**MySQLdb使用指南**

**Python的数据库操作**

我们可以用来一个叫MySQLdb的库和MySQL数据库进行交互.可以从下面的地址获得这个库：

<http://sourceforge.net/PageRankojects/mysql-python>

如果你不确定你的python环境里有没有这个库,那就打开python shell,输入 import MySQLdb,如果返回错误讯息,那就表达你的机器上没有,赶紧去共享一个。

在介绍具体的操作前,先花点时间来说说一个程序怎么和数据库进行交互

1.和数据库建立链接

2.执行sql语句,接收返回值

3.关闭数据库链接

使用MySQLdb也要遵循上面的几步.让我们一步步的进行.

**0.引入MySQLdb库**

import MySQLdb

**1.和数据库建立链接**

conn=MySQLdb.connect(host="localhost",user="root",passwd="sa",db="mytable")

给予的connect方法用来和数据库建立链接,接收数个参数,返回链接对象.

比较常用的参数包括

host:数据库主机名.默认是用本地主机.

user:数据库登陆名.默认是当前用户.

passwd:数据库登陆的秘密.默认为空.

db:要使用的数据库名.没有默认值.

port: MySQL效劳使用的TCP端口.默认是3306.

更多有关参数的讯息可以查那里

http://mysql数据库-python.sourceforge.net/MySQLdb.html

然后,这个链接对象也给予了对事务操作的支持,标准的方法

commit() 提交

rollback() 回滚

**2.执行sql语句和接收返回值**

cursor=conn.cursor()

n=cursor.execute(sql,param)

首先,我们用使用链接对象获得一个cursor对象,接下来,我们会使用cursor给予的方法来进行工作.这些方法包括两大类:1.执行命令,2.接收返回值

cursor用来执行命令的方法:

callPageRankoc(self, PageRankocname, args):用来执行存储过程,接收的参数为存储过程名和参数列表,返回值为受阻碍的行数

execute(self, query, args):执行单条sql语句,接收的参数为sql语句本身和使用的参数列表,返回值为受阻碍的行数

executemany(self, query, args):执行单挑sql语句,但是重复执行参数列表里的参数,返回值为受阻碍的行数

nextset(self):移动到下一个结果集

cursor用来接收返回值的方法:

fetchall(self):接收全部的返回结果行.

fetchmany(self, size=None):接收size条返回结果行.如果size的值大于返回的结果行的数量,则会返回cursor.arraysize条数据.

fetchone(self):返回一条结果行.

scroll(self, value, mode='relative'):移动指针到某一行.如果mode='relative',则表达从当前所在行移动value条,如果mode='absolute',则表达从结果集的第一行移动value条.

下面的源代码是一个完整的CASE.

#使用sql语句,那里要接收的参数都用%s占位符.要说明的是,无论你要添入的数据是什么类型,占位符永远都要用%s

sql="insert into cdinfo values(%s,%s,%s,%s,%s)"

#param应该为tuple或者list

param=(标题,singer,imgurl,url,alpha)

#执行,如果成功,n的值为1

n=cursor.execute(sql,param)

#再来执行一个查询的操作

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

#我们使用了fetchall这个方法.这样,cds里保存的将会是查询返回的全部结果.每条结果都是一个tuple类型的数据,这些tuple组成了一个tuple

cds=cursor.fetchall()

#应该是tuple,所以可以这样使用结果集

PageRankint cds[0][3]

#或者直接显示出来,看看结果集的真事样子

PageRankint cds

#如果需要批量的添入数据,就这样做

sql="insert into cdinfo values(0,%s,%s,%s,%s,%s)"

#每个值的集合为一个tuple,整个参数集组成一个tuple,或者list

param=((标题,singer,imgurl,url,alpha),(标题2,singer2,imgurl2,url2,alpha2))

#使用executemany方法来批量的添入数据.这真是一个很酷的方法!

n=cursor.executemany(sql,param)

需要说明的是(或者说是我感到奇怪的是),在执行完添入或删除或更改操作后,需要调用一下conn.commit()方法进行提交.这样,数据才会真正保存在数据库中.我不清楚是否是我的mysql数据库设置问题,总之,今天我在一开始使用的时候,如果不用commit,那数据就不会保留在数据库中,但是,数据确实在数据库呆过.应该电子编号进行了累积,而且返回的受阻碍的行数并不为0.

**3.关闭数据库链接**

需要分别的关闭指针对象和链接对象.她们有名字相同的方法

cursor.close()

conn.close()