# 1.数据库管理系统的引入

|  |
| --- |
|  |

# 2.MySQL的安装,其实下载一个免费的压缩包,然后解压缩,配置好数据库的路径然后配置用户名和密码,并且把数据库软件安装为范围就可以了,mysql语法具体可以参考: <https://github.com/kennycaiguo/kenny-learn-mysql>, 下面是服务的安装启动和停止

|  |
| --- |
|  |

# 3.mysql数据库相关SQL语句的学习

## 1>数据库相关的SQL语句

### 1)查看服务器上面有哪些数据库 : show databases;

### 2)创建数据库: create database 数据库名称 default charset utf8 collate utf8\_general\_ci;

#### 比如我们新建一个叫做students的数据库: create database students default charset utf8 collate utf8\_general\_ci;

|  |
| --- |
|  |

#### 然后我们用navicat工具可以看到新建的数据库,不过此时它只是一个文件夹,里面没有任何东西

|  |
| --- |
|  |

### 3)删除数据库: drop database 数据库名称; 如我们把students删除: drop database students;

|  |
| --- |
|  |

#### 然后这个students数据库就被删除了.用navicat已经看不见它了

|  |
| --- |
|  |

### 4)使用数据库: use 数据库名称; 如: use company; 使用(或者说切换到)company数据库

|  |
| --- |
|  |

### 5)查看当前数据库里面的所有数据表: show tables;

|  |
| --- |
|  |

#### 我们用navicat查看一下,效果是一样的

|  |
| --- |
|  |

## 2>数据表相关的SQL语句

### 1)创建数据表:

|  |
| --- |
| **CREATE TABLE `account` (**  **`id` int(11) NOT NULL AUTO\_INCREMENT,**  **`accountName` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL,**  **`amount` int(20) NULL DEFAULT NULL,**  **PRIMARY KEY (`id`) USING BTREE**  **) ENGINE = InnoDB AUTO\_INCREMENT = 3 CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Compact;** |

#### 如我们在company数据库里面新建一个emp数据表,代码可以这么写

|  |
| --- |
|  |

#### 点击运行,发现运行成功,然后我们刷新数据库,就可以打开emp数据表(navicat真的很方便)

|  |
| --- |
|  |

#### 当然也可以简单一点,但是charset是utf8这个必须得写.

|  |
| --- |
|  |

#### 注意:我们创建列的时候还可以给他传递默认值,也可以有一下限制

|  |
| --- |
|  |

#### 还可以创建主键

|  |
| --- |
|  |

#### 而且还需要设置id能够自增长

|  |
| --- |
|  |

#### 这样子,我们往数据表里面插入数据的时候,就不需要传递id了,系统会自动维护这个id,还可以加上not null

|  |
| --- |
|  |

### 2)删除数据表: drop table 数据表名称; 比如我们删除刚刚创建的emp表格

|  |
| --- |
|  |

#### 然后我们用navicat就看不见这个表了

|  |
| --- |
|  |

### 3)在mysql命令行工具里面查看数据表的所有列的信息: desc 表格名称; 如,desc employee

|  |
| --- |
|  |

## 3>mysql数据库的常用数据类型,其实常用的只有四种:int,float,string,日期,但是每一种都有很多子类型.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
| 对于一些定长的数据,如电话号码,密码等等建议使用char因为他的速度比较快,对于一些无法固定长度的数据,如用户名,邮箱等等就只能使用varchar了 |
|  |
|  |

### 参考地址: <https://blog.csdn.net/qq_43145072/article/details/127166717>

### 我们需要根据实际的需要来使用数据表里面列的数据类型,比如年龄只需要tinyint即可,因为int的取值非常大,所以如果你使用它,就会占用很多内存.会影响性能.这个需要注意.

### 小练习,创建一个worker表,SQL语句如下

|  |
| --- |
| **CREATE TABLE worker(**  **id int PRIMARY KEY auto\_increment,**  **name VARCHAR(50) not null,**  **password CHAR(64) Not null,**  **gender char(6) not null,**  **email VARCHAR(80) not null,**  **age TINYINT,**  **salary DECIMAL(10,2),**  **created datetime**  **) ENGINE=INNODB CHARACTER SET=utf8 COLLATE=utf8\_general\_ci;** |

### 运行代码发现成功了,然后我们就可以看到整个worker表

|  |
| --- |
|  |

### 然后我们插入3条数据

|  |
| --- |
| insert into worker (name,password,gender,email,age,salary,created) values ("Jackline","123","female","jl123@gmail.com",18,2000.50,"2022-09-30 08:12:30"),("Denny","123","male","dy123bb@hotmail.com",18,2100.50,"2023-03-30 08:12:30"),("Miky","123","male","mky123456@hotmail.com",18,2000.50,"2022-09-30 08:12:30"); |
|  |

### 注意:在实际开发中数据库和数据表右边需要提前创建好的.

## 4>数据的增删改查SQL语句

### 1)新增也就是插入语句: insert into 表名称(字段1,字段2,...) values(value1,value2,...); 如,我们往employee数据表里面插入一条数据: insert into employee (name,depId,gender,salary) values("Jacklou",5,"male",5000);

|  |
| --- |
|  |

#### 此时执行成功,我们可以在navicat上面看到新增的数据

|  |
| --- |
|  |

#### 注意:其实这个命令可以插入多条数据,比如我们再插入2条数据,可以这么写

#### INSERT INTO employee (name,depId,gender,salary) values("Andylou",5,"male",4000),("Meggie Qu",5,"female",3000);

|  |
| --- |
|  |

#### 运行成功后,我们可以看到新增的数据

|  |
| --- |
|  |

### 2)删除数据: delete from 表名 where 条件(一定要有条件,否则全部数据都会被删除)

#### 比如我们删除id为57的员工: delete from employee where id=57; 运行代码,执行成功

|  |
| --- |
|  |

#### 然后我们使用navicat来查看,发现57号员工就被删除了

|  |
| --- |
|  |

### 3)修改数据: update 表名 set 列1=值1,列2=值2 where 条件,比如我们把id为50的员工的工资改为2000,把他的id设置为5,可以这么写: update employee set id=5,salary=2000 where id=50;执行代码,显示成功

|  |
| --- |
|  |

#### 然后我们用navicat查看,发现数据的确被修改了

|  |
| --- |
|  |

#### 然后我们也可以把id为52的员工的名字改为kelly chen: update employee set name="Kelly chen" where id=52;

|  |
| --- |
|  |

### 4)查询数据

#### 1.查看一个数据表里面的所有数据(这个比较复杂我们使用worker表来学习,为了方便学习,我新增了3条数据): select \* from 表名; 如,select \* from worker;

|  |
| --- |
|  |

#### 2.select后面除了\* ,还可以写具体的列名,如: select name,gender,email,salary from worker; 效果如下:

|  |
| --- |
|  |

#### 3,可以添加条件,如: select id, name,gender,email,salary from worker where id<4;

|  |
| --- |
|  |

#### 4.也可以在where条件里面使用in关键字,如select id, name,gender,email,salary from worker where id in(2,4,5);

|  |
| --- |
|  |

#### 5.也可以在where文件里面使用between...and,如: select id, name,gender,email,salary from worker where salary between 2200 and 4000;

|  |
| --- |
|  |

##### 注意: between..and是包头不包尾,and后面的是不包括的.

# 小结:

## 一般来说,我们在实际开发中,可以使用mysql控制台来创建数据库和表结构,但是数据一般是通过程序来进行增删改查的.一般不要预先添加数据.

## 用python代码来操作数据库的笔记参考flask相关或者django相关的笔记