Python访问数据库MySQL

一、课前必知

1.1授课环境

操作系统：window72

Python：3.6

MySql：5.7.15

1.2 授课内容

使用Python访问MySQL数据库，实现查询、添加、删除、修改功能。

1.3 重点内容

1、安装PyMySQL

2、Python操作数据库的整体步骤

1.4 难点内容

1、中文乱码问题的解决

2、Linux中安装PyMySQL

二、Python操作数据库的整体步骤

1、安装PyMySQL

2、导入PyMySQL模块

3、创建连接对象

4、由连接对象获取游标对象

5、由游标对象执行sql语句

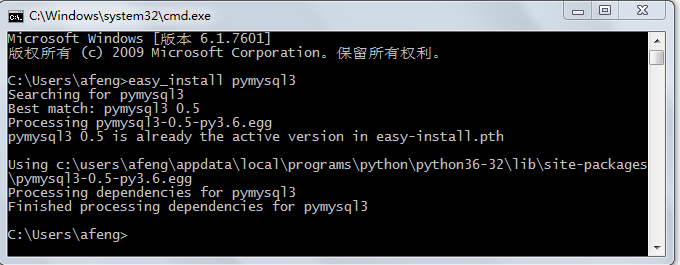
6、在游标对象中获取结果

7、关闭资源(包括游标和连接对象)

三、安装PyMySQL

在python3之后，MySQLdb被pymysql取代了。根据使用版本的不同，安装不同的连接器。  
    在命令窗口(cmd)使用命令如下：easy\_install pymysql3 。

装好以后，使用easy\_install pymysql3，会显示如下结果：



四、Python操作数据库的具体实现

# 1、导入PyMySQL模块

>>> import pymysql

>>> # 2、创建连接对象

>>> conn=pymysql.connect(host="127.0.0.1",port=3306,user="root",passwd="root",db="czf")

>>> # 3、由连接对象获取游标对象

>>> cursor=conn.cursor();

>>> # 4、由游标对象执行sql语句

>>> cursor.execute("select count(\*) from citys")

1

>>> # 5、在游标对象中获取结果

>>> row=cursor.fetchone();

>>> print(row[0])

43

>>> # 6、关闭资源(包括游标和连接对象)

>>> cursor.close()

>>> conn.close()

>>>

需要注意的是，使用 fetchone() 方法获取单条数据.

保存文件为testDB.py，并把它放在C:\Users\afeng目录下，  
# 1、导入PyMySQL模块

import pymysql

# 2、创建连接对象

conn=pymysql.connect(host="127.0.0.1",port=3306,user="root",passwd="root",db="czf")

# 3、由连接对象获取游标对象

cursor=conn.cursor();

# 4、由游标对象执行sql语句

cursor.execute("select count(\*) from citys")

# 5、在游标对象中获取结果

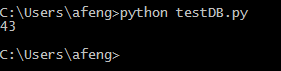
row=cursor.fetchone();

print(row[0])

# 6、关闭资源(包括游标和连接对象)

cursor.close()

conn.close()

执行testDB.py  


五、获取表的所有数据

# 1、导入PyMySQL模块

import pymysql

# 2、创建连接对象

conn=pymysql.connect(host="127.0.0.1",port=3306,user="root",passwd="root",db="czf",charset='utf8')

# 3、由连接对象获取游标对象

cursor=conn.cursor();

# 4、由游标对象执行sql语句

cursor.execute("select \* from citys")

# 5、在游标对象中获取结果

rows=cursor.fetchall()

# 遍历结果集

for row in rows:

cid=row[0]

cname=row[1]

pid=row[2]

print(cid,cname,pid)

# 6、关闭资源(包括游标和连接对象)

cursor.close()

conn.close()

需要注意的是，在连接对象里设置charset='utf8'解决中文乱码问题。

六、添加表的数据

# 1、导入PyMySQL模块

import pymysql

# 2、创建连接对象

conn=pymysql.connect(host="127.0.0.1",port=3306,user="root",passwd="root",db="czf",charset='utf8')

# 3、由连接对象获取游标对象

cursor=conn.cursor();

# 4、由游标对象执行sql语句

cursor.execute("insert into citys(c\_name,p\_id) values ('衡阳',4)")

# 5、提交数据

try:

conn.commit()

except:

# 发生错误时回滚

conn.rollback()

# 6、关闭资源(包括游标和连接对象)

cursor.close()

conn.close()

七、修改表的数据

# 1、导入PyMySQL模块

import pymysql

# 2、创建连接对象

conn=pymysql.connect(host="127.0.0.1",port=3306,user="root",passwd="root",db="czf",charset='utf8')

# 3、由连接对象获取游标对象

cursor=conn.cursor();

# 4、由游标对象执行sql语句

cursor.execute("update citys set c\_name='郴州' where c\_id=44")

# 5、提交数据

try:

conn.commit()

except:

# 发生错误时回滚

conn.rollback()

# 6、关闭资源(包括游标和连接对象)

cursor.close()

conn.close()

八、删除表的数据

# 1、导入PyMySQL模块

import pymysql

# 2、创建连接对象

conn=pymysql.connect(host="127.0.0.1",port=3306,user="root",passwd="root",db="czf",charset='utf8')

# 3、由连接对象获取游标对象

cursor=conn.cursor();

# 4、由游标对象执行sql语句

cursor.execute("delete from citys where c\_id=44")

# 5、提交数据

try:

conn.commit()

except:

# 发生错误时回滚

conn.rollback()

# 6、关闭资源(包括游标和连接对象)

cursor.close()

conn.close()