

We do have connectors for every language to connect to databases

19.2.3 Connecting MySQL With Python

Let's use python to connect with this database.

- First we need to install "mysql-connector-python" package to establish a connection with MySQL.

```
pip install mysql-connector-python
```

```
(pyMySQL) D:\Project_PythonToMySQL>pip install mysql-connector-python
Collecting mysql-connector-python
```

Basically it is a library

We do search in google for the library

Python Software Foundation 20th Year Anniversary Fundraiser

Search projects

mysql-connector-python 8.0.25

pip install mysql-connector-python

Released: May 11, 2021

MySQL driver written in Python

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage

Project description

MySQL driver written in Python which does not depend on MySQL C client libraries and implements the DB API v2.0 specification (PEP-249).

```
In [1]: 1 !pip install mysql-connector-python

Requirement already satisfied: mysql-connector-python in c:\programdata\anaconda3\lib\site-packages (8.0.24)
Requirement already satisfied: protobuf>=3.0.0 in c:\programdata\anaconda3\lib\site-packages (from mysql-connector-python) (3.15.8)
Requirement already satisfied: six>=1.9 in c:\programdata\anaconda3\lib\site-packages (from protobuf>=3.0.0->mysql-connector-python) (1.15.0)
```

By using pip we can install the package for that.

```
Requirement already satisfied: mysql-connector-python in c:\programdata\anaconda3\lib\ (8.0.24)
Requirement already satisfied: protobuf>=3.0.0 in c:\programdata\anaconda3\lib\site-pa (3.15.8)
Requirement already satisfied: six>=1.9 in c:\programdata\anaconda3\lib\site-packages (3.0.0->mysql-connector-python) (1.15.0)
```

```
In [ ]: 1 import mysql.connector as connection
2
3 try:
4     mydb = connection.connect(host="localhost",user="root", passwd="mysql",use_pur
5     # check if the connection is established
6
7     query = "SHOW DATABASES"
8
9     cursor = mydb.cursor() #create a cursor to execute queries
10    cursor.execute(query)
```

1. First we need to import my sql connector helps to start communication between notebook and database.
- 2.

```
4 mydb = connection.connect(host="localhost",user="root", passwd="mysql",use_pure=True)
5 # check if the connection is established
6
7 query = "SHOW DATABASES"
8
9 cursor = mydb.cursor() #create a cursor to execute queries
10 cursor.execute(query)
11 print(cursor.fetchall())
12
13 except Exception as e:
14     mydb.close()
15     print(str(e))

[2]: 1 import mysql.connector as connection
2

[ ]: 1 |
```

- 3.
4. Connector alisa name connnect helps us to connect to it

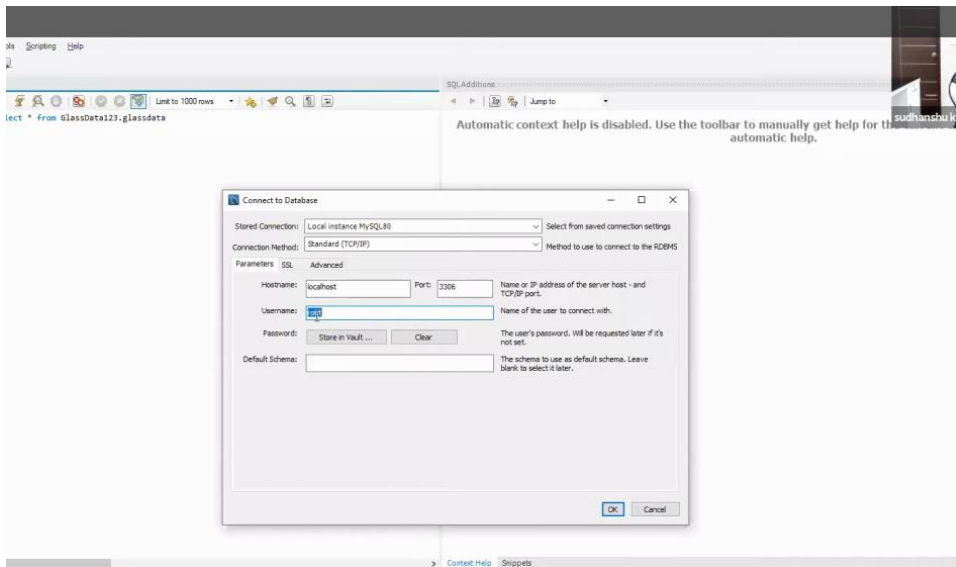
```
Requirement already satisfied: mysql-connector-python in c:\programdata\anaconda3\lib\site-packages (from my (8.0.24)
Requirement already satisfied: protobuf>=3.0.0 in c:\programdata\anaconda3\lib\site-packages (from mysql-connector-python) (3.15.8)
Requirement already satisfied: six>=1.9 in c:\programdata\anaconda3\lib\site-packages (from protobuf>=3.0.0->mysql-connector-python) (1.15.0)
```

```
1 import mysql.connector as connection
2
3 try:
4     mydb = connection.connect(host="localhost",user="root",passwd="mysql",use_pure=True)
5     # check if the connection is established
6
7     query = "SHOW DATABASES"
8
9     cursor = mydb.cursor() #create a cursor to execute queries
10    cursor.execute(query)
11    print(cursor.fetchall())
12
13 except Exception as e:
14     mydb.close()
15     print(str(e))
```

```
1 import mysql.connector as connection
2
```

5. connection.

Usually we get this host, user and all in from sql



We define those from here

If we have the cloud environment just in place of localhost we do have the url inplace of local host

```

sql-connector-python) (3.15.8)
Requirement already satisfied: six>=1.9 in c:\programdata\anaconda3\lib\site-packages (from protobuf>
=3.0.0->mysql-connector-python) (1.15.0)

In [ ]: 1 import mysql.connector as connection
        2
        3 try:
        4     mydb = connection.connect(host="localhost", user="root", passwd="mysql", use_pure=True)
        5     # check if the connection is established
        6
        7     query = "SHOW DATABASES"
        8
        9     cursor = mydb.cursor() #create a cursor to execute queries
        10    cursor.execute(query)
        11    print(cursor.fetchall())
        12
        13 except Exception as e:
        14    mydb.close()
        15    print(str(e))

```

```

5 mydb = connection.connect(host="localhost", database = 'GlassData', user="root", passwd="n
6 #check if the connection is established
7 print(mydb.is_connected())
8 query = "Select * from GlassData;"
9 cursor = mydb.cursor() #create a cursor to execute queries
10 cursor.execute(query)
11 for result in cursor.fetchall():
12     print(result)
13 mydb.close() #close the connection
14
15
16 except Exception as e:
17     #mydb.close()
18     print(str(e))

]: 1 cur = mydb.cursor()
   2 cursor.execute("select * from test")

```

Once we establish the connection we do create cursor to execute queries.

Once cursor execute the queries we need fetch the data for that we use fetchall()


```

7]: 1 import pandas as pd

i8]: 1 mydb
i8]: <mysql.connector.connection.MySQLConnection at 0x272908db6a0>

i9]: 1 pd.read_sql("select * from sudhanshu12345.test",mydb)
i9]:

```

	x1	x2	x3
0	4564	sudh	2021-05-15
1	4545	sudhdfgd	2021-05-15
2	4545	sudhdfgd	2021-05-15
3	4545	sudhdfgd	2021-05-15
4	4545	sudhdfgd	2021-05-15
5	4545	sudhdfgd	2021-05-15
6	4545	sudhdfgd	2021-05-15

All the connection things are same just passing query is different no need of fetch and everything as previous.

```

7]: 1 import pandas as pd

i8]: 1 mydb
i8]: <mysql.connector.connection.MySQLConnection at 0x272908db6a0>

i9]: 1 pd.read_sql("select * from sudhanshu12345.test",mydb)
i9]:

```

	x1	x2	x3
0	4564	sudh	2021-05-15
1	4545	sudhdfgd	2021-05-15
2	4545	sudhdfgd	2021-05-15
3	4545	sudhdfgd	2021-05-15
4	4545	sudhdfgd	2021-05-15
5	4545	sudhdfgd	2021-05-15
6	4545	sudhdfgd	2021-05-15

To read data and write to a csv file/table..

```

7]: 1 import pandas as pd

3]: 1 mydb
3]: <mysql.connector.connection.MySQLConnection at 0x272908db6a0>

5]: 1 a = pd.read_sql("select x3,x1 from test",mydb)

7]: 1 a.to_csv("mydata.csv")

]: 1 |

```

We can even create a table using cursur api


```

n [101]: 1 mydb = connection.connect(host="localhost", database='sudhanshu12345', user="root", passwd="mysql"
2 mydb

ut[101]: <mysql.connector.connection.MySQLConnection at 0x272934951c0>

n [102]: 1 cur = mydb.cursor()
2 cur.execute("create table mydata(n1 int(20) , n2 varchar(20) , n3 date)")

In [ ]: 1

```

Creating table using cursor and connection api

```

26 #read from the file
27 with open('glass.data', "r") as f:
28     next(f)
29     glass_data = csv.reader(f, delimiter="\n")
30     for line in enumerate(glass_data):
31         for list_ in (line[1]):
32             cursor.execute('INSERT INTO GlassData values ({values})'.format(values=list_))
33     print("Values inserted!!")
34 mydb.commit()
35 cursor.close()
36 mydb.close()
37
38 except Exception as e:
39     #mydb.close()
40     print(str(e))

In [120]: 1 import mysql.connector as connection
2 import pandas as pandas
3 import csv
4 mydb = connection.connect(host="localhost", user="root", passwd="mysql",use_pure=True)
5

In [123]: 1 cur = mydb.cursor()
2 cur.execute("CREATE TABLE IF NOT EXISTS sudhanshu12345.GlassData (Index_Number INT(10),RI float(10,5), Na float(10,5), Mg fl
3         " Si float(10,5), K float(10,5), Ca float(10,5), Ba float(10,5), Fe float(10,5), Class INT(5))")

In [ ]: 1

```

After that we can open the created table and list all the tables which are available

```

In [124]: 1 cur = mydb.cursor()
2 cur.execute("CREATE TABLE sudhanshu12345.GlassData1 (Index_Number INT(10),RI float(10,5), Na float(10,5), Mg float(10,5),Al
3         " Si float(10,5), K float(10,5), Ca float(10,5), Ba float(10,5), Fe float(10,5), Class INT(5))")

In [ ]: 1 with open('glass.data')

In [125]: 1 !ls
19.2.MySQL.ipynb
imgs
tmp files
mydata.csv
MySQL_application.docx
MySQL_application.pdf
untitled.ipynb

```

Creating table, opening it and iterating over index by index

```

1 import mysql.connector as connection
2 import pandas as pandas
3 import csv
4 mydb = connection.connect(host="localhost", user="root", passwd="mysql",use_pure=True)
5

1 cur = mydb.cursor()
2 cur.execute("CREATE TABLE sudhanshu12345.GlassData1 (Index_Number INT(10),RI float(10,5), Na float(10,5), Mg float(10,5),Al
3         " Si float(10,5), K float(10,5), Ca float(10,5), Ba float(10,5), Fe float(10,5), Class INT(5))")

1 with open('glass.data',"r") as data :
2     data_csv = csv.reader(data, delimiter= "\n")
3     print(data_csv)
4     for i in enumerate(data_csv):
5         print(i)

< csv.reader object at 0x0000027293E69DC0>
(0, ['index,RI,Na,Mg,Al,Si,K,Ca,Ba,Fe,Class'])
(1, ['1,1.52101,13.64,4.49,1.10,71.78,0.06,8.75,0.00,1'])
(2, ['2,1.51761,13.89,3.60,1.36,72.73,0.48,7.83,0.00,1'])

```

Taking data from one csv table and inserting into new created table using two for loops

Imp files
mydata.csv
MySQL application.docx
MySQL application.pdf
untitled.ipynb

```
In [139]: 1 with open('glass.data','r') as data :  
2     next(data)  
3     data_csv = csv.reader(data, delimiter= "\n")  
4     print(data_csv)  
5     for i in enumerate(data_csv):  
6         print(i)  
7         for j in i[1]:  
8             print(type(j))  
9             cur.execute('insert into sudhanshu12345.GlassData1 values ({data})'.format(data=j))  
10        print("all the data inserted")  
11    mydb.commit()
```

```
(46, ['47,1.51869,13.19,3.37,1.18,72.72,0.57,8.83,0.00,0.16,1'])  
<class 'str'>  
(47, ['48,1.52667,13.99,3.70,0.71,71.57,0.02,9.82,0.00,0.10,1'])  
<class 'str'>  
(48, ['49,1.52223,13.21,3.77,0.79,71.99,0.13,10.02,0.00,0.00,1'])  
<class 'str'>  
(49, ['50,1.51898,13.58,3.35,1.23,72.08,0.59,8.91,0.00,0.00,1'])  
<class 'str'>  
(50, ['51,1.52320,13.72,3.72,0.51,71.75,0.09,10.06,0.00,0.16,1'])  
<class 'str'>  
(51, ['52,1.51926,13.20,3.33,1.28,72.36,0.60,9.14,0.00,0.11,1'])  
<class 'str'>  
(52, ['53,1.51888,13.43,2.87,1.10,72.84,0.55,9.03,0.00,0.00,1'])
```

Taking data from one csv table and inserting into new created table using one for loops

```
In [146]: 1 with open('glass.data','r') as data :  
2     next(data)  
3     data_csv = csv.reader(data, delimiter= "\n")  
4     print(data_csv)  
5     for j in data_csv :  
6         print(type(str(j)))  
7         print(j)  
8         cur.execute('insert into sudhanshu12345.GlassData values ({data})'.format(data=str(j[0])))  
9         #print(type(str(j)))  
10        print("all the data inserted")  
11    mydb.commit()
```

```
<class 'str'>  
['8,1.51756,13.15,3.61,1.05,73.24,0.57,8.24,0.00,0.00,1']  
<class 'str'>  
['9,1.51918,14.04,3.58,1.37,72.08,0.56,8.30,0.00,0.00,1']  
<class 'str'>  
['10,1.51755,13.00,3.60,1.36,72.99,0.57,8.40,0.00,0.11,1']  
<class 'str'>  
['11,1.51571,12.72,3.46,1.56,73.20,0.67,8.09,0.00,0.24,1']  
<class 'str'>  
['12,1.51763,12.80,3.66,1.27,73.01,0.60,8.56,0.00,0.00,1']  
<class 'str'>  
['13,1.51509,12.88,3.43,1.40,73.28,0.69,8.05,0.00,0.24,1']  
<class 'str'>  
['14,1.51748,12.86,3.56,1.27,73.71,0.64,8.30,0.00,0.17,1']
```