



廣州軟件學院

SOFTWARE ENGINEERING INSTITUTE OF GUANGZHOU

课程名称： 网络应用开发与系统集成 课程代码： NT3009

任课老师： 曹春华 实训指导老师： 曹春华

实训名称： 数据库编程（Mysql 与 Python 交互）

学生姓名： 曾倩怡

学号： 2240233148

教学班： ATX

递交日期： 2024. 11. 19

签收人： 曹春华

实训报告评语与评分：

11. 数据库编程（Mysql 与 Python 交互）

实验目的

- 1、了解下载和安装 pymysql
- 2、了解安装 pymysql
- 3、掌握 python 对象操作数据

实验环境

- 1、硬件环境：i386 兼容机，2G 剩余硬盘空间。
- 2、软件环境：Windows 10 以上兼容系统
- 3、使用资源：教师提供的 Python 环境软件

实验要求

按要求完成课堂上的实训。

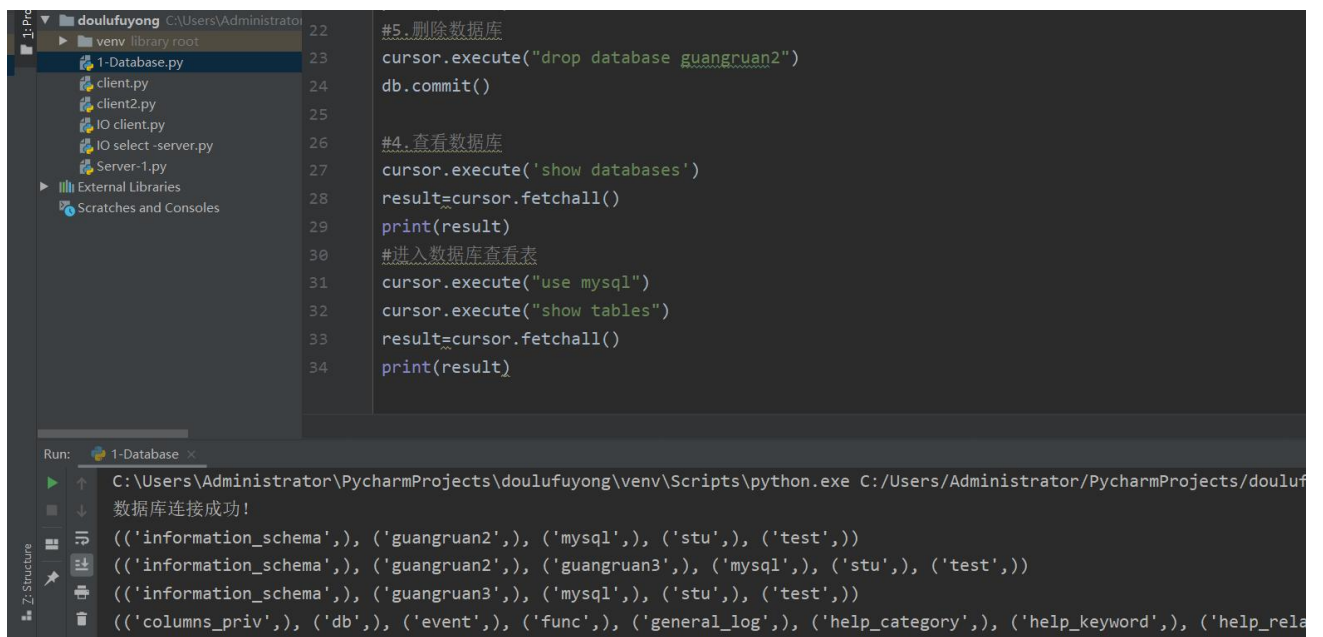
实验内容

一、根据 PPT 示例，结合教材，练习 Mysql 与 pymysql 的安装以及 python 来操作数据库的基本操作，为实现 Mysql 与 Python 交互作知识铺垫。

1、Mysql 与 python 连接

```
1  #数据库操作，作者：曹春华
2  import pymysql
3  #1.连接数据库
4  db=pymysql.connect(host='localhost',
5                      user='root',
6                      passwd='123',
7                      database='mysql',
8                      charset='utf8')
9  print('数据库连接成功! ')
10 cursor=db.cursor()
11 #2.查看数据库
12 cursor.execute('show databases')
13 result=cursor.fetchall()
14 print(result)
15 #3.创建数据库
16 cursor.execute("create database guangruan3 charset utf8 collate utf8_general_ci")
17 db.commit()
18 #4.查看数据库
19 cursor.execute('show databases')
20 result=cursor.fetchall()
21 print(result)
```

2、python 操作 Mysql 数据库



```
22 #5. 删除数据库
23 cursor.execute("drop database guangruan2")
24 db.commit()
25
26 #4. 查看数据库
27 cursor.execute('show databases')
28 result=cursor.fetchall()
29 print(result)
30 #进入数据库查看表
31 cursor.execute("use mysql")
32 cursor.execute("show tables")
33 result=cursor.fetchall()
34 print(result)
```

Run: 1-Database x

C:\Users\Administrator\PycharmProjects\doulufuyong\venv\Scripts\python.exe C:/Users/Administrator/PycharmProjects/doulufuyong/venv/Scripts/python.exe

数据库连接成功!

```
((('information_schema',), ('guangruan2',), ('mysql',), ('stu',), ('test',)))
((('information_schema',), ('guangruan2',), ('guangruan3',), ('mysql',), ('stu',), ('test',)))
((('information_schema',), ('guangruan3',), ('mysql',), ('stu',), ('test',)))
((('columns_priv',), ('db',), ('event',), ('func',), ('general_log',), ('help_category',), ('help_keyword',), ('help_rela
```

二、知识应用

Mysql 与 Python 交互

```
15.py x
1  #数据库操作 作者: 曾倩怡
2  import pymysql
3
4  # 数据库连接配置
5  config = {
6      'host': 'localhost',
7      'user': 'root',
8      'password': 'abc123',
9      'database': 'aa',
10     'charset': 'utf8',
11     'cursorclass': pymysql.cursors.DictCursor,
12 }
13
14 # 连接到数据库
15 connection = pymysql.connect(**config)
16
17 try:
18     # 创建表
19     with connection.cursor() as cursor:
20         sql_create_table = """
21         CREATE TABLE IF NOT EXISTS students (
```

15 x

D:\taidii\python.exe "D:\大学\大三上\系统集成-2158- ATX\15.py"

```
{'id': 1, 'name': 'Alice', 'age': 20, 'grade': 'A'}
{'id': 2, 'name': 'Bob', 'age': 22, 'grade': 'B'}
```

进程已结束,退出代码0

```
运行: 15 x
D:\taidii\python.exe "D:\大学\大三上\系统集成-2158- ATX\15.py"
{'id': 1, 'name': 'Alice', 'age': 20, 'grade': 'A'}
{'id': 2, 'name': 'Bob', 'age': 22, 'grade': 'B'}

进程已结束,退出代码0
```

1、实现思路：

3、代码测试

使用 Python 完成 MySQL 数据库操作的实现思路如下：

一、实验准备

安装 MySQL：确保已经在本地或服务器上安装了 MySQL 数据库，并创建了一个测试数据库以及相应的测试表。

安装 Python 及 pymysql 库：确保 Python 环境已经配置好，并通过 pip 安装 pymysql 库。可以使用以下命令进行安装：

bash 复制代码

```
pip install pymysql
```

二、实现思路

导入 pymysql 库：

在 Python 脚本中导入 pymysql 库，以便使用其提供的函数和类来操作 MySQL 数据库。

python 复制代码

```
import  
pymysql
```

建立数据库连接：

使用 pymysql.connect() 函数建立与 MySQL 数据库的连接。需要指定数据库的主机名、用户名、密码、数据库名等参数。

python 复制代码

```
# 建立数据库连接  
  
connection = pymysql.connect(  
  
    host='localhost', # 数据库主机名  
  
    user='root', # 数据库用户名  
  
    password='123', # 数据库密码  
  
    database='aa' # 数据库名
```

```
)
```

创建游标对象：

通过连接对象的 `cursor()` 方法创建一个游标对象，用于执行 SQL 语句并获取结果。

python 复制代码

```
# 创建游标对象

cursor = connection.cursor()
```

执行 SQL 语句：

使用游标对象的 `execute()` 方法执行 SQL 语句。可以是查询语句（如 `SELECT`），也可以是插入、更新或删除语句（如 `INSERT`、`UPDATE`、`DELETE`）。

python 复制代码

```
# 执行查询语句

sql_query = "SELECT * FROM yourtable"

cursor.execute(sql_query)


# 执行插入语句（示例）

# sql_insert = "INSERT INTO yourtable (column1, column2) VALUES ('value1',
'value2')"
```

```
# cursor.execute(sql_insert)

# connection.commit() # 提交事务
```

获取查询结果：

如果执行的是查询语句，可以使用游标对象的 `fetchall()`、`fetchone()` 或 `fetchmany()` 方法获取查询结果。

python 复制代码

```
# 获取查询结果

results =
cursor.fetchall()

for row in results:

    print(row)
```

关闭游标和连接：

完成数据库操作后，使用游标对象的 `close()` 方法关闭游标，然后使用连接对象的 `close()` 方法关闭连接。

python 复制代码

```
# 关闭游标和连接  
cursor.close()  
connection.close()
```

三、代码测试

将上述代码整合到一个 Python 脚本中，并运行该脚本。观察输出结果，确保数据库连接成功，SQL 语句执行正确，并能够获取到预期的查询结果。

【分析与总结】

本次实验

1、实现了一个什么功能模块？

本次实验实现了一个 Python 与 MySQL 数据库交互的功能模块。这个模块能够建立与 MySQL 数据库的连接，执行 SQL 语句（包括查询、插入、更新、删除等），并获取查询结果。通过该模块，用户可以在 Python 程序中实现对 MySQL 数据库的操作。

2、应用了哪些知识点？

Python 编程语言基础：包括变量、数据类型、控制结构、函数等。

MySQL 数据库基础：包括数据库的安装与配置、SQL 语句的编写与执行等。

pymysql 库的使用：了解 pymysql 库的安装、导入、连接数据库、创建游标、执行 SQL 语句、获取查询结果以及关闭连接等操作方法。

3、提升了哪方面的技能？

数据库操作能力：通过本次实验，学生掌握了如何在 Python 程序中操作 MySQL 数据库，包括连接数据库、执行 SQL 语句以及获取查询结果等。

编程实践能力：学生将所学的 Python 编程知识和 MySQL 数据库知识结合起来，通过编写代码实现了数据库操作的功能模块，提升了编程实践能力。

问题解决能力：在实验过程中，学生可能会遇到各种问题，如数据库连接失败、SQL 语句执行错误等。通过解决问题，学生提升了问题解决能力和调试技巧。

4、为以后开发项目提供哪些帮助？

数据库交互能力：在开发项目中，经常需要与数据库进行交互。本次实验为学生提供了 Python 与 MySQL 数据库交互的基础知识和实践经验，为以后的开发项目提供了有力的支持。

模块化编程思想：通过实现一个功能模块，学生了解了模块化编程的思想和方法，为以

后的开发项目提供了更好的代码组织和维护方式。

拓展能力: 学生可以根据本次实验的基础, 进一步学习其他数据库(如 Oracle、SQL Server 等)的操作方法, 以及更复杂的数据库操作技巧(如事务处理、存储过程等), 为开发更复杂的项目提供帮助。

5、对本次的学习是否满意? (从教师教学的严格度、课堂气氛、专业能力、教学过程等方面进行简要描述)

答: 我对本次的学习非常满意。教师在教学过程中非常严格, 对实验的要求很高, 这促使我更加认真地对待实验和学习。课堂气氛活跃, 教师经常与学生进行互动和交流, 这使我能够更好地理解和掌握所学知识。教师的专业能力很强, 对 Python 和 MySQL 数据库都有深入的了解和丰富的实践经验, 能够为学生提供很好的指导和帮助。教学过程清晰明了, 教师先介绍了相关的理论知识和操作方法, 然后引导学生进行实验和练习, 最后还进行了总结和反思, 这使我能够系统地掌握所学知识并提升实践能力。