

# 课程目标：MySQL 基本操作

- 1) 数据库的操作；
- 2) 表的操作；
- 3) 数据的操作（数据的增删改查，查询是重点的重点）；
- 4) 常用的数据库对象。

## 1. 做测试为什么要学习数据库？

- 你会数据库的操作，才能更全面准确的开展软件测试（测试环境的搭建、功能测试、性能测试、安全性测试）；
- 许多公司在面试或者笔试的时候，会问到数据库相关的知识

## 2. 跟数据库相关的一些基本概念

- 1) **数据（Data）**：描述事物的符号记录。注意，数据不仅仅指的是数字，还包括文本、声音、图片、视频等；
- 2) **数据库（Database:DB）**：存储数据的仓库。**长期存储**在计算机内的**有组织**的**可共享**的大量数据的集合。
- 3) **数据库管理系统（Database Management System:DBMS）**：是一种用于操纵和管理数据库的大型软件，用于建立、使用和维护数据库。  
常见的数据库管理软件：MySQL、Oracle、SQLServer、DB2……  
数据库管理软件可分为：
  - 关系型数据库管理软件：以表的方式来存储数据。MySQL、Oracle、SQLServer、DB2……
  - 非关系型数据库管理软件：不以表的方式存储数据。MongoDB、Redis……
- 4) **数据库系统（Database System:DBS）**：在计算机系统中引入了数据库之后的系统。一般由数据库、数据库管理系统、应用系统、数据库管理员（DBA）等所组成。

注：关系型数据库管理系统通过使用 **SQL** 来完成对数据库的管理。

## 3. SQL：Structured Query Language：结构化的查询语言

**含义**:是用于管理和操作关系型数据库的一门语言。  
**语言**：单词+语法

### SQL 主要包括：

- **DQL：Data Query Language:数据查询语言**。用于查询数据库的数据。在所有的 SQL 语句之中，这一部分的语法最为复杂和灵活，也是我们学习的重点。包括：select

- **DML:Data Manipulation Language:数据操作语言**。用于改变数据库中的数据。包括：insert(插入)、update(修改字段)、delete(删除)
- **DDL:Data Definition Language:数据定义语言**。用于定义数据的结构。包括：create(创建)、alter(修改表)、drop(删除)
- **DCL: Data Control Language: 数据控制语言**。用于定义数据库用户的权限。包括：grant(赋权限)、revoke(回收权限)——DBA 去维护

## 4. 数据库环境的搭建

### 1) 安装 MySQL 需要注意：

- 建议安装前关闭杀毒软件（360、火绒、QQ 管家、金山毒霸一类）
- Win7 系统安装的适合，建议在安装软件上右击、以管理员身份运行
- MySQL 需要.net framework 4.5.2 的支持（5.7 版本的 MySQL）

### 2) 安装的顺序：

安装 .net framework4.5.2——》MySQL(Server only) ——》Navicat for MySQL(客户端)

### 3) 卸载的步骤：

进到控制面板卸载 MySQL Server 和 MySQL installer——》删除 MySQL 的安装目录和数据文件夹——》删 MySQL 的注册表信息

参考：[https://blog.csdn.net/qq\\_23329167/article/details/81515671](https://blog.csdn.net/qq_23329167/article/details/81515671)

## 5. 数据库的基本操作

### 1) 数据库的管理

#### —— 创建数据库

语法：create database 数据库名;

举例：

create database a;

#### —— 切换/选择/使用数据库

语法：use 数据库名;

举例：

use a;

#### —— 删除数据库

语法：drop database 数据库名;

举例：

drop database a;

## 2)表的管理

数据库无法直接存储数据，我们需要先建表，使用表来存储数据。

表由行和列所组成，行我们也称之为记录，列我们也称之为字段。

### ——简单的建表语法：

```
create table 表名称(  
    字段 1 字段 1 的数据类型,  
    字段 2 字段 2 的数据类型,  
    .....  
    字段 n 字段 n 的数据类型  
);
```

注解：

### MySQL 常用的数据类型：

——整型：int

——浮点型(小数)：float ( m , n ) : 比如 ( 7 , 2 ) 也就是占位 7 , 小数点为 2 / double(范围更大)

——字符型：char / varchar(节省空间)

——日期型：date

举例：

use a; (切换数据库 a)

```
create table student(  
    sid int,  
    sname varchar(20),  
    grade float
```

); (创建一个整数型的字段,创建一个字符型的字段,创建一个字符型的字段)

### 一个汉字两个字符

建表通常伴随着约束。约束指的是对表中字段的数据的一系列的规定和限制。

约束的目的在于保证数据的完整性（正确的有意义的）。

### 常用的约束如下：（前面 4 种针对单表，后面 1 种针对多表）

#### ——主键约束(不能重复)： primary key 。

主键指的是主关键字，它是表中的一个或多个字段，它的值可以用于唯一的标识表中的每一条记录。如果你对某个字段设置了主键约束，那么这个字段既不能为空，也不能重复。

——非空约束: **not null**。

如果你给某个字段设置了该约束, 那么该字段不能取空值(null)

——唯一约束(不能重复): **unique**。

如果你给某个字段设置了该约束, 那么该字段禁止更新为重复的值

——默认值约束: **default** 默认值。

如果你给某个字段设置了该约束, 那么当你没有给该字段赋值的时候, 它使用默认值。

——外键约束: **foreign key**。

如果某个字段, 它在一张表里是主键, 然后它又出现在另外一张表里, 那么就称之为另外一张表的外键。外键跟主键之间有一个参照关系, 外键的取值必须参照主键的取值。

举例: 创建一个 student1 的表格

```
create table student1(  
    sid int primary key,  
    sname varchar(20) not null,  
    grade float default 0,  
    semail varchar(50) unique  
);
```

注解: 字段内使用了主键约束(不唯一且不能为空), sname 字段设置了不能为 null, grade 设置了默认值(如果该字段没有设置参数, name 默认就位 grade), semail 字段设置了唯一约束(修改后的值不能重复);

```
insert into student1(sid,sname,grade,semail)  
values(1,'zhangsan',80,'zs@163.com'),  
(2,'lisi',78,'ls@163.com'),  
\(3,'wangwu',90,'ww@163.com'\);
```

```
insert into student1(sid,sname,grade,semail)  
values(4,null,77,'ss'); #sname 不能为 null
```

```
select * from student1;
```

```
create table question(  
    sid int,  
    ques varchar(100),  
    foreign key(sid) references student1(sid)  
);
```

注解: 建立第二张表格, 因为 sid 和上一张表 student1 的 sid 是一样的, 那么就可以给这两个字段作为外键约束(比如, 一个班有一百个学员用字段记录, 那么第二张表的相同名称字段数据(学号)就不可以超过 100 这个范围), 格式就是 foreign key(字段名) references 想要关联的表名(字段名)

```
insert into question(sid,ques) values(4,'what is sql?'); #因为外键约束, 4 号学员的信息不让插入, 因为最大为 3
```

```
insert into question(sid,ques) values(3,'what is sql?'); #可以插入, 因为有 3 号学员
```

```
select * from question;
```

字段内插入数据格式:

```
insert into 表名(字段 1,字段 2,字段 3.....,字段 n)
Values(数据 1,数据 2,数据 3.....,数据 4),
(数据 1,数据 2,数据 3.....,数据 4),
(数据 1,数据 2,数据 3.....,数据 4)
可以插入多个数据信息,要按照字段格式来插入
```

## ——查看表结构:

语法: desc 表名称;

举例:

```
desc question;
```

信息	结果1	概况	状态			
Field	Type	Null	Key	Default	Extra	
sid	int(11)	YES	MUL	(Null)		
ques	varchar(11)	YES		(Null)		

## ——显示数据库中所有的表:

语法: show tables;

举例:

```
show tables;
```

## ——表的修改:

### 1) 修改表名:

语法: alter table 旧表名 rename to 新表名;

举例:

```
alter table student rename to student2;
```

### 2) 修改字段名:

语法: alter table 表名 change 旧字段名 新字段名 新的数据类型;

举例:

```
alter table student2 change grade fenshu double;
```

### 3) 添加字段:

语法: alter table 表名 add 字段名 数据类型;

举例：

```
alter table student2 add gender char(5);
```

#### 4) 删除字段：

语法：alter table 表名 drop 字段名；

举例：

```
alter table student2 drop gender;
```

#### ——表的删除：

语法：drop table 表名称；

举例：

```
drop table student2;
```

小结：

- 1) 数据库相关的基本概念；
- 2) 软件的获取及安装；
- 3) 数据库及表的常用操作。