[**通过mysqlclient操作MySQL数据库**](https://www.cnblogs.com/kingwangzhen/p/9395914.html)

Posted on 2018-07-31 15:03  [kingwangzhen](https://www.cnblogs.com/kingwangzhen/)  阅读(11310)  评论(1)  [编辑](https://i.cnblogs.com/EditPosts.aspx?postid=9395914)  收藏

**一，安装mysql**

如果是windows 用户，mysql 的安装非常简单，直接下载安装文件，双击安装文件一步一步进行操作即可。

Linux 下的安装可能会更加简单，除了下载安装包进行安装外，一般的linux 仓库中都会有mysql ，我们只需要通过一个命令就可以下载安装：

Ubuntu\deepin

>>sudo apt-get install mysql-server

>>Sudo apt-get install  mysql-client

centOS/redhat

>>yum install mysql

**二，安装MySQL-python**

要想使python可以操作mysql 就需要MySQL-python驱动，它是python 操作mysql必不可少的模块。

下载地址：<https://pypi.python.org/pypi/MySQL-python/>

下载MySQL-python-1.2.5.zip 文件之后直接解压。进入MySQL-python-1.2.5目录:

>>python setup.py install

**三，测试**

测试非常简单，检查MySQLdb 模块是否可以正常导入。

fnngj@fnngj-H24X:~/pyse$ python

Python 2.7.4 (default, Sep 26 2013, 03:20:56)

[GCC 4.7.3] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>> import MySQLdb

没有报错提示MySQLdb模块找不到，说明安装OK ，下面开始使用python 操作数据库之前，我们有必要来回顾一下mysql的基本操作：

**四，mysql 的基本操作**

$ mysql -u root -p  （有密码时）

$ mysql -u root     （无密码时）

复制代码

复制代码

mysql> show databases; // 查看当前所有的数据库

+--------------------+

| Database |

+--------------------+

| information\_schema |

| csvt |

| csvt04 |

| mysql |

| performance\_schema |

| test |

+--------------------+

6 rows in set (0.18 sec)

mysql> use test; //作用与test数据库

Database changed

mysql> show tables; //查看test库下面的表

Empty set (0.00 sec)

//创建user表，name 和password 两个字段

mysql> CREATE TABLE user (name VARCHAR(20),password VARCHAR(20)); Query OK, 0 rows affected (0.27 sec)

//向user表内插入若干条数据

mysql> insert into user values('Tom','1321');

Query OK, 1 row affected (0.05 sec)

mysql> insert into user values('Alen','7875');

Query OK, 1 row affected (0.08 sec)

mysql> insert into user values('Jack','7455');

Query OK, 1 row affected (0.04 sec)

//查看user表的数据

mysql> select \* from user;

+------+----------+

| name | password |

+------+----------+

| Tom | 1321 |

| Alen | 7875 |

| Jack | 7455 |

+------+----------+

3 rows in set (0.01 sec)

//删除name 等于Jack的数据

mysql> delete from user where name = 'Jack';

Query OK, 1 rows affected (0.06 sec)

//修改name等于Alen 的password 为 1111

mysql> update user set password='1111' where name = 'Alen';

Query OK, 1 row affected (0.05 sec)

Rows matched: 1 Changed: 1 Warnings: 0

//查看表内容

mysql> select \* from user;

+--------+----------+

| name | password |

+--------+----------+

| Tom | 1321 |

| Alen | 1111 |

+--------+----------+

3 rows in set (0.00 sec)

复制代码

复制代码

**五，python 操作mysql数据库基础**

复制代码

复制代码

#coding=utf-8

import MySQLdb

conn= MySQLdb.connect(

host='localhost',

port = 3306,

user='root',

passwd='123456',

db ='test',

)

cur = conn.cursor()

#创建数据表

#cur.execute("create table student(id int ,name varchar(20),class varchar(30),age varchar(10))")

#插入一条数据

#cur.execute("insert into student values('2','Tom','3 year 2 class','9')")

#修改查询条件的数据

#cur.execute("update student set class='3 year 1 class' where name = 'Tom'")

#删除查询条件的数据

#cur.execute("delete from student where age='9'")

cur.close()

conn.commit()

conn.close()

复制代码

复制代码

>>> conn = MySQLdb.connect(host='localhost',port = 3306,user='root', passwd='123456',db ='test',)

Connect() 方法用于创建数据库的连接，里面可以指定参数：用户名，密码，主机等信息。

这只是连接到了数据库，要想操作数据库需要创建游标。

>>> cur = conn.cursor()

通过获取到的数据库连接conn下的cursor()方法来创建游标。

>>> cur.execute("create table student(id int ,name varchar(20),class varchar(30),age varchar(10))")

通过游标cur 操作execute()方法可以写入纯sql语句。通过execute()方法中写如sql语句来对数据进行操作。

>>>cur.close()

cur.close() 关闭游标

>>>conn.commit()

conn.commit()方法在提交事物，在向数据库插入一条数据时必须要有这个方法，否则数据不会被真正的插入。

>>>conn.close()

Conn.close()关闭数据库连接

**六，插入数据**

通过上面execute()方法中写入纯的sql语句来插入数据并不方便。如：

>>>cur.execute("insert into student values('2','Tom','3 year 2 class','9')")

我要想插入新的数据，必须要对这条语句中的值做修改。我们可以做如下修改：

复制代码

复制代码

#coding=utf-8

import MySQLdb

conn= MySQLdb.connect(

host='localhost',

port = 3306,

user='root',

passwd='123456',

db ='test',

)

cur = conn.cursor()

#插入一条数据

sqli="insert into student values(%s,%s,%s,%s)"

cur.execute(sqli,('3','Huhu','2 year 1 class','7'))

cur.close()

conn.commit()

conn.close()

复制代码

复制代码

假如要一次向数据表中插入多条值呢？

复制代码

复制代码

#coding=utf-8

import MySQLdb

conn= MySQLdb.connect(

host='localhost',

port = 3306,

user='root',

passwd='123456',

db ='test',

)

cur = conn.cursor()

#一次插入多条记录

sqli="insert into student values(%s,%s,%s,%s)"

cur.executemany(sqli,[

('3','Tom','1 year 1 class','6'),

('3','Jack','2 year 1 class','7'),

('3','Yaheng','2 year 2 class','7'),

])

cur.close()

conn.commit()

conn.close()

复制代码

复制代码

executemany()方法可以一次插入多条值，执行单挑sql语句,但是重复执行参数列表里的参数,返回值为受影响的行数。

**七，查询数据**

也许你已经尝试了在python中通过

>>>cur.execute("select \* from student")

来查询数据表中的数据，但它并没有把表中的数据打印出来，有些失望。

来看看这条语句获得的是什么

>>>aa=cur.execute("select \* from student")

>>>print aa

5

它获得的只是我们的表中有多少条数据。那怎样才能获得表中的数据呢？进入python shell

复制代码

复制代码

>>> import MySQLdb

>>> conn = MySQLdb.connect(host='localhost',port = 3306,user='root', passwd='123456',db ='test',)

>>> cur = conn.cursor()

>>> cur.execute("select \* from student")

5L

>>> cur.fetchone()

(1L, 'Alen', '1 year 2 class', '6')

>>> cur.fetchone()

(3L, 'Huhu', '2 year 1 class', '7')

>>> cur.fetchone()

(3L, 'Tom', '1 year 1 class', '6')

...

>>>cur.scroll(0,'absolute')

复制代码

复制代码

　　fetchone()方法可以帮助我们获得表中的数据，可是每次执行cur.fetchone() 获得的数据都不一样，换句话说我没执行一次，游标会从表中的第一条数据移动到下一条数据的位置，所以，我再次执行的时候得到的是第二条数据。

　　scroll(0,'absolute') 方法可以将游标定位到表中的第一条数据。

还是没解决我们想要的结果，如何获得表中的多条数据并打印出来呢？

复制代码

复制代码

#coding=utf-8

import MySQLdb

conn= MySQLdb.connect(

host='localhost',

port = 3306,

user='root',

passwd='123456',

db ='test',

)

cur = conn.cursor()

#获得表中有多少条数据

aa=cur.execute("select \* from student")

print aa

#打印表中的多少数据

info = cur.fetchmany(aa)

for ii in info:

print ii

cur.close()

conn.commit()

conn.close()

复制代码

复制代码

　　通过之前的print aa 我们知道当前的表中有5条数据，fetchmany()方法可以获得多条数据，但需要指定数据的条数，通过一个for循环就可以把多条数据打印出啦！执行结果如下：

复制代码

复制代码

5

(1L, 'Alen', '1 year 2 class', '6')

(3L, 'Huhu', '2 year 1 class', '7')

(3L, 'Tom', '1 year 1 class', '6')

(3L, 'Jack', '2 year 1 class', '7')

(3L, 'Yaheng', '2 year 2 class', '7')

[Finished in 0.1s]