DataBase

Introduction to MySQL Python connector

Installation

The MySQL Python Connector is available on pypi.org, therefore, you can install it using the pip command.

The pip command allows you to install MySQL Python connector on any Operating system including Windows, macOS, Linux, and Unix:

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

If you found any issue while installing, you can explicitly specify the module version

```
pip install mysql-connector-python==8.0.17
```

Note that to uninstall current MySQL Connector/Python, you use the following command:

```
pip uninstall mysql-connector-python
```

It will prompt you for confirmation, you type y to confirm.

```
Proceed (y/n)? y
```

Verifying MySQL Connector/Python installation

After installing the MySQL Python connector, you need to test it to make sure that it is working correctly and you are able to connect to the MySQL database server without any issues.

To verify the installation, you use the following steps:

1. Open Python command line
2. Type the following code

```
import mysql.connector as con
conn=con.connect(host="localhost",user="root",password="",database="creative")
if conn.is_connected():
    print('connected')
else:
    print('not')
```

Creating a Database

To create a database in MySQL, use the "CREATE DATABASE" statement:

```
import mysql.connector as con

mydb=con.connect(host="localhost",user="gautam",password="")

mycursor=mydb.cursor()
mycursor.execute("create database school")
```

Insert data

```
import mysql.connector as con
conn=con.connect(host="localhost",user="root",password="",database="creative")
if conn.is_connected():
    print('connected')
else:
    print('not')

sql="insert into student(name,email,contact)
values('gautam','gautam@gmail.com','9797987')"
mycursor=conn.cursor()
mycursor.execute(sql)
conn.commit()

print(mycursor.rowcount,' record inserted')
mycursor.close()
```

value from the user data

```
import mysql.connector as con

conn=con.connect(host='localhost',user='root',password='',database='creative')
if conn.is_connected():
    print('connected')
else:
    print('not connected')

name1=input("enter name=")
email1=input('enter email=')
contact1=input('enter contact=')
```

```
qry="insert into student(name,email,contact) values(%s,%s,%s)"
val=(name1,email1,contact1)

cursor=conn.cursor()
cursor.execute(qry,val)
conn.commit()

print(cursor.rowcount,' record inserted')
cursor.close()
```

Insert Multiple Rows

To insert multiple rows into a table, use the `executemany()` method.

The second parameter of the `executemany()` method is a list of tuples, containing the data you want to insert: