# 通用：打开文件

## open()

### 参数：

w：以写方式打开，

a：以追加模式打开 (从 EOF 开始, 必要时创建新文件)

r+：以读写模式打开

w+：以读写模式打开 (参见 w )

a+：以读写模式打开 (参见 a )

rb：以二进制读模式打开

wb：以二进制写模式打开 (参见 w )

ab：以二进制追加模式打开 (参见 a )

rb+：以二进制读写模式打开 (参见 r+ )

wb+：以二进制读写模式打开 (参见 w+ )

ab+：以二进制读写模式打开 (参见 a+ )

# CSV文件：

## 引入库：

import csv

## 读取：

* 方法一：

import csv

#读取csv至字典

csvFile = open(

"C:\\Users\\Simon Fong\\Desktop\\Nation\_list.csv",

"r")

reader = csv.reader(csvFile)

# 建立空字典

result = {}

for item in reader:

#忽略第一行

if reader.line\_num == 1:

continue

result[item[0]] = item[1]

csvFile.close()

print(result)

* 方法二：

import csv

csvFile = open("C:\\Users\\Simon Fong\\Desktop\\Nation\_list.csv", "r",encoding="utf-8")

reader = csv.reader(csvFile)

for item in reader:

print(item)

## 创建：

import csv

path = "country.csv"                            #文件路径

with open(path,'wb') as f:                      #打开文件，以二进制写入打开

    csv\_write = csv.writer(f)                   #调用csv模块

    csv\_head = ["good","bad"]                   #创建一个列表作为表头

    csv\_write.writerow(csv\_head)                #向csv写入行

## 写入：

# sqlite3文件

## 基本概念

### 1）引入库：

#导入sqlite3库文件

import sqlite3

### 2）connection --- 数据库连接对象

连接对象：建立python客户端与数据库的网络连接

创建方法：sqlite3.connect(参数)

path = "C:\\Users\\Simon Fong\\source\\repos\\ArgusTest\\db.sqlite3"

conn = sqlite3.connect(path)

connection对象支持的方法：

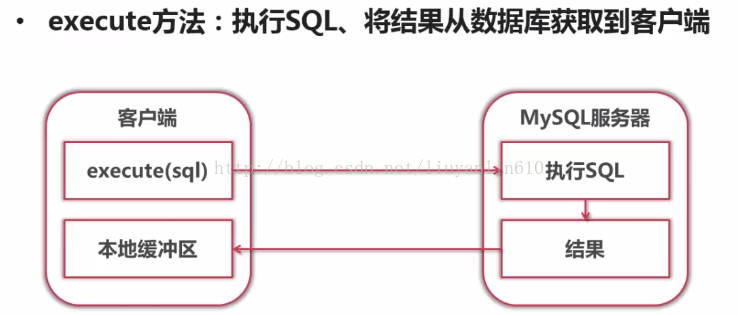
|  |  |
| --- | --- |
| 方法名 | 说明 |
| Cursor() | 使用该连接创建并返回游标 |
| Commit() | 提交当前事务 |
| Rollback() | 回滚当前事务 |
| Close() | 关闭连接 |

### 3）cursor --- 游标对象

游标对象：用于执行查询和获取结果

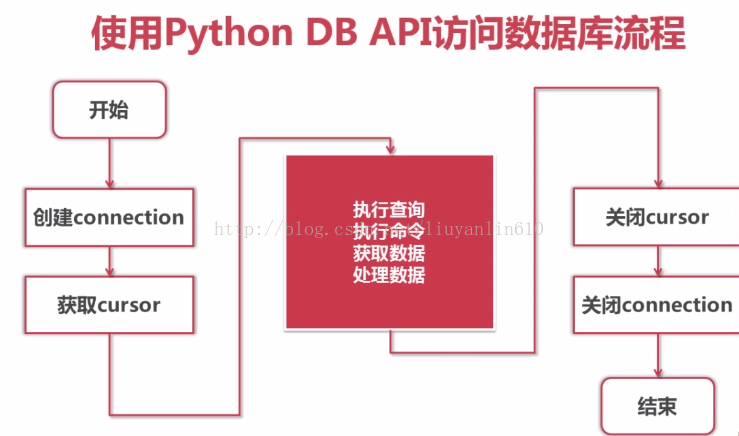
cursor对象支持的方法：

|  |  |
| --- | --- |
| 参数名 | 说明 |
| Execute(op[,args]) | 执行一个数据库查询和命令 |
| Fetchone() | 取得结果集的下一行 |
| Fetchmany(size) | 获取结果集的下几行 |
| Fetchall() | 获取结果集中剩下的所有行 |
| Rowcount | 最近一次execute返回数据的行数或影响行数 |
| Close() | 关闭游标对象 |

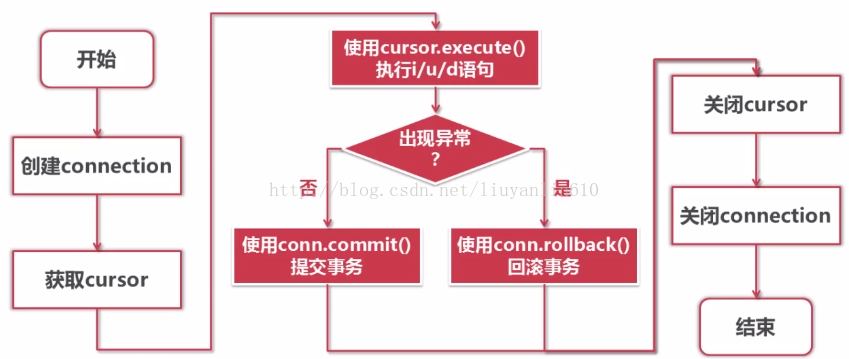




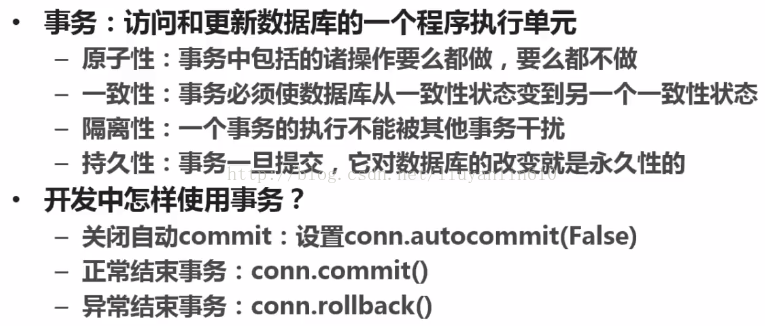
### 4）增删改查的流程



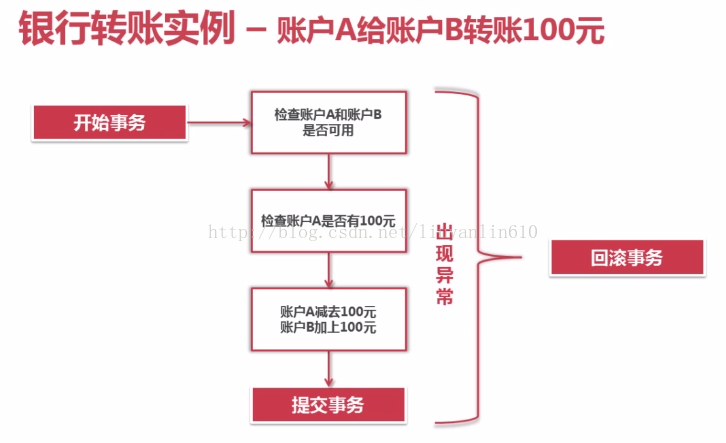
insert/upgrade/delete语句



### 5）事务:保证一组操作要么同时执行，要么都不执行



实例：



## 数据表操作

### 创建数据表：

#导入sqlite3库文件

import sqlite3

#数据库存在时，直接连接；不存在时，创建相应数据库，此时当前目录下可以找到对应的数据库文件。

path = "C:\\Users\\Simon Fong\\source\\repos\\PythonCMD\\test.sqlite3"

conn = sqlite3.connect(path)

cursor = conn.cursor() #创建游标对象

#执行查询语句

sql1 = """create table category(

id int primary key,

sort int,

name text)"""

cursor.execute(sql1) #执行sql语句

sql2 = """create table book(

id int primary key,

sort int,

name text,

price real,

category int,

foreign key (category) references category(id))"""

cursor.execute(sql2) #执行sql语句

cursor.close() #关闭游标对象

conn.commit() #提交事务

conn.close() #关闭数据连接

### 删除数据表：

SQLite 的 **DROP TABLE** 语句用来删除表定义及其所有相关数据、索引、触发器、约束和该表的权限规范。

DROP TABLE database\_name.table\_name;

#导入sqlite3库文件

import sqlite3

#数据库存在时，直接连接；不存在时，创建相应数据库，此时当前目录下可以找到对应的数据库文件。

path = "C:\\Users\\Simon Fong\\source\\repos\\PythonCMD\\test.sqlite3" #数据库路径

conn = sqlite3.connect(path) #创建数据库连接

cursor = conn.cursor() #创建游标对象

sql = "drop table category"

cursor.execute(sql)

conn.commit()

cursor.close() #关闭游标对象

conn.close() #关闭数据连接

### 查询所有数据表名：

path = "C:\\Users\\Simon Fong\\source\\repos\\ArgusTest\\db.sqlite3" #数据库路径

conn = sqlite3.connect(path) #创建数据库连接

cursor = conn.cursor() #创建游标对象

sql = "select name from sqlite\_master where type='table'order by 'name'"

try:

cursor.execute(sql) #执行SQL语句

print(cursor.fetchall()) #返回所有记录

except Exception as e:

print(e)

conn.rollback

cursor.close() #关闭游标对象

conn.close() #关闭数据连接

## 数据操作：

### 插入数据insert：

#导入sqlite3库文件

import sqlite3

#数据库存在时，直接连接；不存在时，创建相应数据库，此时当前目录下可以找到对应的数据库文件。

path = "C:\\Users\\Simon Fong\\source\\repos\\PythonCMD\\test.sqlite3" #数据库路径

conn = sqlite3.connect(path) #创建数据库连接

cursor = conn.cursor() #创建游标对象

#创建列表

books = [(1,1,'Cook Recipe',3.12,1),

(2,3,'Python Intro',17.5,2),

(3,2,'OS Intro',13.6,2),

]

#执行查询语句

#插入一行数据

#cursor.execute("insert into category values(1,1,'kitchen')")

#插入一行数据

cursor.execute("insert into category values(?,?,?)",(2,2,'computer'))

#插入多行数据

cursor.executemany("insert into book values(?,?,?,?,?)",books)

cursor.close() #关闭游标对象

conn.commit() #提交事务

conn.close() #关闭数据连接

### 查询数据select

#导入sqlite3库文件

import sqlite3

#数据库存在时，直接连接；不存在时，创建相应数据库，此时当前目录下可以找到对应的数据库文件。

path = "C:\\Users\\Simon Fong\\source\\repos\\PythonCMD\\test.sqlite3" #数据库路径

conn = sqlite3.connect(path) #创建数据库连接

cursor = conn.cursor() #创建游标对象

sql = "select name from category order by sort"

cursor.execute(sql) #执行SQL语句

print(cursor.fetchone()) #返回一个记录

print(cursor.fetchall()) #返回所有记录

sql = "select \* from book where book.category = 1"

cursor.execute(sql) #执行SQL语句

print(cursor.fetchall()) #返回所有记录

#使用for循环执行sql语句

for row in cursor.execute("select name, price from book order by sort"):

print(row)

cursor.close() #关闭游标对象

conn.close() #关闭数据连接

### 更新与删除update/delete

#导入sqlite3库文件

import sqlite3

#数据库存在时，直接连接；不存在时，创建相应数据库，此时当前目录下可以找到对应的数据库文件。

path = "C:\\Users\\Simon Fong\\source\\repos\\PythonCMD\\test.sqlite3" #数据库路径

conn = sqlite3.connect(path) #创建数据库连接

cursor = conn.cursor() #创建游标对象

sql = "update book set price = ? where id = ?"

cursor.execute(sql,(1000,1)) #执行SQL语句

cursor.execute("delete from book where id = 2")

conn.commit() #提交事务

cursor.close() #关闭游标对象

conn.close() #关闭数据连接

### 异常处理:增删改查

#导入sqlite3库文件

import sqlite3

#数据库存在时，直接连接；不存在时，创建相应数据库，此时当前目录下可以找到对应的数据库文件。

path = "C:\\Users\\Simon Fong\\source\\repos\\PythonCMD\\test.sqlite3" #数据库路径

conn = sqlite3.connect(path) #创建数据库连接

cursor = conn.cursor() #创建游标对象

com = [(1,'Lili'),

(2,'Tom'),

(3,'Bob')

]

sql\_insert = "insert into company values(?,?)"

sql\_select = "select \* from company"

sql\_upgrade = "update company set name = 'A' where id = 2"

sql\_delete = "delete from company where id < 2"

try:

cursor.executemany(sql\_insert,com)

cursor.execute(sql\_upgrade)

cursor.execute(sql\_delete)

conn.commit() #提交事务

except Exception as e:

print(e) #输出异常信息：no such table:company

conn.rollback #回滚当前事务

cursor.execute(sql\_select)

print(cursor.fetchall())

cursor.close() #关闭游标对象

conn.close() #关闭数据连接