



ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN  
VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

한-베정보통신기술대학교

# Chapter 6

## Part 1: Connecting MySQL



ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN  
VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

한-베정보통신기술대학교

# Part 1: Connecting MySQL

- 1. Introduction
- 2. Connecting MySQL and Python
- 3. Creating a Connection
- 4. Database Operations
- 5. Basic Queries using Python and MySQL

- When we create useful applications using Python, we need to store the data somewhere in a database so that it can be used later. One great relational database that we can use is MySQL
- We can easily connect Python with the MySQL database and do the database operations using Python. For that, we need to set up the MySQL database on our system and connect it to Python.
- In this chapter, let's see how we can set up the MySQL database and connect it to Python. We will also learn how we can do basic operations like creating databases, creating tables, inserting data, retrieving data, etc.

- Step 1: Install MySQL ver 9.0.1
  - Download: <https://dev.mysql.com/downloads/file/?id=531675>
  - Guide to install:  
<https://dev.mysql.com/doc/mysql-installation-excerpt/8.0/en/mysql-installer-setup.html>
- Step 2: Install Workbench (it is a visual design and modeling database access tool for the MySQL server).
  - Download: Mysql Workbench:  
<https://dev.mysql.com/downloads/workbench/>
  - User manual to MySQL Workbench:  
<https://dev.mysql.com/doc/workbench/en/>

- Step 1: install a MySQL connector package

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

```
(base) C:\Users\Lab>pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading https://files.pythonhosted.org/packages/43/97/fce1e40eff7da567f45f99364b00044134a1cb32c2caba7c2cc552557595
/mysql_connector_python-8.0.24-cp37-cp37m-win_amd64.whl (793kB)
    100% |#####| 798kB 1.1MB/s
Collecting protobuf>=3.0.0 (from mysql-connector-python)
  Downloading https://files.pythonhosted.org/packages/1d/9c/2609ddd0774f721a028352fcf864b9e4ec762099cdce0ea75462e00a7f57
/protobuf-3.16.0-cp37-cp37m-win_amd64.whl (904kB)
    100% |#####| 911kB 978kB/s
Requirement already satisfied: six>=1.9 in c:\users\lab\miniconda\lib\site-packages (from protobuf>=3.0.0->mysql-connect
or-python) (1.12.0)
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.0.24 protobuf-3.16.0

(base) C:\Users\Lab>
```



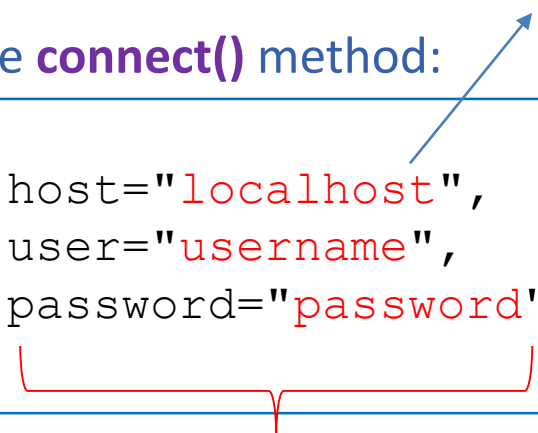
- Import the **mysql.connector** module:

```
import mysql.connector
```

Name of dataserver  
Or ip address dataserver

- Create a connection for dataserver using the **connect()** method:

```
mydb = mysql.connector.connect (  
    host="localhost",  
    user="username",  
    password="password")  
  
print (mydb)
```



We setup this value at time installing mySQL

- Create a connection for database:

```
import mysql.connector
mydb = mysql.connector.connect (
    host="localhost",
    user="username",
    password="password",
    database="namedatabase")

print(mydb)
```



- Create a database using `cursor()` method:

```
mycursor = mydb.cursor()  
mycursor.execute("CREATE DATABASE namedatabase")
```

- Display the databases that are existing:

```
mycursor = mydb.cursor()  
mycursor.execute("SHOW DATABASES")  
for x in mycursor:  
    print(x)
```

- Create a table:

```
mycursor = mydb.cursor()  
mycursor.execute('Create table Student(RollNo int(5),  
                                         Name varchar(30))')
```

- Show a table:

```
mycursor = mydb.cursor()  
mycursor.execute('SHOW TABLES')  
for x in mycursor:  
    print(x)
```

- Insert data into the table:

```
mycursor = mydb.cursor()
sql = 'INSERT INTO student (RollNo, Name) VALUES (%s, %s)'
val = ("1", "John")
mycursor.execute(sql, val)
mydb.commit()
```

- Insert multiple rows:

```
mycursor = mydb.cursor()
sql = 'INSERT INTO student (RollNo, Name) VALUES (%s, %s)'
val = [("2", "Bob"), ("3", "Elisa"), ("4", "Maxx")]
mycursor.executemany(sql, val)
mydb.commit()
```

- Retrieve data from the table:

```
mycursor = mydb.cursor()
mycursor.execute("SELECT * FROM student")
myresult = mycursor.fetchall()
for x in myresult:
    print(x)
```

- `fetchall()` method will return all rows in table
- `fetchone()` method will return the first row

- Sort the result alphabetically:

```
mycursor = mydb.cursor()
sql = "SELECT * FROM customers ORDER BY name"
mycursor.execute(sql)
myresult = mycursor.fetchall()
for x in myresult:
    print(x)
```

- Use the DESC keyword to sort the result in a descending order:

```
mycursor = mydb.cursor()
sql = "SELECT * FROM customers ORDER BY name DESC"
mycursor.execute(sql)
myresult = mycursor.fetchall()
for x in myresult:
    print(x)
```

- Delete record:

```
mycursor = mydb.cursor()
sql="DELETE FROM customers WHERE address =' 21 Tran Phu'"
mycursor.execute(sql)
mydb.commit()
```

- Delete a Table:

```
mycursor = mydb.cursor()
sql = "DROP TABLE customers"
mycursor.execute(sql)
mydb.commit()
```

- Drop Only if Exist:

```
mycursor = mydb.cursor()
sql = "DROP TABLE IF EXISTS customers"
mycursor.execute(sql)
```

- Update Table:

```
mycursor = mydb.cursor()
sql = "UPDATE customers SET address = 'Canyon 123' WHERE
      address = 'Valley 345'"

mycursor.execute(sql)
mydb.commit()
```

- Using the placeholder **%s** method

```
mycursor = mydb.cursor()
sql = "UPDATE customers SET address=%s WHERE address=%s"
val = ("Valley 345", "Canyon 123")
mycursor.execute(sql, val)
mydb.commit()
```



- Limit the Result:

```
mycursor = mydb.cursor()  
mycursor.execute("SELECT * FROM customers LIMIT 5")  
myresult = mycursor.fetchall()
```

- Start from position 3, and return 5 records:

```
mycursor = mydb.cursor()  
mycursor.execute("SELECT * FROM customers LIMIT 5 OFFSET 2")  
myresult = mycursor.fetchall()
```

- Join Two or More Tables:

```
mycursor = mydb.cursor()
sql = "SELECT users.name AS user, products.name AS favorite \
FROM users INNER JOIN products ON users.fav = \
products.id"
mycursor.execute(sql)
myresult = mycursor.fetchall()
```

- LEFT JOIN:

```
sql = "SELECT users.name AS user, products.name AS favorite\
FROM users LEFT JOIN products ON users.fav = products.id"
```

- RIGHT JOIN:

```
sql = "SELECT users.name AS user, products.name AS favorite\
FROM users RIGHT JOIN products ON users.fav = products.id"
```

# Practice Part 1