

Use Stack Overflow for Teams at work to find answers in a private and secure environment. Get your first 10 users free. [Sign up.](#)



Ask Question

python-pandas and databases like mysql

Asked 7 years, 5 months ago Active 3 months ago Viewed 110k times



96



75

The documentation for Pandas has numerous examples of best practices for working with data stored in various formats.

However, I am unable to find any good examples for working with databases like MySQL for example.

Can anyone point me to links or give some code snippets of how to convert query results using **mysql-python** to data frames in Pandas efficiently ?

python

pandas

share improve this question

edited Aug 23 '18 at 14:10



Scott Boston

69k ● 7 ● 43 ● 69

asked Apr 8 '12 at 18:01



user1320615

602 ● 1 ● 7 ● 5

1 Documentation: pandas.pydata.org/pandas-docs/stable/io.html#sql-queries – Mechanical snail Jun 1 '13 at 13:26

See also: stackoverflow.com/questions/15231646/... – Mechanical snail Jun 1 '13 at 13:29

Also take a look at [Blaze](#). – osa Dec 6 '14 at 3:59

If you're willing to spend money, I believe that Wes McKinney's book ("Python for Data Analysis") has some useful examples. – MTrenfield Dec 17 '14 at 18:00

add a comment

12 Answers



98

As Wes says, io/sql's `read_sql` will do it, once you've gotten a database connection using a DBI compatible library. We can look at two short examples using the `MySQLdb` and `cx_Oracle` libraries to connect to Oracle and MySQL and query their data dictionaries. Here is the example for `cx_Oracle`:

```
import pandas as pd
import cx_Oracle

ora_conn = cx_Oracle.connect('your_connection_string')
df_ora = pd.read_sql('select * from user_objects', con=ora_conn)
print 'loaded dataframe from Oracle. # Records: ', len(df_ora)
ora_conn.close()
```

And here is the equivalent example for `MySQLdb`:

```
import MySQLdb
mysql_cn= MySQLdb.connect(host='myhost',
                           port=3306,user='myusername', passwd='mypassword',
                           db='information_schema')
df_mysql = pd.read_sql('select * from VIEWS;', con=mysql_cn)
print 'loaded dataframe from MySQL. records:', len(df_mysql)
mysql_cn.close()
```

share improve this answer

edited Dec 29 '16 at 2:34

answered Jun 21 '12 at 12:34



Owen

1,086 ● 1 ● 11 ● 23



Keith C Campbell

996 ● 6 ● 3

add a comment

56

For recent readers of this question: pandas have the following warning in their [docs for version 14.0](#):

Warning: Some of the existing functions or function aliases have been deprecated and will be removed in future versions. This includes: `tquery`, `uquery`, `read_frame`, `frame_query`, `write_frame`.

And:

Warning: The support for the 'mysql' flavor when using DBAPI connection objects has been deprecated. MySQL will be further supported with SQLAlchemy engines (GH6900).

This makes many of the answers here outdated. You should use `sqlalchemy`:

```
from sqlalchemy import create_engine
import pandas as pd
engine = create_engine('dialect://user:pass@host:port/schema', echo=False)
f = pd.read_sql_query('SELECT * FROM mytable', engine, index_col = 'ID')
```

share improve this answer

edited Sep 4 '14 at 5:11

answered Jul 30 '14 at 7:04



Korem

6,853 ● 5 ● 38 ● 61

loading a table with 133 rows and 7 columns takes around 30 secs.. can you give some insights regarding why is that?
– [idoda](#) Sep 16 '14 at 13:43

@Korem I did thought about opening a new one, but I wanted to make sure it is not a trivial one first. When I use an mySql client (Sequel pro) and query the database, reuslts come up much faster. When you say "simply sending and then retrieving", is that what you mean? (using a client) – [idoda](#) Sep 17 '14 at 8:33

@idoda I mean comparing the time it takes to execute `engine.execute("select * FROM mytable")` with the time it takes to execute `pd.read_sql_query('SELECT * FROM mytable', engine)` – [Korem](#) Sep 17 '14 at 10:45

Can one pass a sqlalchemy query (session.query as in my answer below) directly to a pandas method? That would be a ripper! – [dmvianna](#) Feb 5 '15 at 2:06

[add a comment](#)

For the record, here is an example using a sqlite database:

23

```
import pandas as pd
import sqlite3

with sqlite3.connect("whatever.sqlite") as con:
    sql = "SELECT * FROM table_name"
    df = pd.read_sql_query(sql, con)
    print df.shape
```

[share](#) [improve this answer](#)

edited Nov 1 '15 at 22:15



[propjk007](#)

365 ● 7 ● 16

answered Feb 26 '13 at 18:58



[mbatchkarov](#)

11k ● 5 ● 46 ● 73

2 BTW `frame_query` is a synonym for `read_frame()` – [lexual](#) Apr 29 '13 at 5:43

1 You can specify the column to use as an index by specifying `index_col='timestamp'` in `frame_query`. – [Mechanical snail](#) Jun 1 '13 at 13:25

[add a comment](#)

I prefer to create queries with [SQLAlchemy](#), and then make a DataFrame from it. **SQLAlchemy** makes it easier to combine **SQL** conditions Pythonically if you intend to mix and match things over and over.

19

```
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy import Table
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
from pandas import DataFrame
import datetime

# We are connecting to an existing service
engine = create_engine('dialect://user:pwd@host:port/db', echo=False)
Session = sessionmaker(bind=engine)
session = Session()
Base = declarative_base()

# And we want to query an existing table
tablename = Table('tablename',
    Base.metadata,
    autoload=True,
    autoload_with=engine,
```



```
# create joins and limit results
us = tablename.c.country_code.in_(['US', 'MX'])
dc = tablename.c.locn_name.like('%DC%')
dt = tablename.c.arr_date >= datetime.date.today() # Give me convenience or...

q = session.query(tablename).\
    filter(us & dc & dt) # That's where the magic happens!!!

def querydb(query):
    """
    Function to execute query and return DataFrame.
    """
    df = DataFrame(query.all());
    df.columns = [x['name'] for x in query.column_descriptions]
    return df
```

share improve this answer

edited Apr 28 '14 at 3:21

answered Apr 28 '14 at 3:08



dmvianna

7,126 ● 12 ● 62 ● 93

Also you must specify the driver if it's not the same as [the SQLAlchemy's default one](#):

dialect+driver://user:pwd@host:port/db – Nuno André Sep 15 at 20:03

add a comment

MySQL example:

10

```
import MySQLdb as db
from pandas import DataFrame
from pandas.io.sql import frame_query

database = db.connect('localhost', 'username', 'password', 'database')
data      = frame_query("SELECT * FROM data", database)
```

share improve this answer

answered Jan 21 '14 at 14:03



aerkenemesis

487 ● 3 ● 13

6 `frame_query` is now deprecated. Now use `pd.read_sql(query, db)` instead. – Robert Smith Apr 24 '15 at 22:43

add a comment

The same syntax works for Ms SQL server using pyodbc also.

8

```
import pyodbc
import pandas.io.sql as psql

cnxn = pyodbc.connect('DRIVER={SQL Server};SERVER=servername;DATABASE=mydb;UID=username;PWD=')
cursor = cnxn.cursor()
sql = ("select * from mytable")

df = psql.frame_query(sql, cnxn)
cnxn.close()
```



hedgcutter
89 ● 1 ● 1

[add a comment](#)



5



And this is how you connect to PostgreSQL using psycopg2 driver (install with "apt-get install python-psycopg2" if you're on Debian Linux derivative OS).

```
import pandas.io.sql as psql
import psycopg2

conn = psycopg2.connect("dbname='datawarehouse' user='user1' host='localhost' password='ube")

q = """select month_idx, sum(payment) from bi_some_table"""

df3 = psql.frame_query(q, conn)
```

[share](#) [improve this answer](#)

answered Sep 11 '13 at 22:30

Will
1,636 ● 17 ● 18

[add a comment](#)



4



For Sybase the following works (with <http://python-sybase.sourceforge.net>)

```
import pandas.io.sql as psql
import Sybase

df = psql.frame_query("<Query>", con=Sybase.connect("<dsn>", "<user>", "<pwd>"))
```

[share](#) [improve this answer](#)

edited Jan 9 '13 at 22:48

answered Jan 9 '13 at 22:36

user1827356
4,566 ● 1 ● 14 ● 25

[add a comment](#)



4



`pandas.io.sql.frame_query` is deprecated. Use `pandas.read_sql` instead.

[share](#) [improve this answer](#)

edited Feb 14 '16 at 16:17

answered Dec 17 '14 at 17:34

joris
73.7k ● 27 ● 178 ● 162

ajkl
546 ● 1 ● 3 ● 19

[add a comment](#)



1



import the module

```
import pandas as pd
import oursql
```

connect



```
df_mysql = pd.read_sql(sql,conn)
print df_mysql
```

That works just fine and using pandas.io.sql frame_works (with the deprecation warning). Database used is the sample database from mysql tutorial.

share improve this answer

edited Feb 14 '16 at 15:52



Tobin Nary

4,046 ● 4 ● 24 ● 45

answered Feb 14 '16 at 14:57



user5925400

11 ● 2

add a comment

▲ This should work just fine.

0

```
import MySQLdb as mdb
import pandas as pd
con = mdb.connect('127.0.0.1', 'root', 'password', 'database_name');
with con:
    cur = con.cursor()
    cur.execute("select random_number_one, random_number_two, random_number_three from randomn
    rows = cur.fetchall()
    df = pd.DataFrame( [[ij for ij in i] for i in rows] )
    df.rename(columns={0: 'Random Number One', 1: 'Random Number Two', 2: 'Random Number Three
    print(df.head(20))
```

share improve this answer

answered Jan 31 '17 at 4:20



MontyPython

1,520 ● 9 ● 28 ● 48

add a comment

▲ This helped for me for connecting to **AWS MYSQL(RDS)** from **python 3.x** based **lambda function** and loading into a pandas DataFrame

0

```
import json
import boto3
import pymysql
import pandas as pd
user = 'username'
password = 'XXXXXXX'
client = boto3.client('rds')
def lambda_handler(event, context):
    conn = pymysql.connect(host='xxx.xxxxus-west-2.rds.amazonaws.com', port=3306, user=user
    df= pd.read_sql('select * from TableName limit 10',con=conn)
    print(df)
    # TODO implement
    #return {
    #    'statusCode': 200,
    #    'df': df
    #}
```

share improve this answer

answered May 28 at 8:20












Dheeraj


389 ● 3 ● 7





Your Answer

B *I*         

Sign up or [log in](#)

 Sign up using Google

 Sign up using Facebook

 Sign up using Email and Password

Post as a guest

Name

Email


Required, but never shown

Post Your Answer

By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#)


Not the answer you're looking for? Browse other questions tagged [python](#) [pandas](#) or [ask your own question](#).


Blog


 [Meet the Bots that Help Moderate Stack Overflow](#)



Featured on Meta

 Unicorn Meta Zoo #8: What does leadership look like in our communities?

 Stack Exchange and Stack Overflow are moving to CC BY-SA 4.0

 Join our dedicated Meta Stack Overflow chat room!

 2020 Developer Survey: any topic suggestions?


























Linked

- 9 [read frame with sqlalchemy, mysql and pandas](#)
- 7 [Using pandas.io.sql.read_frame, can I parse_dates, as in read_csv?](#)
- 2 [Simple Python-MySQL Bridge?](#)
- 4 [combine/average multiple data files](#)
- 2 [Select 7, 14, 20, 50, 200 day prices from OHLC data in SQL database.](#)
- 0 [How to train a DNN classification model using arbitrary data features stored in mysql?](#)

Related

- 4442 [Calling an external command in Python](#)
- 5315 [What are metaclasses in Python?](#)
- 2804 [Finding the index of an item given a list containing it in Python](#)
- 3119 [What is the difference between Python's list methods append and extend?](#)
- 3772 [How can I safely create a nested directory?](#)
- 1101 [How do I connect to a MySQL Database in Python?](#)
- 5448 [Does Python have a ternary conditional operator?](#)
- 3602 [Does Python have a string 'contains' substring method?](#)
- 1528 [Renaming columns in pandas](#)

Hot Network Questions

-  Is it right to extend flaps only in the white arc?
-  When is it acceptable to write a bad letter of recommendation?
-  How to say "cheat sheet" in French
-  Subverting the emotional woman and stoic man trope
-  Is this Portent-like spell balanced?
-  Worms crawling under skin
-  What is the difference between an astronaut in the ISS and a freediver in perfect neutral buoyancy?
-  Hilbert's hotel: why can't I repeat it infinitely many times?
-  Do we know the situation in Britain before Sealion (summer 1940)?
-  What's the lowest risk highest reward I can invest in for 5 years?
-  A high quality contribution but an annoying error is present in my published article
-  Would the same counter be used for a slice of pizza and a whole one?
-  Is it a good idea to leave minor world details to the reader's imagination?
-  How do you use the interjection for snorting?
-  Is there any iPhone SE out there with 3D Touch?
-  Could Apollo astronauts see city lights from the moon?
-  Examples of "unsuccessful" theories with afterlives
-  Norwegian refuses EU delay (4.7 hours) compensation because it turned out there was nothing wrong with the aircraft
-  Why are there two fundamental laws of logic?
-  Is this a Sherman, and if so what model?
-  Is it true that, "just ten trading days represent 63 per cent of the returns of the past 50 years"?
-  1, 2, 4, 8, 16, ... 33?
-  Hiking with a mule or two?
-  Functional analysis of the Pink Panther
-  Question feed

STACK OVERFLOW

PRODUCTS

COMPANY

STACK EXCHANGE
NETWORK

[Blog](#) [Facebook](#) [Twitter](#) [LinkedIn](#)

[Questions](#)

[Teams](#)

[About](#)

[Technology](#) >

[Jobs](#)

[Talent](#)

[Press](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



[Help](#)

[Mobile](#)

[Disable Responsiveness](#)

[Privacy Policy](#)

[Contact Us](#)

[Science](#) >

[Other](#) >

site design / logo © 2019 Stack Exchange Inc; user contributions licensed under cc by-sa 4.0 with attribution required.
rev 2019.9.20.34984

