

[专栏首页](#) [凹凸玩数据](#) 通过cursor游标讲解，带你初步搞懂python操作mysql数据库

通过cursor游标讲解，带你初步搞懂python操作mysql数据库

2020-01-16 阅读 1.7K

01

什么是游标？

游标，通俗的解释就是“游动的标志”，这是数据库中一个很重要的概念。

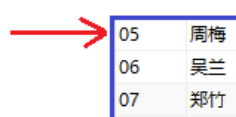
有时候，我们执行一条查询语句的时候，往往会得到N条返回结果，执行sql语句取出这些返回结果的接口(起始点)，就是游标。沿着这个游标，我们可以一次取出一行记录。

当不使用游标功能，我们去执行 `select * from student where age > 20;` 这条语句的时候，如果有1000条返回结果，系统会一次性将1000条记录返回到界面中，你没有选择，也不能做其他操作。

当我们开启了游标功能，再去执行这条语句的时候，系统会先帮你找到这些行，先给你存放起来，然后提供了一个游标接口。当你需要数据的时候，就借助这个游标去一行行的取出数据，你每取出一条记录，游标指针就朝前移动一次，一直到取完最后一行数据后。

一张图讲述游标的功能:

select * from student where ssex = "女";



| | | | |
|----|----|---------------------|---|
| 05 | 周梅 | 1991-12-01 00:00:00 | 女 |
| 06 | 吴兰 | 1992-01-01 00:00:00 | 女 |
| 07 | 郑竹 | 1989-01-01 00:00:00 | 女 |
| 09 | 张三 | 2017-12-20 00:00:00 | 女 |
| 10 | 李四 | 2017-12-25 00:00:00 | 女 |
| 11 | 李四 | 2012-06-06 00:00:00 | 女 |
| 12 | 赵六 | 2013-06-13 00:00:00 | 女 |
| 13 | 孙七 | 2014-06-01 00:00:00 | 女 |

图示说明:

假设我们是在“游标功能开启”的状态下，执行这条SQL语句。此时蓝色方框中的结果集并不会马上打印到屏幕上，而是将这些结果存储起来，提供一个游标接口，图中的红色箭头。当我们每获取其中一行数据的时候，这个红色游标就会沿着这个黑色箭头方向朝下，走一行；当再次获取第二行数据的时候，这个红色游标又会移动到第三行，一直到获取完结果集中的所有数据为止。

2

使用游标的好处？

如果不使用游标功能，直接使用select查询，会一次性将结果集打印到屏幕上，你无法针对结果集做第二次编程。

作者介绍

朱小五

关注

专栏

| 文章 | 阅读量 | 获赞 | 作者排名 |
|-----|-------|-----|------|
| 259 | 82.6K | 549 | 723 |

精选专题

腾讯云原生专题

云原生技术干货，业务实践落地。

活动推荐

云+社区开启竞赛「寻...
快来报名吧!

[立即查看](#)

腾讯云自媒体分享计划
入驻云加社区，共享百万
资源包。

立即入驻

运营活动





3

利用python连接数据库

1) 以python连接mysql数据库为例

使用python连接数据库的时候，会经常使用游标这个功能。我们以python连接mysql数据库来说明使用游标的好处。

当我们使用python连接mysql的时候，那么python就相当于mysql服务器的一个客户端，我们利用python这个client去操纵mysql的server。

在pymysql中操作数据库，就是使用游标这种方式来获取表中的数据。

2) 使用游标的操作步骤

首先，使用pymysql连接上mysql数据库，得到一个数据库对象。

然后，我们必须开启数据库中的游标功能，得到一个游标对象。

接着，使用游标对象中的execute()方法，去执行某个SQL语句，系统会根据你的SQL语句，找到这些匹配行，给你存储起来，而不是一次性的打印到屏幕上。什么时候需要这个结果中的数据的时候，你就去获取它。

最后，就是获取结果集中的数据了，这里有两种方法获取结果集中的数据。一个是fetchone()，该方法一次获取一条记录，每一条记录是一个元组形式的数据，每获取一条记录游标会往前移动一格，等待获取下一条记录；一个是fetchall()方法，能够一次性的获取所有的数据，该方法返回的是一个元组列表。

当完成所有操作后，记得断开数据库的连接，释放资源。

① 连接数据库，创建一个数据库对象

```
db = pymysql.connect(host='localhost',user='root',db='huangwei',  
                    password='123456',port=3306,charset='utf8')
```

注意：

我们要操作的是huangwei这个数据库中的表，因此在连接的时候使db这个参数来指明要使用哪一个数据库；由于mysql数据库就装在本机上，因此可以写localhost，当然你也可以写成主机名，或者主机ip；

② 开启游标功能，创建游标对象

```
# 这里使用的是数据库对象db中的cursor()方法，  
cursor = db.cursor()
```

③ 使用execute()方法，执行SQL语句

```
cursor.execute('select sname,ssex from student')
```

注意：当开启游标功能执行这个SQL语句后，系统并不会将结果直接打印到屏幕上，而是将上述得到的结果，找个地方存储起来，提供一个游标接口给我们，当你需要获取数据的时候，就可以从中拿数据。

④ 使用fetchone()或fetchall()获取数据

```
# 一次性获取一条数据  
a = cursor.fetchone()  
# 一次性获取所有数据  
a = cursor.fetchall()
```

注意：使用游标获取数据，需要用个变量将结果存储起来，才能被我们拿来二次使用，这里在下面的案例中会体现出来。

04

案例讲解

需求：以操作huangwei数据库中的student表为例。完成【筛选出女生，同时将sname 以这种形式输出“我叫周梅，我是一个学生!”】这个需求。

① student表

| 对象 | student @huangwei (localh... | | | |
|------|------------------------------|---------------------|------|-------|
| 开始事务 | 文本 | 筛选 | 排序 | 导入 导出 |
| sid | sname | sage | ssex | |
| 01 | 赵雷 | 1990-01-01 00:00:00 | 男 | |
| 02 | 钱电 | 1990-12-21 00:00:00 | 男 | |
| 03 | 孙风 | 1990-12-20 00:00:00 | 男 | |
| 04 | 李云 | 1990-12-06 00:00:00 | 男 | |
| 05 | 周梅 | 1991-12-01 00:00:00 | 女 | |
| 06 | 吴兰 | 1992-01-01 00:00:00 | 女 | |
| 07 | 郑竹 | 1989-01-01 00:00:00 | 女 | |
| 09 | 张三 | 2017-12-20 00:00:00 | 女 | |
| 10 | 李四 | 2017-12-25 00:00:00 | 女 | |
| 11 | 李四 | 2012-06-06 00:00:00 | 女 | |
| 12 | 赵六 | 2013-06-13 00:00:00 | 女 | |
| 13 | 孙七 | 2014-06-01 00:00:00 | 女 | |

② 使用fetchone()方法，获取SQL查询结果集中的数据

```
db = pymysql.connect(host='localhost',user='root',db='huangwei',
                      password='123456',port=3306,charset='utf8')

cursor = db.cursor()
cursor.execute('select count(*) from student')
aa = cursor.fetchone()
print(aa)
# 注意这一句一定是在循环之外，不能放到循环里面。
# 想想这是为什么？
cursor.execute('select sname,ssex from student')
for i in range(aa[0]):
    a,b = cursor.fetchone()
    if b == "女":
        a = "我叫{}, 我是一个学生!".format(a)
        print(a)
db.close()
```

结果如下：



```
cursor.execute('select count(*) from student')
aa = cursor.fetchone()
print(aa)
cursor.execute('select sname,ssex from student')
for i in range(aa[0]):
    a,b = cursor.fetchone()
    if b == "女":
        a = "我叫{},我是一个学生!".format(a)
        print(a)
db.close()
```

```
(12,)
我叫周梅,我是一个学生!
我叫吴兰,我是一个学生!
我叫郑竹,我是一个学生!
我叫张三,我是一个学生!
我叫李四,我是一个学生!
我叫李四,我是一个学生!
我叫赵六,我是一个学生!
我叫孙七,我是一个学生!
```

③ 使用fetchall()方法，获取SQL查询结果集中的数据

```
db = pymysql.connect(host='localhost',user='root',db='huangwei',
                      password='123456',port=3306,charset='utf8')

cursor = db.cursor()
cursor.execute('select sname,ssex from student')
aa = cursor.fetchall()
# print(aa)
for a,b in aa:
    if b == "女":
        a = "我叫{},我是一个学生!".format(a)
        print(a)
db.close()
```

结果如下：

```
In [8]: db = pymysql.connect(host='localhost',user='root',db='huangwei',
                             password='123456',port=3306,charset='utf8')

        cursor = db.cursor()
        cursor.execute('select sname,ssex from student')
        aa = cursor.fetchall()
        # print(aa)
        for a,b in aa:
            if b == "女":
                a = "我叫{},我是一个学生!".format(a)
                print(a)
        db.close()
```

```
我叫周梅,我是一个学生!
我叫吴兰,我是一个学生!
我叫郑竹,我是一个学生!
我叫张三,我是一个学生!
我叫李四,我是一个学生!
我叫李四,我是一个学生!
我叫赵六,我是一个学生!
我叫孙七,我是一个学生!
```

④ 结果分析

通过上述操作可以发现，这种游标功能可以帮助我们对SQL语句得到的结果集，进行二次开发，然后通过灵活的python语法，得到任何我们最终想要的结果集。

作者：Huang supreme，个人博客地址：https://blog.csdn.net/weixin_41261833

微信首发于公众号【凹凸玩数据】，有趣的不像个技术号

朱小五

259 篇文章

通过cursor游标讲解，带你初步搞懂python操作mysql数...

转到我的清单

明天的推文会详细讲解如何利用python操作mysql"建、查、增、删、改"！

读者交流群已建立，后台回复「微信群」即可

加入与作者一起讨论交流

在看+分享，人间真情

本文分享自微信公众号 - 凹凸数据（alltodata），作者：Huang supreme
原文出处及转载信息见文内详细说明，如有侵权，请联系 yunjia_community@tencent.com 删除。
原始发表时间：2020-01-08
本文参与[腾讯云自媒体分享计划](#)，欢迎正在阅读的你也加入，一起分享。

Python

举报

点赞 2

分享

0 条评论

我来说两句

[登录](#) 后参与评论

相关文章

python与MySQL数据库的交互实战

如果你想要使用python操作MySQL数据库，就必须先要安装pymysql库，这个

数据森麟

干货！python与MySQL数据库的交互实战

如果你想要使用python操作MySQL数据库，就必须先要安装pymysql库，这个

朱小五

scrapy爬虫框架（四）： scrapy中 yield使用详解

MySQL下载：点我 python MySQL驱动下载：pymysql（pyMySQL，直接用pip方式安装）

259 篇文章

朱小五

通过cursor游标讲解，带你初步搞懂python操作mysql数...

转到我的清单

2

1

python接口自动化（三十八）-python操作...

现在的招聘要求对QA人员的要求越来越高，测试的一些基础知识就不必说了，来说测试知识以外的，会不会一门...

北京-宏哥

干货 | 利用Python操作mysql数据库

本文主要讲解如何利用python中的pymysql库来对mysql数据库进行操作。

朱小五

python mysql模块

多次使用python操作Mysql数据库，先与大家分享一下，关于如何使用python操作mysql数据库。mysql并不是python自带的模块，因此需...

py3study

Day24访问数据库

使用SQLite SQLite是一种嵌入式数据库，它的数据库就是一个文件。由于SQLite本身是C写的，而且体积很小，所以，经常被集成到各种应用程序中，甚至在i...

林清猫耳

Python学习之旅（三十七）

程序运行的时候，数据都是在内存中的。当程序终止的时候，通常都需要将数据保存到磁盘上，无论是保存到本地磁盘，还是通过网络保存到服务器上，最终都会将数据写...

py3study

Python 连接MySQL

数据库技术(例如MySQL)在气象业务和其他商业行业中都有着广泛的应用，气象与电网结合的大项目甚至都用上了hadoop分布式存储，Hadoop中的Hive...

zhangqibot

专栏：008：MySQLdb及其银行模拟转账

谢伟

Python实现数据库一键导出为Excel表格！...

数据库数据导出为excel表格，也可以说是一个很常用的功能了。毕竟不是任何人都懂数据库操作语句的。

IT派

https://cloud.tencent.com/developer/article/1575066

6/8

2

1

朱小五
259 篇文章

通过cursor游标讲解，带你初步搞懂python操作mysql数...

转到我的清单

来说分为三步：

py3study

一周学习小总结，MySQL进阶~
今天的这篇推文，是对前六天MySQL学习的一个进阶学习和小总结，前两天是端午节，大家过得如何？休息之余，不要忘记坚持学习，加油！

DataScience

「Python爬虫系列讲解」六、Python 数据...
前文回顾： 「Python爬虫系列讲解」一、网络数据爬取概述
「Python爬虫系列讲解」二、Python知识初学 「Python...

荣仔_最靓的仔

Django学习-第六讲（上）:Django数据库原生的方法操作
我们使用Django来操作MySQL，实际上底层还是通过Python来操作的。因此我们想要用Django来操作MySQL，首先还是需要安装一个驱动程序。在Pyt...

小海怪的互联网

Python小白的数据库入门
sqlite3 以上两个是主流的关系型数据库，我们观察之后发现，它们与我们熟知的Excel好像也没有什么不同。关系型...

arcticfox

Postgresql 有了 psycopg2 操作测试更方便...
随着对DB们的要求越来越高，测试，查找问题，监控，做一些比较复杂的事情，没有程序来加入，模拟一些比较复杂...

AustinDatabases

Python快速学习第九天--安装并操作Mysql数据库
python操作mysql数据库 Python 标准数据库接口为 Python DB-API，Python DB-API 为开发人员提供了数据库应用编程接口。...

汤高

用python实现接口测试（四、操作MySQL）
使用Python进行MySQL的库主要有三个，Python-MySQL（更熟悉的名字可能是MySQLdb），PyMySQL和SQLAlchemy。

用户6367961

社区

专栏

互动问答

技术快讯

团队主页

开发者手册

智能钛AI

活动

原创分享计划

自媒体分享计划

邀请作者入驻

自荐上首页

在线直播

生态合作计划

资源

技术周刊

社区标签

开发者实验室

关于

视频介绍

社区规范

免责声明

联系我们

友情链接

云+社区



扫码关注云+社区
领取腾讯云代金券

| | | | | | | | |
|------|--------|------|-------|--------|------|------|-----------|
| 热门产品 | 域名注册 | 云服务器 | 区块链服务 | 消息队列 | 网络加速 | 云数据库 | 域名解析 |
| | 云存储 | 视频直播 | | | | | |
| 热门推荐 | 人脸识别 | 腾讯会议 | 企业云 | CDN 加速 | 视频通话 | 图像分析 | MySQL 数据库 |
| | SSL 证书 | 语音识别 | | | | | |
| 更多推荐 | 数据安全 | 负载均衡 | 短信 | 文字识别 | 云点播 | 商标注册 | 小程序开发 |
| | 网站监控 | 数据迁移 | | | | | |