# 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")
```

## Create Table

```
import mysql.connector as c

mydb=c.connect(host='localhost',user='root',password='',database='school')

mycursor=mydb.cursor()

mycursor.execute('create table student(id INT AUTO_INCREMENT PRIMARY KEY,name
VARCHAR(100),contact VARCHAR(100),email VARCHAR(100),password VARCHAR(100))')
```

## Check if Table Exists

You can check if a table exist by listing all tables in your database with the "SHOW TABLES" statement:

```
mydb=c.connect(host='localhost',user='root',password='',database='school')

mycursor=mydb.cursor()

mycursor.execute("SHOW TABLES")

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

If the table already exists, use the ALTER TABLE keyword:

Create primary key on an existing table:

```
import mysql.connector as c

mydb=c.connect(host='localhost',user='root',password='',database='school')

mycursor=mydb.cursor()

mycursor.execute("ALTER TABLE student ADD COLUMN id INT AUTO_INCREMENT PRIMARY
KEY")
```

## Insert data

```
import mysql.connector as con

conn=con.connect(host="localhost",user="root",password="",database='datavidya')

if conn.is_connected():
    print('connected')
else:
    print('not')

mycursor=conn.cursor()

sql="insert into student(name,contact, email, password) values('mohit',
'9909151897','mohit@gmail.com','123')"

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='datavidya')

if conn.is_connected():
    print('connected')
else:
    print('not')

mycursor=conn.cursor()

name=input("enter name=")
email=input('enter email=')
contact=input('enter contact=')
password=input('enter password=')

sql="insert into student(name,contact, email, password) values(%s,%s,%s,%s)"
val=(name,contact,email,password)

mycursor.execute(sql,val)
conn.commit()

print(mycursor.rowcount,' record inserted')
mycursor.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:

```
import mysql.connector as con

conn=con.connect(host="localhost",user="root",password="",database='datavidya')

if conn.is_connected():
    print('connected')
else:
    print('not')

mycursor=conn.cursor()


sql="insert into student(name,contact, email, password) values(%s,%s,%s,%s)"
val=[('haresh','398457','haresh@gmail.com','123456'),('jaydip','345678','jaydip
@gmail.com','567')]

mycursor.executemany(sql,val)
conn.commit()

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

## Select From a Table

“select * from table name”

```
import mysql.connector as con

conn=con.connect(host="localhost",user="root",password="",database="creative")

if conn.is_connected:
    print('database connected')
else:
    print('database not connected')

mydb=conn.cursor()

qry="select * from student"

mydb.execute(qry)

res=mydb.fetchall()

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

# MySQL Where

sql = "SELECT * FROM customers WHERE email ='emailid'"
mycursor.execute(sql)

## Sort the Result

Use the ORDER BY statement to sort the result in ascending or descending order.

The ORDER BY keyword sorts the result ascending by default. To sort the result in descending order, use the DESC keyword.

```
import mysql.connector as con

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

mydb=conn.cursor()

qry='select * from student order by name'

mydb.execute(qry)

res=mydb.fetchall()

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

## ORDER BY DESC

Use the DESC keyword to sort the result in a descending order.

```

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

cursor=conn.cursor()
qry="select * from student order by name DESC"
cursor.execute(qry)
res=cursor.fetchall()

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

## Delete Record

You can delete records from an existing table by using the "DELETE FROM" statement:

```
import mysql.connector as con

conn=con.connect(host="localhost",user="root",password="",database='datavidya')

mycursor=conn.cursor()

sql="delete from student where id='1'"
mycursor.execute(sql)
conn.commit()
```

## Delete a Table

You can delete an existing table by using the "DROP TABLE" statement:

```
import mysql.connector as con

conn=con.connect(host="localhost",user="root",password="",database='demo')

mycursor=conn.cursor()

sql = "drop table student"
mycursor.execute(sql)
```

If the the table you want to delete is already deleted, or for any other reason

does not exist, you can use the IF EXISTS keyword to avoid getting an error.

```
import mysql.connector as con

conn=con.connect(host="localhost",user="root",password="",database='demo')

mycursor=conn.cursor()

sql = "drop table if exists student"
mycursor.execute(sql)
```

## Update Table

You can update existing records in a table by using the "UPDATE" statement:

```
import mysql.connector as con

conn=con.connect(host="localhost",user="root",password="",database='datavidya')

mycursor=conn.cursor()

sql="update student set name='mohitkumar' where id='2'"
mycursor.execute(sql)
conn.commit()
```

## Limit the Result

You can limit the number of records returned from the query, by using the "LIMIT" statement

```
import mysql.connector as con

conn=con.connect(host="localhost",user="root",password="",database='datavidya')
mycursor=conn.cursor()
sql="select * from student limit 3"

mycursor.execute(sql)

myresult=mycursor.fetchall()

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

If you want to return 2 records, starting from the 2nd record, you can use the "OFFSET" keyword:

```
import mysql.connector as con

conn=con.connect(host="localhost",user="root",password="",database='datavidya')

mycursor=conn.cursor()

sql="select * from student limit 2 offset 2"
mycursor.execute(sql)

myresult=mycursor.fetchall()

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