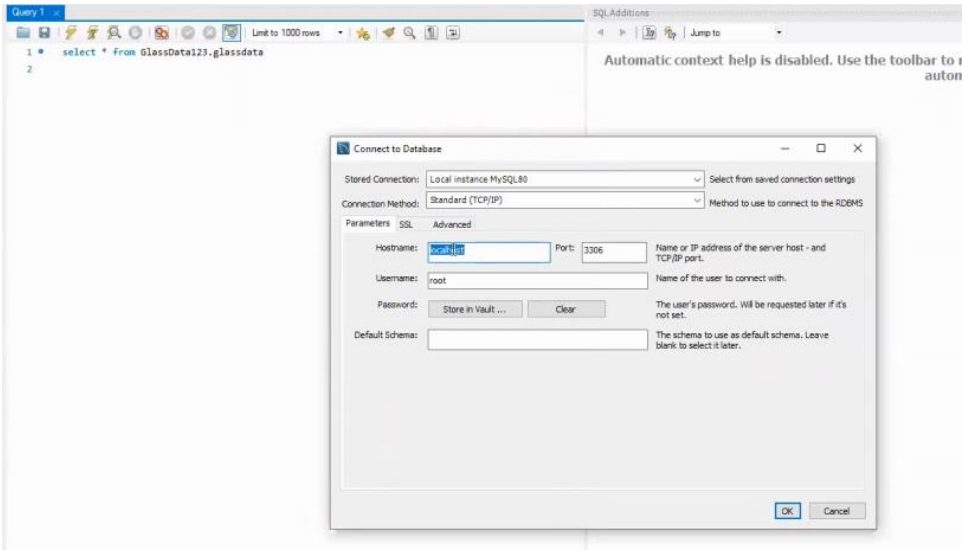
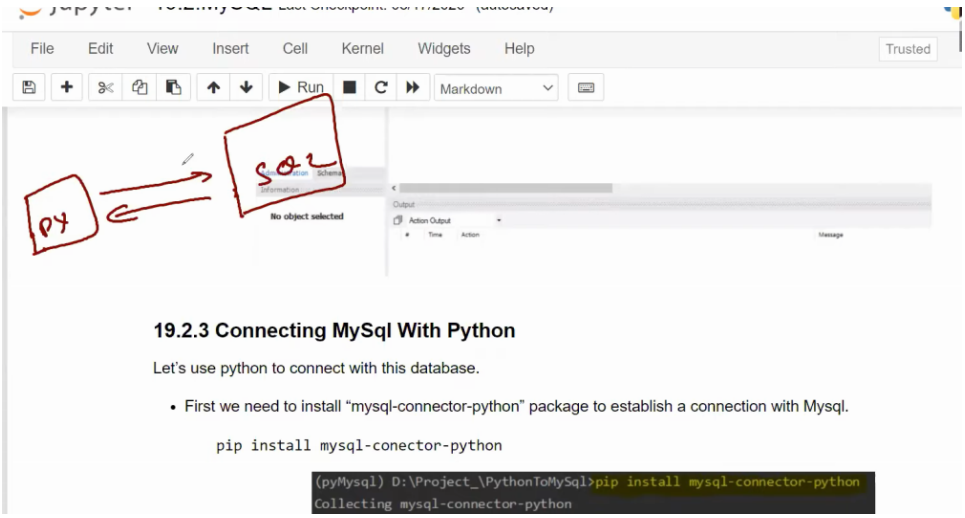
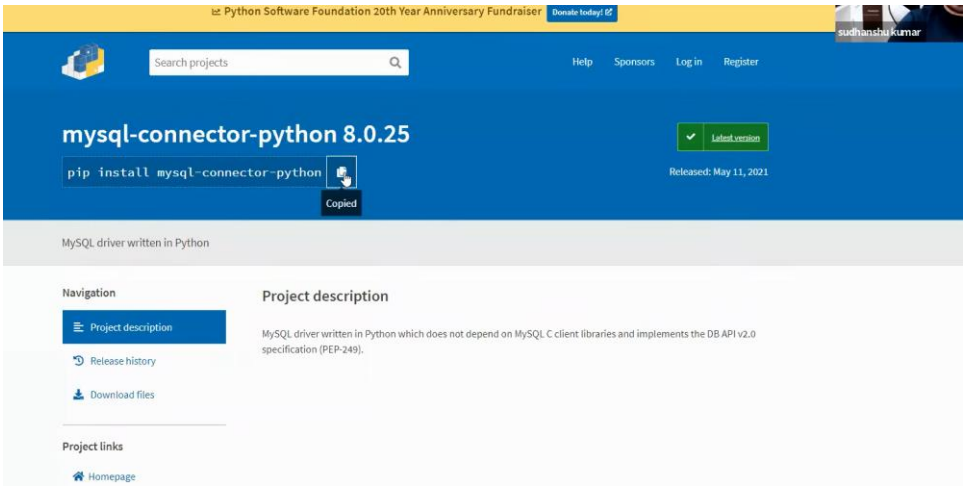


We do have connectors for every language to connect to databases



Basically it is a library

We do search in google for the ilbrary



```
n [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 my
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)
```

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
sql-connector-python) (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 [ ]:
import mysql.connector as connection

try:
    mydb = connection.connect(host="localhost",user="root", passwd="mysql",use_pur
    # check if the connection is established

    query = "SHOW DATABASES"

    cursor = mydb.cursor() #create a cursor to execute queries
    cursor.execute(query)
```

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

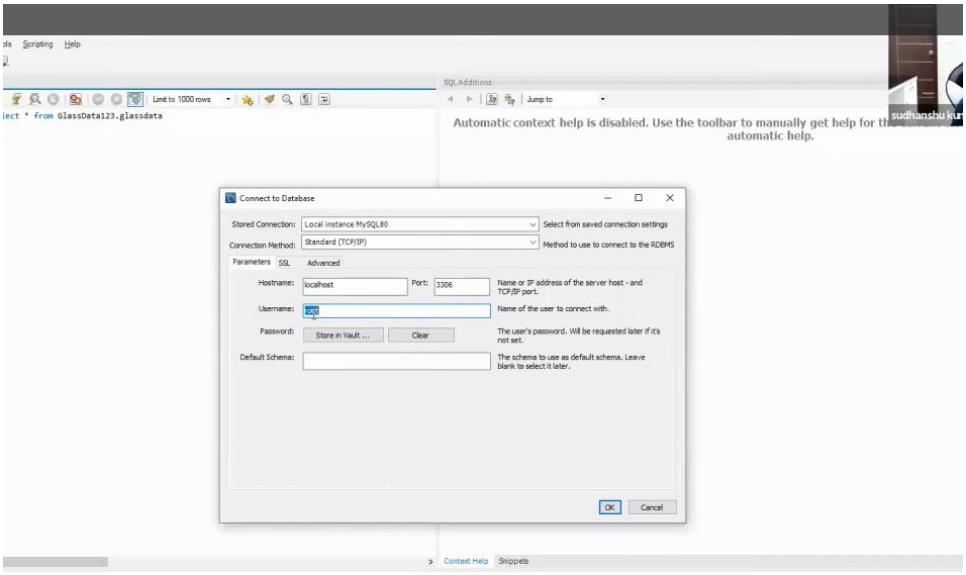
```
Run  Code
    mydb = connection.connect(host="localhost",user="root", passwd="mysql",use_pure=True)
    # check if the connection is established

    query = "SHOW DATABASES"

    cursor = mydb.cursor() #create a cursor to execute queries
    cursor.execute(query)
    print(cursor.fetchall())

except Exception as e:
    mydb.close()
    print(str(e))

[2]: import mysql.connector as connection

[ ]:
```

3.
4. Connector alisa name.comnnect helps us to connect to it

```
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)

import mysql.connector as connection

try:
    mydb = connection.connect(host="localhost",user="root",Ipasswd="mysql",use_pure=True)
    # check if the connection is established

    query = "SHOW DATABASES"

    cursor = mydb.cursor() #create a cursor to execute queries
    cursor.execute(query)
    print(cursor.fetchall())

except Exception as e:
    mydb.close()
    print(str(e))

import mysql.connector as connection

connection.
```

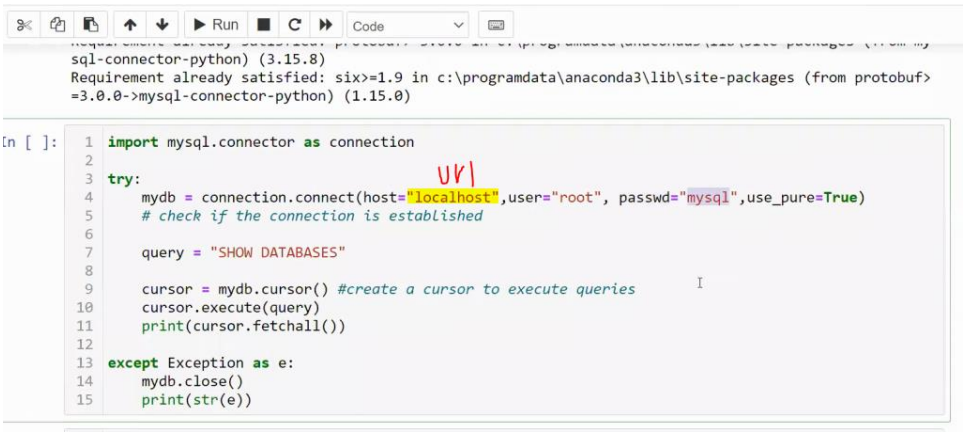
5.

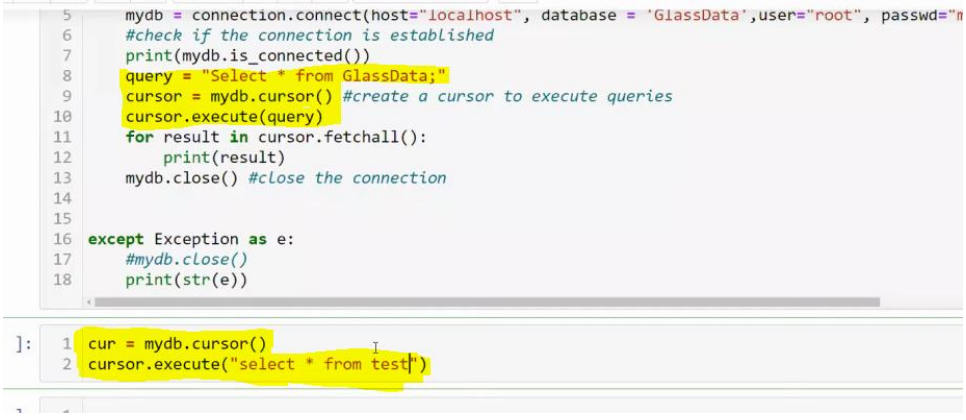
Usually we get this host, user and all in from sql



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





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()

```
try:
    mydb = connection.connect(host="localhost", database = 'GlassData',user="root", passwd="mysql"
    #check if the connection is established
    print(mydb.is_connected())
    query = "Select * from GlassData;"
    cursor = mydb.cursor() #create a cursor to execute queries
    cursor.execute(query)
    for result in cursor.fetchall():
        print(result)
    mydb.close() #close the connection


except Exception as e:
    #mydb.close()
    print(str(e))
```

```
cur1 = mydb.cursor()
out = cur1.execute("select * from test")
```

```
cur1.execute("select * from test")
```

```
cur1.fetchall()
```

```
[(4564, 'sudh', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
 (4545, 'sudhdfgd', datetime.date(2021, 5, 15)),
```

Till now selection and fetching of dataset is done

==

We can also write queries over pandas

```
import pandas as pd
```

```
mydb
```

```
<mysql.connector.connection.MySQLConnection at 0x272908db6a0>
```

```
pd.read_sql("select * from sudhanshu12345.test",mydb)
```

	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.

```
import pandas as pd
```

```
mydb
```

```
<mysql.connector.connection.MySQLConnection at 0x272908db6a0>
```

```
pd.read_sql("select * from sudhanshu12345.test",mydb)
```

	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..

```
import pandas as pd
```

```
mydb
```

```
<mysql.connector.connection.MySQLConnection at 0x272908db6a0>
```

```
a = pd.read_sql("select x3,x1 from test",mydb)
```

```
a.to_csv("mydata.csv")
```

We can even create a table using cursur api

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

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

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

In [ ]:
```

Creating table using cursor and connection api

```
    #read from the file
    with open('glass.data', "r") as f:
        next(f)
        glass_data = csv.reader(f, delimiter="\n")
        for line in enumerate(glass_data):
            for list_ in (line[1]):
                cursor.execute('INSERT INTO GlassData values ({values})'.format(values=(list_)))
    print("Values inserted!!")
    mydb.commit()
    cursor.close()
    mydb.close()

except Exception as e:
    #mydb.close()
    print(str(e))
```

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

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

```
In [ ]:
```

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

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

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

In [125]:
!ls

19.2.MySQL.ipynb
imgs
Imp files
mydata.csv
MYSQL_application.docx
MYSQL application.pdf
Untitled.ipynb
```

Creating table, opening it and iterating over index by index

```
import mysql.connector as connection
import pandas as pandas
import csv
mydb = connection.connect(host="localhost", user="root", passwd="mysql",use_pure=True)
```

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

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

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

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

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