# 一、命令

mysql -uroot -p 进入数据库

create database ---name---创建数据库

use ---datebaeseName--使用数据库

show tables 显示表

show database 显示数据库

truncate table 你的表名 #索引从0开始

# 二、运行sql文件

1/创建数据

2/进入库

3/source /opt/mysql.sql

# 三、导出

mysqldump -u用戶名 -p密码 -d 数据库名 表名 > 脚本名;

导出整个数据库结构和数据  
mysqldump -h localhost -uroot -p123456 database > dump.sql

导出单个数据表结构和数据  
mysqldump -h localhost -uroot -p123456  database table > dump.sql

导出整个数据库结构（不包含数据）  
mysqldump -h localhost -uroot -p123456  -d database > dump.sql

导出单个数据表结构（不包含数据）  
mysqldump -h localhost -uroot -p123456  -d database table > dump.sql

# 四、MySQL-python

import MySQLdb

# 打开数据库连接

db = MySQLdb.connect(

host="localhost",

user="root",

port="3306",

password="humingfei212697~",

db="codelion",

charset='utf8'

)

# 使用cursor()方法获取操作游标

cursor = db.cursor()

#执行

cursor.execute("")

# 使用 fetchone() 方法获取一条数据

# data = cursor.fetchone()

print "Database version : %s " % data

# 关闭数据库连接

db.close()

execute(sql)

　　可接受一条语句从而执行

executemany(templet,args)

　　能同时执行多条语句，执行同样多的语句可比execute()快很多，强烈建议执行多条语句时使用executemany

　　templet : sql模板字符串,

                 例如     'insert into table(id,name) values(%s,%s)'

　　args: 模板字符串的参数，是一个列表，列表中的每一个元素必须是元组！！！

                例如：  [(1,'小明'),(2,'zeke'),(3,'琦琦'),(4,'韩梅梅')]

# 五、pymysql

python3不支持mysql-python,需要使用pymysql

## 1、连接

import pymysql

conn = pymysql.connect(

host='192.168.9.99', #数据库主机名.默认是用本地主机

port=3306, #MySQL服务使用的TCP端口.默认是3306

user='root', #数据库登陆名.默认是当前用户

password='123', #数据库登陆的秘密.默认为空

db = 'dbname', #要使用的数据库名.没有默认值

charset='utf8', #数据库编码

)

cursor = conn.cursor()

## conn对象

conn.commit() #提交

conn.roback() #滚回

## 执行命令

**cursor.mogrify(sql, value)**

显示sql语句是怎么写的

**cursor.execute(sql,param)**

select返回值是受影响行数

**executemany(self, query, args)**

执行单条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

---------------------------------------------------------------------------------------

**callproc(procname, args)**

用来执行存储过程

------------------------------------------------------------------------------------------

**nextset(self)**移动到下一个结果集

## 4、获取查询结果

**fetchone(self):**

返回一条结果行 type：

--------------------------------------

**fetchall(self):**

全部的返回结果行 type：list

------------------------------------------

**fetchmany(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.

-----------------------------------------------------

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

## 5、关闭数据库连接

需要分别的关闭指针对象和连接对象.他们有名字相同的方法   
cursor.close()

conn.close()

## 6、注意：

**使用sql语句,这里要接收的参数都用%s占位符.要注意的是,无论你要插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=(title,singer,imgurl,url,alpha)

#执行,如果成功,n的值为1

n=cursor.execute(sql,param)

#如果需要批量的插入数据,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=((title,singer,imgurl,url,alpha),(title2,singer2,imgurl2,url2,alpha2))

#使用executemany方法来批量的插入数据.这真是一个很酷的方法!   
n=cursor.executemany(sql,param)