



Database with python MySQL

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by circles of varying sizes, some with concentric rings, and the lines are thin and grey. The diagram is partially cut off by the left edge of the slide.

1.

What is Database

Let's start

Database Structure



A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some with concentric rings, and the lines are thin and grey.

2.

Type of Database

◎ SQL

ข้อดี

- มีการเก็บข้อมูลเป็นตารางใช้งานง่าย
- มีมาตรฐาน
- เชื่อมต่อกับเครื่องมืออื่น ๆ ได้ง่าย
- ใช้งานได้หลากหลาย

เช่น Mysql, Oracle, Sybase etc

◎ NoSQL

ข้อดี

- ออกแบบมาเฉพาะทาง
- มีความรวดเร็วในการใช้งานสูง
- เหมาะสำหรับการใช้งานที่มีข้อมูลจำนวนมาก

เช่น Cassandra, MongoDB, Redis etc

◎ New SQL

เป็นคำนิยามใหม่ ซึ่งจะรวม ข้อดี ของ Database ทั้ง 2 ชนิดมารวมกัน
แต่ยังคงเป็น Database ที่ไม่ได้มาตรฐาน และยังมีข้อจำกัดบางอย่าง แตกต่าง
ตามผู้ใช้งาน

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by circles of varying sizes, some with concentric rings, and the lines are thin and grey. The overall structure is organic and sprawling.

3.

Relationship of Table in Database

1 to 1



1 to Many



Many to Many



A decorative network diagram in the top-left corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow with a grey outline. The connections form a complex, branching structure.

4.

MySQL Database



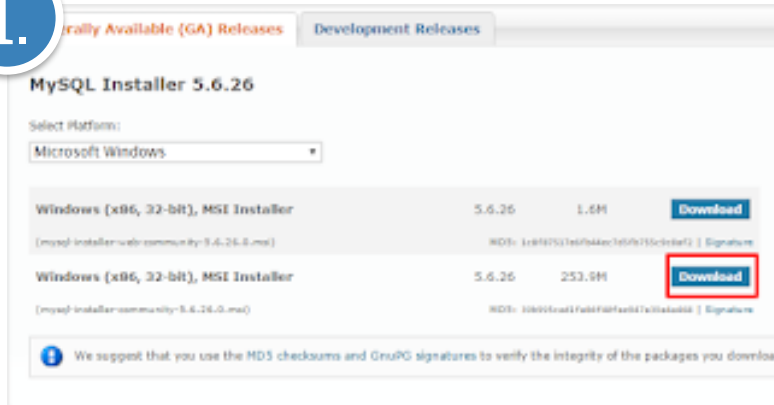
“

MySQL

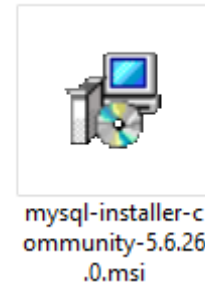
ทำไมต้อง MySQL เพราะว่า ใช้งานง่าย
เข้ากับหลาย ๆ Program มีFunction ใน
การทำงานครบครัน และเป็นที่นิยมใช้

การติดตั้ง MySQL และ MySQL Workbench

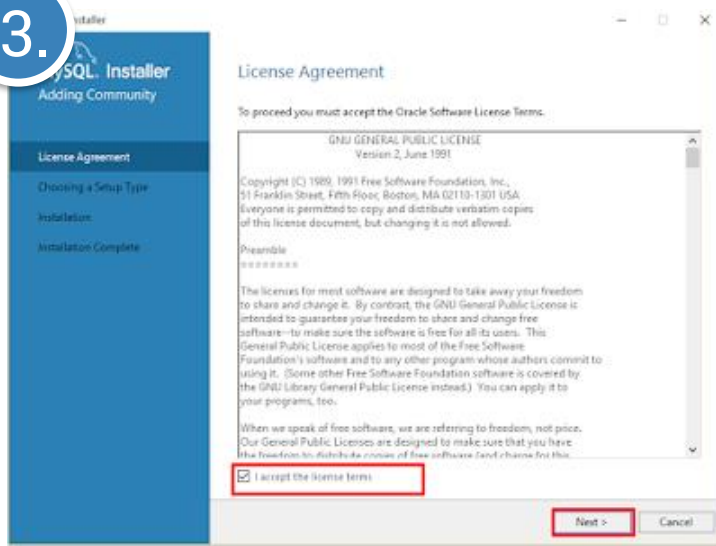
1.



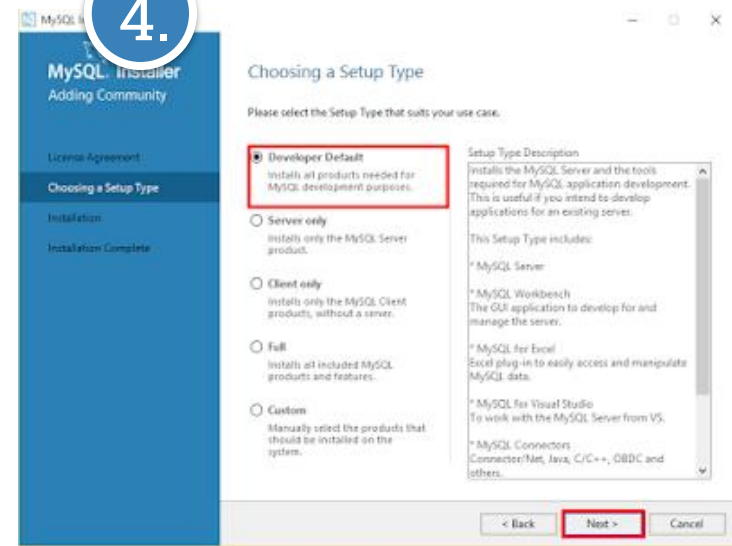
2.



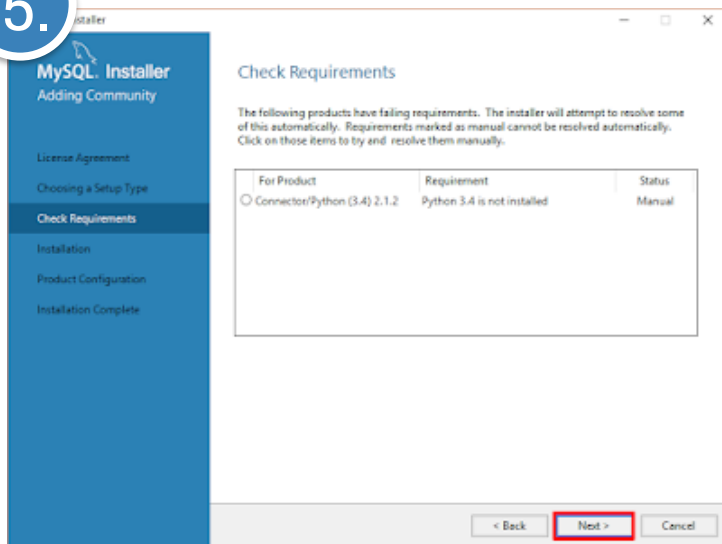
3.



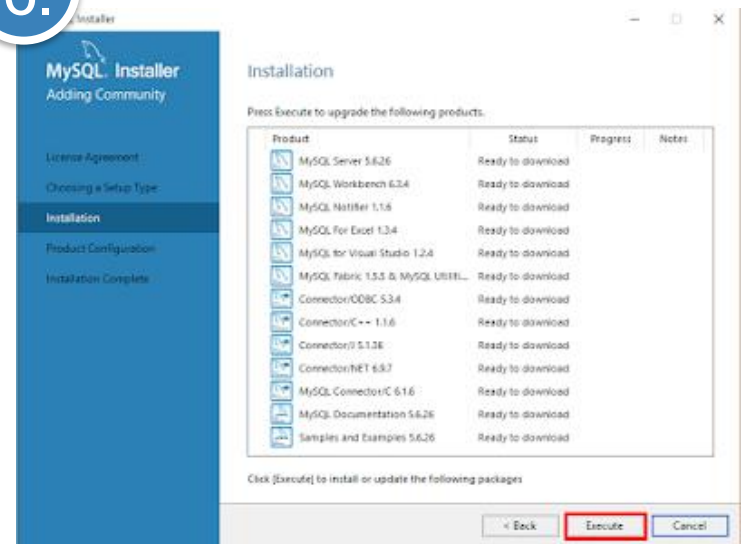
4.



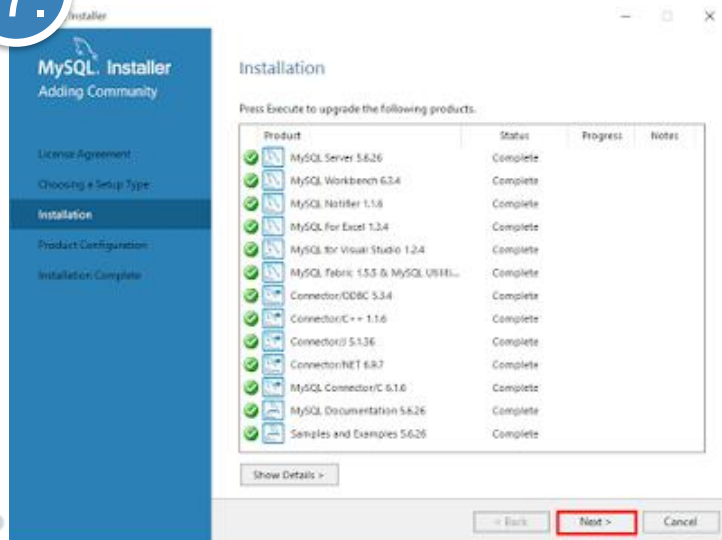
5.



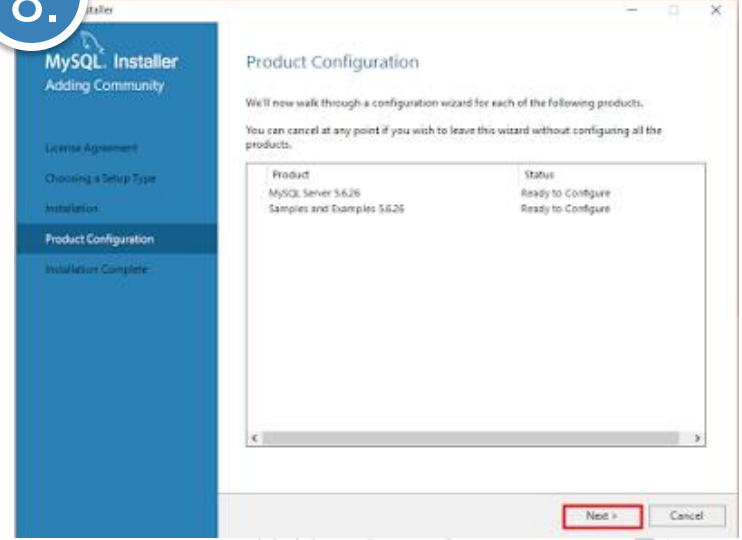
6.



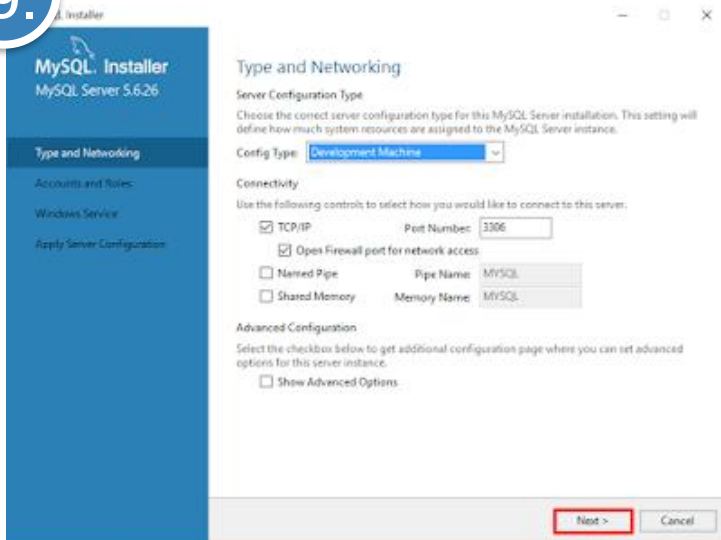
7.



8.



9.



MySQL Installer
MySQL Server 5.6.26

Type and Networking

Server Configuration Type
Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type: **Development Machine**

Connectivity
Use the following controls to select how you would like to connect to this server.

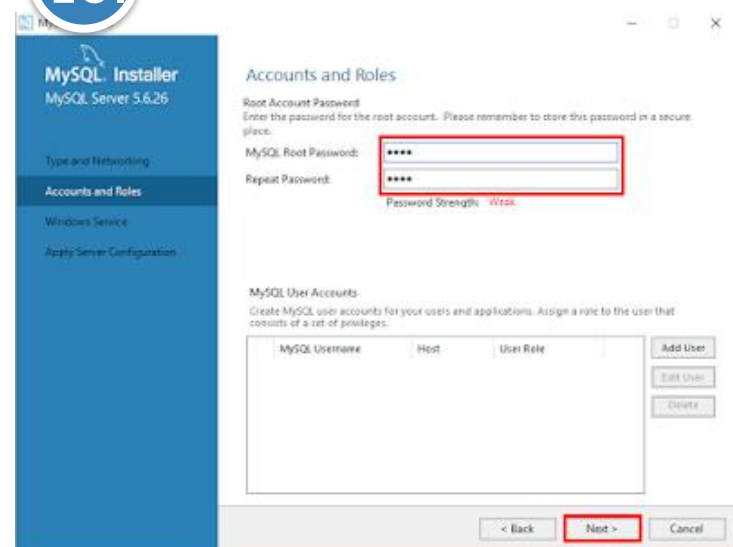
☒ TCP/IP Port Number: 3306
☒ Open Firewall port for network access.
☐ Named Pipe Pipe Name: MYSQL
☐ Shared Memory Memory Name: MYSQL

Advanced Configuration
Select the checkbox below to get additional configuration page where you can set advanced options for this server instance.

☐ Show Advanced Options

Next > Cancel

10.



MySQL Installer
MySQL Server 5.6.26

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: ****
Repeat Password: ****
Password Strength: 0/23

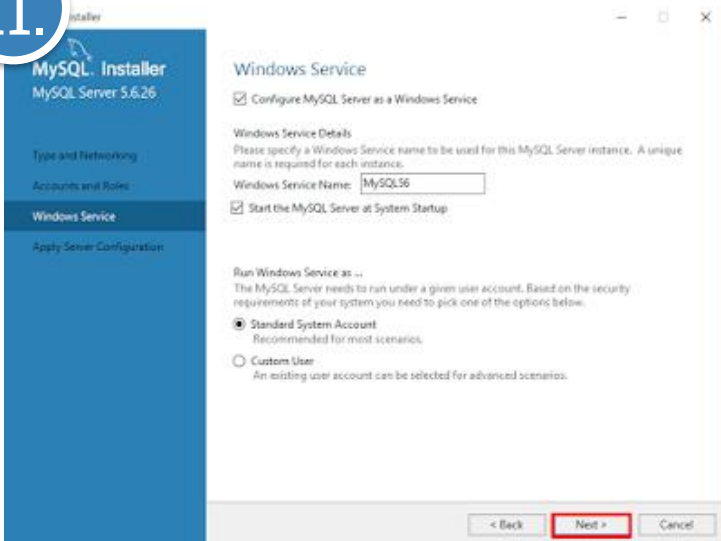
MySQL User Accounts
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

MySQL Username	Host	User Role

Add User Edit User Delete

< Back Next > Cancel

11.



MySQL Installer
MySQL Server 5.6.26

Windows Service

☒ Configure MySQL Server as a Windows Service

Windows Service Details
Please specify a Windows Service name to be used for this MySQL Server instance. A unique name is required for each instance.

Windows Service Name:

☒ Start the MySQL Server at System Startup

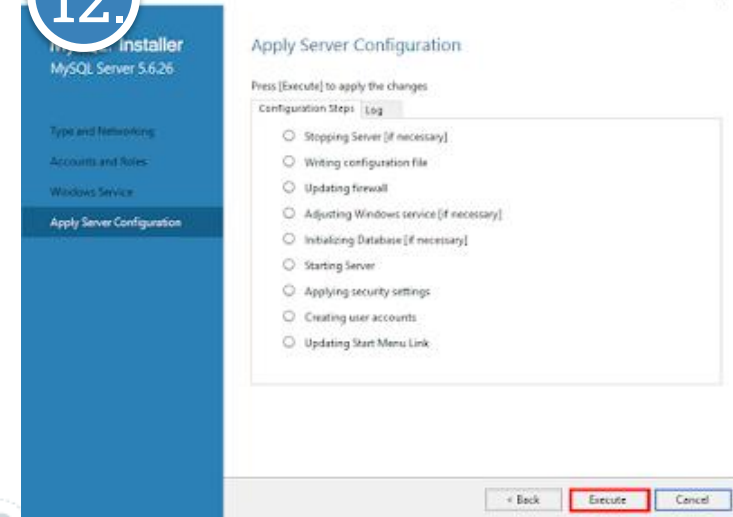
Run Windows Service as ...
The MySQL Server needs to run under a given user account. Based on the security requirements of your system you need to pick one of the options below.

☒ Standard System Account
Recommended for most scenarios.

☐ Custom User
An existing user account can be selected for advanced scenarios.

< Back Next > Cancel

12.



MySQL Installer
MySQL Server 5.6.26

Apply Server Configuration

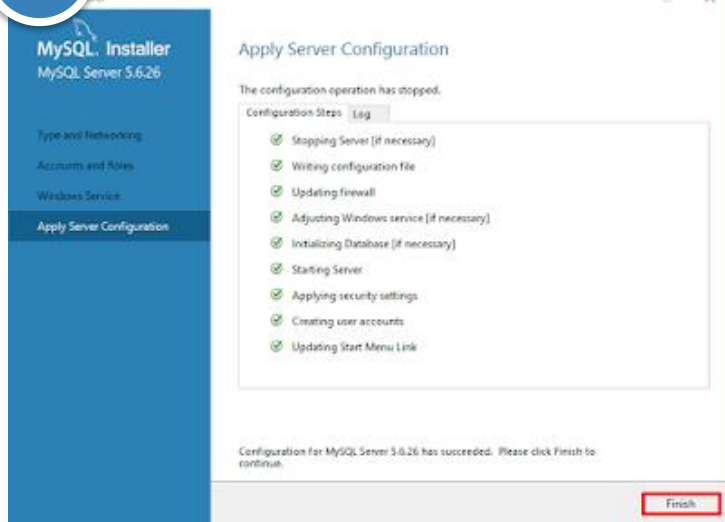
Press [Execute] to apply the changes

Configuration Steps Log

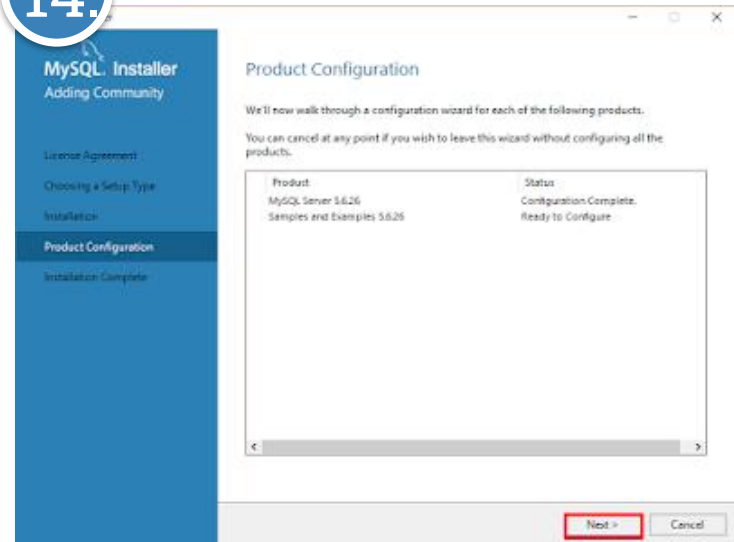
- ☐ Stopping Server [if necessary]
- ☐ Writing configuration file
- ☐ Updating firewall
- ☐ Adjusting Windows service [if necessary]
- ☐ Initializing Database [if necessary]
- ☐ Starting Server
- ☐ Applying security settings
- ☐ Creating user accounts
- ☐ Updating Start Menu Link

< Back Execute Cancel

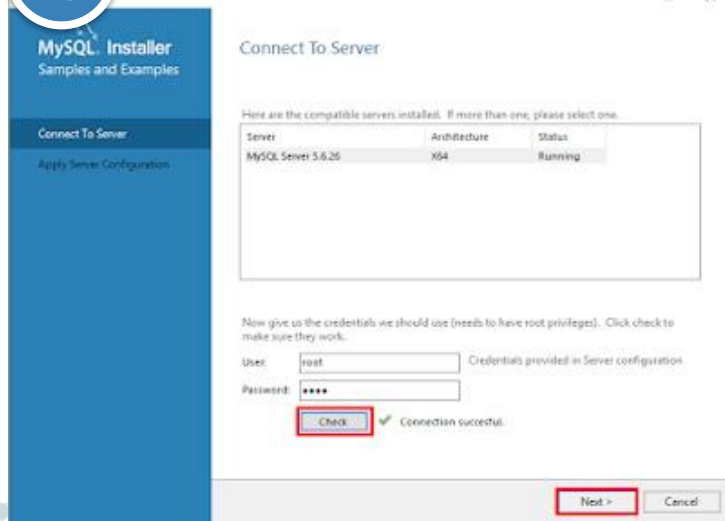
13.



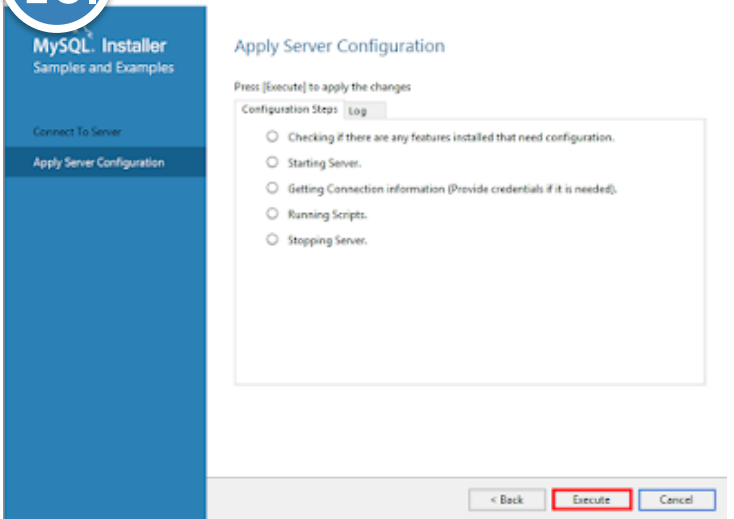
14.

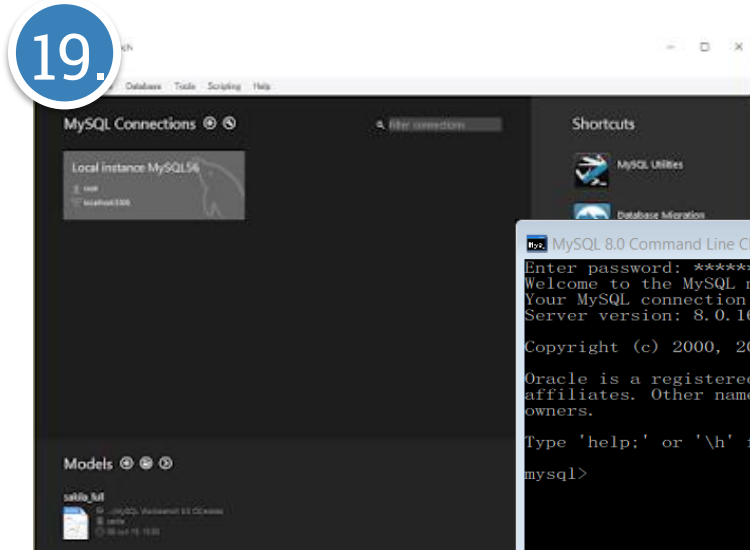
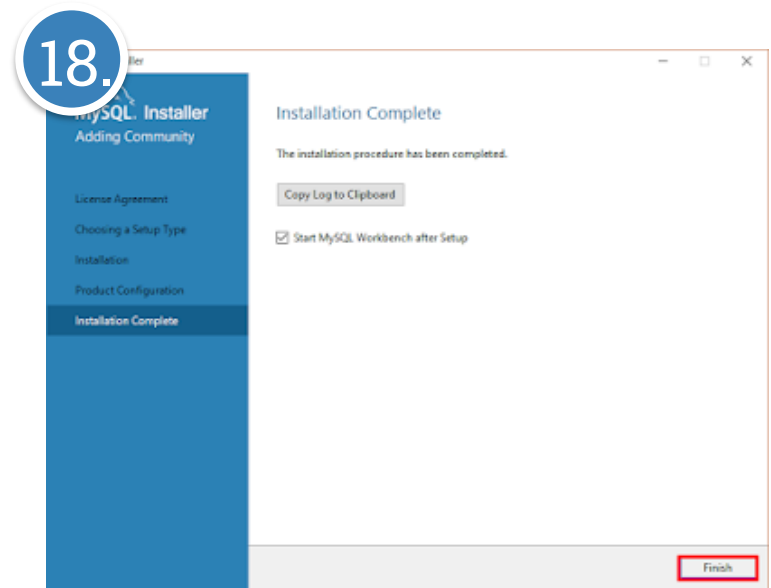
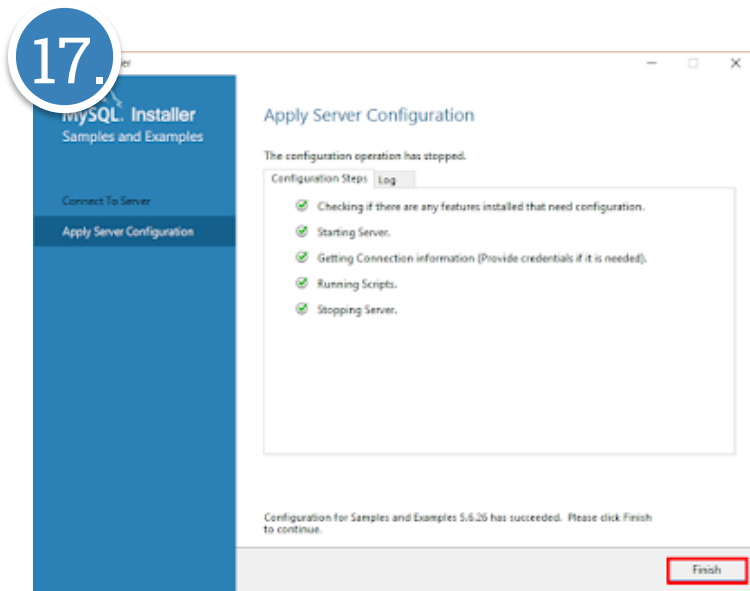


15.

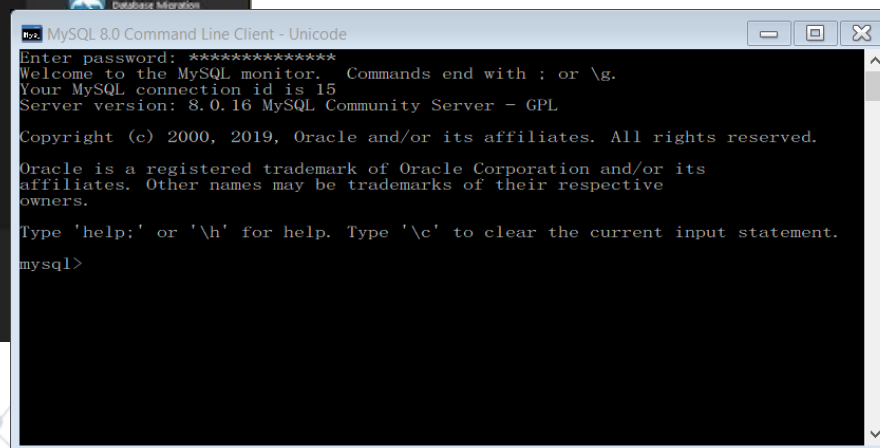


16.

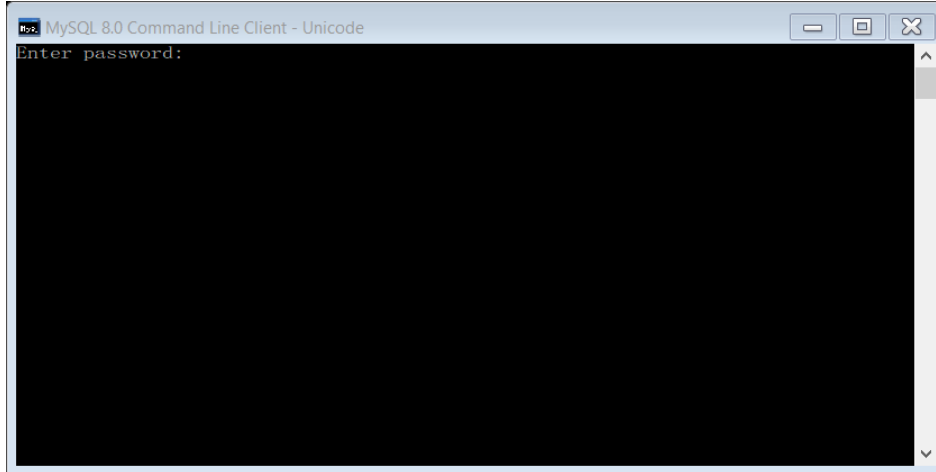




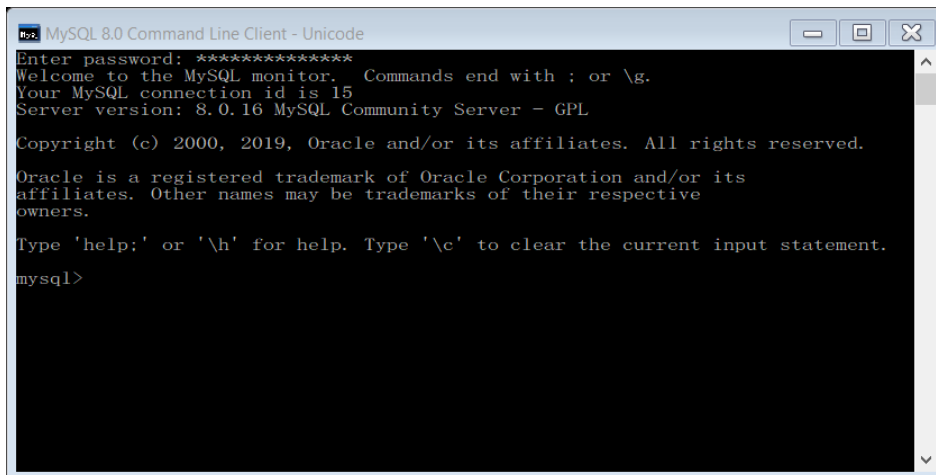
เข้าผ่าน MySQL Workbench หรือ MySQL Command line - Unicode



Login ด้วย password ที่เราตั้งไว้



```
MySQL 8.0 Command Line Client - Unicode
Enter password:
```



```
MySQL 8.0 Command Line Client - Unicode
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.16 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

ได้นี้แบบนี้

ใช้คำสั่ง

show databases;

เพื่อดู Databases ทั้งหมด ที่มี



ใช้คำสั่ง

create database <database name>;

เพื่อสร้าง Database ที่ต้องการ

```
MySQL 8.0 Command Line Client - Unicode
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
mysql>
mysql> show databases;
+-----+
| Database |
+-----+
| cnxtraining |
| information_schema |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
7 rows in set (0.15 sec)

mysql>
```


ใช้คำสั่ง

use <database name>;

เพื่อใช้ Database นั้น ๆ

```
MySQL 8.0 Command Line Client - Unicode
mysql>
mysql>
mysql> show databases:
+-----+
| Database |
+-----+
| cnxtraining |
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
7 rows in set (0.15 sec)

mysql>
mysql>
mysql> use cnxtraining
Database changed
mysql>
mysql>
mysql>
mysql>
mysql>
```

```
MySQL 8.0 Command Line Client - Unicode
sys
world
+-----+
| sys |
| world |
+-----+
7 rows in set (0.15 sec)

mysql>
mysql>
mysql> use cnxtraining
Database changed
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> show tables:
+-----+
| Tables_in_cnxtraining |
+-----+
| citizen |
| city |
| person |
| user |
+-----+
4 rows in set (0.17 sec)

mysql>
```

ใช้คำสั่ง

show tables;

เพื่อดู Table ทั้งหมด ถ้าไม่
มีให้ทำการสร้างก่อน

การสร้าง Table

ใช้คำสั่ง

```
create table <table name> (  
<column name 1> <type> <type> ...,  
<column name 2> <type> <type>...,  
... );
```

เช่น

```
create table user (  
user_id int(20) primary key,  
username varchar(255) not null  
);
```

ใช้คำสั่ง

```
describe <table name>;
```

เพื่อตรวจสอบลักษณะต่าง ๆ ของ Table ที่ทำการสร้างไว้

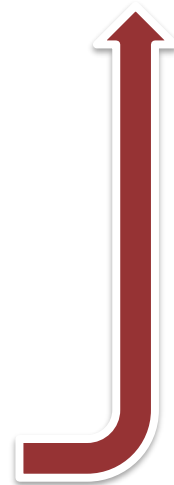
Data type	Function
int(number)	กำหนดค่าใน Table ให้เป็น int
varchar(number)	กำหนดค่าใน Table ให้เป็น string
float(number)	กำหนดค่าใน Table ให้เป็น float
primary key	กำหนดให้ค่าใน Table ไม่ซ้ำกัน
auto increment	กำหนดให้เพิ่มเลขบวกขึ้นไปเรื่อย ๆ
unsigned	กำหนดให้ค่ามีตั้งแต่ 0 และเป็น จำนวนเต็มบวก
default	กำหนดค่า default ให้ เมื่อไม่มีข้อมูลใส่มา
not null	กำหนดค่าให้ห้ามเป็นค่าว่าง
datetime(0-6)	กำหนดค่าใน Table ให้เป็น datetime

Insert

คือ การเพิ่มข้อมูลลงไปใน Database
ใช้คำสั่ง

insert into <table name> (column 1, column 2, ...) values (value 1, value 2, ...);

จำนวน column และ
value ต้องเท่ากัน และ
value ต้องตรงกับ
type ที่กำหนดไว้ตั้งแต่
ตอนสร้าง table



Select

คือ การเลือกข้อมูลใน Database ออกมาดู
ใช้คำสั่ง

```
select * from <table name> where <column name> = <value>;
```

สามารถใช้ ชื่อ
column แทน * ได้
เช่น **user_id**,
username

คำสั่ง **where** คือ
กำหนดไปว่า จะเจาะจง
ไปที่ข้อมูลตัวไหน

Update

คือ การเปลี่ยนแปลงค่าใน Database
ใช้คำสั่ง

```
update <table name> set column 1 = value 1, column 2 = value  
2, ... where <some column> = <some value>;
```

Delete

คือ การลบค่าใน Database
ใช้คำสั่ง

```
delete form <table name> where <some column> = <some  
value>;
```



การนำ Python มาเชื่อมกับ Database

1. Install lib mysql-connector
2. Import mysql.connector

การเชื่อม Database

```
mydb = mysql.connector.connect(  
    host='host_number',  
    port='3306',  
    database='database_name',  
    user='username',  
    password='password',  
    auth_plugin="mysql_native_password"  
)
```

Select

```
mycursor = mydb.cursor(dictionary=True)
text_command = "select * from <table name>;"
mycursor.execute(text_command)
```

```
myresult = mycursor.fetchall()
```

```
print myresult
```

Insert

```
mycursor = mydb.cursor(dictionary=True)
sql = "insert into <table name> (name, address) values ('name',  
'address');"
mycursor.execute(sql)
```

```
mydb.commit()
```

```
print mycursor.rowcount, "record inserted."
```


Update

```
mycursor = mydb.cursor(dictionary=True)
sql = "update <table name> set column 1 = value 1 where <some column> = <some value>;"
mycursor.execute(sql)
```

```
mydb.commit()
```

```
print mycursor.rowcount, "record(s) affected"
```

Delete

```
mycursor = mydb.cursor(dictionary=True)
sql = "delete from <table name> where <some column> = <some value>;"
mycursor.execute(sql)
```

```
mydb.commit()
```

```
print(mycursor.rowcount, "record(s) deleted")
```

Ex:

1. จงสร้าง Table ที่มีชื่อว่า user (จาก Code หรือ แบบปกติ) โดยต้องมี Column **user id, username, name** และ **password** โดย กำหนดให้ column user id และ username ห้ามมีค่าซ้ำกันในแต่ละแถว
2. จงสร้าง API ตามหลัก Standard Coding ให้สามารถ
 1. **Post** user เข้าไปได้
 2. **Get** user ออกมาได้
 3. **Put** user ได้
 4. **Delete** user ได้โดย การ Get user ออกมานั้น ให้ return เฉพาะ username และ name ออกมาทาง Postman ส่วน API ที่เหลือ ให้ return ออกมาว่า Success หรือ Fail รวมไปถึง Status code ด้วย

