

*Using MySql
connector in Python
Updated March
2020*

All rights reserved by Somnath PaulChoudhury

Version 1.2

Follow the steps

1. Start PC and type cmd in search, click upon it to bring the command prompt



2. Locate where pip is within Python sub directory it can be in

C:\Program Files\XYZ\Python37\Scripts

3. Go to the sub directory by giving DOS command

Dos Prompt>cd\program files\xyz\python37\scripts

4. Then type the following to install mysql connector

python -m pip install mysql-connector

```
python37\Scripts>python -m pip install mysql connector
```

How to fix the problems

First upgrade pip to latest version if required, you can do this every week

```
m Files\Python37\Scripts>python -m pip install --upgrade pip
g pip
ding https://files.pythonhosted.org/packages/54/0c/d01aa759fdc501a58f43
495f15b88da142ce14b5845662c13f3/pip-20.0.2-py2.py3-none-any.whl (1.4MB)
| 1.0MB 262kB/s eta 0:00:02
```

For applicable for all users give the command

```
Files\Python37\Scripts>python -m pip install --upgrade pip --user
pip
hed https://files.pythonhosted.org/packages/54/0c/d01aa759fdc501a58f4
95f15b88da142ce14b5845662c13f3/pip-20.0.2-py2.py3-none-any.whl
collected packages: pip
y installed pip-20.0.2
```

You should get this message

```
Requirement already up-to-date: pip in
n37\site-packages (20.0.2)
```

Finally run the command to install mysql connector as shown

```
rogram Files\Python37\Scripts>python -m pip install mysql-connector --user
ecting mysql-connector
nloading mysql-connector-2.2.9.tar.gz <11.9 MB>
! ██████████ | 2.8 MB 544 kB/s eta 0:00:17
```

If successful you should get this, pretty simple isn't?

```
Installing collected packages: mysql-connector
Running setup.py install for mysql-connector ... done
Successfully installed mysql-connector-2.2.9
```

Open mysql prompt and see the list of databases

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.64 sec)
```

Check the user and localhost also using this command

```
mysql> select USER();
+-----+
| USER() |
+-----+
| root@localhost |
+-----+
1 row in set (1.13 sec)
mysql>
```

Now lets see it from Python! Open a new file in Python (.py) and type as it is

```
import mysql.connector

spcdatabase = mysql.connector.connect ( # just giving a name to my connection
    host="localhost", # as you can see in select USER() command
    user="root",
    password=""
)

spccur=spcdatabase.cursor() # cursor() method is called by the connection again stored in spccur
spccur.execute("show databases")

for i in spccur:
    print(i)
```

For your convinence I'm attaching a .txt also of the code

```
import mysql.connector
```

```

spcdatabase = mysql.connector.connect ( # just giving a name to my connection
    host="localhost", # as you can see in select USER() command
    user="root",
    password=""
)

```

```

spccur=spcdatabase.cursor() # cursor() method is called by the connection again stored in spccur
spccur.execute("show databases")

```

```

for i in spccur:
    print(i)

```

When you run you see

```

('information_schema',)
('mysql',)
('test',)
>>>

```

Cross check it's same databases as you have seen.

Lets us try to create a database name **virus** and a table under the database as **coronavirus**. We will add the fields **province, country and transmission_type** in the table initially and add few records. After that we will alter the table and add another field which is primary key name is record_id type integer. For simplicity we have taken all other fields as varchar.

The following script will create a database and also will display the list of databases present.

```

import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password=""
)

spccur=spcdatabase.cursor()
spccur.execute("create database virus")

spccur.execute("show databases")

for i in spccur:
    print(i)

```

It will display the output as shown below, notice that database virus is created. [You can actually at this point check it in mysql also]

```
('information_schema',)  
('mysql',)  
(test',)  
(virus',)
```

Cross check in MySQL and its there

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| test |  
| virus |  
+-----+  
4 rows in set (0.00 sec)
```

Remember to run it only once otherwise it will display an error as the database is already created.

mysql.connector.errors.DatabaseError: 1007 (HY000): Can't create database 'virus'; database exists

Once the database is there we can have the table **coronavirus**

The following code will create a table **corona** under database **virus**

```
import mysql.connector  
  
spcdatabase=mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password=" ",  
    database="virus"  
)  
  
spccur=spcdatabase.cursor()  
spccur.execute("create table corona(province varchar(30), country varchar(30), transmission_type varchar(30))")  
  
spccur.execute("show tables") # our database is set to virus  
  
for i in spccur:  
    print(i)
```

The output is shown below

```
('corona',)
```

We can cross check in MySQL the see the table created

```
mysql> use virus;  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_virus |  
+-----+  
| corona |  
+-----+  
1 row in set (0.00 sec)
```

For your convenience I'm attaching a .txt also of the Python code

```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password=" ",
    database="virus"
)

spccur=spcdatabase.cursor()
spccur.execute("create table corona(province varchar(30), country varchar(30), transmission_type
varchar(30))")

spccur.execute("show tables") # our database is set to virus

for i in spccur:
    print(i)
```

Suppose we need to display the table structure we change the code a little

```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password=" ",
    database="virus"
)

spccur=spcdatabase.cursor()
spccur.execute("show columns from corona")

for i in spccur:
    print(i)
```

The output is shown as below

```
('province', 'varchar(30)', 'YES', '', None, '')
('country', 'varchar(30)', 'YES', '', None, '')
('transmission_type', 'varchar(30)', 'YES', '', None, '')
```

In case you want to see the same in MySQL

```
mysql> show columns from corona;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| province | varchar(30) | YES | | NULL | |
| country | varchar(30) | YES | | NULL | |
| transmission_type | varchar(30) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

If we now require to alter the table and add a primary key we modify the code as

```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password=" ",
    database="virus"
)

spccur=spcdatabase.cursor()
spccur.execute("alter table corona add column record_id int primary key first")

spccur.execute("show columns from corona")
for i in spccur:
    print(i)
```

The output is as below, a primary key is added as shown

```
('record_id', 'int(11)', 'NO', 'PRI', None, ")
('province', 'varchar(30)', 'YES', "", None, ")
('country', 'varchar(30)', 'YES', "", None, ")
('transmission_type', 'varchar(30)', 'YES', "", None, ")
```

Lets add some records now

```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password=" ",
    database="virus"
)

spccur=spcdatabase.cursor()
spccur.execute("insert into corona values(1,'Wuhan','China','Local')")

spccur.execute("select *from corona")
for i in spccur:
    print(i)
```

Output is

```
(1, 'Wuhan', 'China', 'Local')
```

But if we check in MySQL we don't find any records

```
mysql> select * from corona;
Empty set (0.00 sec)
```

because to write in the table we must add the statement `spcdatabase.commit()` so the code must be modified as

```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password=" ",
    database="virus"
)

spccur=spcdatabase.cursor()
spccur.execute("insert into corona values(1,'Wuhan','China','Local')")
spcdatabase.commit()
spccur.execute("select *from corona")
for i in spccur:
    print(i)
```

```
mysql> select * from corona;
+-----+-----+-----+-----+
| record_id | province | country | transmission_type |
+-----+-----+-----+-----+
|          1 | Wuhan    | China   | Local              |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Lets add few more records and display all records

```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password=" ",
    database="virus"
)

spccur=spcdatabase.cursor()
spccur.execute("select *from corona")
for i in spccur:
    print(i)
```

Output is as below

```
(1, 'Wuhan', 'China', 'Local')
(2, 'Huanan', 'China', 'Local')
(3, 'Patong', 'Thiland', 'Imported')
(4, 'Washington DC', 'USA', 'Imported')
(5, 'Milan', 'Italy', 'Imported')
(6, 'Mulhouse', 'France', 'Imported')
(7, 'Mitterteich', 'Germany', 'Imported')
(8, 'Hurghada', 'Iran', 'Imported')
```


Lets do some query what about displaying only records where local transmission has happened

```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password=" ",
    database="virus"
)

spccur=spcdatabase.cursor()
spccur.execute("select * from corona where transmission_type='Local' ")
for i in spccur:
    print(i)
```

Output is

```
(1, 'Wuhan', 'China', 'Local')
(2, 'Huanan', 'China', 'Local')
```

We can infact do complex queries also

```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password=" ",
    database="virus"
)

spccur=spcdatabase.cursor()
spccur.execute("select * from corona where province like 'M%' or country like 'T%'")
for i in spccur:
    print(i)
```

Output

```
(3, 'Patong', 'Thiland', 'Imported')
(5, 'Milan', 'Italy', 'Imported')
(6, 'Mulhouse', 'France', 'Imported')
(7, 'Mitterteich', 'Germany', 'Imported')
```

No part of this material should be used by any one. Only for private circulation amongst limited individuals. Please inform the author about unauthorize use. Tweet @somnathpc