**MySQL数据库的操作**

MySQL 创建数据库和创建数据表

MySQL 是最常用的数据库，在数据库操作中，基本都是增删改查操作，简称CRUD。

在这之前，需要先安装好 MySQL ，然后创建好数据库、数据表、操作用户。

# 一、数据库操作语言

数据库在操作时，需要使用专门的数据库操作规则和语法，这个语法就是 SQL(Structured Query Language) 结构化查询语言。

SQL 的主要功能是和数据库建立连接，进行增删改查的操作。SQL是关系型数据库管理系统的标准语言。

## SQL 语言的作用：

1. 数据定义语言 DDL(Data Definition Language) 。用于创建数据库，数据表。

2. 数据操作语言 DML(Data Manipulation Language) 。用于从数据表中插入、修改、删除数据。

3. 数据查询语言 DQL(Data Query Language) 。用于从数据表中查询数据。

4. 数据控制语言 DCL(Data Control Language) 。用来设置或修改数据库用户或角色的权限。

使用 SQL 操作数据库时，所有的 SQL 语句都以分号结束。(切换数据库时可以不用分号)

在 SQL 语句中，不区分大小写，编写 SQL 语句时可以根据情况用大小写的区别来增加可读性。

# 二、创建数据库

### 1. 连接 MySQL

输入 mysql -u root -p 命令，回车，然后输入 MySQL 的密码(不要忘记了密码)，再回车，就连接上 MySQL 了。

**mysql -u root -p**

最初，都是使用 root 用户登录，工作中如果一直用 root 用户登录，因为权限太大，风险是很大的，所以等创建好权限适合的用户后，就不要经常登录 root 用户了。

## 查找MySQL数据库的端口号

show global variables like 'port';

### 2. 查看当前的数据库

使用 show databases; 查看当前安装的 MySQL 中有哪些数据库。

**show databases;**

刚安装 MySQL 时，默认有四个数据库，information\_schema，mysql，perfomance\_schema，sys 。通常情况下，我们不会直接使用这四个数据库，但千万不要把这四个数据库删了，否则会带来很多不必要的麻烦。如果不小心删了，建议是重新安装 MySQL ，在重装之前把自己的数据迁移出来备份好，或者从其他服务器上迁移一个相同的数据库过来。

### 3. 创建数据库

使用 create database 数据库名; 创建数据库。

**create database MyDB\_one;**

创建数据库成功后，数据库的数量变成了5个，多了刚才创建的 MyDB\_one 。

### 4. 创建数据库时设置字符编码

使用 create database 数据库名 character set utf8; 创建数据库并设置数据库的字符编码。

**create database MyDB\_two character set utf8;**

直接创建的数据库，数据库的编码方式是 MySQL 默认的编码方式 latin1 (单字节编码) ，通常我们会在数据库中存放中文数据，所以最好把数据库的编码方式设置成 utf-8 ，这样中文才能正常显示。

**create database MyDB\_three charset utf8;**

character set 可以缩写成 charset ，效果是一样的。

### 5. 查看和显示数据库的编码方式

使用 show create database 数据库名; 显示数据库的创建信息。

**show create database MyDB\_one;**

**show create database MyDB\_two;**

如果不知道一个数据库的编码方式是什么，可以使用 show create database 数据库名 来查看数据库的编码方式。可以看到刚才创建的 MyDB\_one 的编码方式是 MySQL 的默认编码 latin1 ，MyDB\_two 的编码方式是 utf-8 。

当然，这种方式不能在创建的同时显示，只能查看一个已经存在的数据库的编码方式。

### 6. 使用 alter database 数据库名 character set utf8； 修改数据库编码

**alter database MyDB\_one character set utf8;**

如果一个数据库的编码方式不符合使用需求，可以进行修改。刚才创建的 MyDB\_one 经过修改后，编码方式也变成了 utf-8 。

### 7. 进入或切换数据库

使用 use 数据库名 进入或切换数据库。

**use MyDB\_one**

**use MyDB\_two;**

刚连接上 MySQL 时，没有处于任何一个数据库中，如果要使用某一个数据库，就需要进入到这个数据库中。

use 数据库名 这个命令后面的分号可以省略，这是 SQL 语句中唯一可以省略分号的语句。

### 8. 显示当前数据库 select database();

**select database();**

进入数据库中，可以使用 select database(); 来查看当前处于哪个数据库中。长时间操作数据库时，在很多数据库中来回切换后，查看当前的数据库，避免操作错了数据库。

# 三、创建数据

### 1. 查看当前数据库中的表

使用 show tables；查看当前数据库中有哪些表。

**show tables;**

在刚才创建的数据库 MyDB\_one 中，还没有创建任何表，所以当前是空的。

### 2. 创建表

使用 create table 表名(字段1 字段类型,字段2 字段类型,字段3 字段类型,…); 来创建一张表。

**create table Phone\_table(pid INT, name CHAR(20), price INT);**

在 MyDB\_one 中创建了一个叫 Phone\_table 的数据表，这张表有三个字段 pid，name，price 。为了增加 SQL 的可读性，字段名我用的是小写，字段类型用大写。

### 3. 显示表信息

用 show create table 表名; 来显示已创建的表的信息。

**show create table Phone\_table;**

使用 show create table 表名; 可以显示表的字段信息， MySQL 的引擎，和默认的字符编码等信息。与显示数据库信息一样，show 只能显示已经创建了的数据表的信息，不能在创建的同时显示信息。

如果想更好地展示表的字段信息，可以使用 desc 表名; 来显示表的字段信息。

### 4. 给表增加字段

使用 alter table 表名 add 字段名 数据类型; 为已存在的表添加一个新字段。

**alter table Phone\_table add color CHAR(20);**

添加后，刚才的表中多了一个字段，新增成功。

### 5. 删除表的字段

使用 alter table 表名 drop 字段名; 删除一个表中已存在的字段。

**alter table Phone\_table drop price;**

删除字段后，表中不再有该字段。

### 6. 修改字段的数据类型

使用 alter table 表名 modify 字段名 数据类型; 修改表中现有字段的数据类型。

**alter table Phone\_table modify name VARCHAR(12);**

修改之后，该字段的数据类型发生改变。

### 7. 修改字段的数据类型并且改名

使用 alter table 表名 change 原字段名 新字段名 数据类型; 修改表中现有字段的字段名和类型。

**alter table Phone\_table change name pname CHAR(18);**

现在，将表的 name 改成了 pname ，同时修改了 pname 的数据类型。

# 四、MySQL 常用字段类型

一个数据表是由若干个字段组成的，一个表十几个字段也很正常，每个字段表示不同的信息，需要使用不同类型的数据。

所以在创建表的时候，要为每个字段指定适合的数据类型。

## MySQL 中常用的字段类型有以下这些：

### 1. 整数类型

数据类型 数据范围

TINYINT -128 -- 127

SMALLINT -32768 -- 32767

MEDIUMINT -2^23 -- 2^23-1

INT -2^31 -- 2^31-1

BIGINT -2^63 -- 2^63-1

### 2. 字符串类型

数据类型 字节范围 用途

CHAR(n) 0 -- 255字节 定长字符串

VARCHAR(n) 0 -- 65535字节 变长字符串

TEXT 0 -- 65535字节 长文本数据

LONGTEXT 0 -- 2^32-1字节 极大文本数据

BLOB 0 -- 65535字节 二进制长文本数据

LONGBLOB 0 -- 2^32-1字节 二进制极大文本数据

### 3. 小数类型

m 表示浮点数的总长度，n 表示小数点后有效位数。

数据类型 数据用法 数据范围

Float Float(m,n) 7位有效数

Double Double(m,n) 15位有效数

Decimal Decimal(m,n) 28位有效数

### 4. 时间类型

数据类型 格式 用途

DATE YYYY-MM-DD 日期

TIME HH:MM:SS 时间

YEAR YYYY 年份

DATETIME YYYY-MM-DD HH:MM:SS 日期和时间

TIMESTAMP 10位或13位整数（秒数） 时间戳

### 5. 枚举类型

enum(枚举值1,枚举值2,...)