

一、操作mysql数据库常见第三方库

mysql-connector

pymysql

实现通过python操作mysql数据库

二、mysql-connector的基本应用

1、安装mysql-connector

2、mysql-connector的基本应用

mysql-connector操作mysql的步骤：

1、创建连接

2、创建游标实例

3、调用execute(参数1，参数2，参数3)处理数据库的操作

参数1：sql语句

参数2：类型为元组，元素值为sql语句占位符对应的参数

参数3：参数bool类型，第一个参数是不是多sql语句，如果是则传入True,否则传入

False

```
1
2 import mysql.connector
3
4 """
5 mysql.connector基本应用
6
7 数据库：存储项目数据 验证数据---》代码取数据库的数据来跟实际结果进行比对
8
9 1、连接数据库
10 2、创建流标实例
11 3、 调用游标实例的excute(sql) excute(sql,(values),bool) sql语句 数据库表常见操作
12
13 """
14 # 1、连接数据库
15 def connect():
16     conn=mysql.connector.connect(
17         host="localhost",
18         user="root",
19         password="admin"
```

```
20 )
21 print(conn)
22 return conn
23
```

3、连接某个数据库

```
1
2 """连接指定某个数据库"""
3 def connect_database(database):
4     """
5     连接指定某个数据库
6     :param database: 数据库名称
7     :return:
8     """
9     conn=mysql.connector.connect(
10     host="localhost",
11     user="root",
12     password="admin",
13     database=database
14 )
15 print(conn)
16 return conn
```

4、创建数据库、创建表

```
1 def creatdatabase(databasename):
2     conn=mysql.connector.connect(
3     host="localhost",
4     user="root",
5     password="admin"
6     )
7     # 获取游标实例
8     cursor=conn.cursor()
9     sql=f"create database {databasename}"
10    cursor.execute(sql)
11    cursor.execute("show databases")
12    # 展示执行结果
13    for database in cursor:
14        print(database)
```

```

15
16 """
17 创建表
18 """
19 def createtable():
20     conn = mysql.connector.connect(
21         host="localhost",
22         user="root",
23         password="admin",
24         database="mashang_211"
25     )
26     # 获取游标实例
27     cursor = conn.cursor()
28     #创建表
29     sql = "create table user(id int auto_increment primary key ,name varchar(20),classname varchar(255))"
30     cursor.execute(sql)

```

5、插入数据到表

```

1  """插入数据到表"""
2  def insertdata():
3      conn = mysql.connector.connect(
4          host="localhost",
5          user="root",
6          password="admin",
7          database="mashang_211"
8      )
9      # 获取游标实例
10     cursor = conn.cursor()
11     sql="insert into user(name,classname)values(%s,%s)"
12     svalue=("jing","M211")
13     cursor.execute(sql,svalue)
14     #提交
15     conn.commit()
16     rowcount=cursor.rowcount
17     print(f"{rowcount}行记录插入成功! ")
18     cursor.execute("select * from user where name='zhangsan' ")
19     #查看返回结果
20     res=cursor.fetchall()
21     print(res)

```

6、获取查询的结果数据

常用方法有：

`fetchall()` 获取所有的记录

`fetchone()` 获取第一行记录

`fetchmany(size)` 获取前几行记录

```
1
2 def select_showall(database,sql):
3     """
4     查询数据，并返回查询结果
5     :param database:
6     :param sql:
7     :return:
8     """
9     conn=connect_database(database)
10    # 获取游标实例
11    cursor = conn.cursor()
12    cursor.execute(sql)
13    # 查看所有返回结果
14    allrows = cursor.fetchall()
15    # 返回第一条记录
16    onerow=cursor.fetchone()
17    # 返回多条记录
18    manyrow=cursor.fetchmany(3)
19    print(allrows)
20    print(type(allrows))
21
```

7、批量插入数据

`executemany()`方法，实现批量插入多条数据

第二参数是一个元组列表，包含我们插入的数据

```
1
2 """批量插入数据"""
3 def insertmany(database):
4     conn = connect_database(database)
5     # 2获取游标实例
```

```

6  cursor = conn.cursor()
7  sql="insert into user(name,classname)values(%s,%s)"
8  sqlvalues=[('huahua',"M211"),('DD',"M211"),('xiaoyang',"M211")]
9  cursor.executemany(sql,sqlvalues)
10 # 提交
11 conn.commit()
12 rows=cursor.rowcount
13 print(f"{rows}行记录插入成功! ")
14 select_showall(database,"select * from user")

```

8、防sql注入，一般sql语句中用占位符传值

```

1  databse2="mashang"
2  conn = connect_database(databse2)
3  mycursor=conn.cursor()
4  # 避免sql注入
5  sql="select * from vipinfo where name=%s"
6  sqlvals=("zhangsang",)
7  data=mycursor.execute(sql,sqlvals)
8  print(data)

```

三、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。

我们可以使用命令安装最新版的 PyMySQL：

pip install PyMySQL

实现的逻辑跟mysql-connector一致，及相同的操作，操作方法也一致

```

1  import pymysql
2
3  conn=pymysql.connect(host="localhost",user="root",password="admin",database="mashang")
4  print(conn)
5  # 获取流标实例
6  mycursor=conn.cursor()
7  mycursor.execute("select * from vipinfo")
8  data=mycursor.fetchall()

```

```
9 print(data)
```