Interface of Python with an SQL database

Syllabus

Interface of python with an SQL database:

- connecting SQL with Python,
- performing insert, update, delete queries using cursor,
- display data by using fetchone(), fetchall(), rowcount,
- creating database connectivity applications

Need of Python-MySQL connectivity

- In general, during the execution of a program, data is inputted by the user and output is displayed accordingly.

- But this input and output data is not stored anywhere because all program execution takes place inside the RAM which is a temporary memory and as soon as we close the program/IDE, its contents get erased.

- Thus, when next time program is executed again, it requires a new set of inputs from the user.

Need of Python-MySQL connectivity

- This limitation can be overcome by
  - fetching the input from the user (through a Python program) in a database, and
  - sending the output in a database which is not directly accessed by the user.

Installing MySQL connector

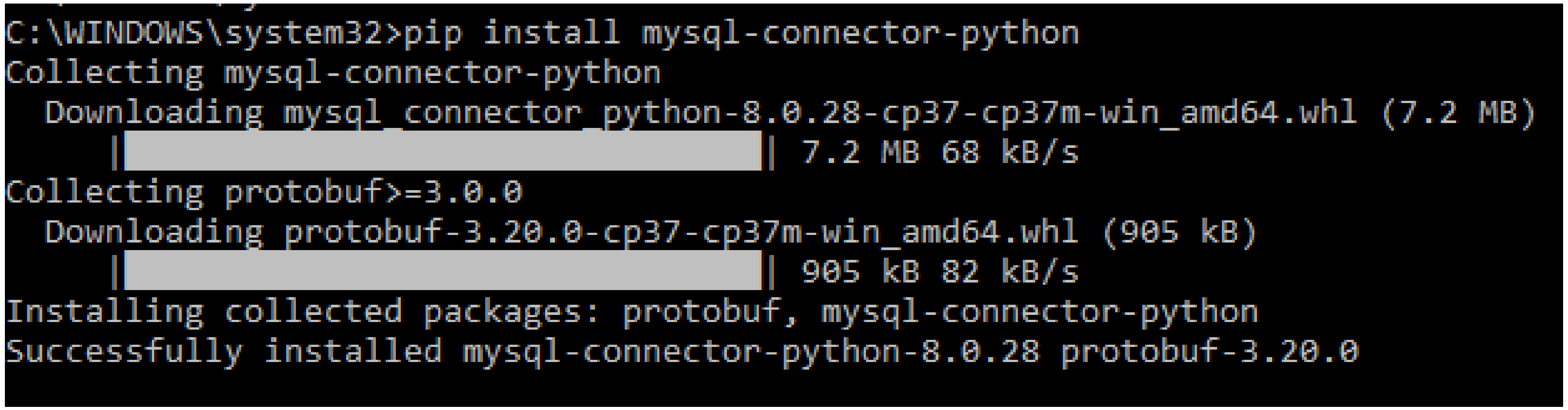
- To establish connectivity between Python and MySQL, we require Python DB-API which is a set of tools used by an Application Program to communicate with the Operating System or other programs such as DBMS.

- This API includes the following:
  - Importing the API module
  - Acquiring a connection with the database
  - Issuing SQL statements and stored procedures
  - closing the connection

Installing MySQL connector

- In order to install MySQL connector, we can use the following command in CMD (run it as an admin):

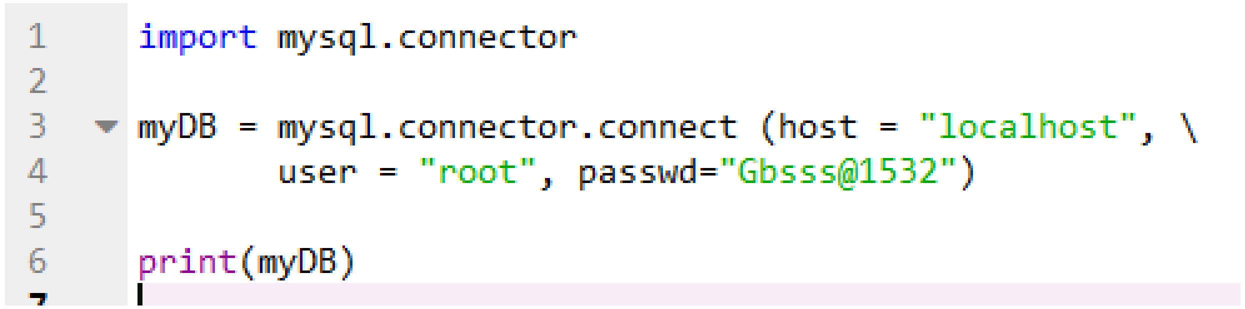
```
pip install mysql-connector-python
```



```
C:\WINDOWS\system32>pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-8.0.28-cp37-cp37m-win_amd64.whl (7.2 MB)
     |████████████████████████████████| 7.2 MB 68 kB/s
Collecting protobuf>=3.0.0
  Downloading protobuf-3.20.0-cp37-cp37m-win_amd64.whl (905 kB)
     |████████████████████████████████| 905 kB 82 kB/s
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.0.28 protobuf-3.20.0
```

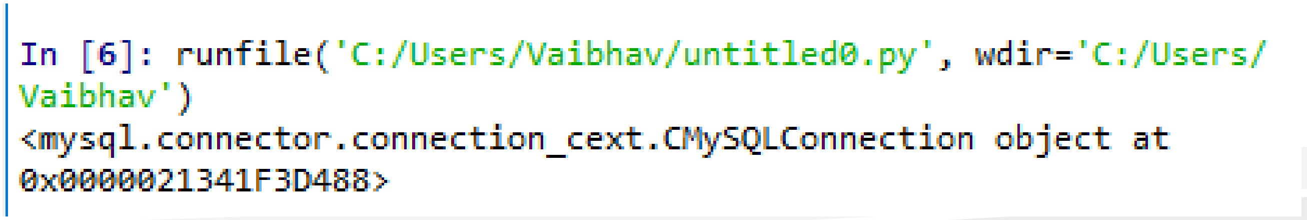
connecting Python and MySQL

- Once we install MySQL connector, let's establish the connection between Python and MySQL from Python IDE:



```python
import mysql.connector

myDB = mysql.connector.connect (host = "localhost", \
        user = "root", passwd="Gbsss@1532")

print(myDB)
```

- if the above statements are executed successfully, then we will received this kind of output..



```
In [6]: runfile('C:/Users/Vaibhav/untitled0.py', wdir='C:/Users/
Vaibhav')
<mysql.connector.connection_cext.CMySQLConnection object at
0x0000021341F3D488>
```

Creating Cursor Object

- In order to execute SQL statements from Python IDE, we need to create a cursor object which will allows Python code to execute database command in a database session.

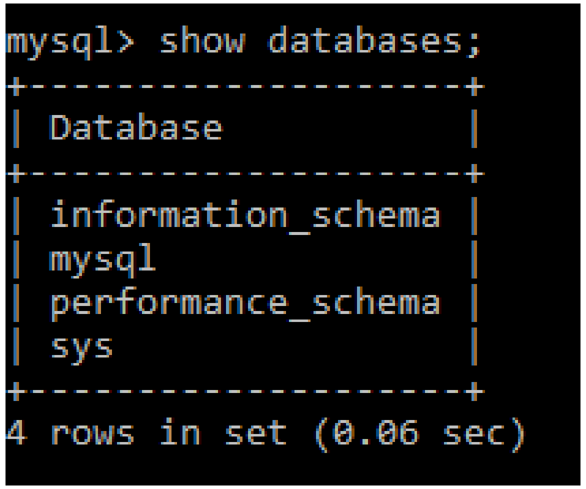
- cursor object is created using cursor method by the connection object returned by connect() method,

```
myCursor = myDB.cursor()
```

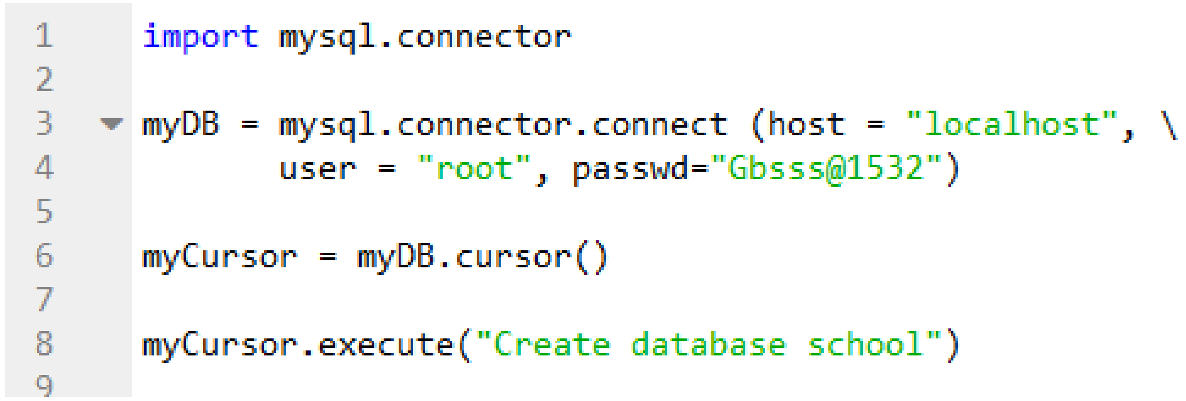
- Once a cursor object is created, we can use execute method to execute SQL queries from Python.

Program-1: Creating a database

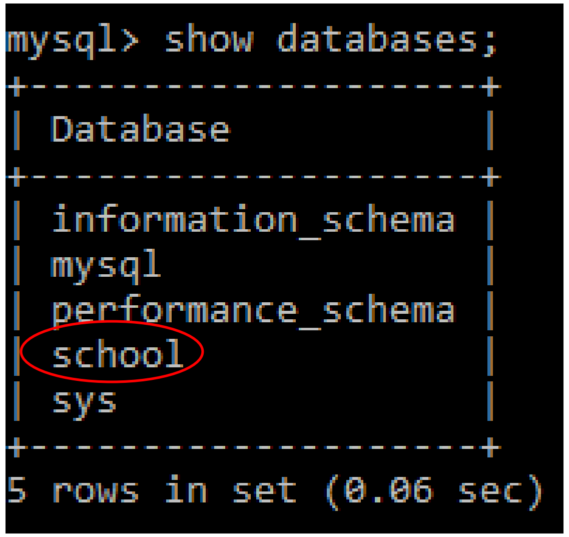
- As mentioned earlier, database queries can be executed in Python using execute() method



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.06 sec)
```

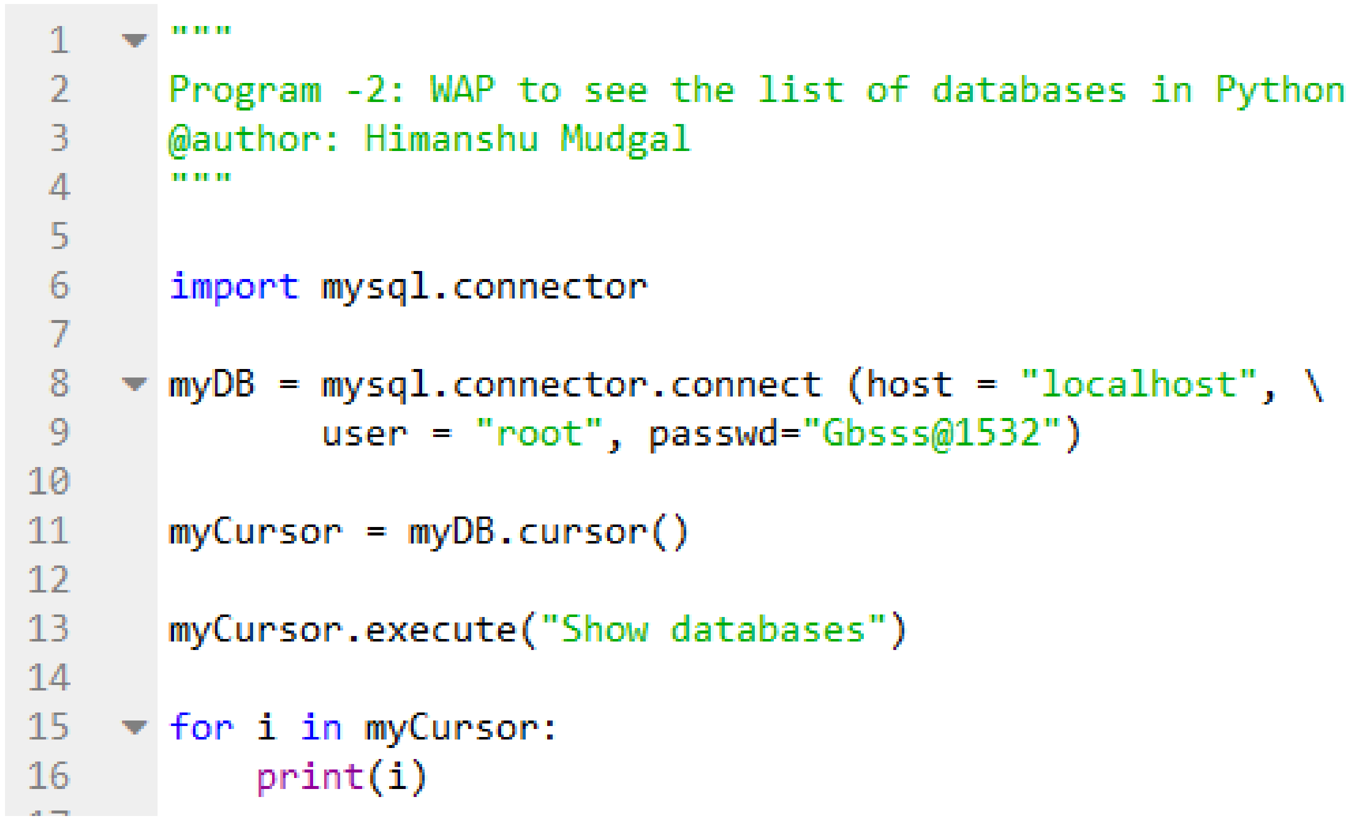


```
import mysql.connector

myDB = mysql.connector.connect (host = "localhost", \
        user = "root", passwd="Gbsss@1532")

myCursor = myDB.cursor()

myCursor.execute("Create database school")

```

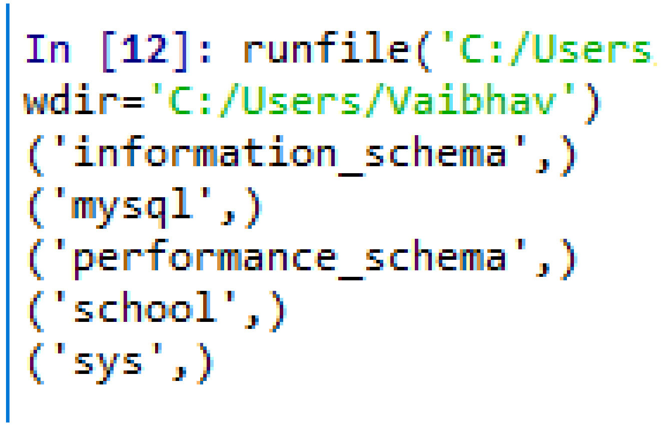


```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| school             |
| sys                |
+--------------------+
5 rows in set (0.06 sec)
```

Program-2: Show databases

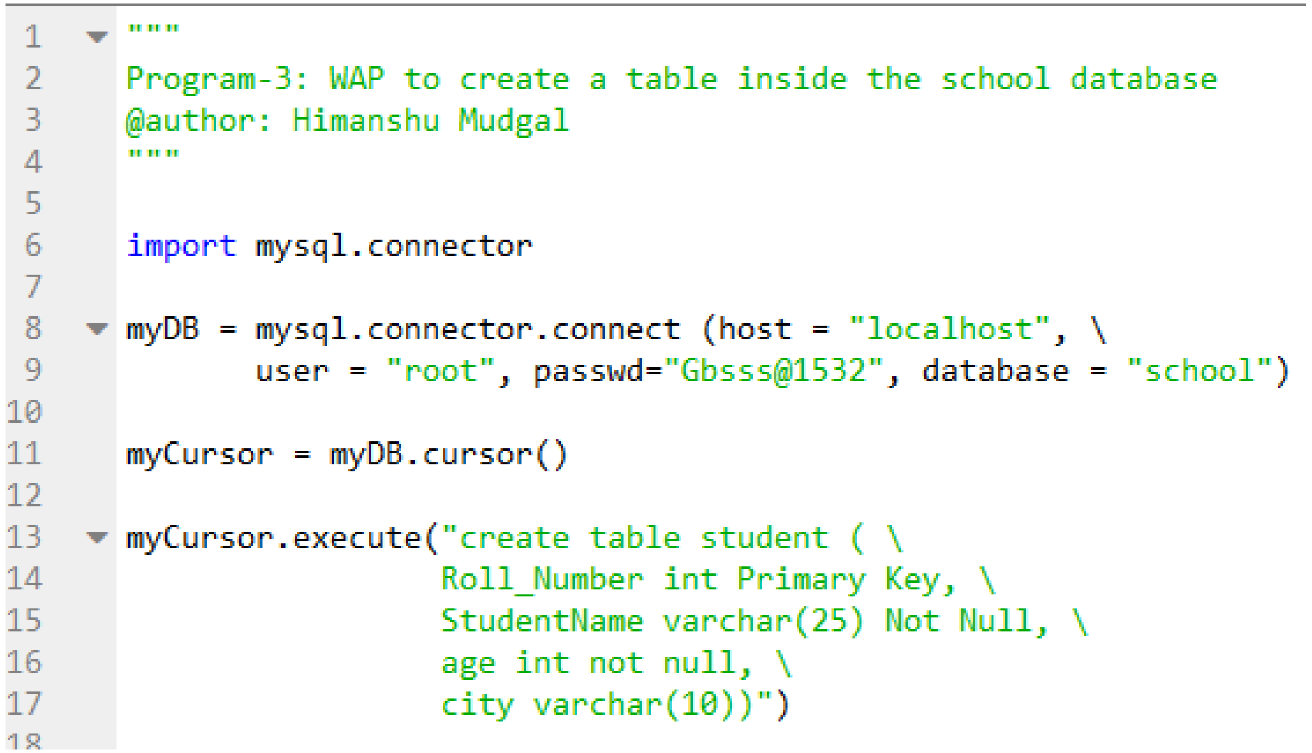


```python
"""
Program -2: WAP to see the list of databases in Python
@author: Himanshu Mudgal
"""

import mysql.connector

myDB = mysql.connector.connect (host = "localhost", \
        user = "root", passwd="Gbsss@1532")

myCursor = myDB.cursor()

myCursor.execute("Show databases")

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

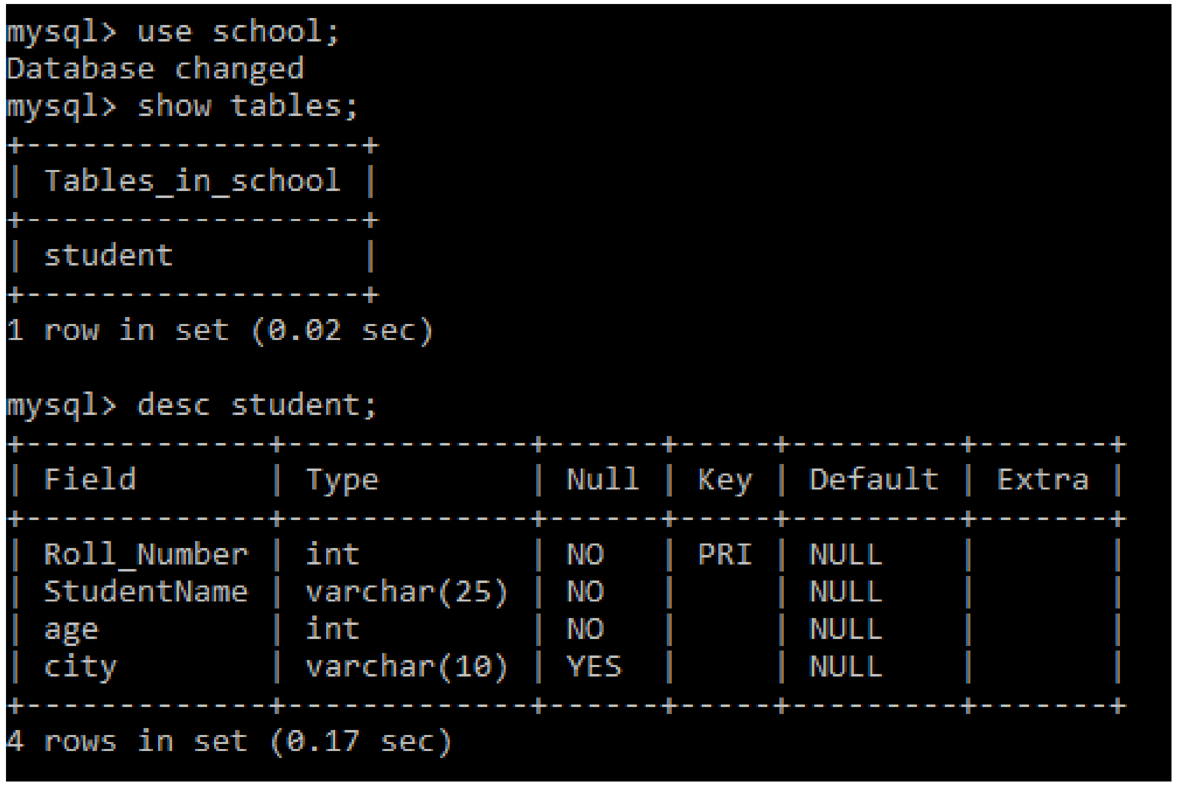


```
In [12]: runfile('C:/Users
wdir='C:/Users/Vaibhav')
('information_schema',)
('mysql',)
('performance_schema',)
('school',)
('sys',)
```

Program-3: create a table inside database

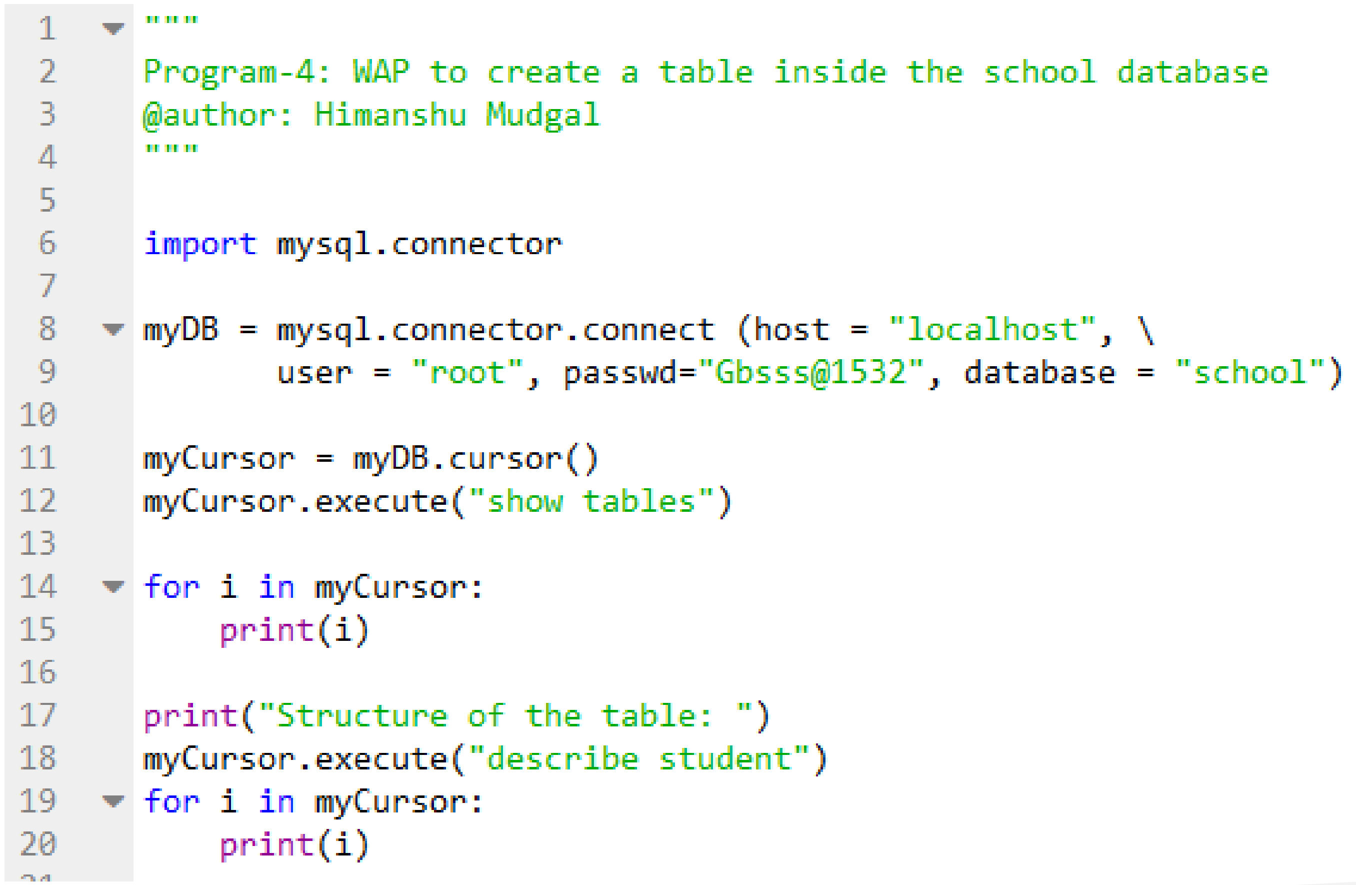


```
"""
Program-3: WAP to create a table inside the school database
@author: Himanshu Mudgal
"""

import mysql.connector

myDB = mysql.connector.connect (host = "localhost", \
          user = "root", passwd="Gbsss@1532", database = "school")

myCursor = myDB.cursor()

myCursor.execute("create table student ( \
                    Roll_Number int Primary Key, \
                    StudentName varchar(25) Not Null, \
                    age int not null, \
                    city varchar(10))")
```

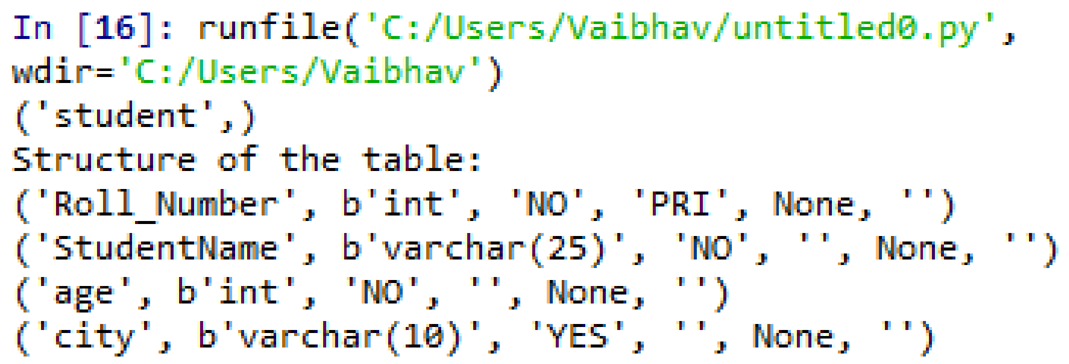


```
mysql> use school;
Database changed
mysql> show tables;
+-------------------+
| Tables_in_school  |
+-------------------+
| student           |
+-------------------+
1 row in set (0.02 sec)

mysql> desc student;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Roll_Number | int         | NO   | PRI | NULL    |       |
| StudentName | varchar(25) | NO   |     | NULL    |       |
| age         | int         | NO   |     | NULL    |       |
| city        | varchar(10) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.17 sec)
```

Program-4: show tables in a database

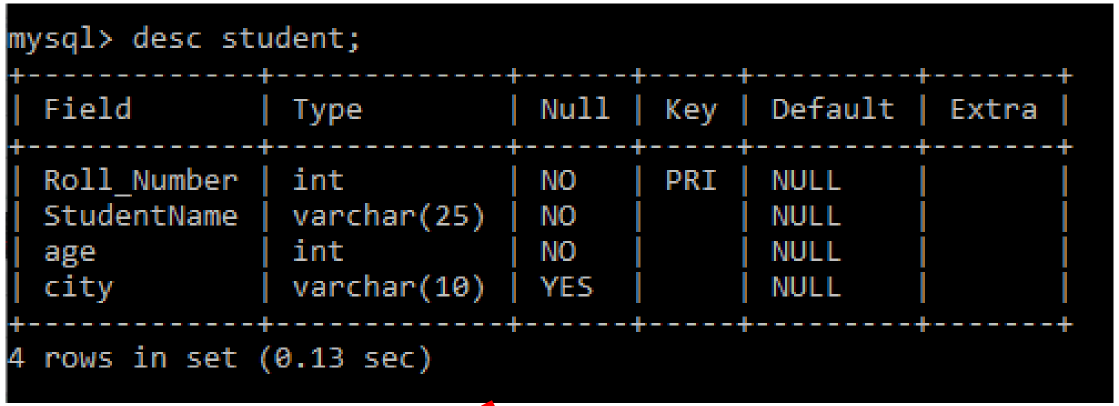


```python
"""
Program-4: WAP to create a table inside the school database
@author: Himanshu Mudgal
"""

import mysql.connector

myDB = mysql.connector.connect (host = "localhost", \
        user = "root", passwd="Gbsss@1532", database = "school")

myCursor = myDB.cursor()
myCursor.execute("show tables")

for i in myCursor:
    print(i)

print("Structure of the table: ")
myCursor.execute("describe student")
for i in myCursor:
    print(i)
```

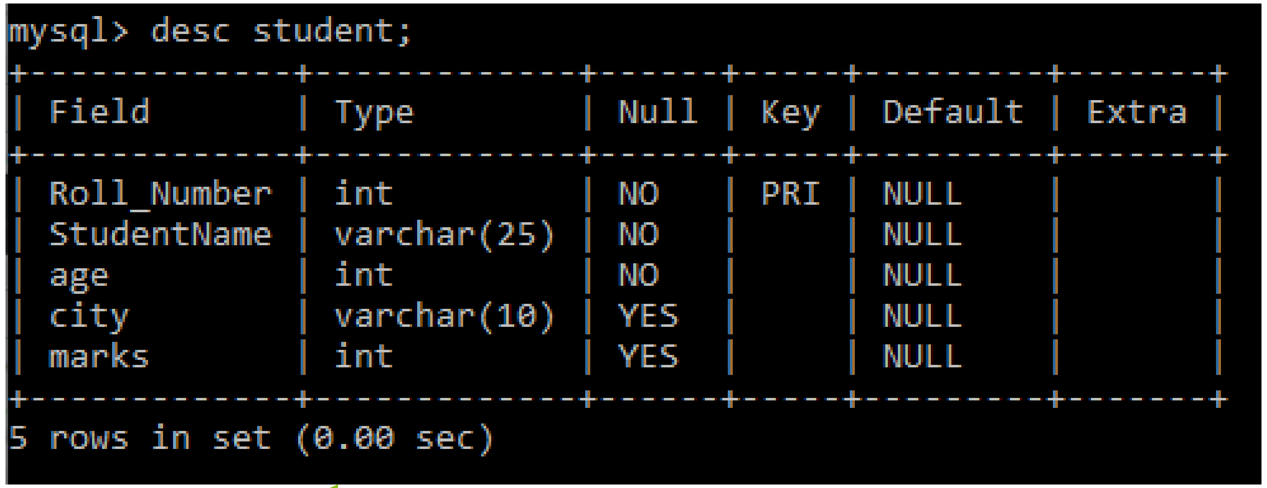


```
In [16]: runfile('C:/Users/Vaibhav/untitled0.py',
wdir='C:/Users/Vaibhav')
('student',)
Structure of the table:
('Roll_Number', b'int', 'NO', 'PRI', None, '')
('StudentName', b'varchar(25)', 'NO', '', None, '')
('age', b'int', 'NO', '', None, '')
('city', b'varchar(10)', 'YES', '', None, '')
```

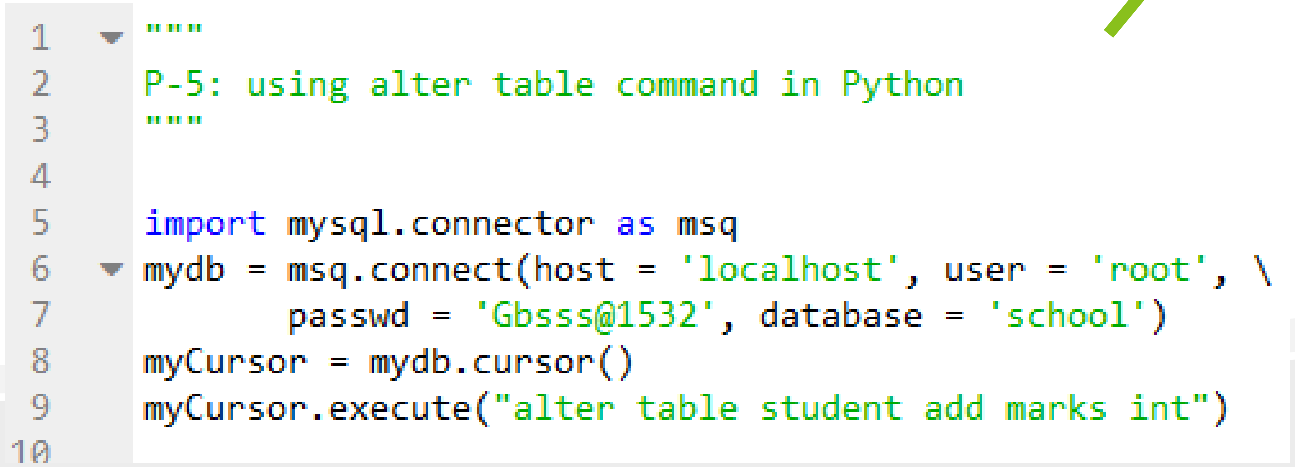
Program-5: using alter table command



```
mysql> desc student;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Roll_Number | int         | NO   | PRI | NULL    |       |
| StudentName | varchar(25) | NO   |     | NULL    |       |
| age         | int         | NO   |     | NULL    |       |
| city        | varchar(10) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.13 sec)
```

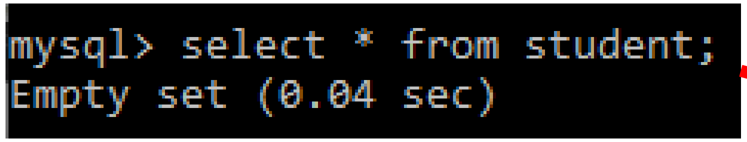


```
mysql> desc student;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Roll_Number | int         | NO   | PRI | NULL    |       |
| StudentName | varchar(25) | NO   |     | NULL    |       |
| age         | int         | NO   |     | NULL    |       |
| city        | varchar(10) | YES  |     | NULL    |       |
| marks       | int         | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

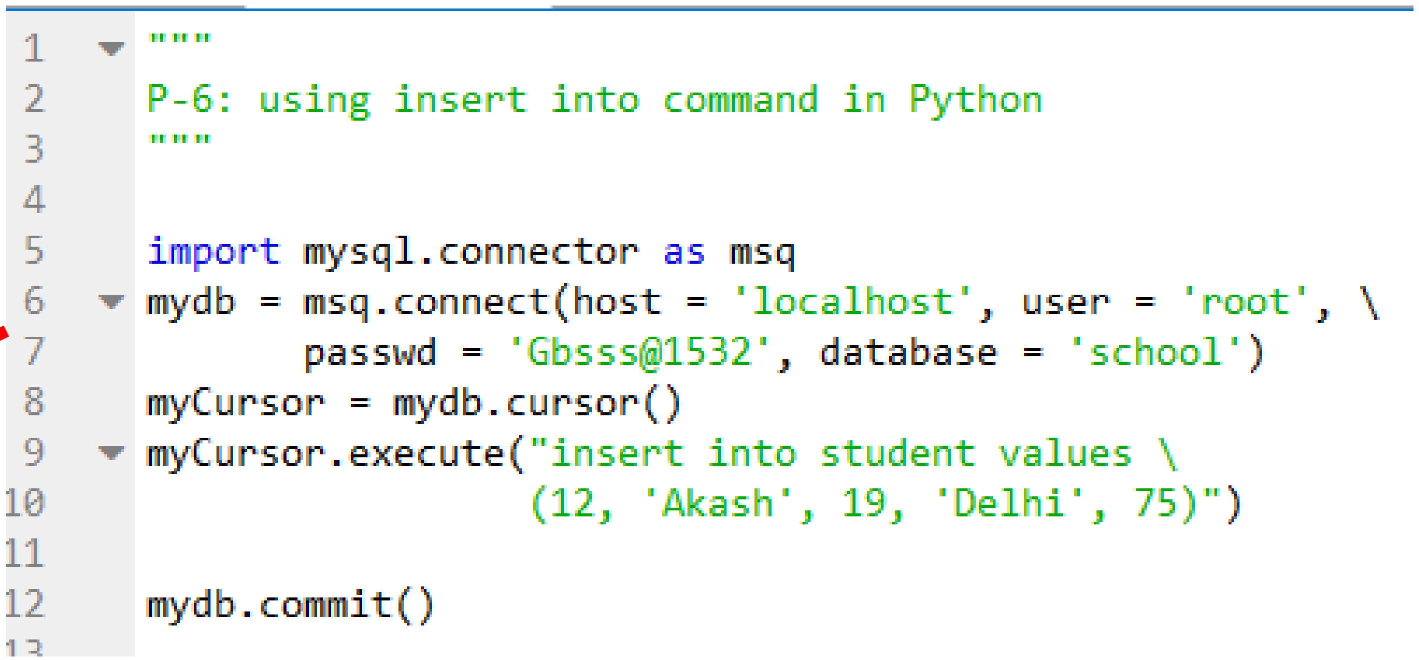


```python
"""
P-5: using alter table command in Python
"""

import mysql.connector as msq
mydb = msq.connect(host = 'localhost', user = 'root', \
        passwd = 'Gbsss@1532', database = 'school')
myCursor = mydb.cursor()
myCursor.execute("alter table student add marks int")
```

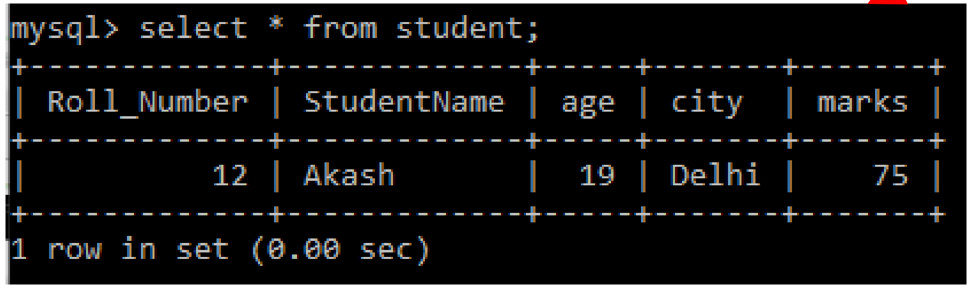
Program-6: Inserting data in table



```
mysql> select * from student;
Empty set (0.04 sec)
```

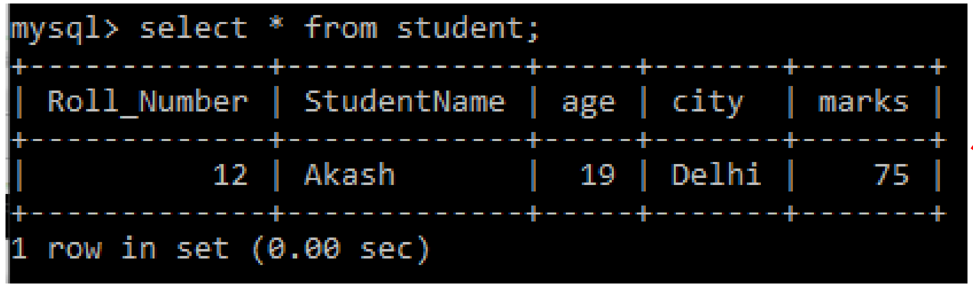


```python
"""
P-6: using insert into command in Python
"""

import mysql.connector as msq
mydb = msq.connect(host = 'localhost', user = 'root', \
        passwd = 'Gbsss@1532', database = 'school')
myCursor = mydb.cursor()
myCursor.execute("insert into student values \
                (12, 'Akash', 19, 'Delhi', 75)")

mydb.commit()
```

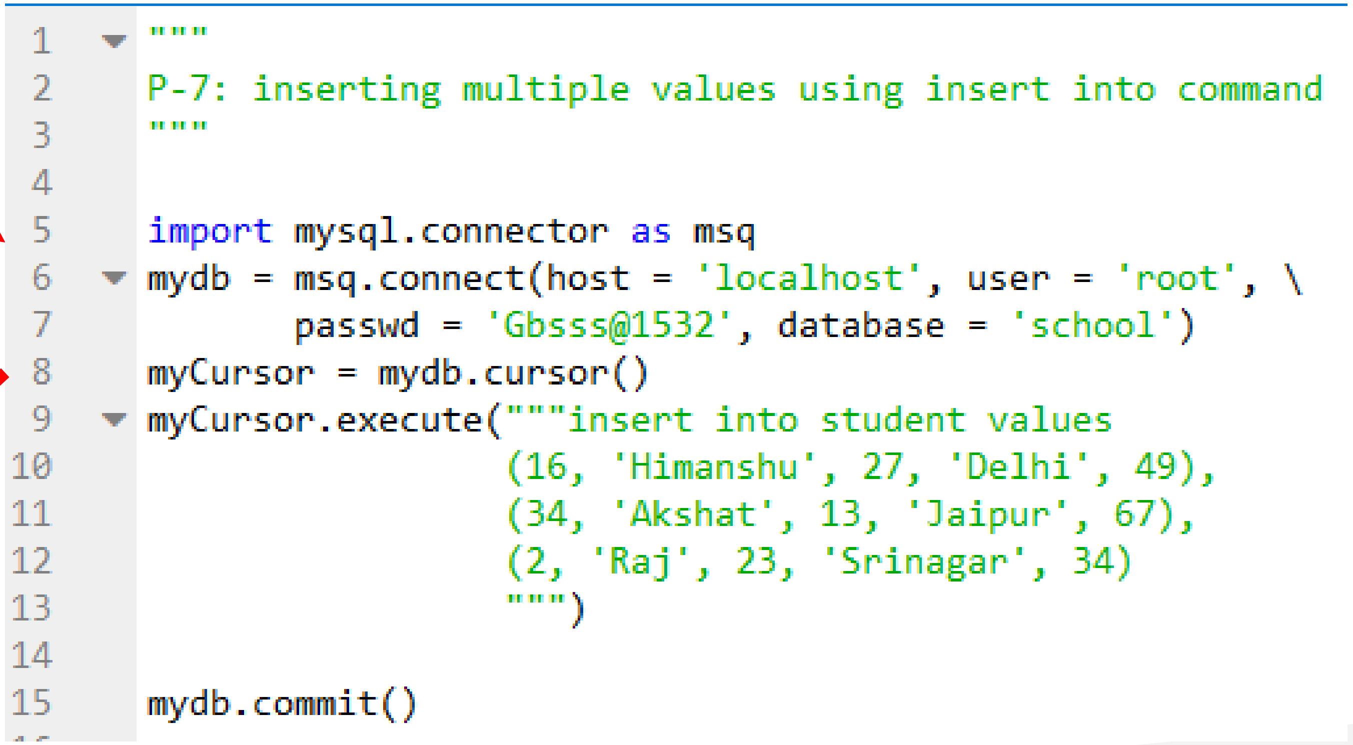


```
mysql> select * from student;
+-------------+-------------+-----+-------+-------+
| Roll_Number | StudentName | age | city  | marks |
+-------------+-------------+-----+-------+-------+
|          12 | Akash       |  19 | Delhi |    75 |
+-------------+-------------+-----+-------+-------+
1 row in set (0.00 sec)
```

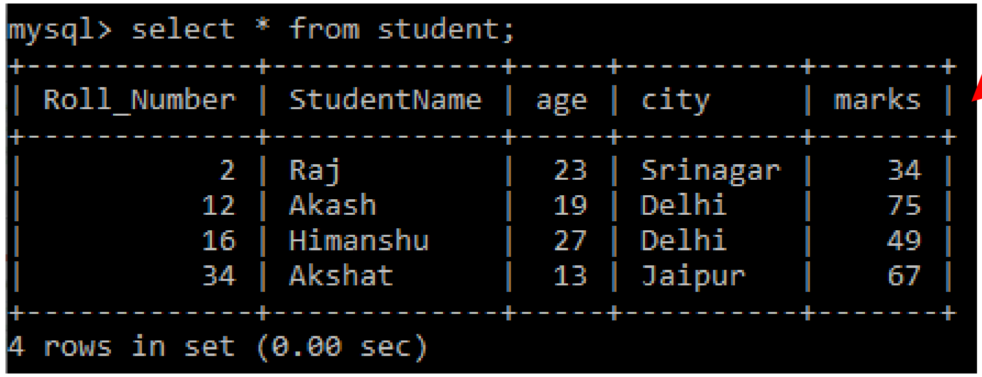
Program-7: inserting multiple values



```
mysql> select * from student;
+-------------+-------------+-----+-------+-------+
| Roll_Number | StudentName | age | city  | marks |
+-------------+-------------+-----+-------+-------+
|          12 | Akash       |  19 | Delhi |    75 |
+-------------+-------------+-----+-------+-------+
1 row in set (0.00 sec)
```

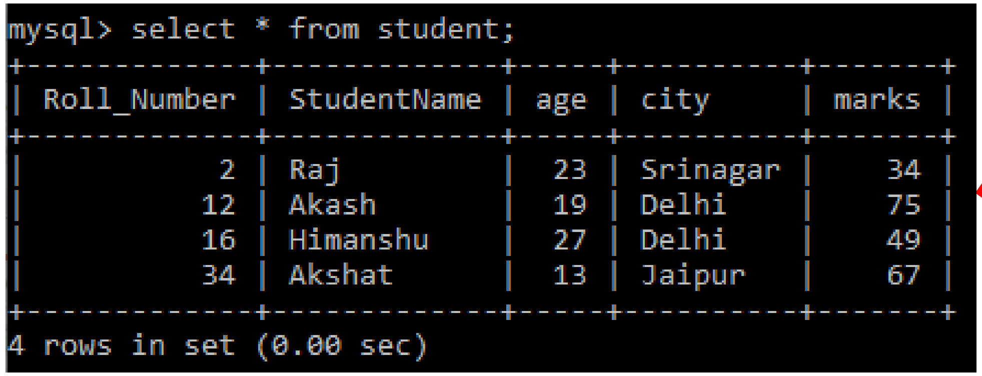


```python
"""
P-7: inserting multiple values using insert into command
"""

import mysql.connector as msq
mydb = msq.connect(host = 'localhost', user = 'root', \
        passwd = 'Gbsss@1532', database = 'school')
myCursor = mydb.cursor()
myCursor.execute("""insert into student values
                 (16, 'Himanshu', 27, 'Delhi', 49),
                 (34, 'Akshat', 13, 'Jaipur', 67),
                 (2, 'Raj', 23, 'Srinagar', 34)
                 """)

mydb.commit()
```

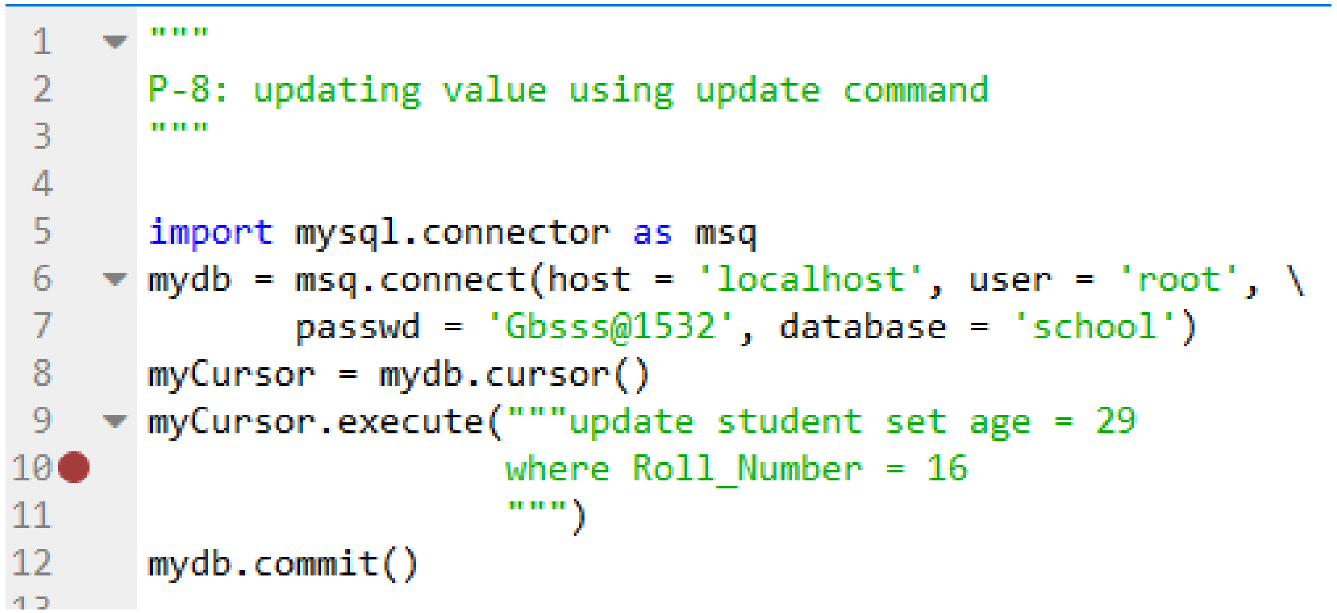


```
mysql> select * from student;
+-------------+-------------+-----+----------+-------+
| Roll_Number | StudentName | age | city     | marks |
+-------------+-------------+-----+----------+-------+
|           2 | Raj         |  23 | Srinagar |    34 |
|          12 | Akash       |  19 | Delhi    |    75 |
|          16 | Himanshu    |  27 | Delhi    |    49 |
|          34 | Akshat      |  13 | Jaipur   |    67 |
+-------------+-------------+-----+----------+-------+
4 rows in set (0.00 sec)
```

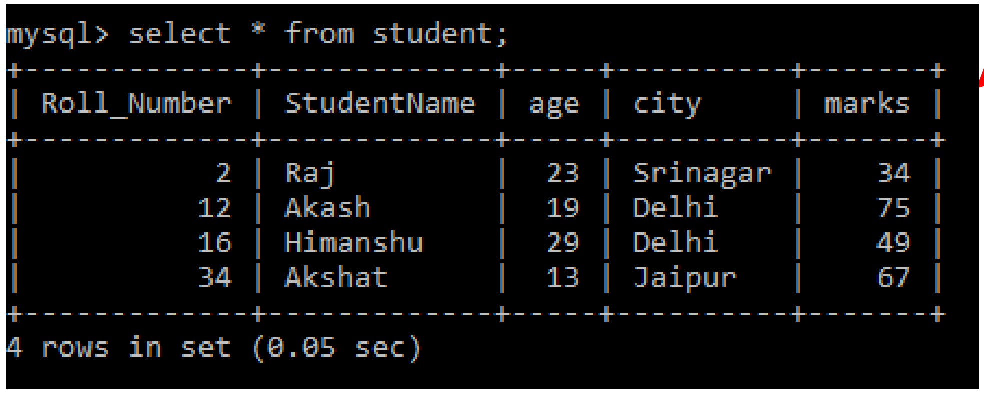
Program-8: updating values



```
mysql> select * from student;
+-------------+-------------+-----+-----------+-------+
| Roll_Number | StudentName | age | city      | marks |
+-------------+-------------+-----+-----------+-------+
|           2 | Raj         |  23 | Srinagar  |    34 |
|          12 | Akash       |  19 | Delhi     |    75 |
|          16 | Himanshu    |  27 | Delhi     |    49 |
|          34 | Akshat      |  13 | Jaipur    |    67 |
+-------------+-------------+-----+-----------+-------+
4 rows in set (0.00 sec)
```

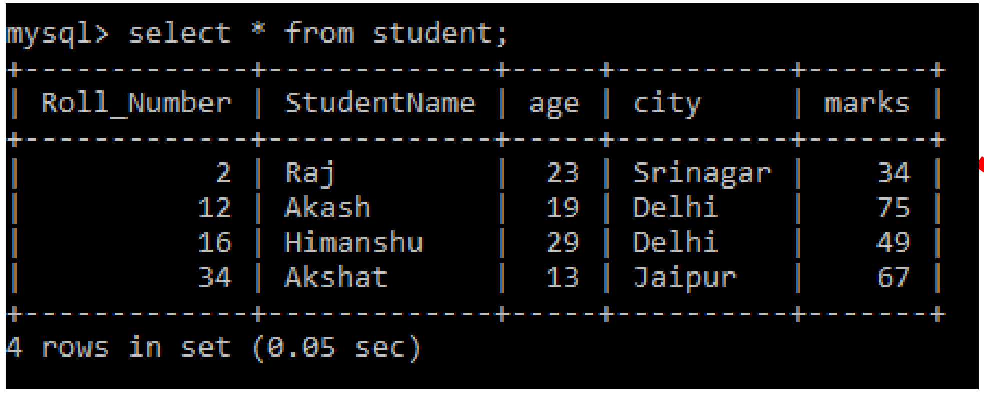


```python
"""
P-8: updating value using update command
"""

import mysql.connector as msq
mydb = msq.connect(host = 'localhost', user = 'root', \
        passwd = 'Gbsss@1532', database = 'school')
myCursor = mydb.cursor()
myCursor.execute("""update student set age = 29
                    where Roll_Number = 16
                    """)
mydb.commit()
```

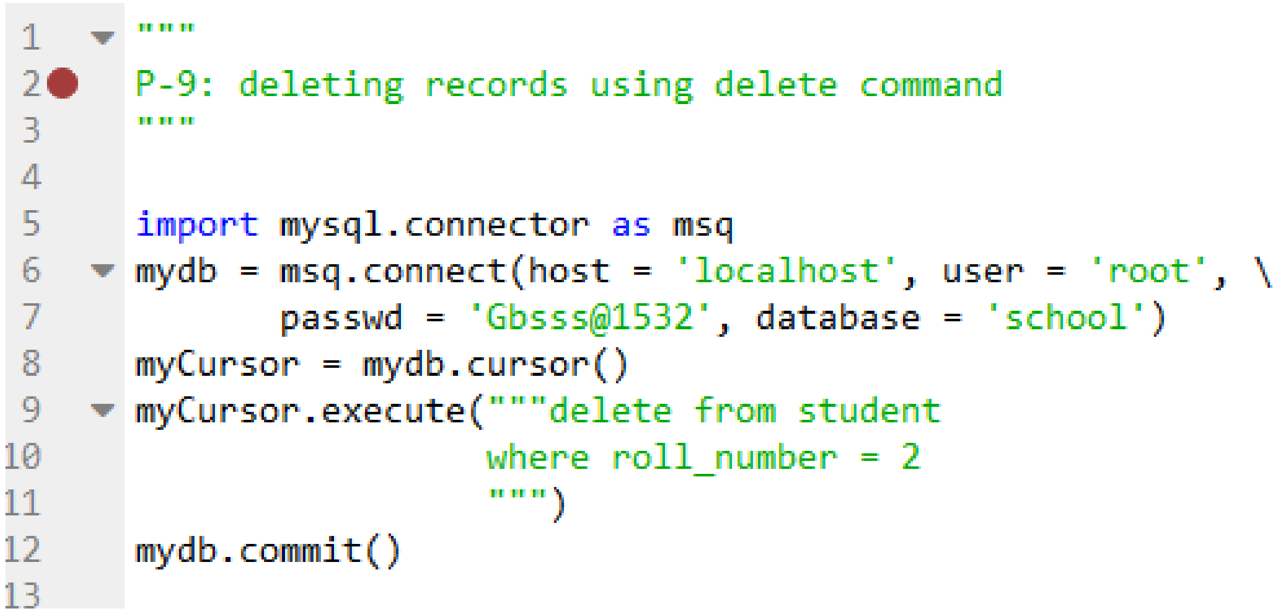


```
mysql> select * from student;
+-------------+-------------+-----+-----------+-------+
| Roll_Number | StudentName | age | city      | marks |
+-------------+-------------+-----+-----------+-------+
|           2 | Raj         |  23 | Srinagar  |    34 |
|          12 | Akash       |  19 | Delhi     |    75 |
|          16 | Himanshu    |  29 | Delhi     |    49 |
|          34 | Akshat      |  13 | Jaipur    |    67 |
+-------------+-------------+-----+-----------+-------+
4 rows in set (0.05 sec)
```

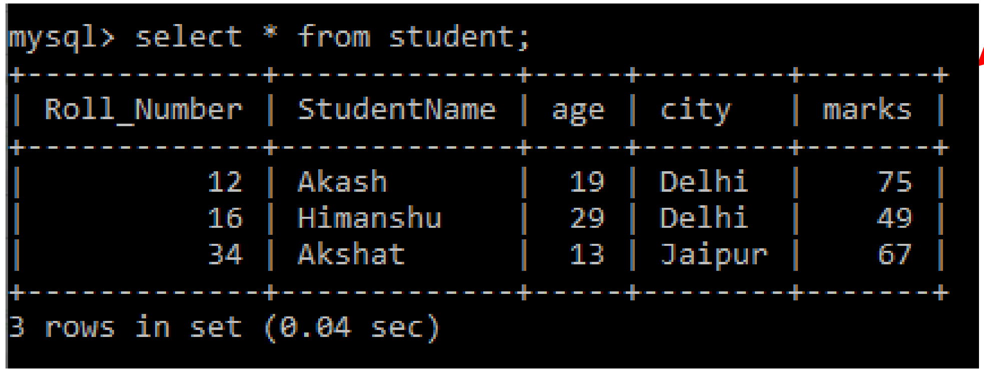
Program-9: deleting records



```
mysql> select * from student;
+-------------+-------------+-----+----------+-------+
| Roll_Number | StudentName | age | city     | marks |
+-------------+-------------+-----+----------+-------+
|           2 | Raj         |  23 | Srinagar |    34 |
|          12 | Akash       |  19 | Delhi    |    75 |
|          16 | Himanshu    |  29 | Delhi    |    49 |
|          34 | Akshat      |  13 | Jaipur   |    67 |
+-------------+-------------+-----+----------+-------+
4 rows in set (0.05 sec)
```



```python
"""
P-9: deleting records using delete command
"""

import mysql.connector as msq
mydb = msq.connect(host = 'localhost', user = 'root', \
          passwd = 'Gbsss@1532', database = 'school')
myCursor = mydb.cursor()
myCursor.execute("""delete from student
                    where roll_number = 2
                    """)
mydb.commit()
```



```
mysql> select * from student;
+-------------+-------------+-----+--------+-------+
| Roll_Number | StudentName | age | city   | marks |
+-------------+-------------+-----+--------+-------+
|          12 | Akash       |  19 | Delhi  |    75 |
|          16 | Himanshu    |  29 | Delhi  |    49 |
|          34 | Akshat      |  13 | Jaipur |    67 |
+-------------+-------------+-----+--------+-------+
3 rows in set (0.04 sec)
```

Reading values from the table

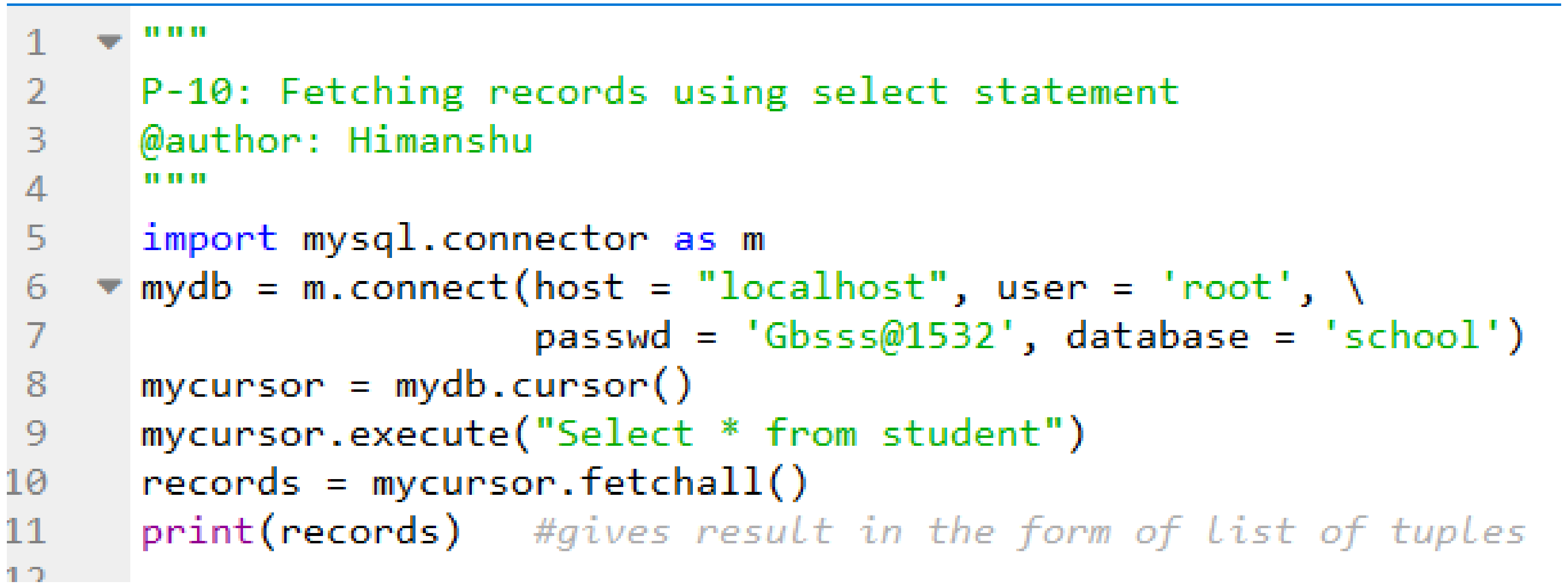
- In order to fetch data from database using Python IDE, we will be using select statement as per the data requirement, like

```
mycursor.execute("select * from student")
```

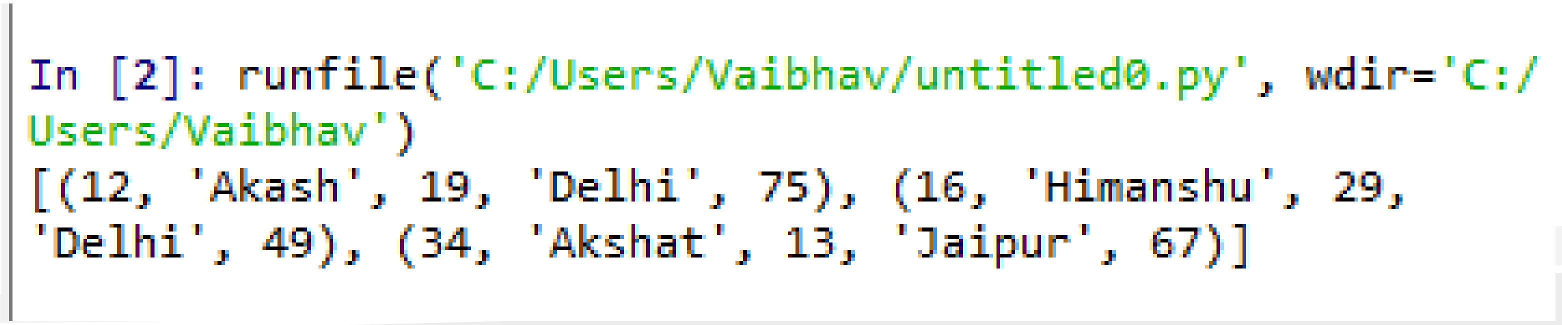
- Data from the database can be retrieved using cursor object along with any of the below functions:

  a.) fetchall()
  b.) fetchone()
  c.) fetchmany()

Reading values from the table: fetchall()

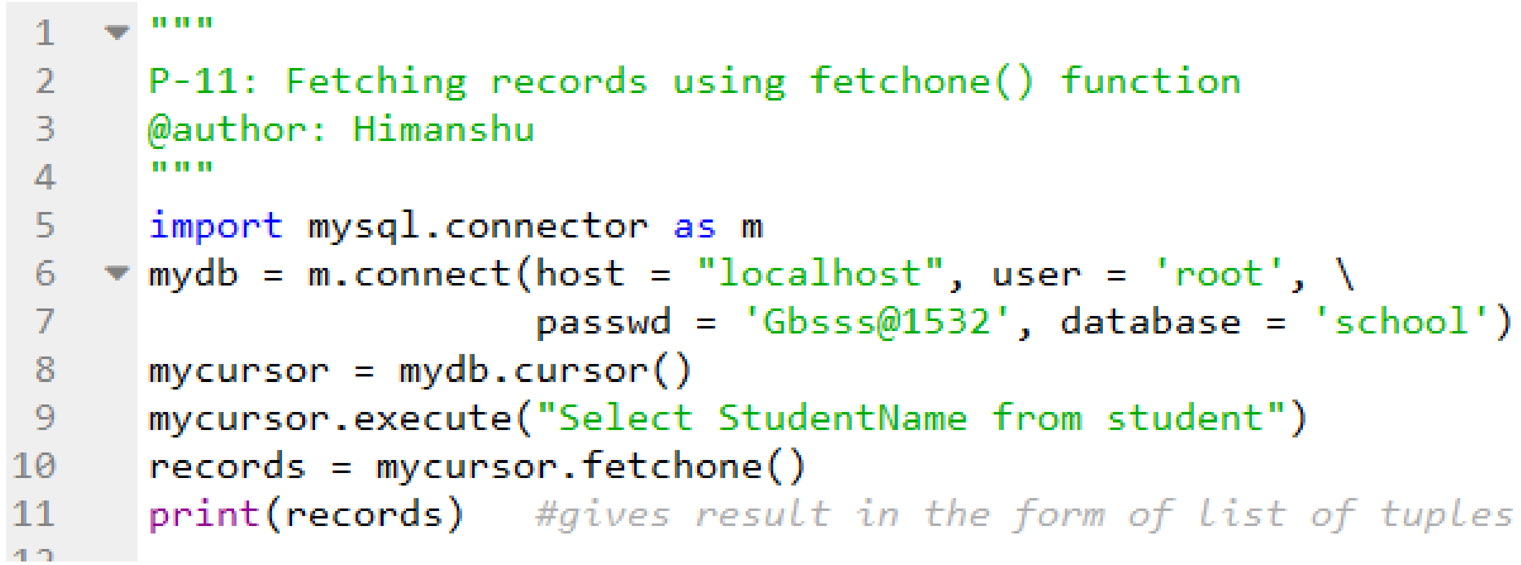


```python
"""
P-10: Fetching records using select statement
@author: Himanshu
"""
import mysql.connector as m
mydb = m.connect(host = "localhost", user = 'root', \
                 passwd = 'Gbsss@1532', database = 'school')
mycursor = mydb.cursor()
mycursor.execute("Select * from student")
records = mycursor.fetchall()
print(records)   #gives result in the form of list of tuples
```

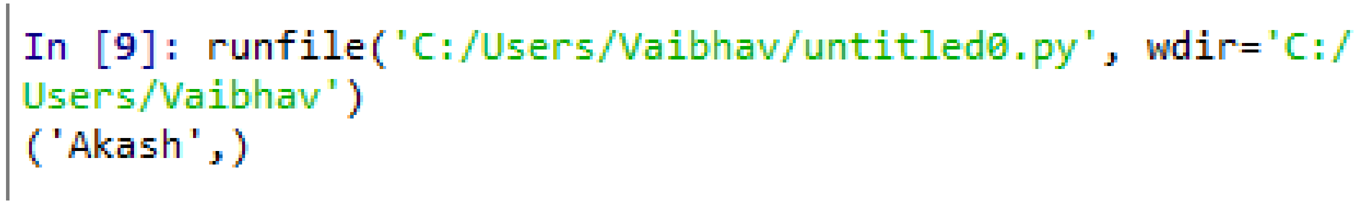


```
In [2]: runfile('C:/Users/Vaibhav/untitled0.py', wdir='C:/
Users/Vaibhav')
[(12, 'Akash', 19, 'Delhi', 75), (16, 'Himanshu', 29,
'Delhi', 49), (34, 'Akshat', 13, 'Jaipur', 67)]
```

Reading values from the table: fetchone()

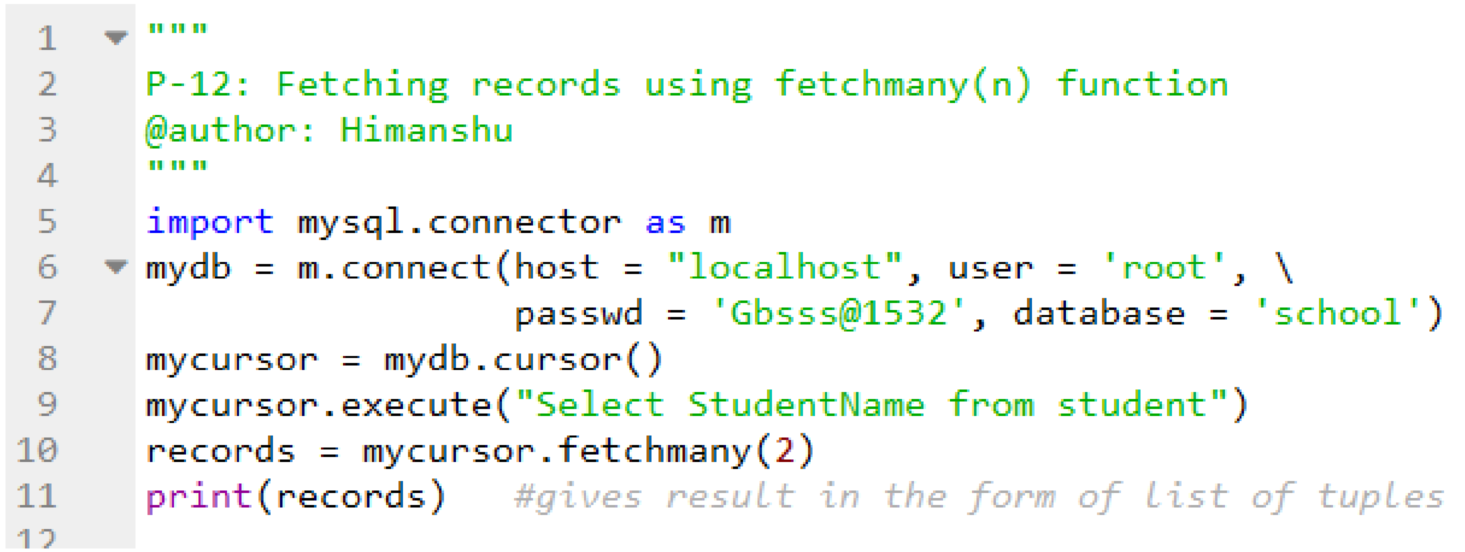


```python
"""
P-11: Fetching records using fetchone() function
@author: Himanshu
"""
import mysql.connector as m
mydb = m.connect(host = "localhost", user = 'root', \
                 passwd = 'Gbsss@1532', database = 'school')
mycursor = mydb.cursor()
mycursor.execute("Select StudentName from student")
records = mycursor.fetchone()
print(records)    #gives result in the form of list of tuples
```

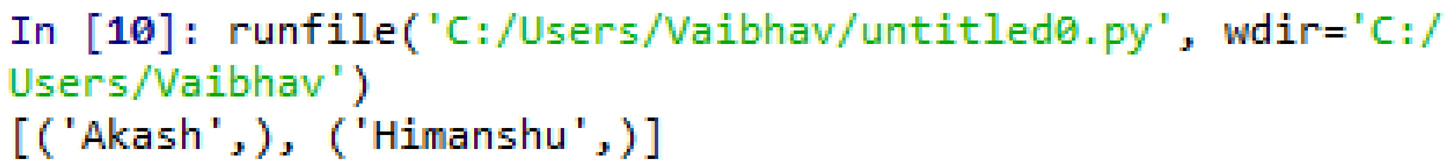


```
In [9]: runfile('C:/Users/Vaibhav/untitled0.py', wdir='C:/
Users/Vaibhav')
('Akash',)
```

Reading values from the table: fetchmany()



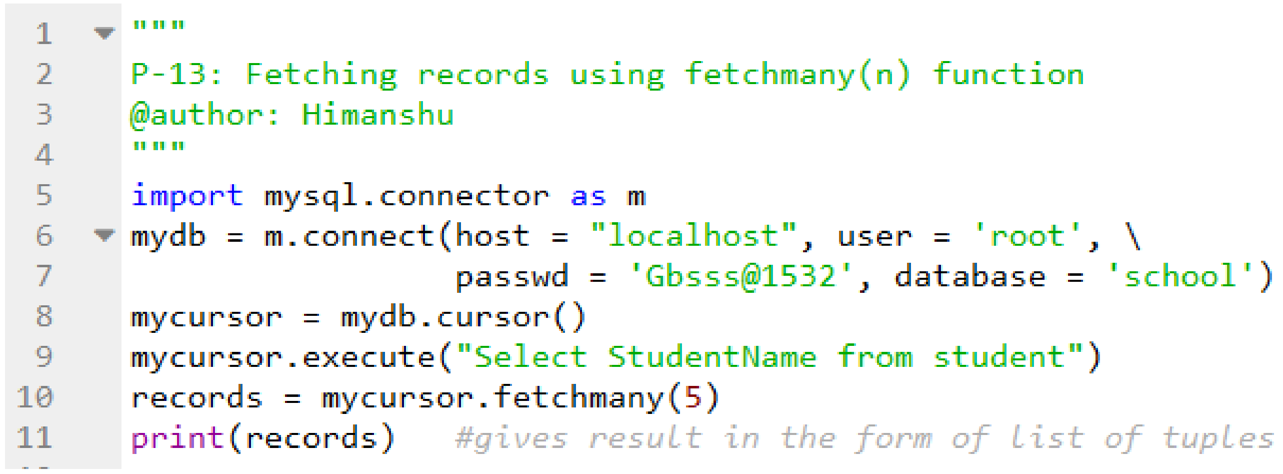
```
"""
P-12: Fetching records using fetchmany(n) function
@author: Himanshu
"""
import mysql.connector as m
mydb = m.connect(host = "localhost", user = 'root', \
                 passwd = 'Gbsss@1532', database = 'school')
mycursor = mydb.cursor()
mycursor.execute("Select StudentName from student")
records = mycursor.fetchmany(2)
print(records)   #gives result in the form of list of tuples
```



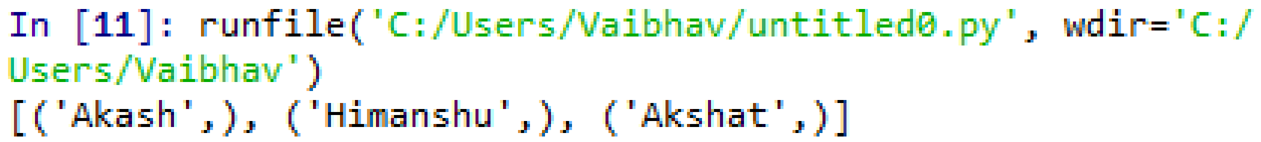
```
In [10]: runfile('C:/Users/Vaibhav/untitled0.py', wdir='C:/
Users/Vaibhav')
[('Akash',), ('Himanshu',)]
```

Reading values from the table: fetchmany()

if we try to fetch more than number of records in the database, then it would return available number of records only.

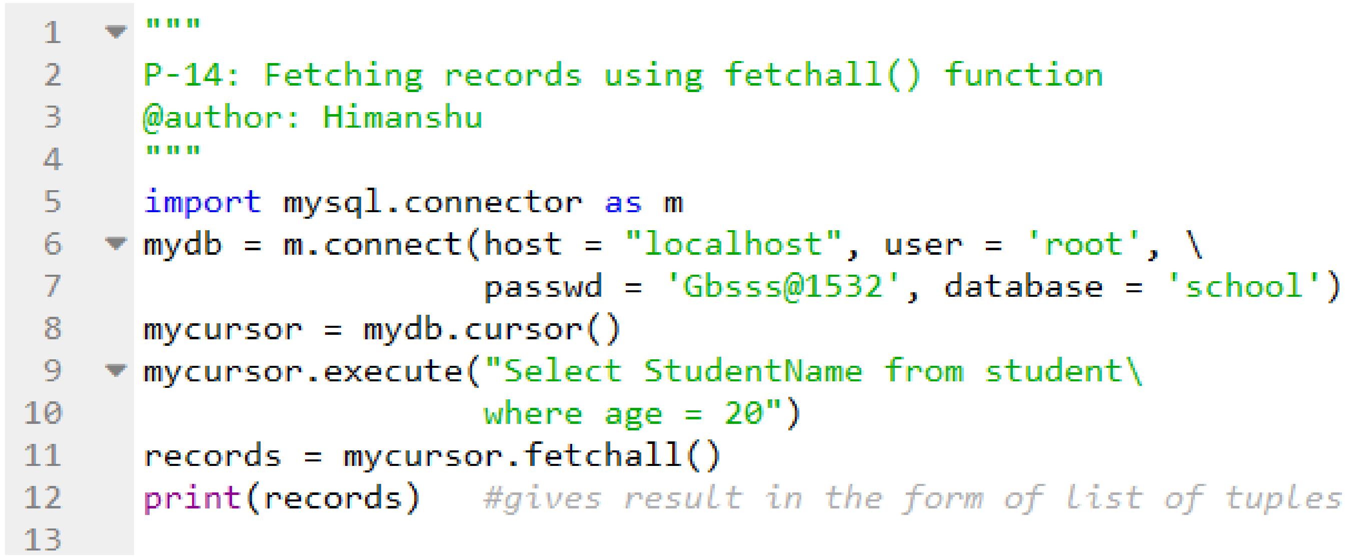


```python
"""
P-13: Fetching records using fetchmany(n) function
@author: Himanshu
"""
import mysql.connector as m
mydb = m.connect(host = "localhost", user = 'root', \
                 passwd = 'Gbsss@1532', database = 'school')
mycursor = mydb.cursor()
mycursor.execute("Select StudentName from student")
records = mycursor.fetchmany(5)
print(records)   #gives result in the form of list of tuples
```

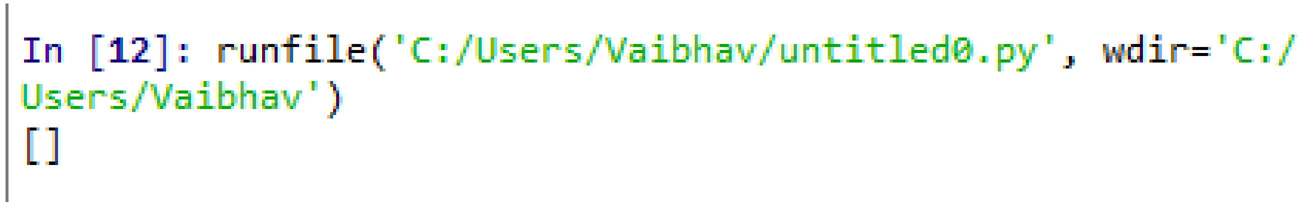


```
In [11]: runfile('C:/Users/Vaibhav/untitled0.py', wdir='C:/
Users/Vaibhav')
[('Akash',), ('Himanshu',), ('Akshat',)]
```

Return of datatype incase there is no value in the database: fetchall()

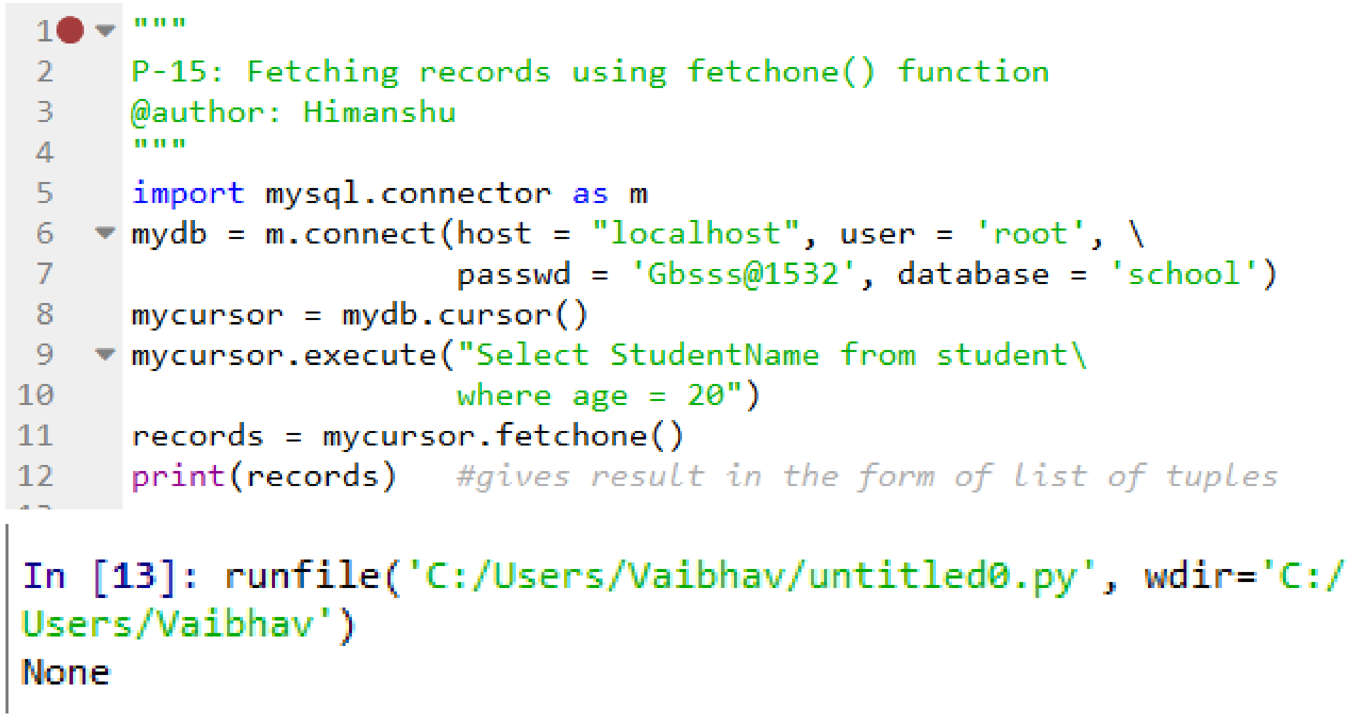


```python
"""
P-14: Fetching records using fetchall() function
@author: Himanshu
"""
import mysql.connector as m
mydb = m.connect(host = "localhost", user = 'root', \
                 passwd = 'Gbsss@1532', database = 'school')
mycursor = mydb.cursor()
mycursor.execute("Select StudentName from student\
                 where age = 20")
records = mycursor.fetchall()
print(records)   #gives result in the form of list of tuples

```



```
In [12]: runfile('C:/Users/Vaibhav/untitled0.py', wdir='C:/
Users/Vaibhav')
[]
```

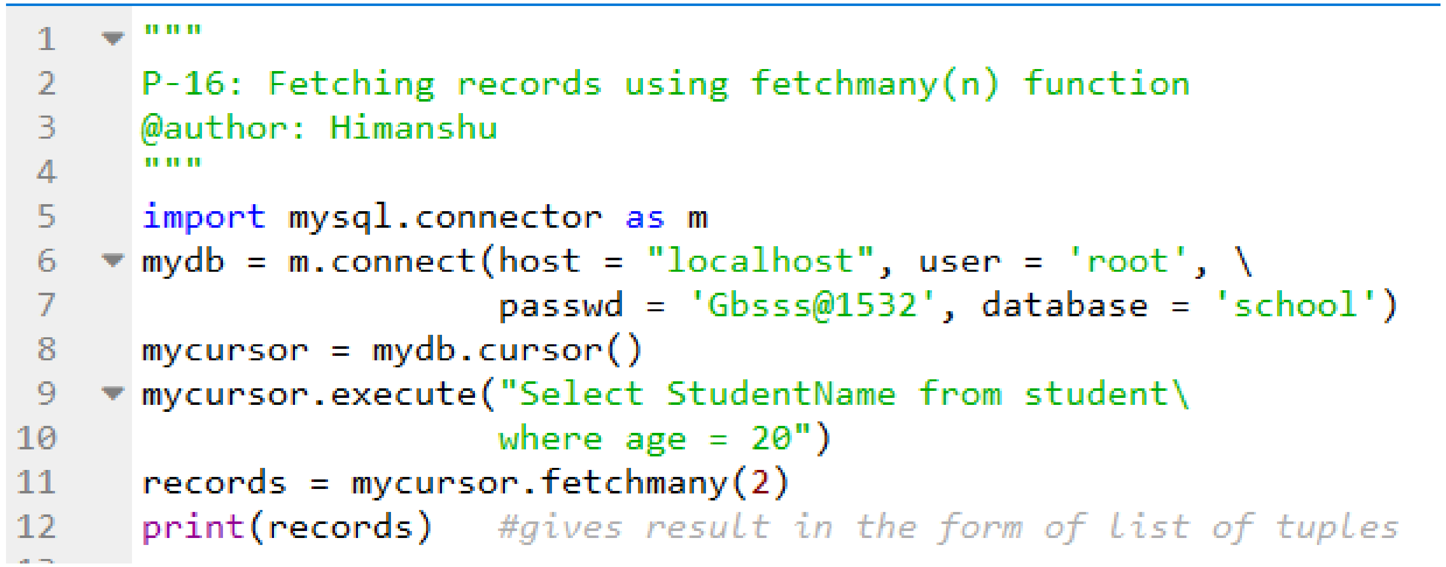
Return of datatype incase there is no value in the database: fetchone()



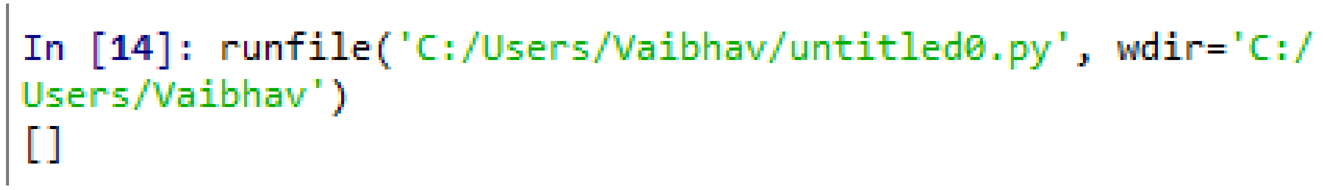
```python
"""
P-15: Fetching records using fetchone() function
@author: Himanshu
"""
import mysql.connector as m
mydb = m.connect(host = "localhost", user = 'root', \
                 passwd = 'Gbsss@1532', database = 'school')
mycursor = mydb.cursor()
mycursor.execute("Select StudentName from student\
                 where age = 20")
records = mycursor.fetchone()
print(records)   #gives result in the form of list of tuples
```

```
In [13]: runfile('C:/Users/Vaibhav/untitled0.py', wdir='C:/
Users/Vaibhav')
None
```

Return of datatype incase there is no value in the database: fetchmany(n)



```python
"""
P-16: Fetching records using fetchmany(n) function
@author: Himanshu
"""
import mysql.connector as m
mydb = m.connect(host = "localhost", user = 'root', \
                 passwd = 'Gbsss@1532', database = 'school')
mycursor = mydb.cursor()
mycursor.execute("Select StudentName from student\
                 where age = 20")
records = mycursor.fetchmany(2)
print(records)   #gives result in the form of list of tuples
```



```
In [14]: runfile('C:/Users/Vaibhav/untitled0.py', wdir='C:/
Users/Vaibhav')
[]
```

Reading values from the table

function	returns	data type to return
fetchall()	all the rows of a query result set	list of tuples
fetchone()	next row of a query result set	tuple/None
fetchmany(n)	specified number of rows	list of tuples

Note:

- default value of n is 1
- If there is no value in a resultset, an empty list [] is returned in case of fetchall() and fetchmany(), and in case of fetchone(), special data type None is returned