

《用 Python 玩转数据》之 pandas 与 MySQL 数据库交互入门

Dazhuang@NJU

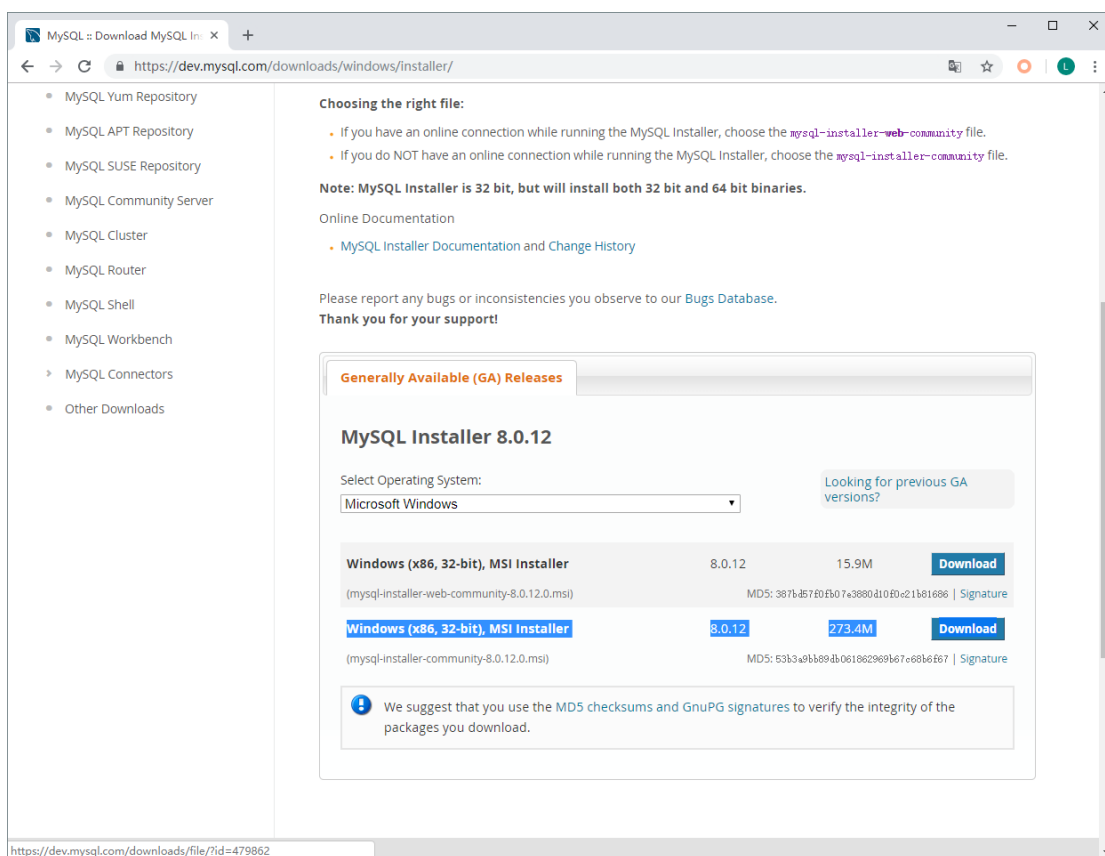
如果数据量较大分类较多，常常会把数据放入数据库中，也常常需要把数据库中的数据读出后进行处理，Python 和 pandas 中有方便的数据库连接和读写函数，以下简要介绍 pandas 与 MySQL 数据库交互的方式。

1. 安装 MySQL Server（如果已经安装则忽略）

下载相应操作系统的 MySQL 数据库安装包（MySQL Community Server 版）

url: <https://dev.mysql.com/downloads/mysql/>

Windows 版安装包示意如下：



下载后安装。

2. 测试 MySQL 数据库

安装后使用 MySQL Command Line 客户端进行功能测试，示意如下：

①用 `show databasename` 和 `use databasename` 查看所有的数据库名和使用某一个数据库；用 `show tables` 显示当前数据库中的数据表；

```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.12 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use mysql
Database changed
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| sys       |
+-----+
4 rows in set (0.01 sec)
```

②创建新的数据库 `test_db` 和数据表 `basic_info`，具体方式如下所示，其中使用了“`insert into tablename(各属性) values(各值)`”的方式插入了两条记录：

```
MySQL 8.0 Command Line Client

mysql> create database test_db;
Query OK, 1 row affected (0.07 sec)

mysql> use test_db;
Database changed
mysql> create table basic_info(
  -> id INT(11) PRIMARY KEY,
  -> name VARCHAR(25),
  -> age INT(3)
  -> );
Query OK, 0 rows affected (0.13 sec)

mysql> insert into basic_info(id, name, age) values(123, "Zhangsan", 18);
Query OK, 1 row affected (0.04 sec)

mysql> insert into basic_info(id, name, age) values(456, "Lisi", 23);
Query OK, 1 row affected (0.02 sec)

mysql> select * from basic_info;
+-----+-----+-----+
| id | name | age |
+-----+-----+-----+
| 123 | Zhangsan | 18 |
| 456 | Lisi | 23 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

3. pandas 与 MySQL 数据库交互

ORM（Object-Relational Mapping）对象关系映射技术指把关系数据库的表结构映射到对象上，通过使用描述对象和数据库之间映射的元数据将程序中的对象自动持久化到关系数据库中。Python 中最有名的 ORM 框架是 SQLAlchemy，需要安装该模块，除此之外还需要安装 pymysql 模块，pymysql 模块支持 Python 操作 MySQL 数据库。

```
$ pip install sqlalchemy
```

```
$ pip install pymysql
```

版本之间不兼容还常常需要进行如下安装:

```
$ pip install -i https://pypi.tuna.tsinghua.edu.cn/simple --upgrade sqlalchemy --ignore-installed --user
```

SQLAlchemy 模块提供了 `create_engine()` 函数用来初始化数据库连接, 方式为:

'数据库类型+数据库驱动名称://用户名:口令@机器地址:端口号/数据库名'

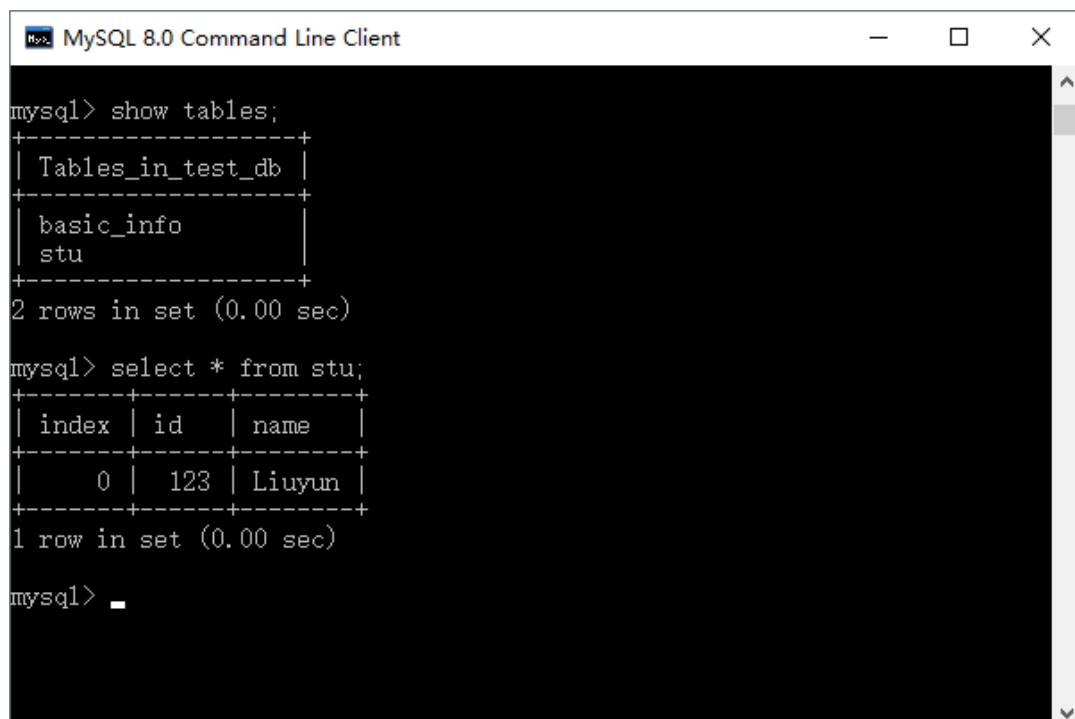
例 1: 将 DataFrame 对象写入一个 MySQL 数据库的新表中。

```
import pandas as pd
from sqlalchemy import create_engine

# 初始化数据库连接, MySQL 的用户为 root, 密码为 test123, 端口为 3306, 数据库为 test_db
engine=create_engine('mysql+pymysql://root:test123@localhost:3306/test_db')
# 查询表 basic_info 中的所有数据
sql = '''
    select * from basic_info;
    ...

df = pd.read_sql_query(sql, engine)
print(df)
# 新建一个 DataFrame
df1 = pd.DataFrame({'id':[123], 'name':['Liuyun']})
# 将新建的 DataFrame 储存到当前数据库的数据表 stu 中
df1.to_sql('stu', engine)
```

查询后发现正确创建了 stu 数据表并添加了内容:



```
mysql> show tables;
+-----+
| Tables_in_test_db |
+-----+
| basic_info         |
| stu                 |
+-----+
2 rows in set (0.00 sec)

mysql> select * from stu;
+-----+-----+-----+
| index | id   | name  |
+-----+-----+-----+
|      0 | 123 | Liuyun |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

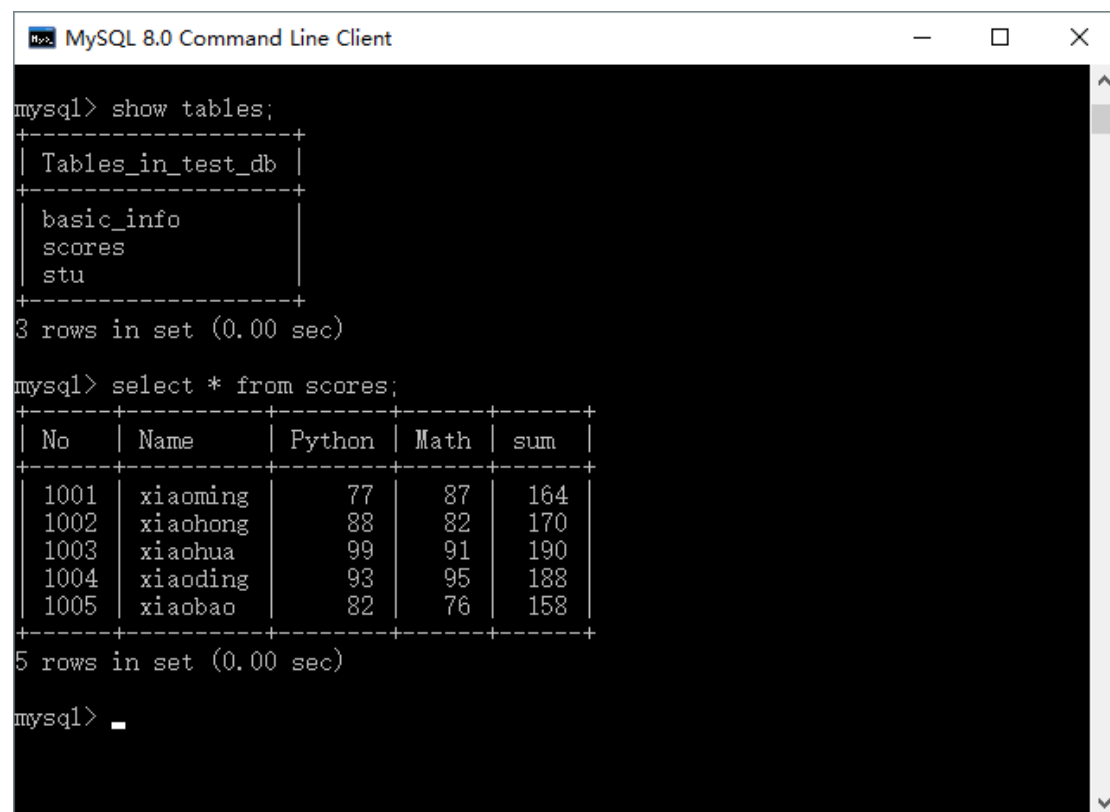
例 2：将 excel 表中数据写入 MySQL 的新表中，例如对于如下 Excel 表：

	A	B	C	D	E	F
1		No	Name	Python	Math	sum
2	0	1001	xiaoming	77	87	164
3	1	1002	xiaohong	88	82	170
4	2	1003	xiachua	99	91	190
5	3	1004	xiaoding	93	95	188
6	4	1005	xiaobao	82	76	158

只要接着执行如下代码：

```
df2 = pd.read_excel("stu_scores.xlsx")
df2.to_sql('scores', engine, index = False)    # 不储存 index 列
```

执行程序后查询发现，在 test_db 中增加了一张数据表 scores，其内容即为正确写入的 excel 表中的数据。



```
MySQL 8.0 Command Line Client

mysql> show tables;
+-----+
| Tables_in_test_db |
+-----+
| basic_info        |
| scores            |
| stu               |
+-----+
3 rows in set (0.00 sec)

mysql> select * from scores;
+----+-----+-----+-----+-----+
| No  | Name  | Python | Math | sum  |
+----+-----+-----+-----+-----+
| 1001 | xiaoming | 77    | 87    | 164  |
| 1002 | xiaohong | 88    | 82    | 170  |
| 1003 | xiachua | 99    | 91    | 190  |
| 1004 | xiaoding | 93    | 95    | 188  |
| 1005 | xiaobao | 82    | 76    | 158  |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

以上虽为两个简单示例，但已可以看到 pandas 如何与 MySQL 数据库进行连接的方式，若有兴趣可继续进行研究，其他与常用的与 Python 交互的数据库还有 MongoDB 和 Redis 等。