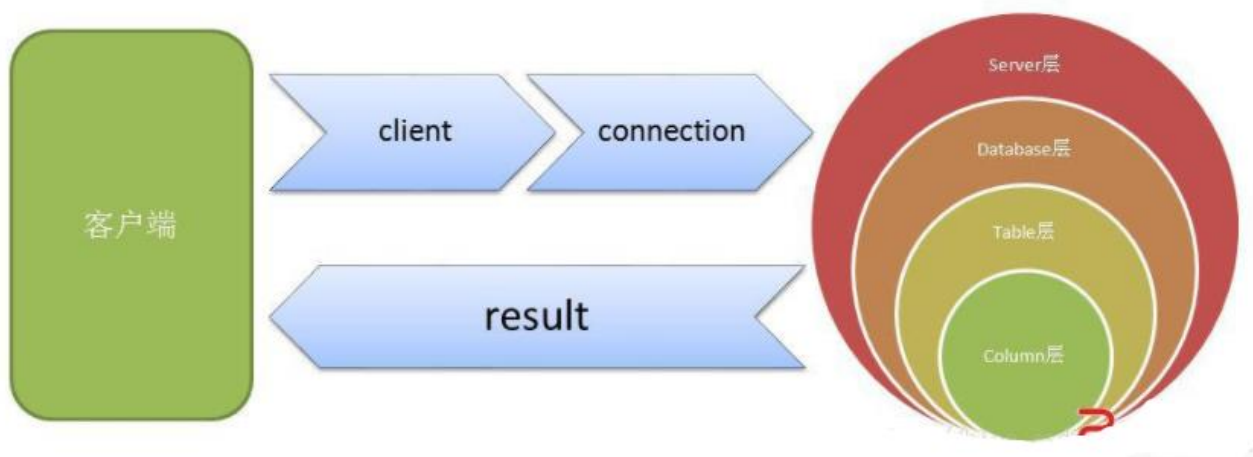
简称 DCL(Data Control Language)，用来定义数据库的访问权限和安全级别，及创建用户。

1.3 SQL 通用语法

```
SELECT * FROM `student`;
```

- SQL 语句可以单行或多行书写，以分号结尾
- 一般会使用空格和缩进来增强语句的可读性
- MySQL 数据库的 SQL 语句不区分大小写，关键字建议使用大写
例如：SELECT * FROM user。
- 可以使用“--”的方式进行注释

2. MySQL 客户端

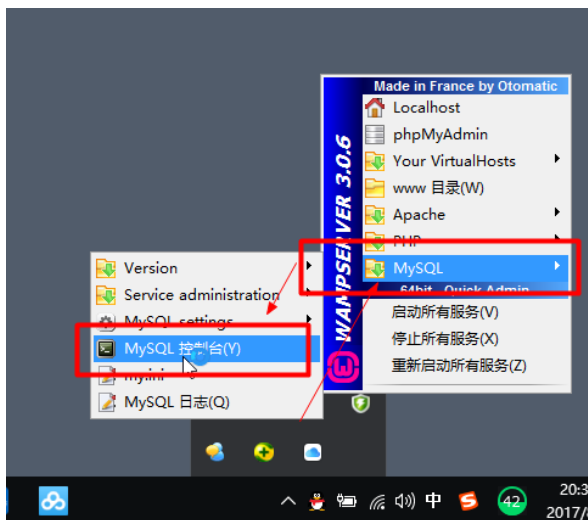


MySQL 是一个需要账户名密码登录的数据库，登陆后使用，它提供了一个默认的 root 账号，使用安装时设置的密码即可登录。

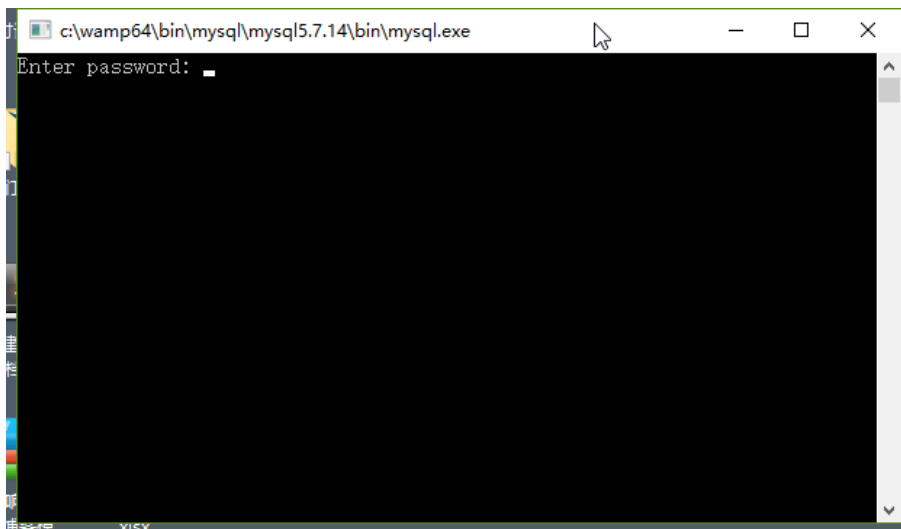
mysql.exe 客户端 （mysql 软件安装后，自带的客户端）
mysqld.exe 服务端

方式一：wamp 的方式登录：

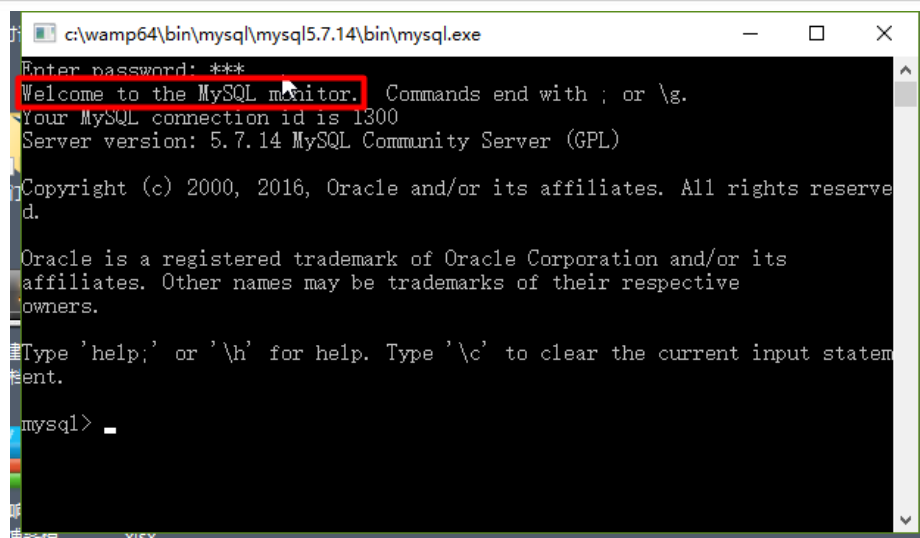
第一步：打开 MySQL 客户端



第二步：输入密码

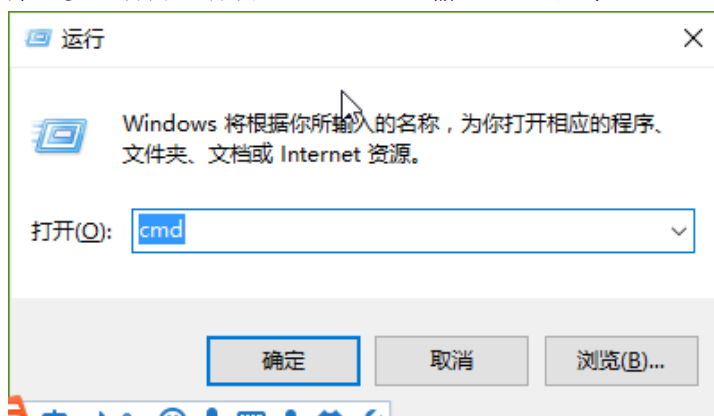


第三步：输入密码登录

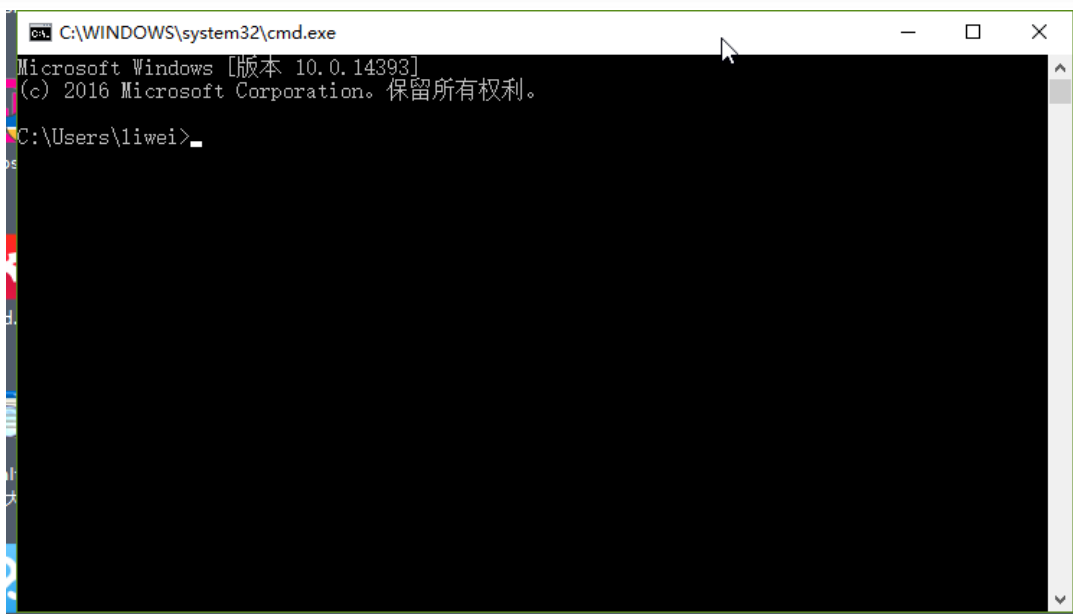


方式二：Dos 命令行中打开

第一步：打开运行窗口（win+R），输入 CMD 回车



第二步：打开 Dos 命令行



第三步：登录 mysql 服务器

格式 1: `cmd> mysql.exe 路径 -u 用户名 -p 密码`

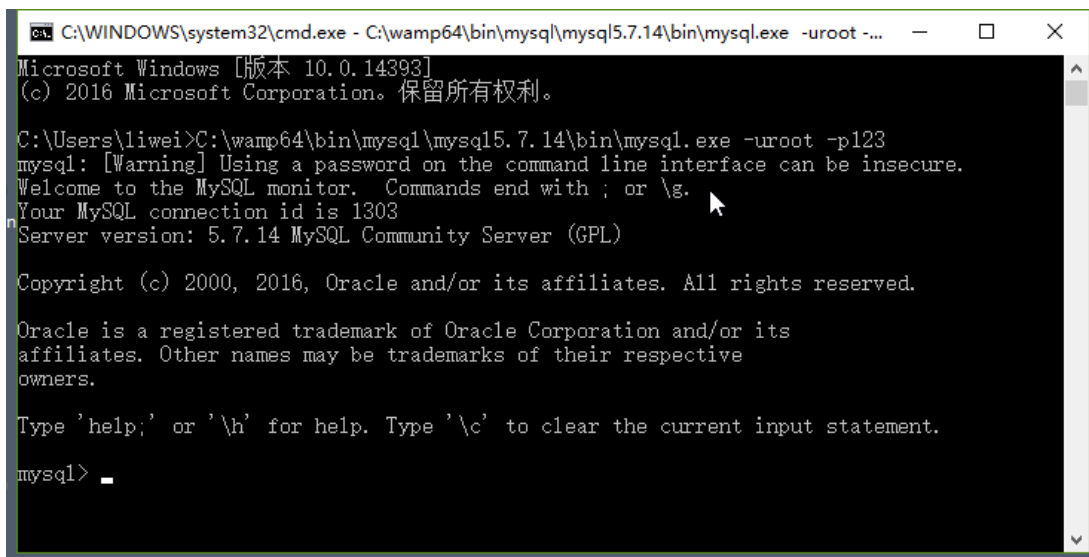
或者: `cmd> mysql.exe 路径 -u 用户名 -p 回车`

输入密码: ***

例如: `C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe -uroot -p123`

输入 `mysqld.exe` 的文件路径

如: `C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe -uroot -p123`



输入时也可以隐藏密码:


```
C:\WINDOWS\system32\cmd.exe - C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe -uroot -p
C:\Users\liwei>C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe -uroot -p
Enter password: ***
```

格式 2: cmd> mysql.exe 路径 --host=ip 地址 --user=用户名 --password=密码

例如: C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe --host=127.0.0.1
--user=root --password=123

```
C:\WINDOWS\system32\cmd.exe
C:\Users\liwei>C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe --host=127.0.0.1 --user=root --password=root
```

登录成功!

```
C:\WINDOWS\system32\cmd.exe - C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe --host=...
C:\Users\liwei>C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe --host=127.0.0.1 --user=root --password=root
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 1304
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, 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>
```

3. show 指令

3.1 显示数据库

```
show databases;
```

show databases 可以在 MySQL 服务器主机上所有数据库



```
C:\WINDOWS\system32\cmd.exe - C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe -uroot -p
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| baijia      |
| czxy_2017_db |
| mydatabase1  |
| mydb        |
| mysql       |
| performance_schema |
| sys         |
+-----+
8 rows in set (0.00 sec)

mysql>
```

3.2 显示数据库中的所有表

```
show tables;
```

show tables 用于查看数据库中的所有的表

```
mysql> use mydb;
Database changed
mysql> show tables;
+-----+
| Tables_in_mydb |
+-----+
| student        |
| test_int       |
+-----+
2 rows in set (0.00 sec)

mysql>
```

3.3 显示指定表的列

```
show columns from 表名;
```

show columns from 表名, 可以查看一个表的字段信息


```

C:\WINDOWS\system32\cmd.exe - C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe -uroot -p
student
test_db
2 rows in set (0.00 sec)

mysql> show columns from student;
+----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+----+-----+-----+-----+-----+-----+
| id    | int(10)       | NO   | PRI | NULL    | auto_increment |
| name  | varchar(30)   | YES  |     | NULL    |                |
| sex   | varchar(3)    | YES  |     | NULL    |                |
| age   | int(10)       | YES  |     | NULL    |                |
| telnum| bigint(11)    | YES  |     | NULL    |                |
| address| varchar(200) | YES  |     | NULL    |                |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
    
```

1. show tables 或 show tables from database_name; -- 显示当前数据库中所有表的名称。
2. show databases; -- 显示 mysql 中所有数据库的名称。
3. show columns from table_name from database_name; 或 show columns from database_name.table_name; -- 显示表中列名称。
4. show grants for user_name; -- 显示一个用户的权限，显示结果类似于 grant 命令。
5. show index from table_name; -- 显示表的索引。
6. show status; -- 显示一些系统特定资源的信息，例如，正在运行的线程数量。
7. show variables; -- 显示系统变量的名称和值。
8. show processlist; -- 显示系统中正在运行的所有进程，也就是当前正在执行的查询。大多数用户可以查看他们自己的进程，但是如果他们拥有 process 权限，就可以查看所有人的进程，包括密码。
9. show table status; -- 显示当前使用或者指定的 database 中的每个表的信息。信息包括表类型和表的最新更新时间。
10. show privileges; -- 显示服务器所支持的不同权限。
11. show create database database_name; -- 显示 create database 语句是否能够创建指定的数据库。
12. show create table table_name; -- 显示 create database 语句是否能够创建指定的数据库。
13. show engines; -- 显示安装以后可用的存储引擎和默认引擎。
14. show innodb status; -- 显示 innnoDB 存储引擎的状态。
15. show logs; -- 显示 BDB 存储引擎的日志。
16. show warnings; -- 显示最后一个执行的语句所产生的错误、警告和通知。
17. show errors; -- 只显示最后一个执行语句所产生的错误。
18. show [storage] engines; --显示安装后的可用存储引擎和默认引擎。

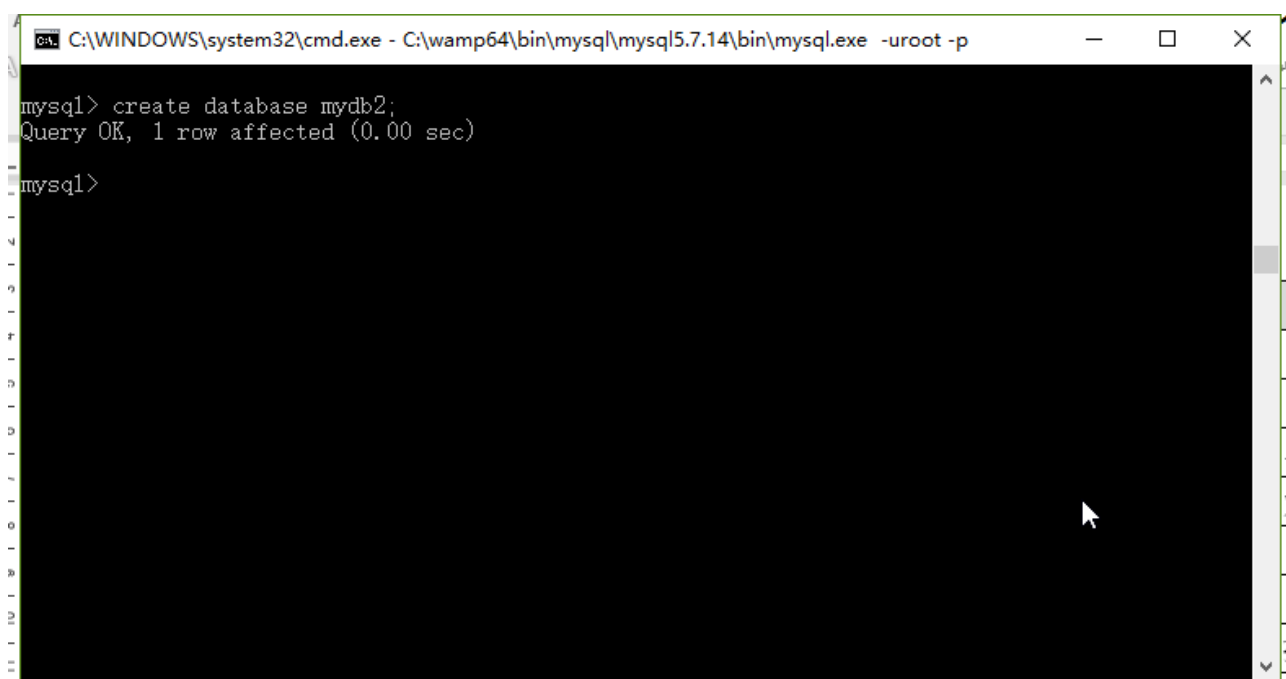
4. DDL SQL 之 database

分类	SQL 语句	描述
创建数据库	create database 数据库名;	

	create database 数据库名 character set 字符集	
查看数据库	show databases;	查看所有的数据库
	show create database 数据库名;	查看某个数据库的定义的信息
删除数据库	drop database 数据库名称	
其他命令	use 数据库名	切换数据库
	select database()	查看正在使用的数据库

4.1 创建数据库

- create database 数据库名;



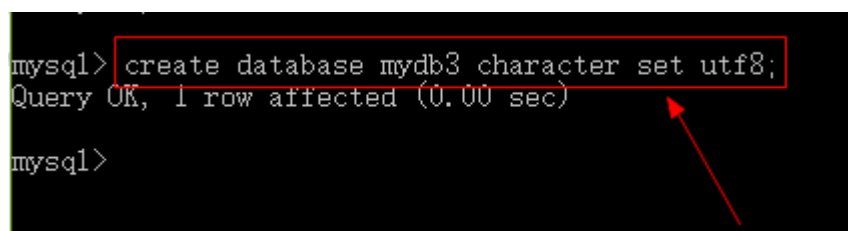
```

C:\WINDOWS\system32\cmd.exe - C:\wamp64\bin\mysql\mysql5.7.14\bin\mysql.exe -uroot -p

mysql> create database mydb2;
Query OK, 1 row affected (0.00 sec)

mysql>
    
```

- create database 数据库名 character set 字符集;



```

mysql> create database mydb3 character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql>
    
```

4.2 查看数据库创建的语句

- show database;



```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| baijia |
| czxy_2017_db |
| mydatabase1 |
| mydb |
| mydb2 |
| mydb3 |
| mysql |
| performance_schema |
| sys |
+-----+
10 rows in set (0.00 sec)

mysql>
```

- show create database mydb2;

```
mysql> show create database mydb2;
+-----+-----+
| Database | Create Database |
+-----+-----+
| mydb2 | CREATE DATABASE `mydb2` /*!40100 DEFAULT CHARACTER SET latin1 */ |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

4.3 切换数据库

ues 数据库名;

可以切换要操作的数据库

4.4 删除数据

drop database mydb2;

```
mysql> drop database mydb2;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

5. DDL 之 table

分类	SQL 语句	描述
创建表	create table 表名(字段名 类型(长度) [约束], 字段名 类型(长度) [约束]);	
查看表	show tables;	查看数据库中的所有表
	desc 表名	查看表结构
删除表	drop table 表名	
修改表	alter table 表名 add 列名 类型(长度) [约束]	修改表添加列
	alter table 表名 modify 列名 类型(长度) 约束	修改表修改列的类型长度及约束
	alter table 表名 change 旧列名 新列名 类型 约束	修改表修改列名
	alter table 表名 drop 列名;	修改表删除列
	rename table 表名 to 新表名	修改表名
	alter table 表名 character set 字符集	修改表的字符集

5.1 创建表

创建 MySQL 数据表需要以下信息：

- 1) 表名
- 2) 表字段名
- 3) 定义每个表字段的类型

```
CREATE TABLE table_name (
    字段名称 字段类型 (长度) 属性...,
    字段名称 字段类型 (长度) 属性...,
);
```

```
-- 创建一个学生表
create table student (
    id int(10) unsigned,
    name varchar(20),
    age tinyint(3) unsigned
);
```



```
mysql> -- 创建一个学生表
mysql> create table student (
  -> id int(10) unsigned,
  -> name varchar(20),
  -> age tinyint(3) unsigned
  -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
```

5.2 查看表

- show tables; 查看所有的表

```
mysql> show tables;
+-----+
| Tables_in_mydb3 |
+-----+
| student          |
+-----+
1 row in set (0.00 sec)

mysql>
```

- show create table student 查看建表语句

- desc 表名; 显示表结构

```
mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int(10) unsigned    | YES  |     | NULL    |       |
| name   | varchar(20)         | YES  |     | NULL    |       |
| age    | tinyint(3) unsigned | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

5.3 修改表

- 为表增加一个新的列

alter table 表名 add 字段名 字段类型 (长度) 属性约束;

如: alter table student add sex enum("男","女");



```
mysql> alter table student add sex enum("男","女");
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

再如: alter table student add address varchar(255);

```
mysql> alter table student add sex enum("男","女");
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table student add address varchar(255);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	YES		NULL	
name	varchar(20)	YES		NULL	
age	tinyint(3) unsigned	YES		NULL	
sex	enum('男','女')	YES		NULL	
address	varchar(255)	YES		NULL	

5 rows in set (0.00 sec)

查看表结构

```
mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	YES		NULL	
name	varchar(20)	YES		NULL	
age	tinyint(3) unsigned	YES		NULL	
sex	enum('男','女')	YES		NULL	
address	varchar(255)	YES		NULL	

5 rows in set (0.00 sec)

```
mysql>
```

● 修改列的长度

alter table 表名 modify 字段名(长度) 约束;

如: 修改 student 表中的 address 字段长度为 300

```
alter table student modify address varchar(300);
```



```
mysql> alter table student modify address varchar(300);
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	YES		NULL	
name	varchar(20)	YES		NULL	
age	tinyint(3) unsigned	YES		NULL	
sex	enum('男','女')	YES		NULL	
address	varchar(300)	YES		NULL	

```
5 rows in set (0.00 sec)

mysql>
```

- 修改列名称

alter table 表名 change 旧列名 新列名 类型(长度) 约束;

如: alter table student change sex gender enum("男","女");

```
mysql> alter table student change sex gender enum("男","女");
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	YES		NULL	
name	varchar(20)	YES		NULL	
age	tinyint(3) unsigned	YES		NULL	
gender	enum('男','女')	YES		NULL	
address	varchar(300)	YES		NULL	

```
5 rows in set (0.00 sec)

mysql>
```

- 删除列

alter table 表名 drop 字段名;

如: alter table student drop address;


```
mysql> alter table student drop address;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) unsigned    | YES  |     | NULL    |       |
| name  | varchar(20)         | YES  |     | NULL    |       |
| age   | tinyint(3) unsigned | YES  |     | NULL    |       |
| gender| enum('男','女')     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> _
```

- 修改表名称

```
rename table 旧表名 to 新表名;
```

如: rename table student to student1;

```
mysql> rename table student to student1;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_mydb3 |
+-----+
| student1         |
+-----+
1 row in set (0.00 sec)

mysql>
```

- 修改字符集

```
alter table 表名 character set 新字符集;
```



```
mysql> alter table student1 character set gbk;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show create table student1;
+-----+-----+
| Table | Create Table |
+-----+-----+
| student1 | CREATE TABLE `student1` (
  `id` int(10) unsigned DEFAULT NULL,
  `name` varchar(20) CHARACTER SET utf8 DEFAULT NULL,
  `age` tinyint(3) unsigned DEFAULT NULL,
  `gender` enum('男','女') CHARACTER SET utf8 DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=gbk |
+-----+-----+
1 row in set (0.00 sec)
```

5.4 删除表

```
drop table 表名;
```

为了配合测试，我们先创建一个名为 test 的表，然后再删除表
create table test (name varchar(10));

```
mysql> create table test (name varchar(10));
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_mydb3 |
+-----+
| student1        |
| test            |
+-----+
2 rows in set (0.00 sec)
```

删除 test 表

```
drop table test;
```

```
mysql> drop table test;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_mydb3 |
+-----+
| student1         |
+-----+
1 row in set (0.00 sec)

mysql>
```

6. DML 之数据增删改

6.1 插入 insert

插入	语法
向表中插入某些列	insert into 表 (列名 1,列名 2,列名 3...) values (值 1,值 2,值 3...)
向表中插入所有列	insert into 表 values (值 1,值 2,值 3...)

- 所有列的情况

insert into student1(id,name,age,gender) values(1,'czxy',18,'男');

```
mysql> insert into student1(id,name,age,gender) values(1,'czxy',18,'男');
Query OK, 1 row affected (0.00 sec)

mysql> select * from student1;
+----+-----+-----+-----+
| id  | name  | age  | gender |
+----+-----+-----+-----+
| 1   | czxy  | 18   | 男     |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

- 部分列的情况

insert into student1(id,name) values(2,'东方不败');

```
mysql> insert into student1(id,name) values(2,'东方不败');
Query OK, 1 row affected (0.00 sec)

mysql> select * from student1;
+----+-----+-----+-----+
| id  | name  | age  | gender |
+----+-----+-----+-----+
| 1   | czxy  | 18   | 男     |
| 2   | 东方不败 | NULL | NULL   |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

- 书写顺序无关，但要对应

insert into student1(id,name) values(2,'东方不败');

```
mysql> insert into student1(name,id) values('黑山老妖',3);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from student1;
```

id	name	age	gender
1	czxy	18	男
2	东方不败	NULL	NULL
3	黑山老妖	NULL	NULL

3 rows in set (0.00 sec)

注意：

1. 列名数与 values 后面的值个数必须保持一致
2. 列的顺序与插入的值得顺序必须保持一致
3. 列名的类型与插入的值要一致.
4. 插入值的时候不能超过最大长度.
5. 值如果是字符串或者日期需要加引号''（一般是单引号）

6.2 更新 update

更新	语法
更新所有记录的指定列	update 表名 set 字段名=值,字段名=值 ...
更新符合条件所有记录的指定列	update 表名 set 字段名=值,字段名=值 ... where 条件

- 更新所有记录的字段值

update student1 set age=20,gender='男';

```
mysql> update student1 set age=20,gender='男';
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

```
mysql> select * from student1;
```

id	name	age	gender
1	czxy	20	男
2	东方不败	20	男
3	黑山老妖	20	男

3 rows in set (0.00 sec)

- 更新复合条件的记录字段值

update student1 set age=18,gender='女' where id=2;

```
mysql> update student1 set age=18,gender='女' where id=2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student1;
+----+-----+-----+-----+
| id | name   | age | gender |
+----+-----+-----+-----+
| 1  | czxy   | 20  | 男     |
| 2  | 东方不败 | 18  | 女     |
| 3  | 黑山老妖 | 20  | 男     |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

注意:

1. 列名的类型与修改的值要一致.
2. 修改值得时候不能超过最大长度.
3. 值如果是字符串或者日期需要加'.

6.3 删除 delete

删除	语法
删除指定表的所有数据	delete from 表名
删除指定表符合条件的所有数据	delete from 表名 where 条件

- 删除表中复合条件记录

delete from student1 where id=3;

```
mysql> delete from student1 where id=3;
Query OK, 1 row affected (0.00 sec)

mysql> select * from student1;
+----+-----+-----+-----+
| id | name   | age | gender |
+----+-----+-----+-----+
| 1  | czxy   | 20  | 男     |
| 2  | 东方不败 | 18  | 女     |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

- 删除表中的所有数据

delete from student1



```
mysql> delete from student1  
->  
Query OK, 2 rows affected (0.00 sec)  
  
mysql> select * from student1;  
Empty set (0.00 sec)  
  
mysql>
```

7. 今日总结