# 数据库编程

## 操作SQLite3数据库

从Python3.x版本开始，在标准库中已经内置了SQLlite3模块，它可以支持SQLite3数据库的访问和相关的数据库操作。在需要操作SQLite3数据库数据时，只须在程序中导入SQLite3模块即可。Python语言操作SQLite3数据库的基本流程如下所示。

1. 导入相关库或模块（SQLite3）。
2. 使用connect()连接数据库并获取数据库连接对象。它提供了以下方法：

.cursor() 方法来创建一个游标对象   
 .commit() 方法来处理事务提交   
 .rollback() 方法来处理事务回滚   
 .close() 方法来关闭一个数据库连接

1. 使用con.cursor()获取游标对象。
2. 使用游标对象的方法(execute()、executemany()、fetchall()等)来操作数据库，实现插入、修改和删除操作，并查询获取显示相关的记录。在Python程序中，连接函数sqlite3.connect()有如下两个常用参数。

database：表示要访问的数据库名。

timeout：表示访问数据的超时设定。

1. 使用close()关闭游标对象和数据库连接。数据库操作完成之后，必须及时调用其close()方法关闭数据库连接，这样做的目的是减轻数据库服务器的压力。

### 使用SQLite3创建表

使用sqlite3模块的connect方法来创建/打开数据库，需要指定数据库路径，不存在则创建一个新的数据库。

|  |
| --- |
| con=sqlite3.connect((r'Python数据库/database/db.db')) |

下面实例代码演示使用SQLite3创建数据库的过程。

**【示例】使用SQLite3创建表**

|  |
| --- |
| #导入sqllite3模块  import sqlite3  # 1.硬盘上创建连接  con = sqlite3.connect(r'Python数据库/database/db.db')  # 获取cursor对象  cur = con.cursor()  # 执行sql创建表  sql = 'create table t\_person(pno INTEGER PRIMARY KEY AUTOINCREMENT ,pname varchar(30) NOT NULL ,age INTEGER)'  try:  cur.execute(sql)  except Exception as e:  print(e)  print('创建表失败')  finally:  # 关闭游标  cur.close()  # 关闭连接  con.close() |

### 使用SQLite3插入数据

调用游标对象的execute执行插入的sql，使用executemany()执行多条sql语句，使用executmany()比循环使用excute()执行多条sql语句效率高。

**【示例】使用SQLite3插入一条数据**

|  |
| --- |
| #导入sqllite3模块  import sqlite3  # 1.硬盘上创建连接  con = sqlite3.connect((r'Python数据库/database/db.db'))  # 获取cursor对象  cur = con.cursor()  # 执行sql创建表  sql = 'insert into t\_person(pname,age) values(?,?)'  try:  cur.execute(sql,('张三',23))  #提交事务  con.commit()  print('插入成功')  except Exception as e:  print(e)  print('插入失败')  con.rollback()  finally:  # 关闭游标  cur.close()  # 关闭连接  con.close() |

**【示例】使用SQLite3插入多条数据**

|  |
| --- |
| #导入sqllite3模块  import sqlite3  # 1.硬盘上创建连接  con = sqlite3.connect(r'Python数据库/database/db.db')  # 获取cursor对象  cur = con.cursor()  try:  #执行sql创建表  sql = 'insert into t\_person(pname,age) values(?,?)'  cur.executemany(sql, [('张三', 23), ('李四', 25), ('小红', 24), ('小李', 12)])  #提交事务  con.commit()  print('插入成功')  except Exception as e:  print('插入失败')  con.rollback()  finally:  # 关闭游标  cur.close()  # 关闭连接  con.close() |

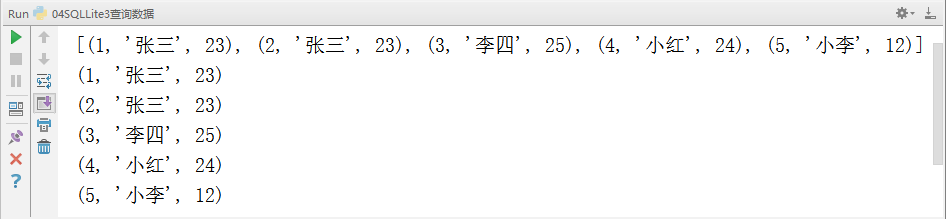
### 使用SQLite3查询数据

查询数据，游标对象提供了fetchall()和fetchone()方法 。fetchall()方法获取所有数据，返回一个列表。fetchone()方法获取其中一个结果，返回一个元组。

**【示例】fetchall()查询所有数据**

|  |
| --- |
| #导入sqllite3模块  import sqlite3  # 1.硬盘上创建连接  con = sqlite3.connect(r'Python数据库/database/db.db')  # 获取cursor对象  cur = con.cursor()  # 执行sql创建表  sql = 'select \* from t\_person'  try:  cur.execute(sql)  # 获取所有数据  person\_all = cur.fetchall()  # print(person\_all)  # 遍历  for p in person\_all:  print(p)  except Exception as e:  print(e)  print('查询失败')  finally:  # 关闭游标  cur.close()  # 关闭连接  con.close() |

执行结果如下图：



**【示例】fetchone()查询一条数据**

|  |
| --- |
| #导入sqllite3模块  import sqlite3  # 1.硬盘上创建连接  con = sqlite3.connect(r'Python数据库/database/db.db')  # 获取cursor对象  cur = con.cursor()  # 执行sql创建表  sql = 'select \* from t\_person'  try:  cur.execute(sql)  # 获取一条数据  person = cur.fetchone()  print(person)  except Exception as e:  print(e)  print('查询失败')  finally:  # 关闭游标  cur.close()  # 关闭连接  con.close() |

执行结果如下图：



**【示例】修改数据**

|  |
| --- |
| #导入sqllite3模块  import sqlite3  #1.硬盘上创建连接  con=sqlite3.connect(r'Python数据库/database/db.db')  #获取cursor对象  cur=con.cursor()  try:  #执行sql创建表  update\_sql = 'update t\_person set pname=? where pno=?'  cur.execute(update\_sql, ('小明', 1))  #提交事务  con.commit()  print('修改成功')  except Exception as e:  print(e)  print('修改失败')  con.rollback()  finally:  # 关闭游标  cur.close()  # 关闭连接  con.close() |

执行结果如下图：



**【示例】删除数据**

|  |
| --- |
| #导入sqllite3模块  import sqlite3  #1.硬盘上创建连接  con=sqlite3.connect( r'Python数据库/database/db.db')  #获取cursor对象  cur=con.cursor()  #执行sql创建表  delete\_sql = 'delete from t\_person where pno=?'  try:  cur.execute(delete\_sql, (2,))  #提交事务  con.commit()  print('删除成功')  except Exception as e:  print(e)  print('删除失败')  con.rollback()  finally:  # 关闭游标  cur.close()  # 关闭连接  con.close() |

执行结果如下图：



在上述实例代码中，首先定义查询所有数据、插入数据、修改数据、删除数据的方法。然后，定义主方法中依次建立连接，获取连接的cursor，通过cursor的execute()等方法来执行SQL语句，调用插入记录、更加记录、删除记录的方法。