



**ĐẠI HỌC
BÁCH KHOA HÀ NỘI**
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Database Practices:

INSTALL & SET UP A RELATIONAL DATABASE MANAGEMENT SYSTEM

Duy-Tung Nguyen
daniel.ng121.ds@gmail.com

ONE LOVE. ONE FUTURE.

Contents

1. Introduction to MySQL
2. Install and Set up MySQL
3. Using MySQL with others Programming Languages
4. Demo
5. Conclusion



A large, semi-transparent watermark of the HUST logo is positioned in the background of the slide. The logo consists of the letters "HUST" in a white, bold, sans-serif font, with a red, dotted, stylized "U" shape behind it.

HUST

1. INTRODUCTION TO MYSQL

Overview of DBMS

DBMS

- ❖ Is a software system for manage and organize data.
- ❖ Allows users to create, modify, query, manage the security and access controls for a database.

Key Features

- ❖ Data modeling
- ❖ Data storage and retrieval
- ❖ Concurrency control
- ❖ Data integrity and security
- ❖ Backup and recovery

02 Types of DBMS (*data model-based*)

- ❖ Relational Database Management System (RDBMS)
- ❖ Non-Relational Database Management System (NoSQL)



DBMS and its components

RDBMS and NoSQL

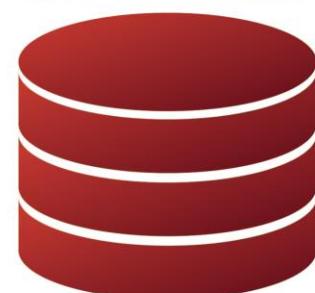
RDBMS



PostgreSQL



ORACLE



NoSQL



redis



elasticsearch



What is MySQL? – MySQL is:

- ❖ first released by MySQL AB (1995),
- ❖ developed, distributed, and supported by Oracle Corporation (now).
- ❖ RDBMS
- ❖ free and open-source¹
- ❖ very fast, reliable, scalable, and easy to use
- ❖ cross-platform: Linux, Windows, MacOS, v.v.
- ❖ compliant with the ANSI SQL standard



Michael "Monty" Widenius

Introduction to MySQL

Who uses MySQL?

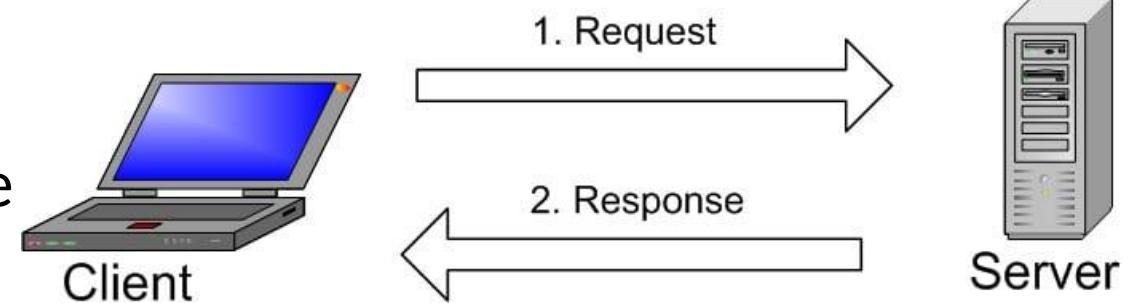
- ❖ Web devs
- ❖ Huge websites: Facebook, Twitters, Airbnb, Uber, Github, YouTube, v.v.
- ❖ Content Management Systems: WordPress, Drupal, v.v.



Introduction to MySQL

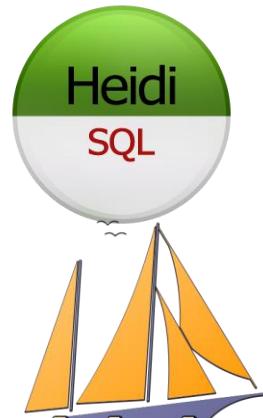
How does MySQL work?

- ❖ MySQL follows Client-Server Architecture
- ❖ MySQL Server: (a systems of) computer(s) that install MySQL server software.
→ store data
- ❖ MySQL Client: any software that can make queries to a MySQL server and receive returned results.



Free tools for MySQL:

- ✓ MySQL Workbench (Mac, Windows, Linux)
- ✓ Sequel Pro (Mac)
- ✓ HeidiSQL (Windows; Mac or Linux using WINE emulator)
- ✓ freephMyAdmin (web app)



Advantages

- ❖ Open-source and cost-effective
- ❖ Easy to use
- ❖ Performance and scalability
- ❖ Cross-platform compatibility
- ❖ Wide range of features
- ❖ Large community and active development

Disadvantages

- ❖ Limited data integrity support
- ❖ Potential data corruption
- ❖ Limited scalability for very large databases
- ❖ Not ideal for unstructured data
- ❖ Limited support for complex queries
- ❖ Requires regular maintenance

A large, semi-transparent watermark of the HUST logo is positioned on the left side of the slide. The logo consists of the letters "HUST" in a white, bold, sans-serif font, with a red, dotted, stylized "U" shape behind it.

HUST

2. INSTALL & SET UP MYSQL

- ❖ Install MySQL on Windows
- ❖ Practice with Workbench
- ❖ Practice with phpMyAdmin

Install MySQL

Install MySQL Server

- ❖ MySQL Server for Windows
- ❖ MySQL Server for Linux [Linux]
- ❖ MySQL Server for MacOS [\[MacOS\]](#)

MySQL for Windows: Download

- ❖ First of all, download the MySQL Installer:
<https://dev.mysql.com/downloads/installer/>
- ❖ Notice: If you use Installer, you also can install MySQL GUI for Clients

The screenshot shows the MySQL Installer download page. At the top, there are tabs for 'General Availability (GA) Releases' (highlighted in orange) and 'Archives'. Below the tabs, it says 'MySQL Installer 8.0.35'. A note states: 'Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.' A red box highlights the 'Select Version:' dropdown menu, which contains '8.0.35'. Another red box highlights the 'Download' button for the 'Windows (x86, 32-bit), MSI Installer' entry, which has a file size of 288.6M. To the right of the page, two annotations point to these highlighted areas.

Version	File Type	File Size	Action
8.0.35	Windows (x86, 32-bit), MSI Installer	2.1M	Download
8.0.35	Windows (x86, 32-bit), MSI Installer	288.6M	Download

Select version you want,
Suggest keep system's choice

Choose the Desktop version

MySQL for Windows: Download

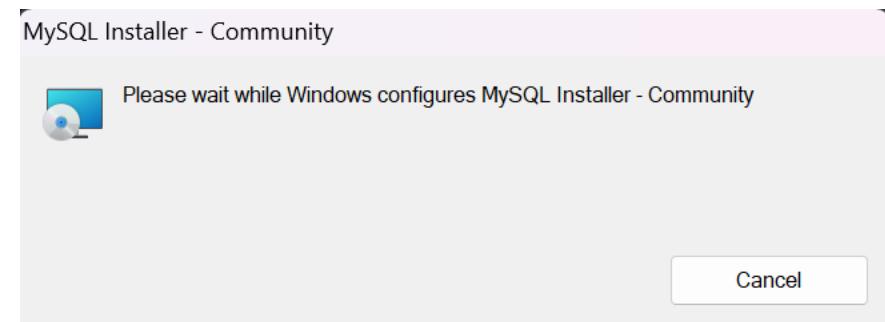
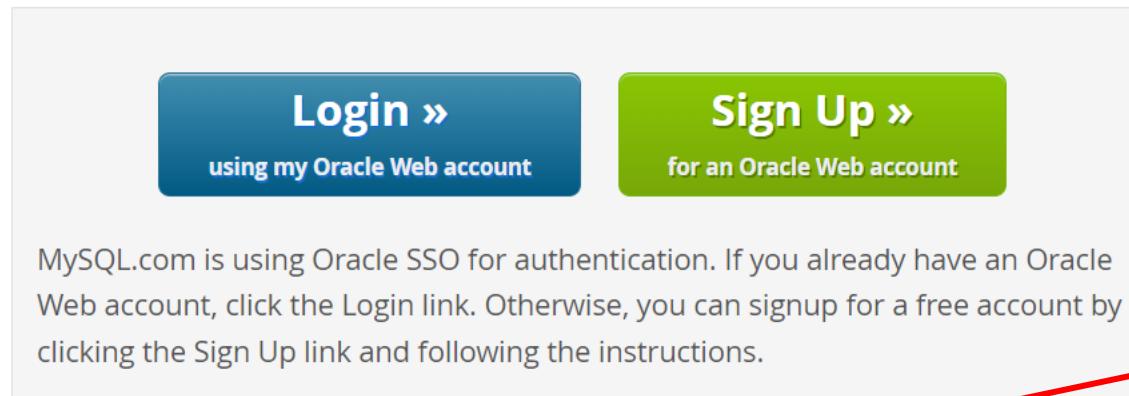
❖ Notice: The website will ask you to log in/ create an account ➔ Not necessary

④ MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system



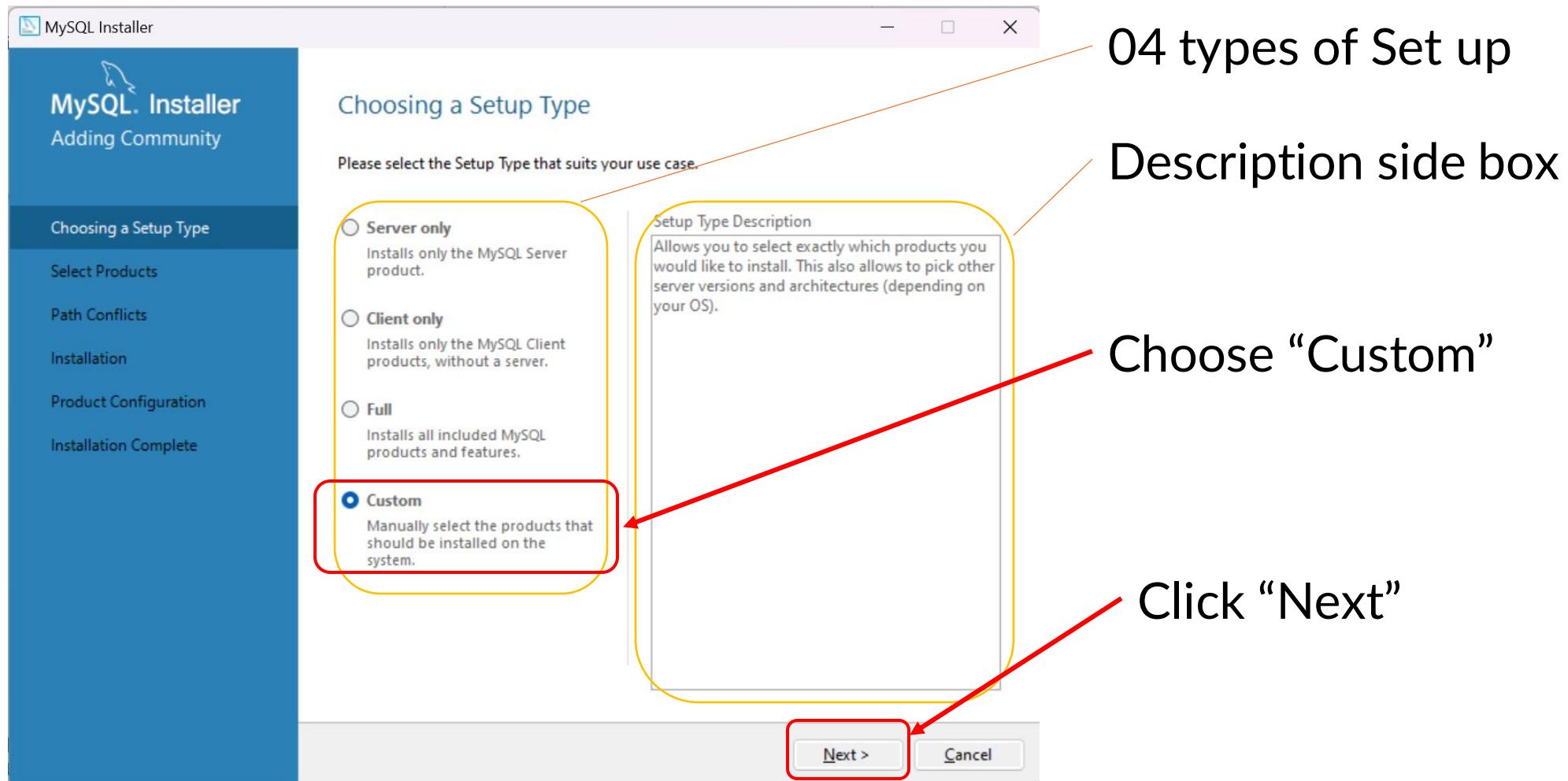
Wait and Allow

Click Here

No thanks, just start my download.

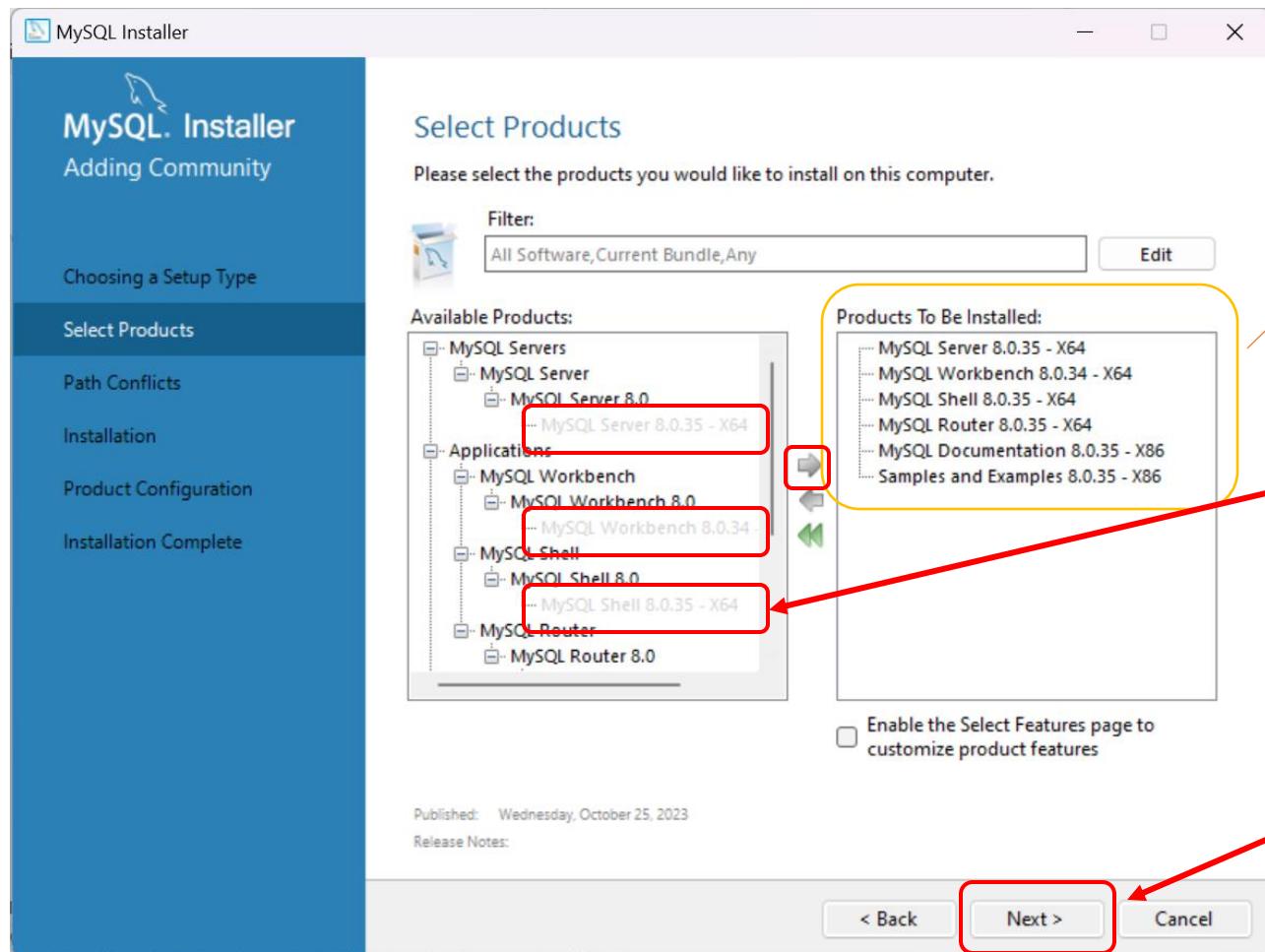
MySQL for Windows: Install

❖ Step 1:



MySQL for Windows: Install

❖ Step 2:



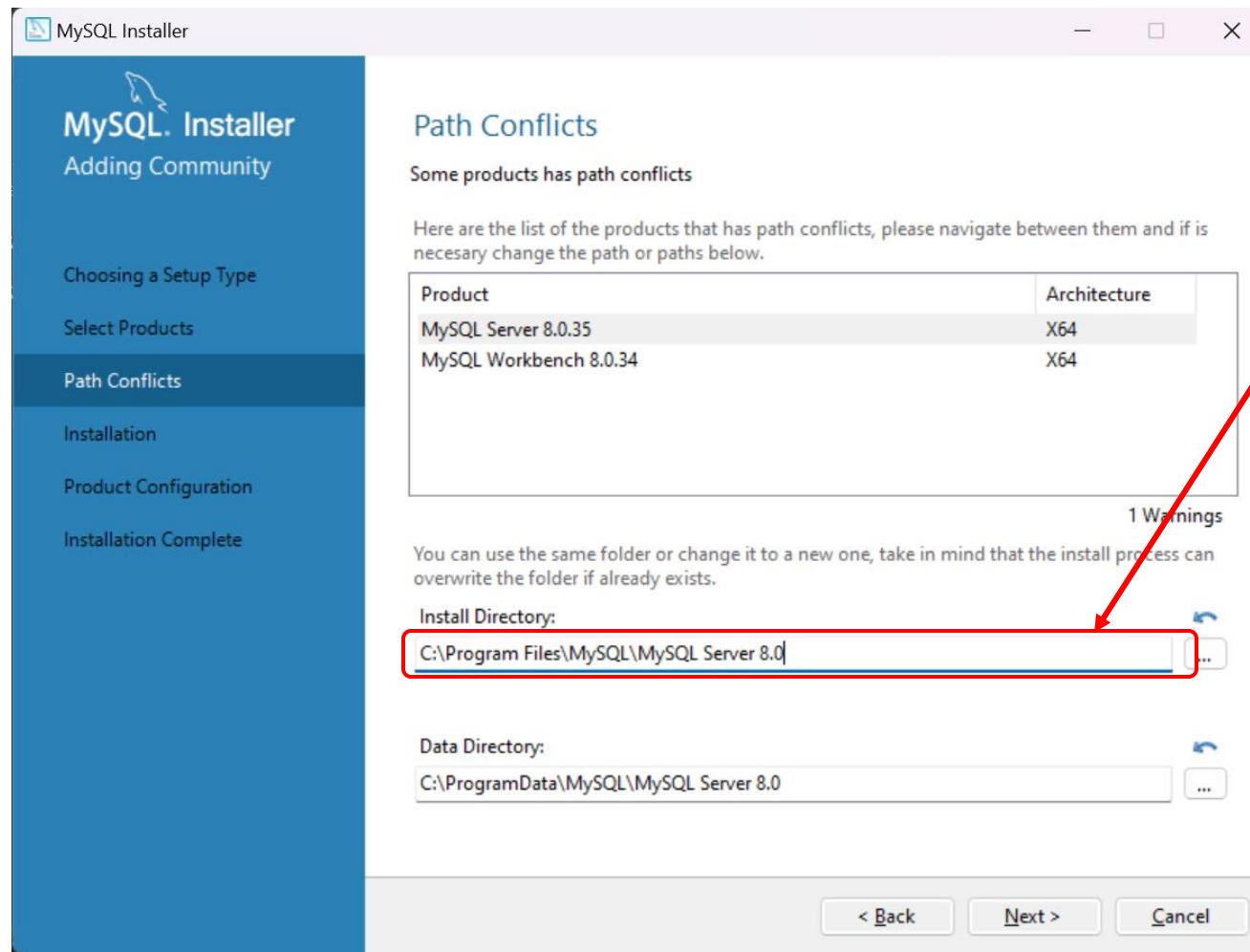
“Products to be installed” box

Choose lastest versions

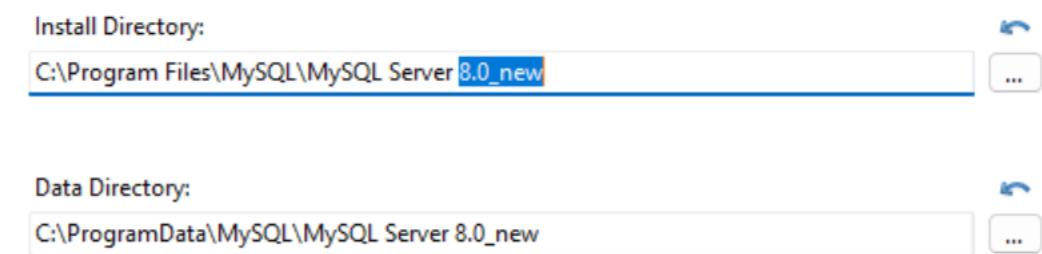
Click “Next”

MySQL Server for Windows: Install

❖ Step 2 (+1): Path Conflicts (*Happened if your path existed*)

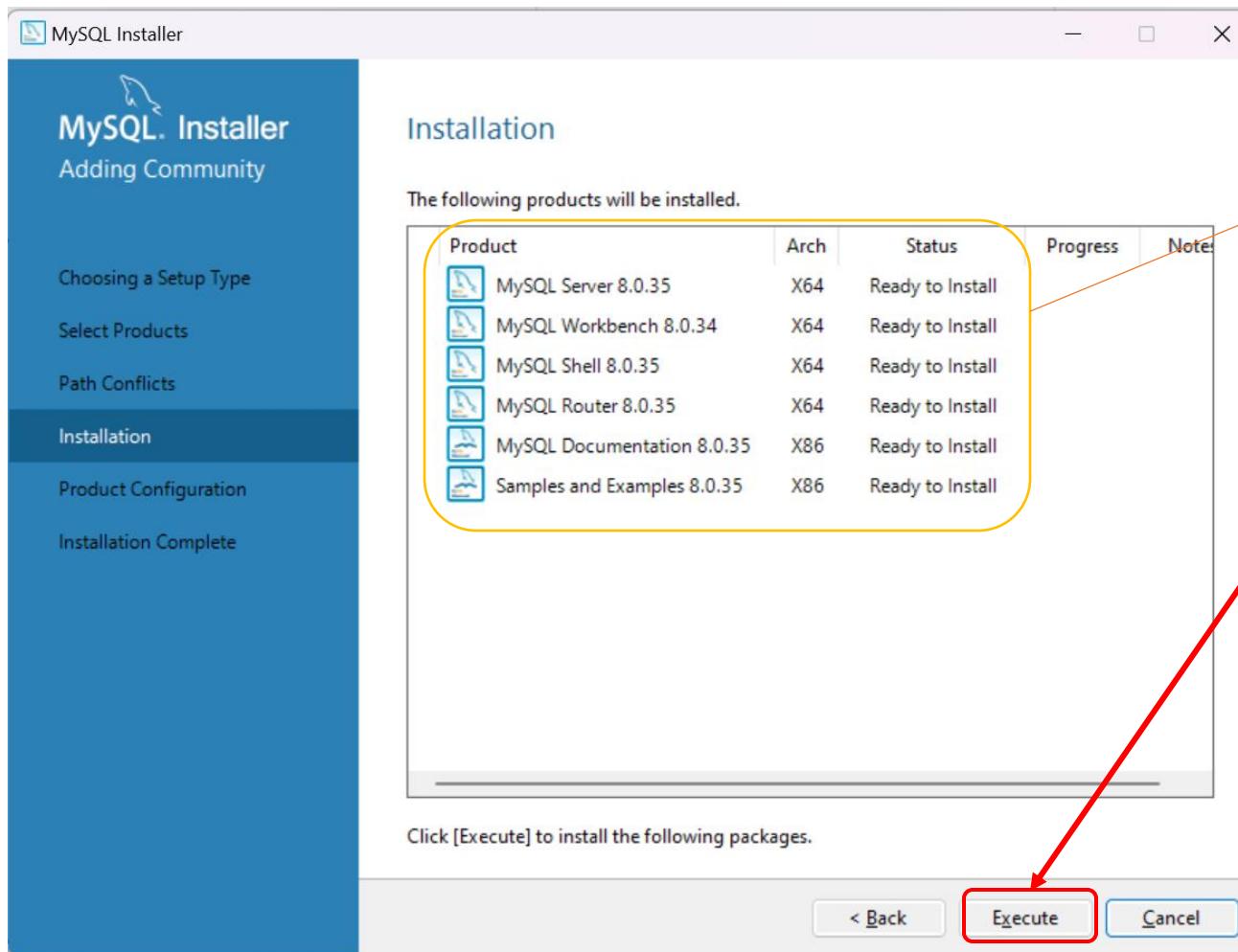


**Change the directories, if necessary
(Or change/delete the old one)**



MySQL for Windows: Install

❖ Step 3:

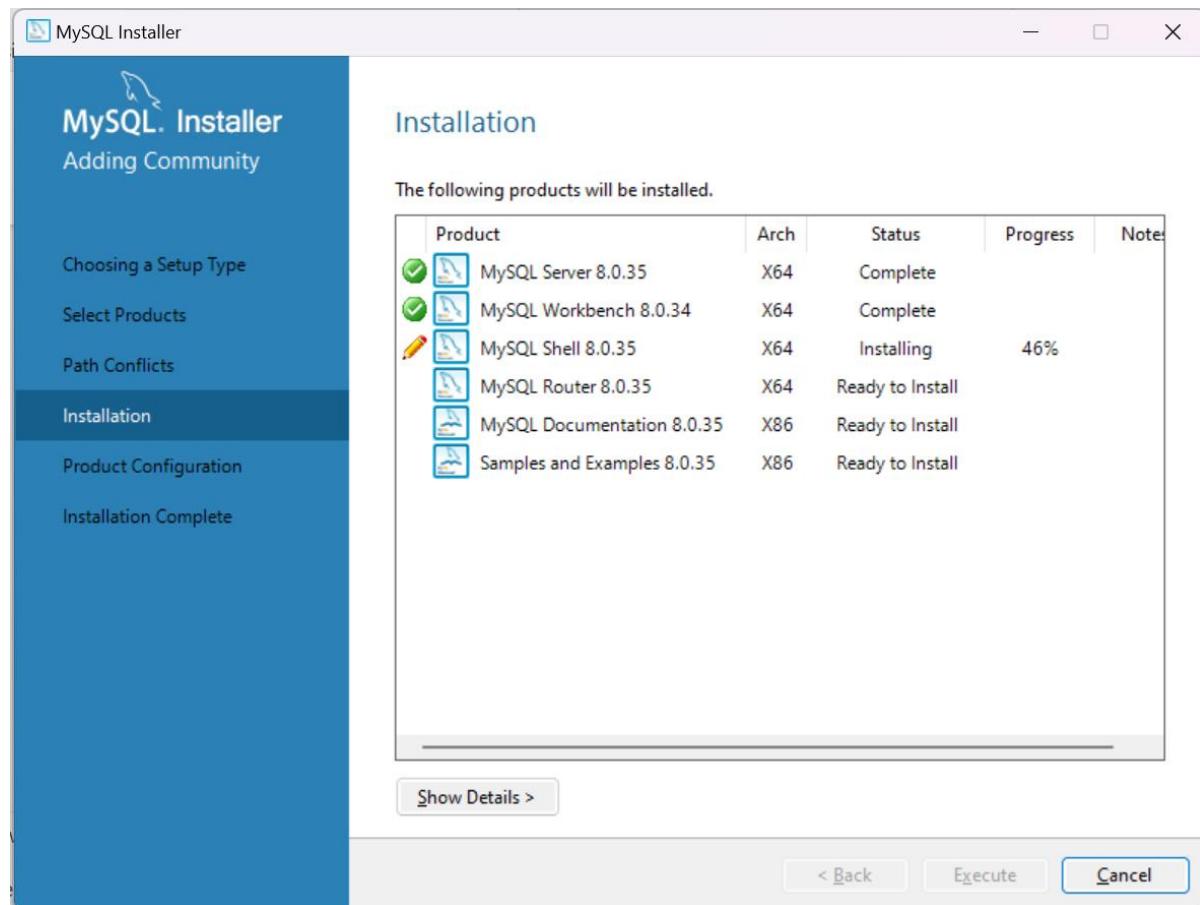


Products is selected

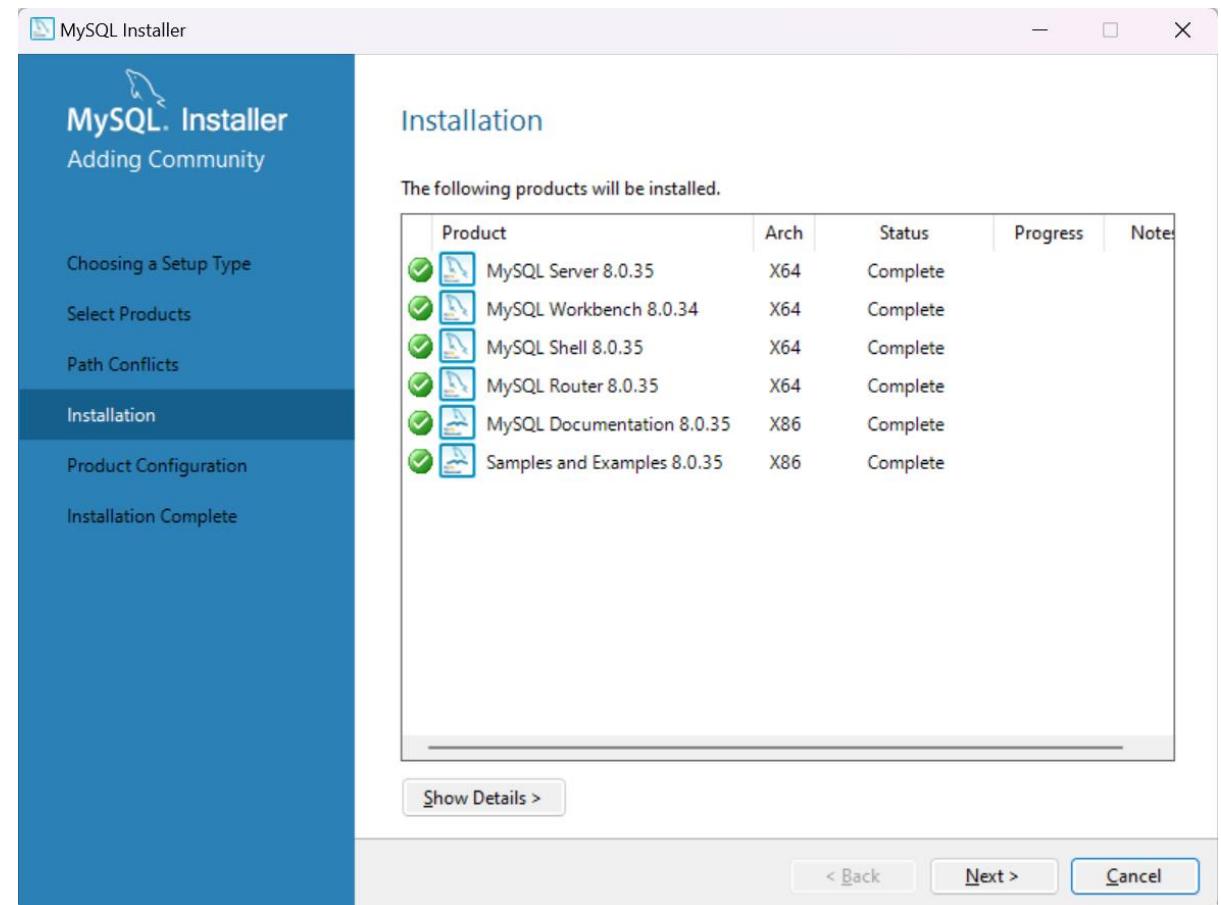
Click “Execute”

MySQL for Windows: Install

❖ Step 3:



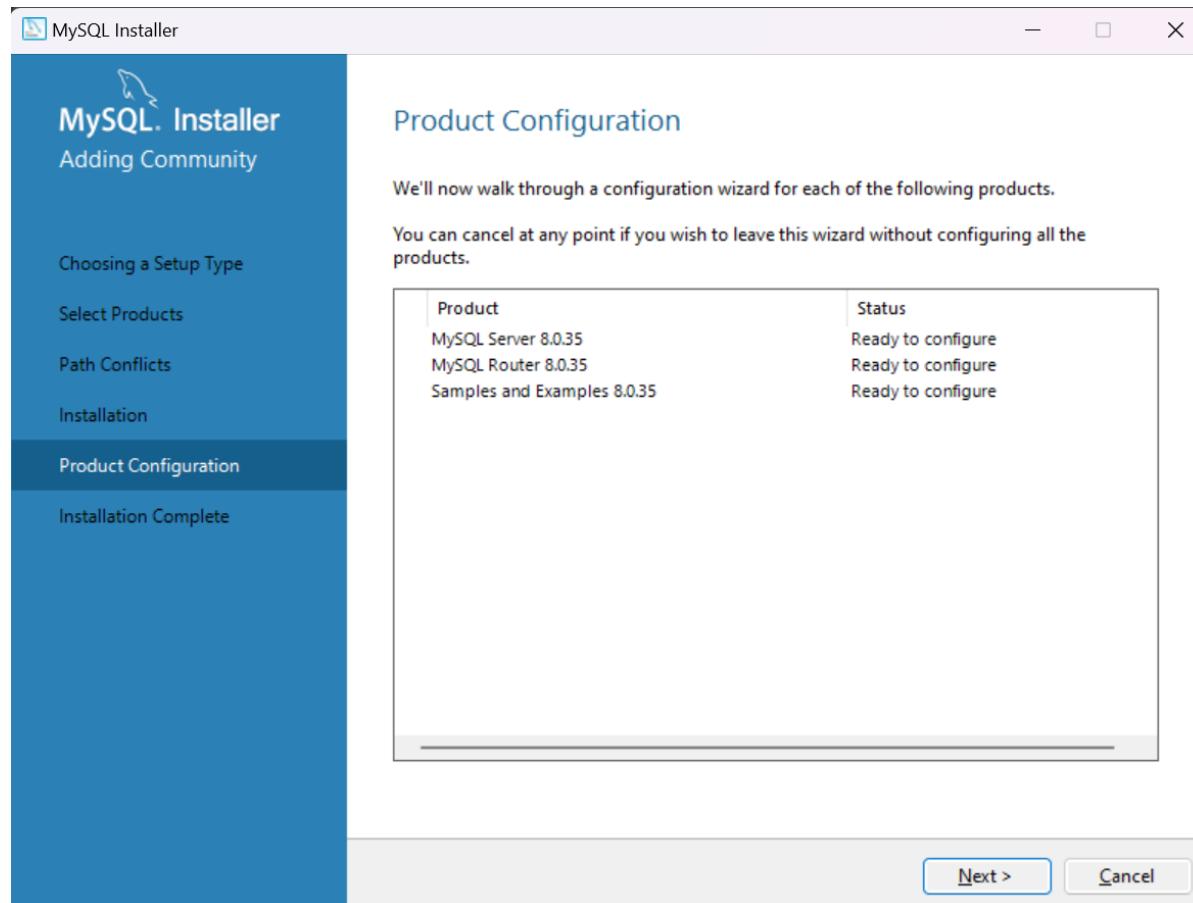
Waiting for finishing



Click "Next"

MySQL for Windows: Install

❖ Step 4-0: Configure the Server

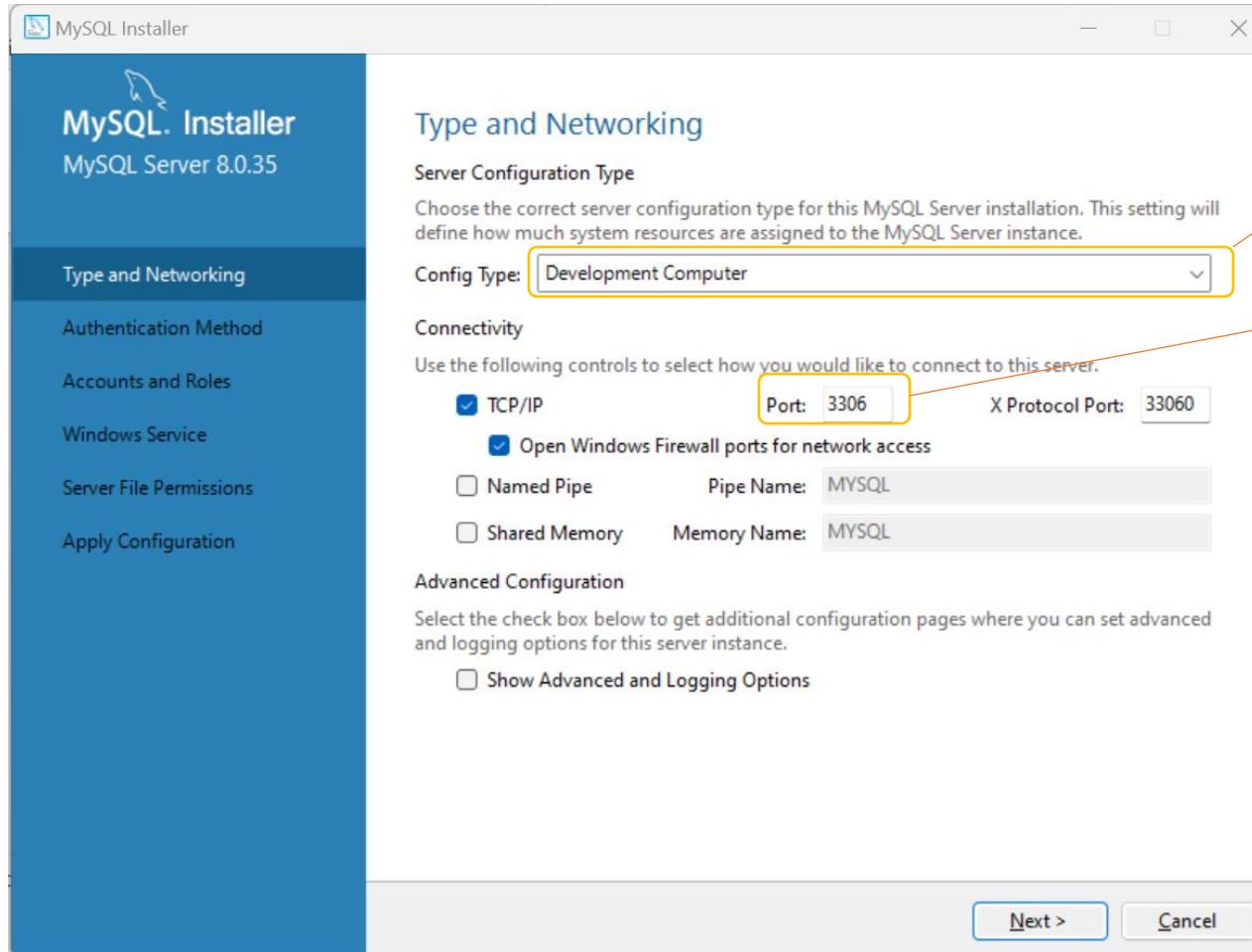


This is the most important big step

Click “Next”

MySQL for Windows: Install

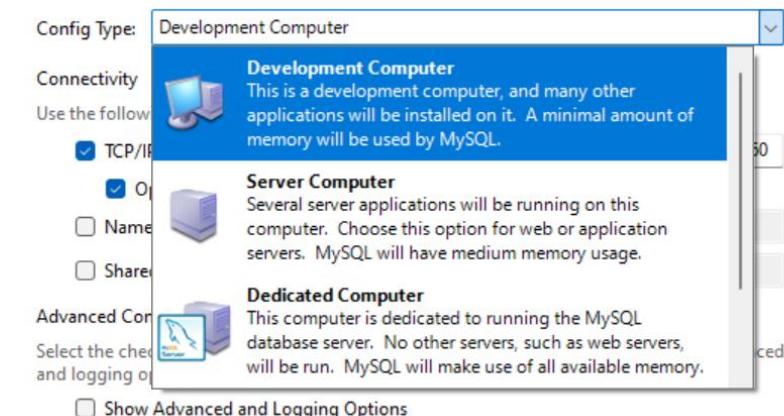
❖ Step 4-1: Configure the Network



Click "Next"

Keep “Development Computer”

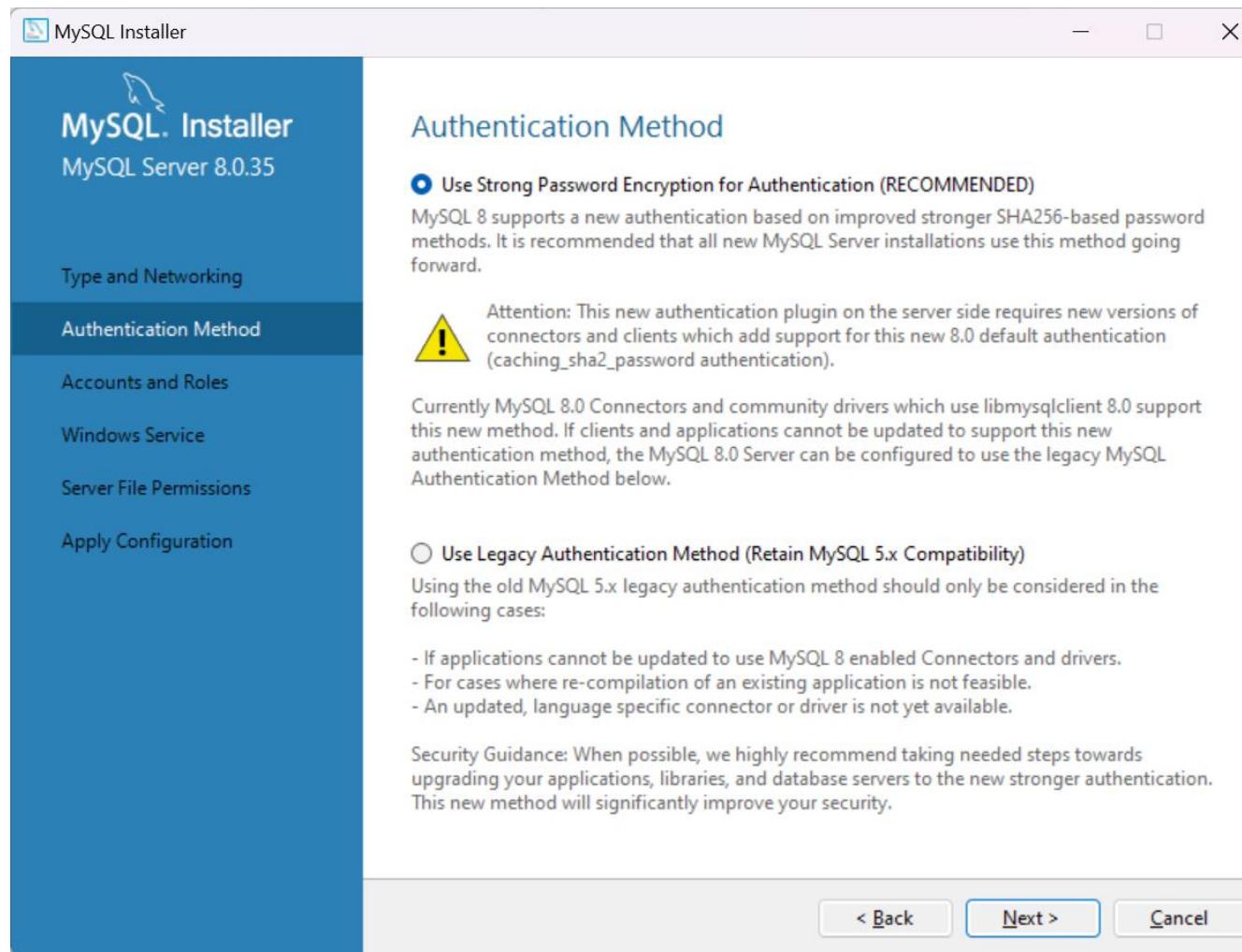
Change if you like



03 types of Computer

MySQL for Windows: Install

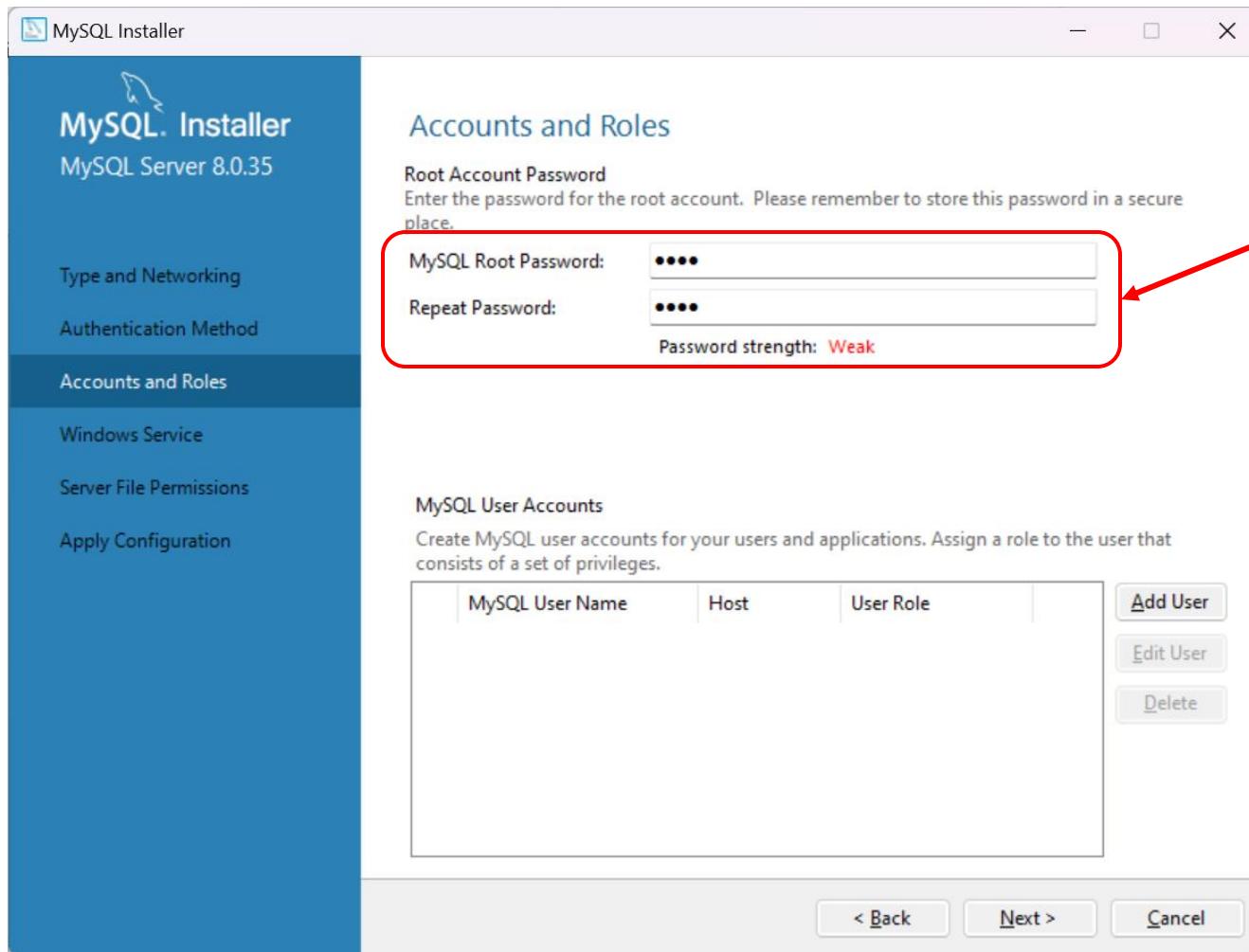
❖ Step 4-2: Authentication Method



Click “Next”

MySQL for Windows: Install

❖ Step 4-3: Create Root account



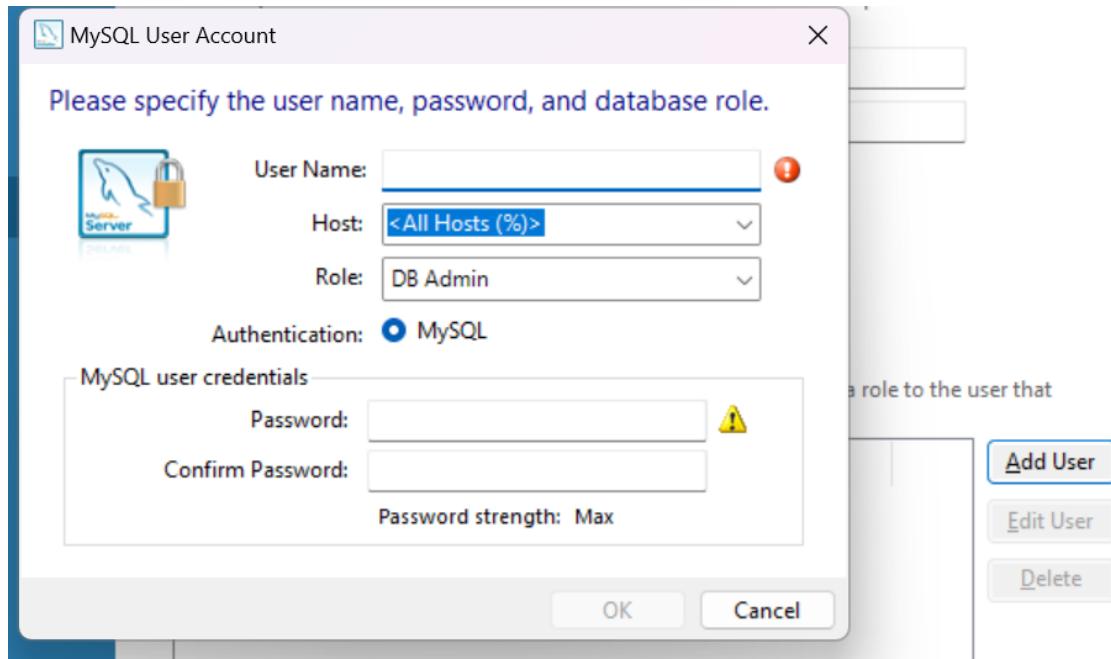
Remember your root's pw

Important step!!!

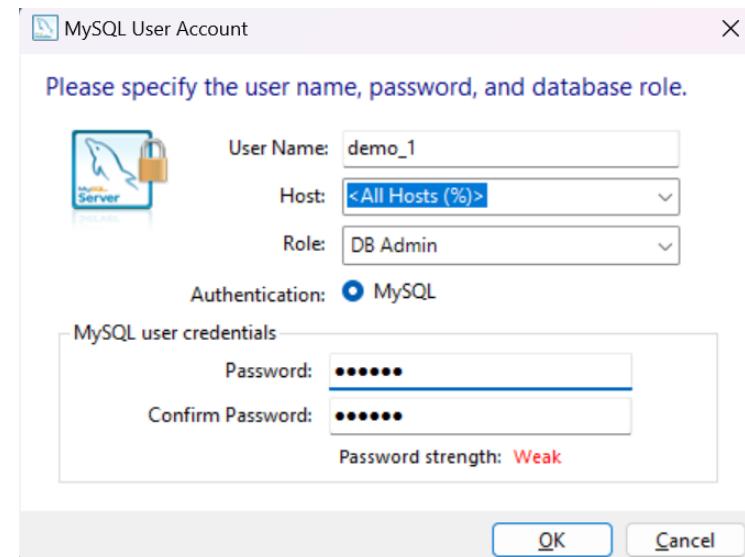
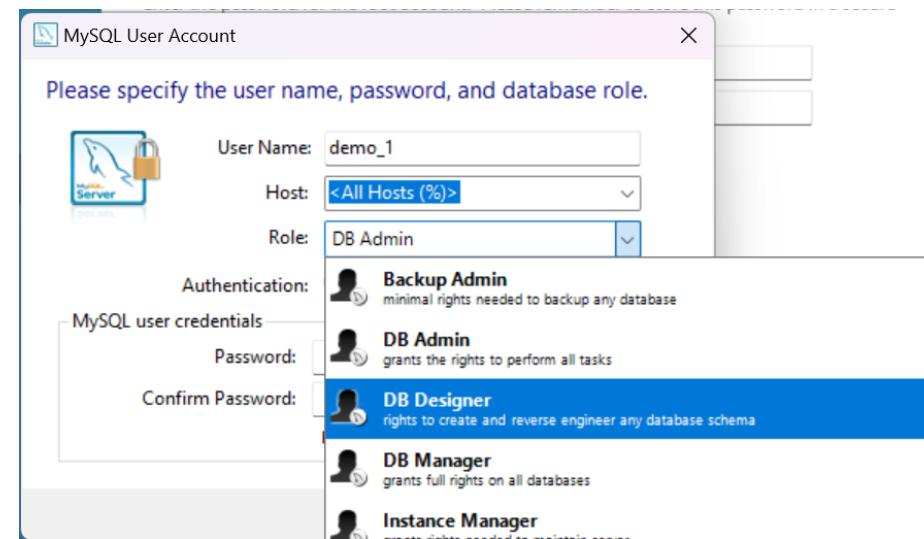
After creating root's password,
Click "Next"

MySQL for Windows: Install

❖ Step 4-3 (+1): Add new user



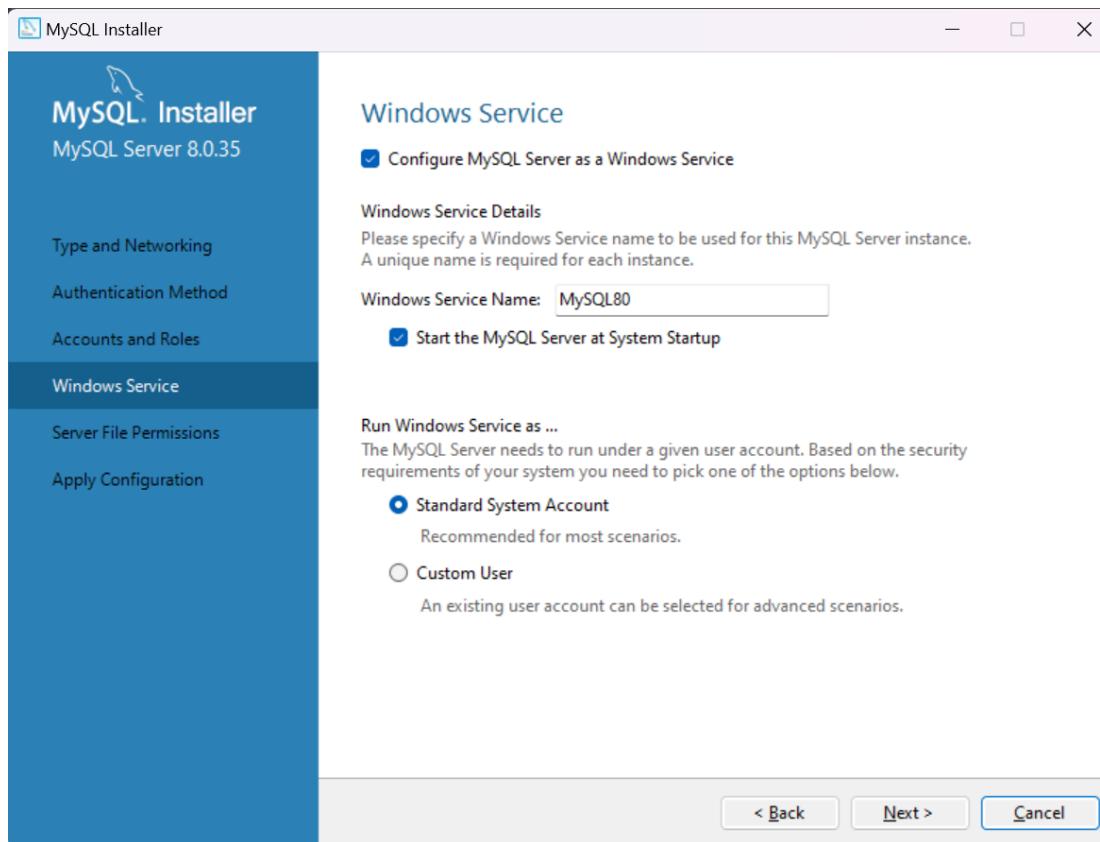
Click “Add User”



Fill Information

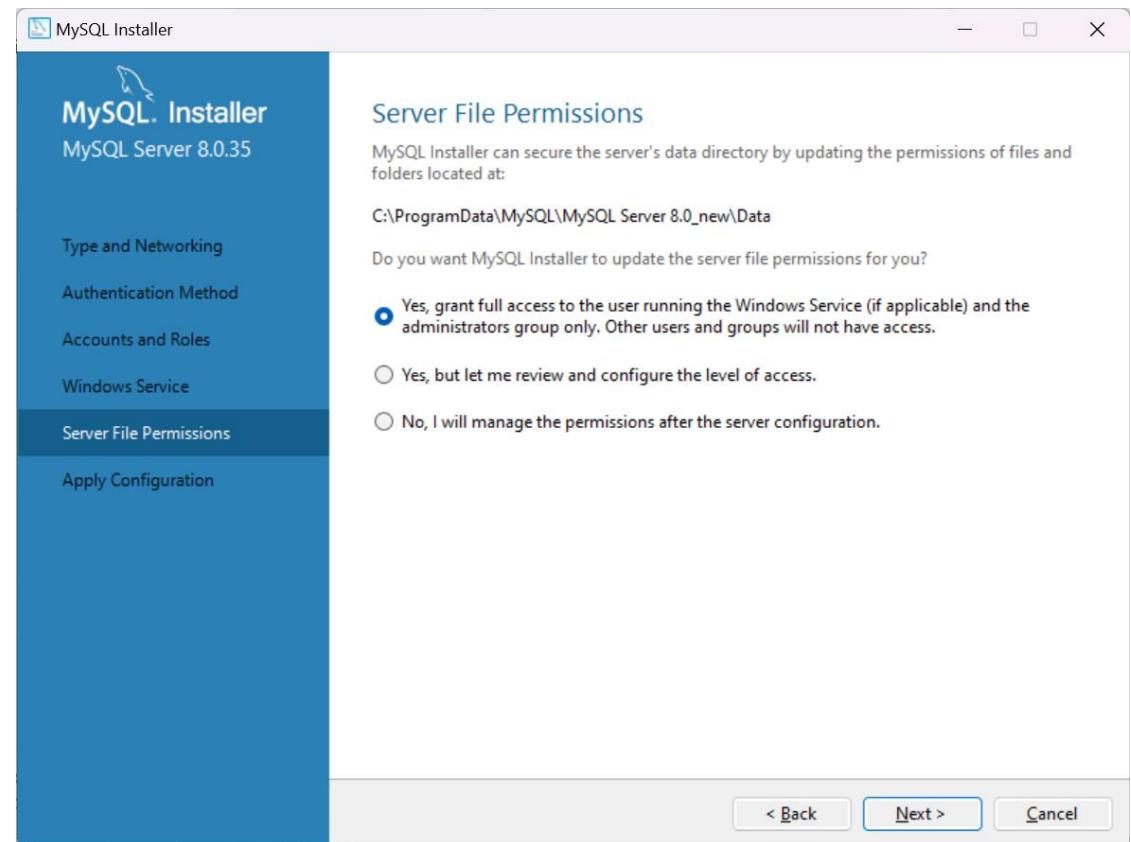
MySQL for Windows: Install

❖ Step 4-4: Windows service details



Click “Next”

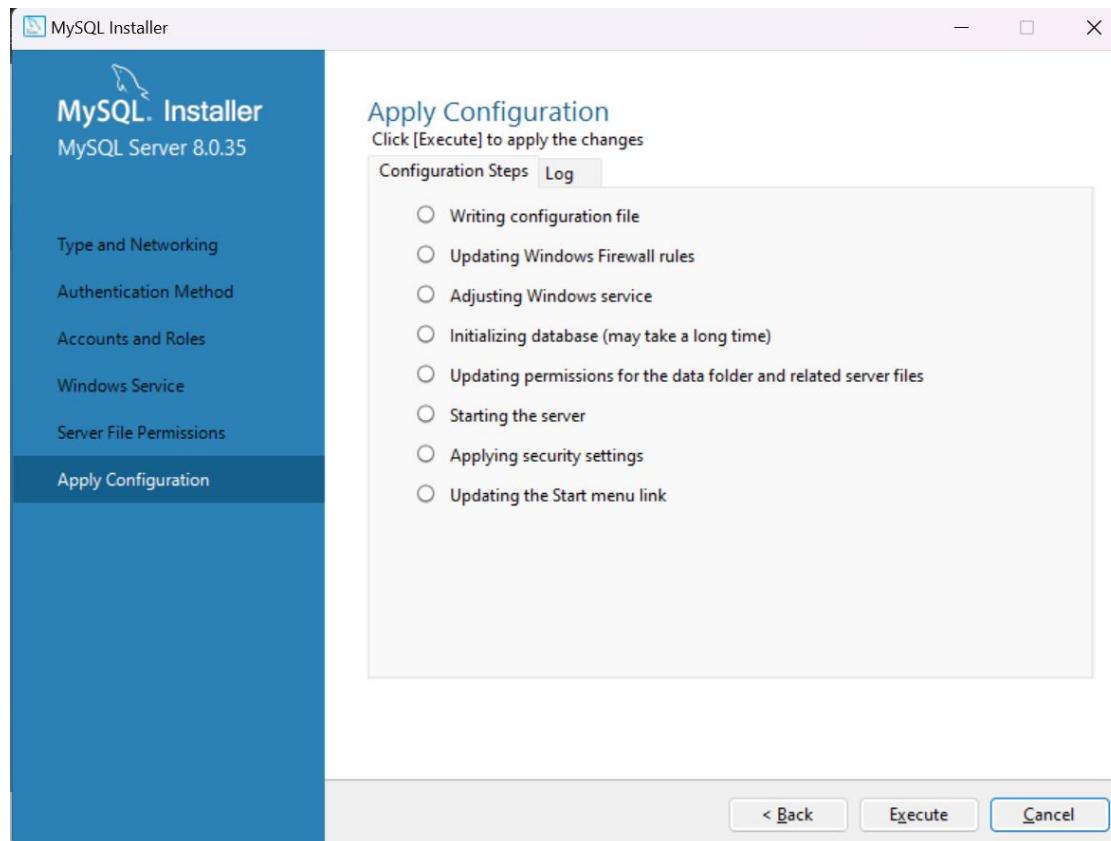
❖ Step 4-5: File permissions



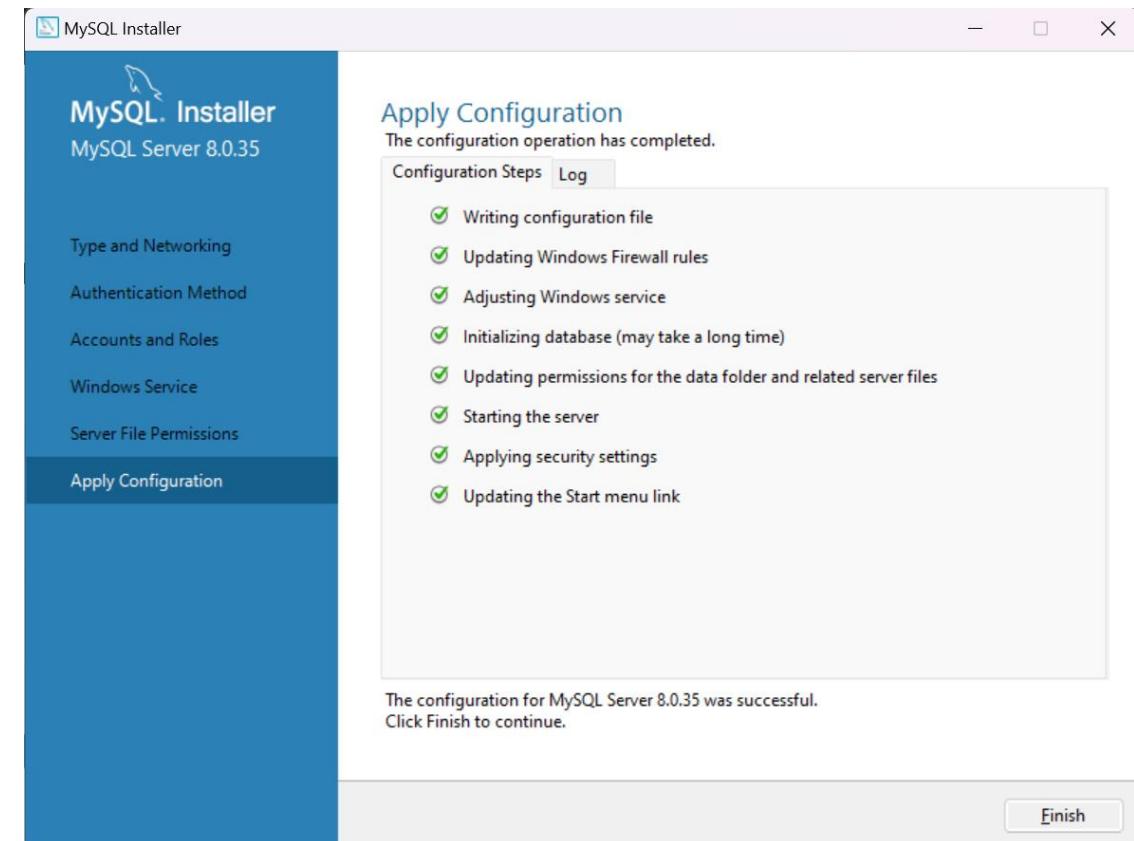
Click “Next”

MySQL for Windows: Install

❖ Step 4-6: Applying



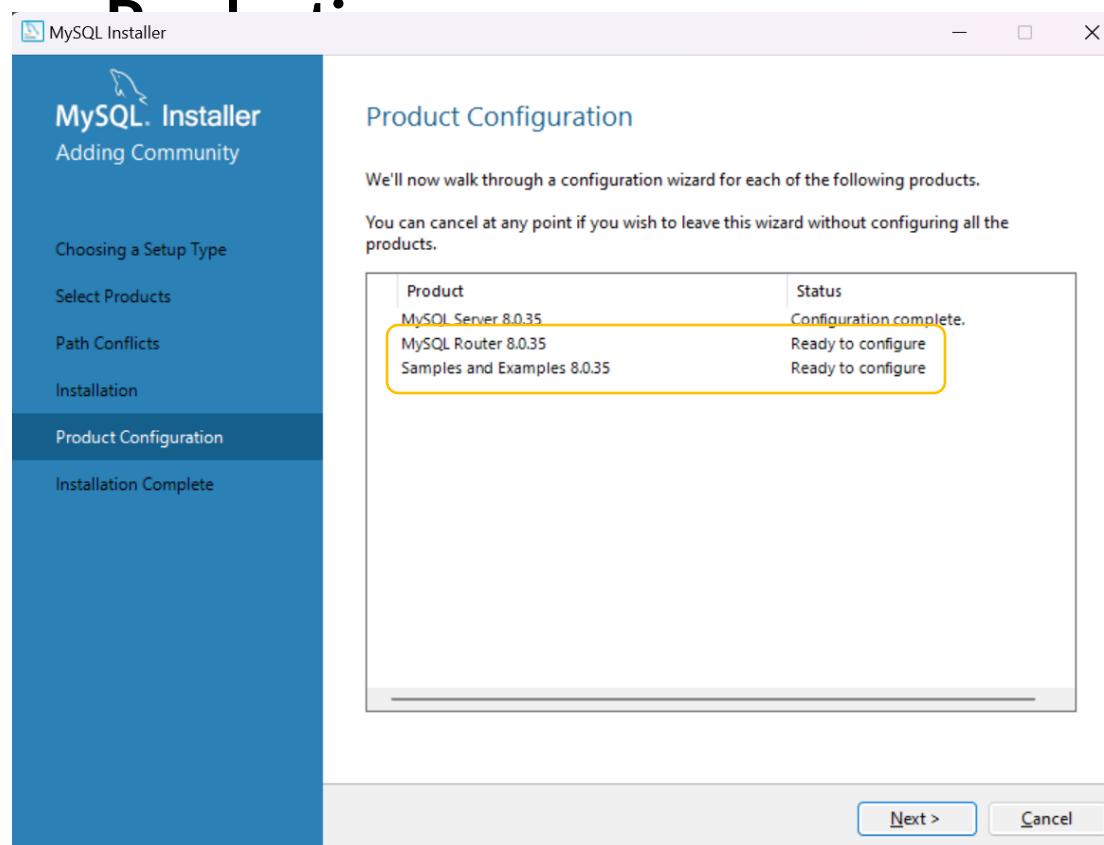
Click “Execute”



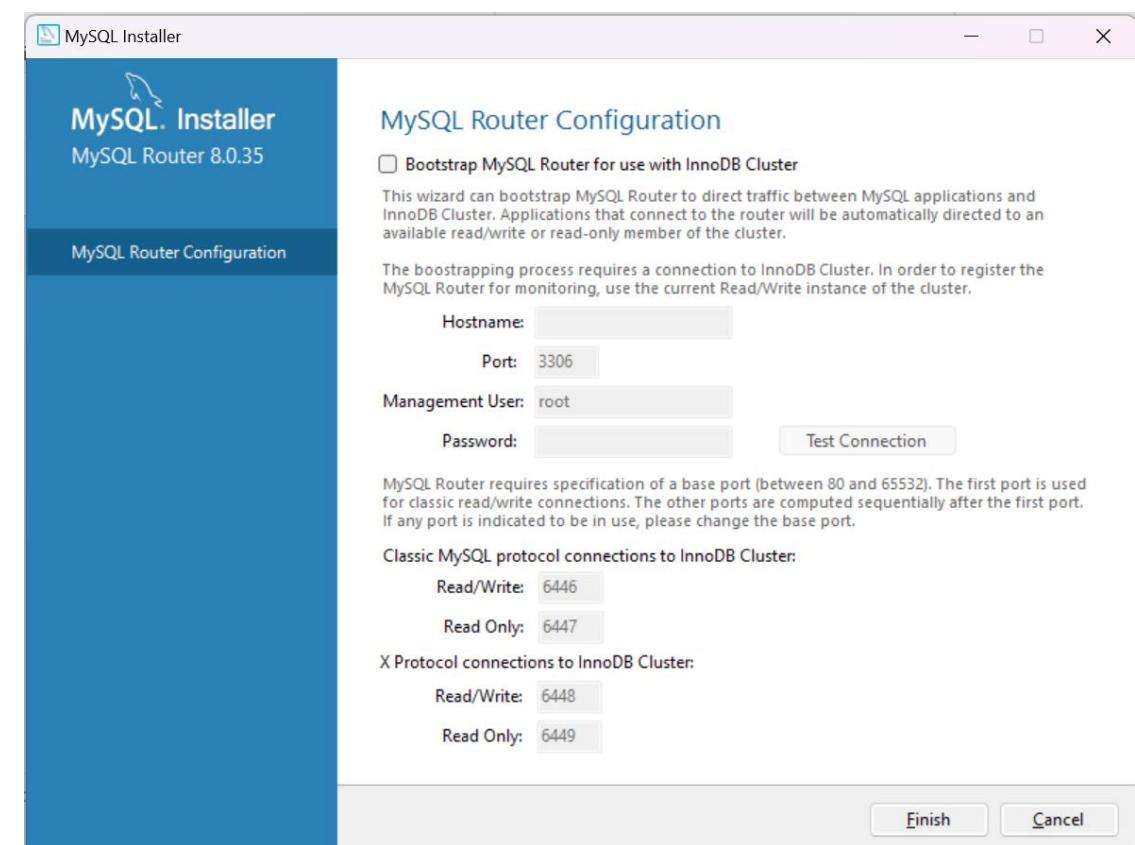
Wait till done and click “Finish”

MySQL for Windows: Install

❖ Step 4-7: Configure the MySQL Router



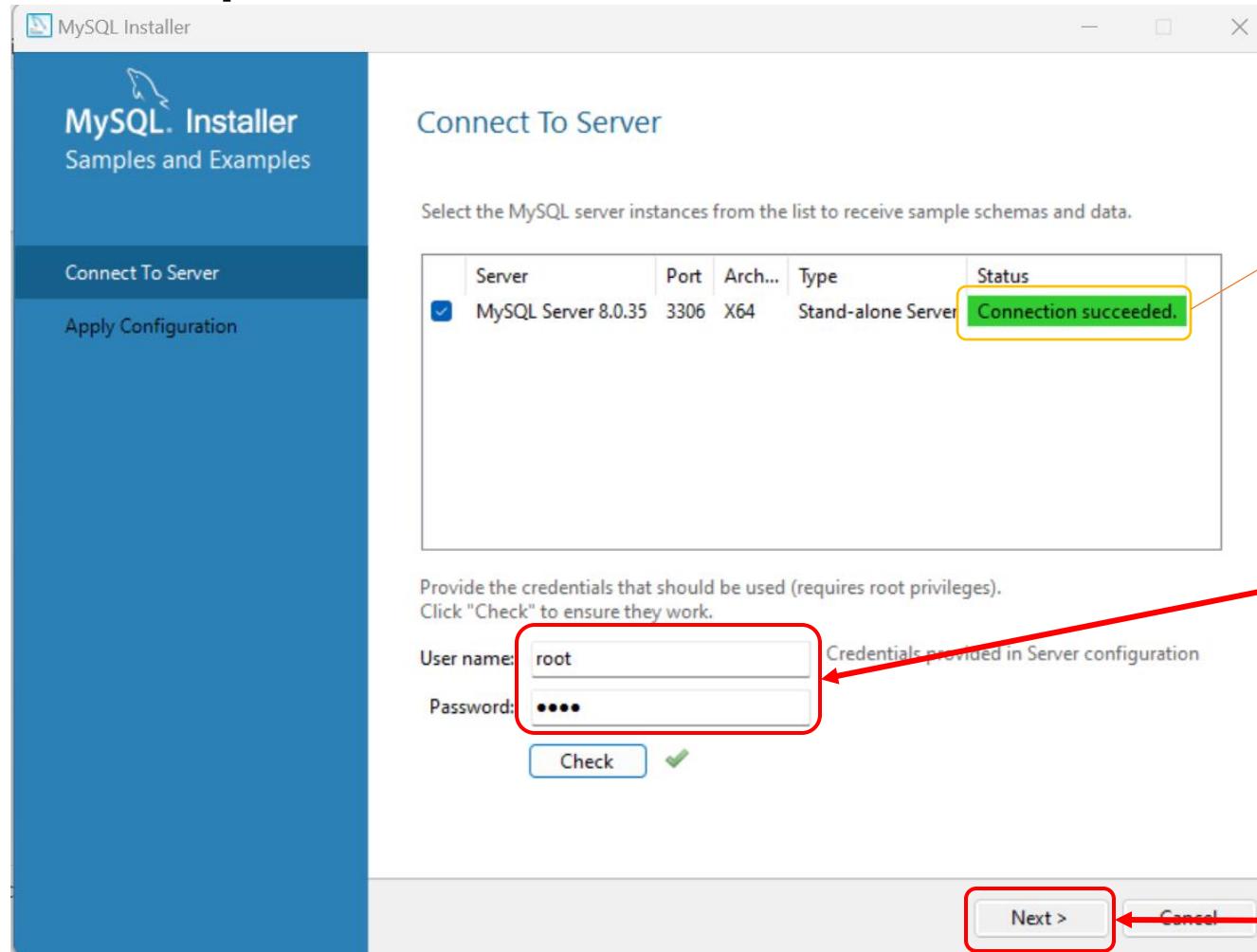
Click “Next”



Click “Finish”

MySQL for Windows: Install

❖ Step 4-8: Connect to Server



Successful connection

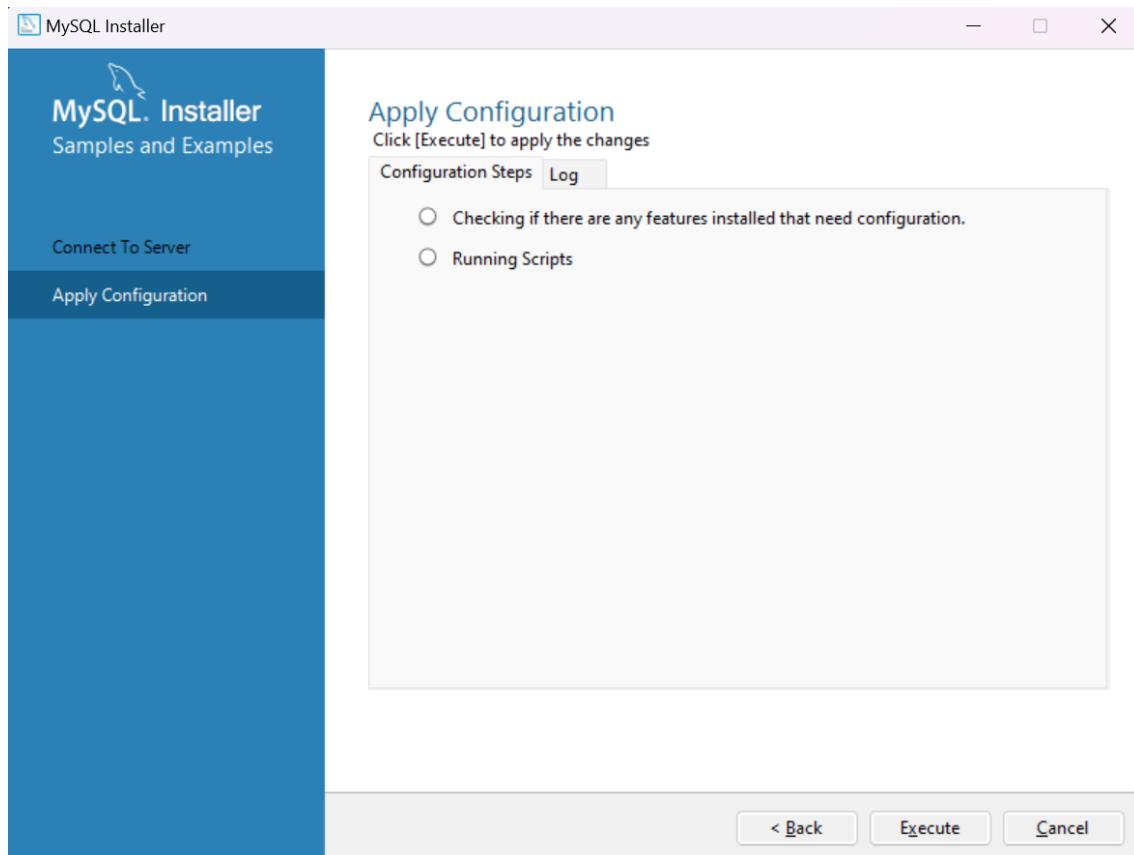
Enter your password

Click “Check” to verify the connection to the Server

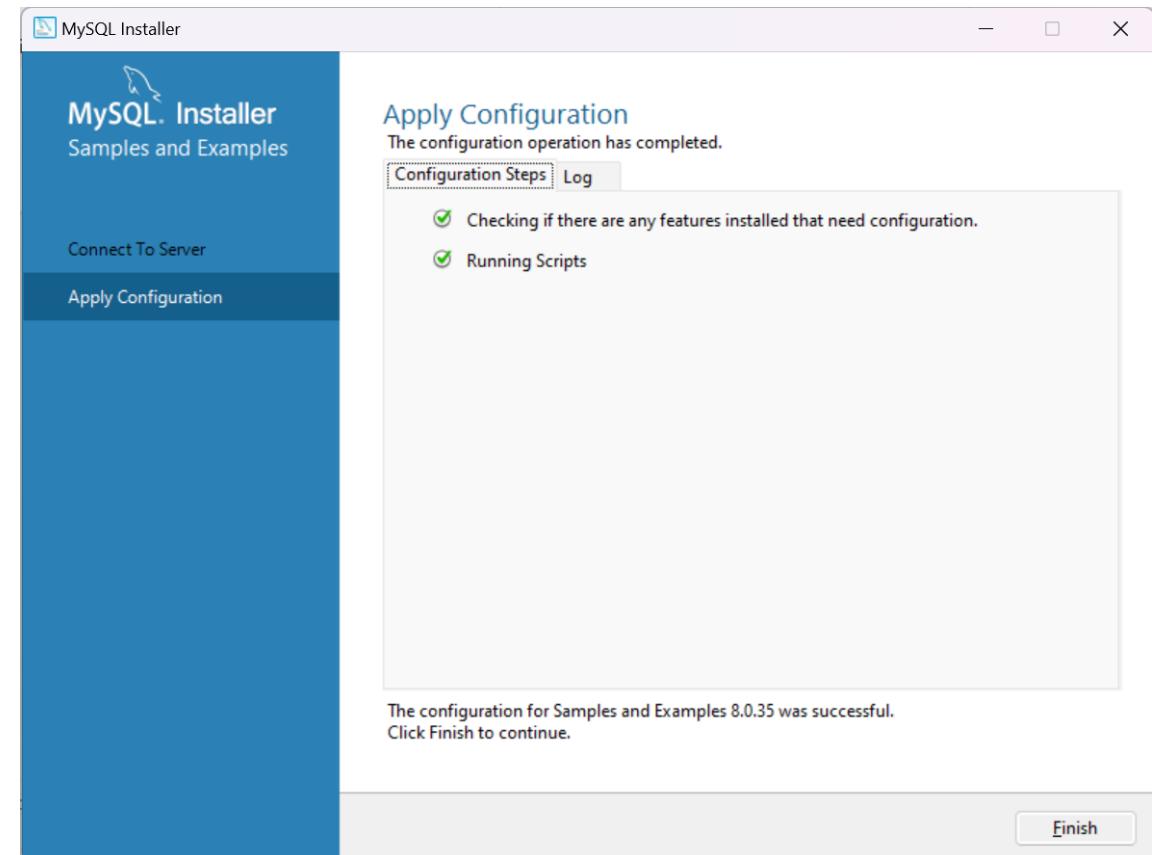
Click “Next”

MySQL for Windows: Install

❖ Step 4-9: Applying configtion



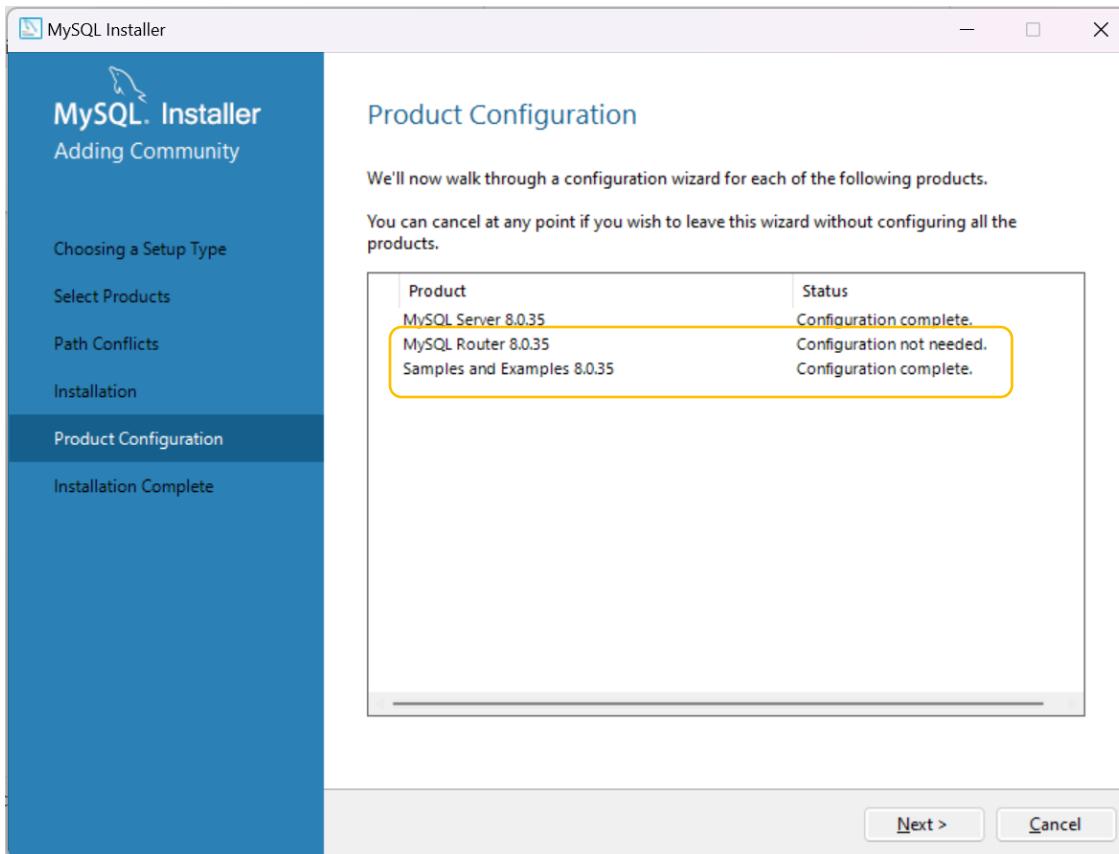
Click “Execute”



Click “Finish”

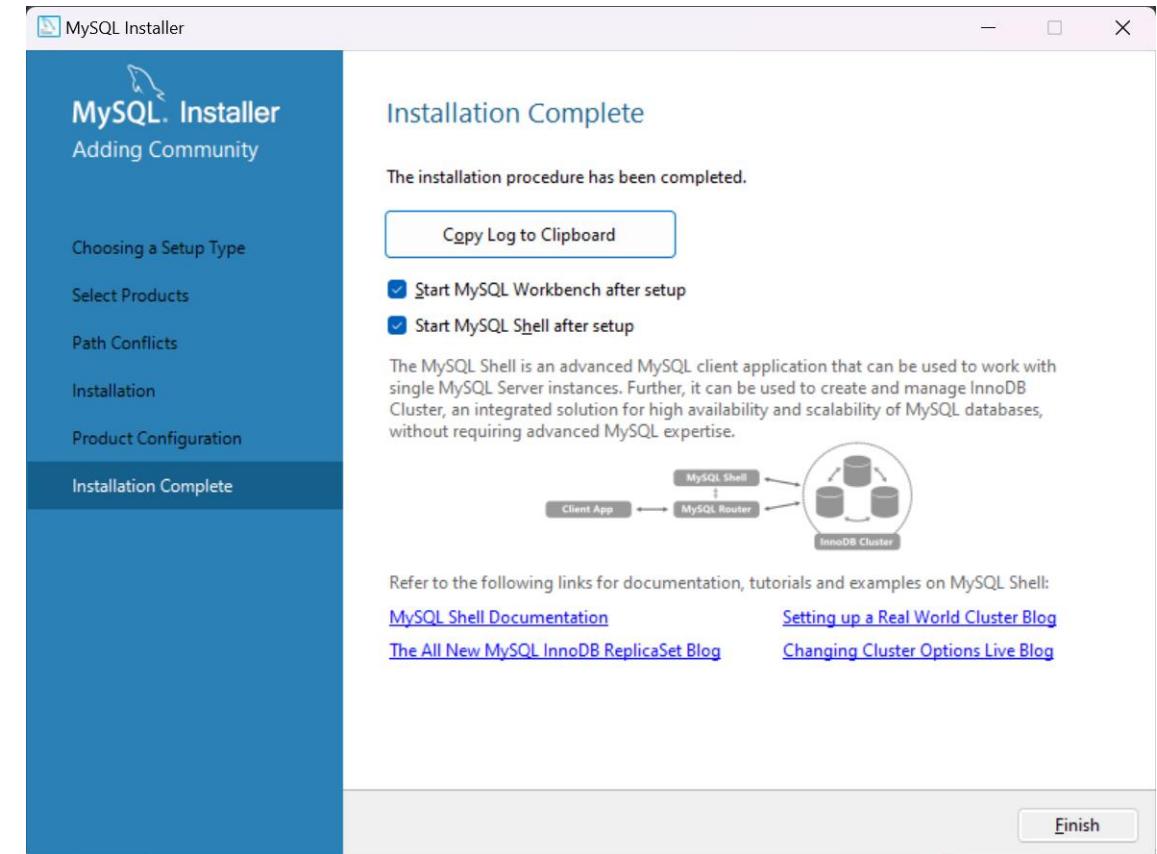
MySQL for Windows: Install

❖ Step 4: Final results



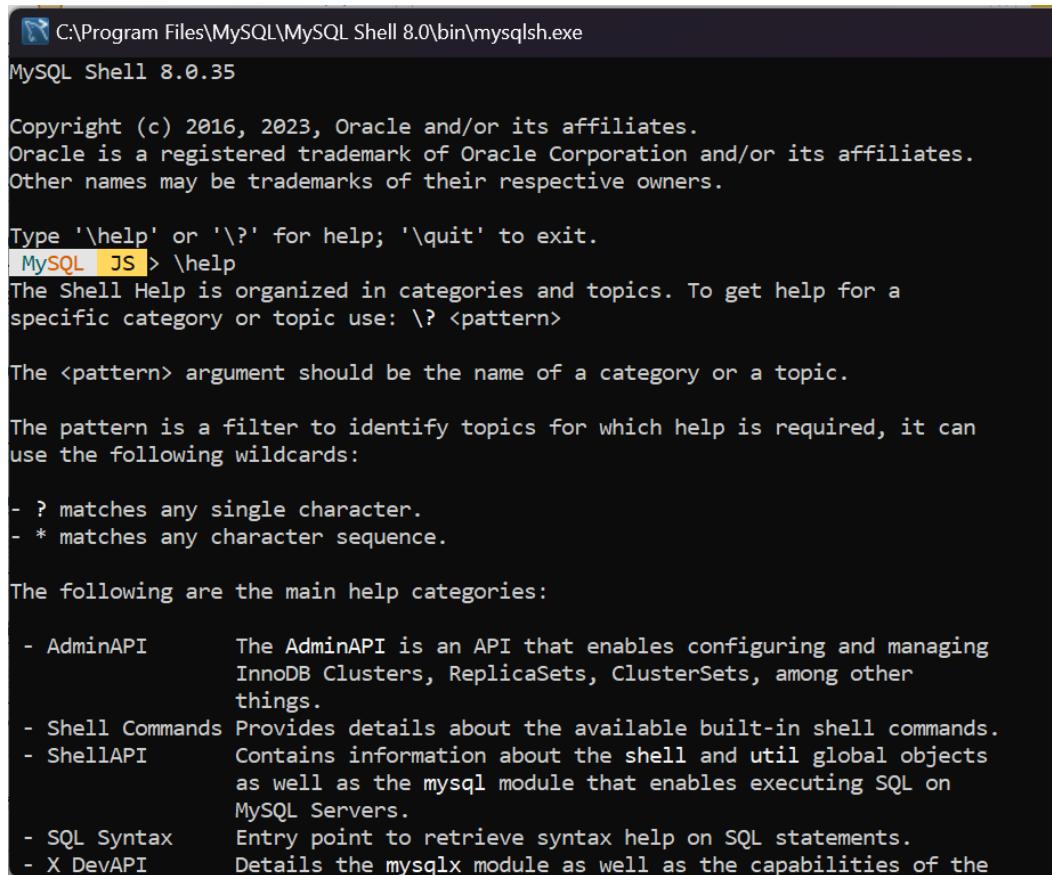
Click “Next”

❖ Step 5: Finish the process



Click “Finish”

MySQL for Windows: After Install



C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe
MySQL Shell 8.0.35

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
MySQL JS > \help
The Shell Help is organized in categories and topics. To get help for a specific category or topic use: \? <pattern>

The <pattern> argument should be the name of a category or a topic.

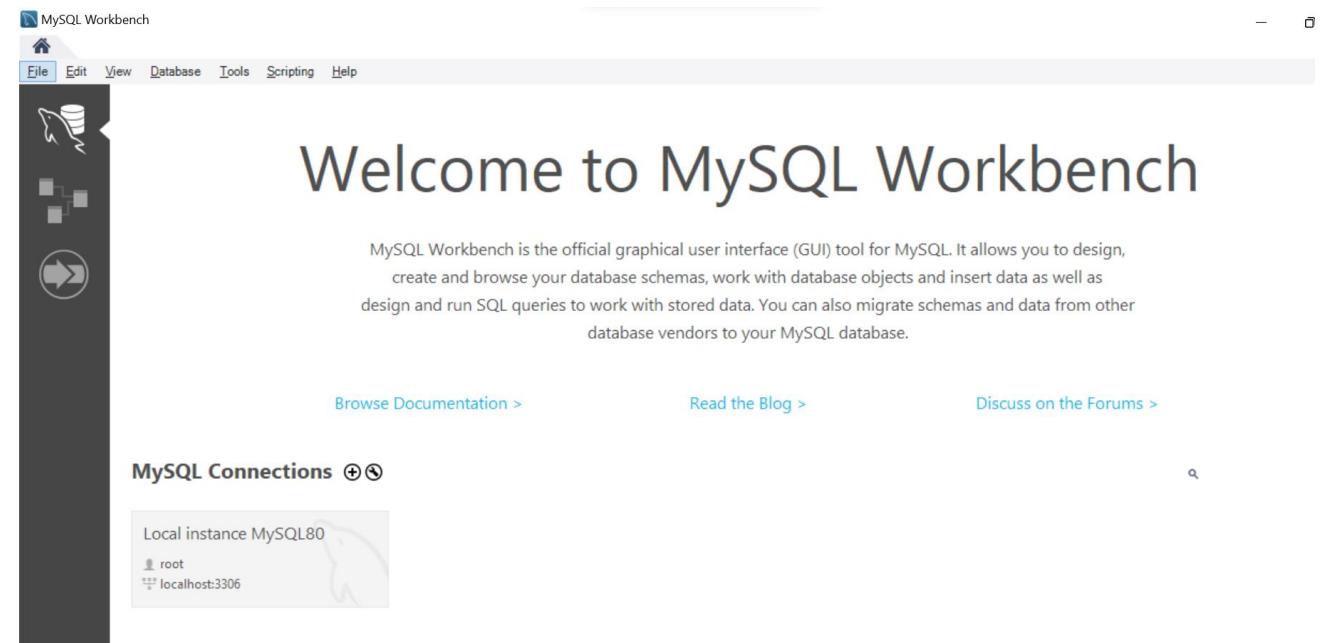
The pattern is a filter to identify topics for which help is required, it can use the following wildcards:

- ? matches any single character.
- * matches any character sequence.

The following are the main help categories:

- AdminAPI The AdminAPI is an API that enables configuring and managing InnoDB Clusters, ReplicaSets, ClusterSets, among other things.
- Shell Commands Provides details about the available built-in shell commands.
- ShellAPI Contains information about the shell and util global objects as well as the mysql module that enables executing SQL on MySQL Servers.
- SQL Syntax Entry point to retrieve syntax help on SQL statements.
- X DevAPI Details the mysqlx module as well as the capabilities of the

MySQL Shell



MySQL Workbench

You now have:

- ❖ MySQL Server:

- Admin account “root” and its password to access
 - Connect via TCP/IP through port 3306 (default port)

- ❖ MySQL Client:

- MySQL Workbench to connect and interactive with databases via GUI.
 - MySQL Shell to connect to Server by command line.

Using MySQL Workbench



Log in MySQL Workbench

The screenshot shows the MySQL Workbench interface. On the left is a dark sidebar with icons for Home, File, Edit, View, Database, Tools, Scripting, and Help. The main area has a large "Welcome to MySQL Workbench" title. Below it is a descriptive paragraph about the tool's features. At the bottom left, there are links to "Browse Documentation >" and "Read the Blog >". The "MySQL Connections" section lists a single entry: "Local instance MySQL80" with details "root" and "localhost:3306". A red box highlights this entry. On the right, a modal dialog titled "Connect to MySQL Server" asks for a password. It shows the MySQL Workbench logo and lists the service as "Mysql@localhost:3306", the user as "root", and the password field containing "****". There is also a checkbox for "Save password in vault".

Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Browse Documentation >

Read the Blog >

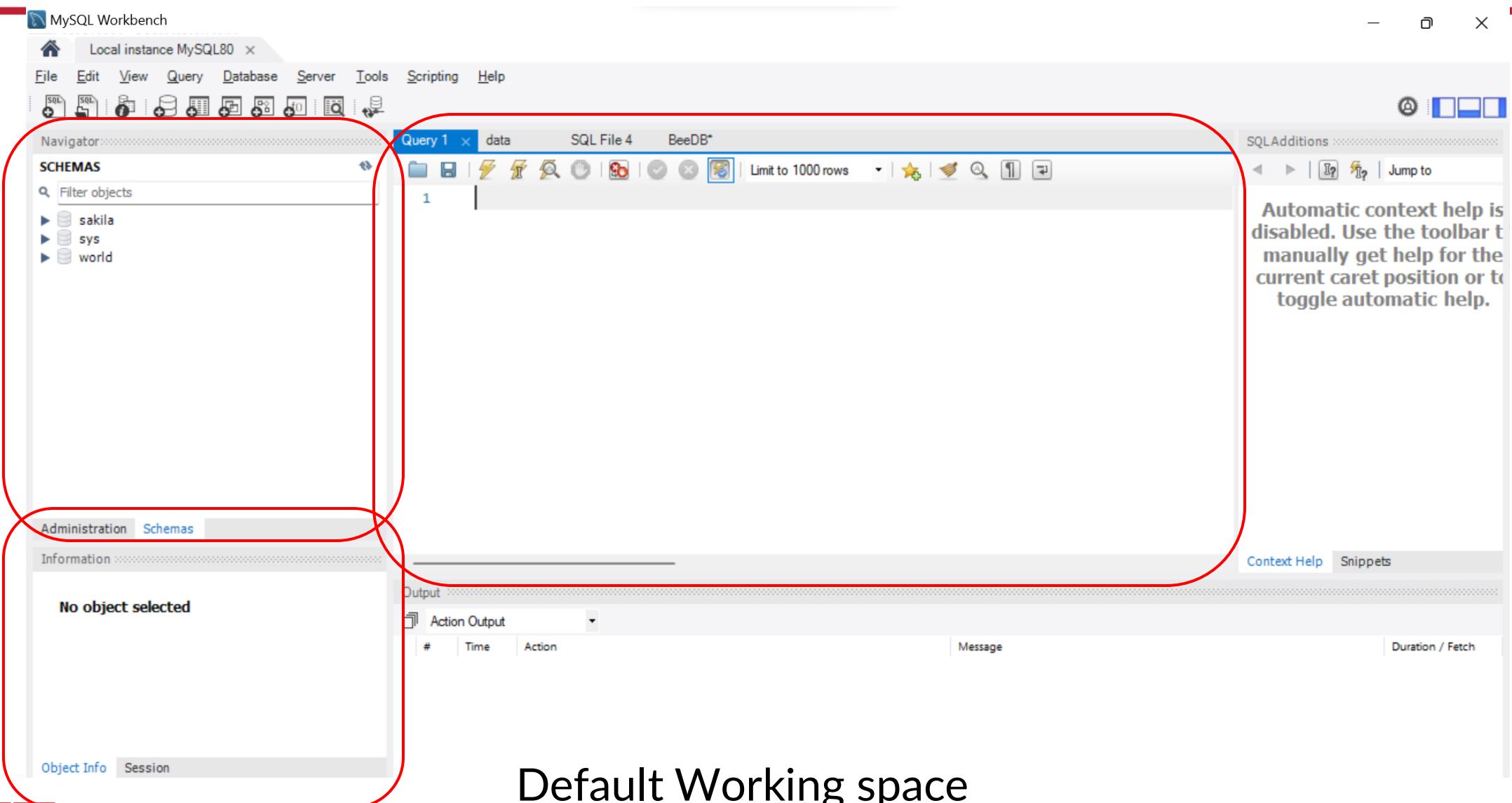
MySQL Connections +
Local instance MySQL80
root
localhost:3306

Please enter password for the following service:
Service: Mysql@localhost:3306
User: root
Password: ****
 Save password in vault
OK Cancel

Click at Local instance

Enter your pw

Log in MySQL Workbench



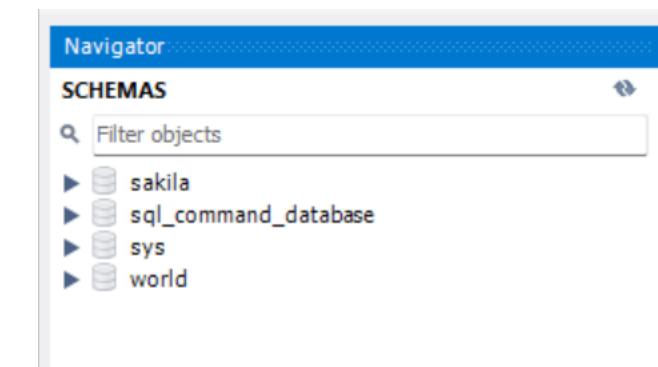
Using SQL command

❖ Type this command to **Create a new database:**

□ **CREATE DATABASE SQL_command_database;**

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the connection is set to "Local instance MySQL80". The main window has tabs for "query 1", "data", "SQL File 4", and "BeeDB*". The "query 1" tab contains two lines of SQL code:
1 Create new database by SQL command that name "SQL_command_database"*/
2 CREATE DATABASE SQL_command_database;The second line is highlighted with a blue selection bar. A red arrow points from the text "1. Click Lightning icon to run" to the lightning bolt icon in the toolbar above the query editor. Another red arrow points from the text "3. Click Refresh icon to see" to the refresh icon in the toolbar.

3. Click Refresh icon to see



1. Click Lightning icon to run

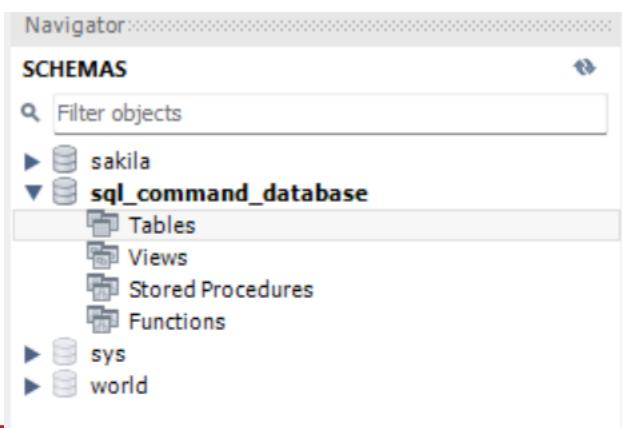
The screenshot shows the "Output" window in MySQL Workbench. It displays the results of the executed SQL command. The table has columns: #, Time, Action, Message, and Duration / Fetch. One row is shown:
1 22:48:33 CREATE DATABASE SQL_command_database
Message 1 row(s) affected
Duration / Fetch 0.000 secA red arrow points from the text "2. Output show up at Output window" to the "Output" window title bar.

2. Output show up at Output window

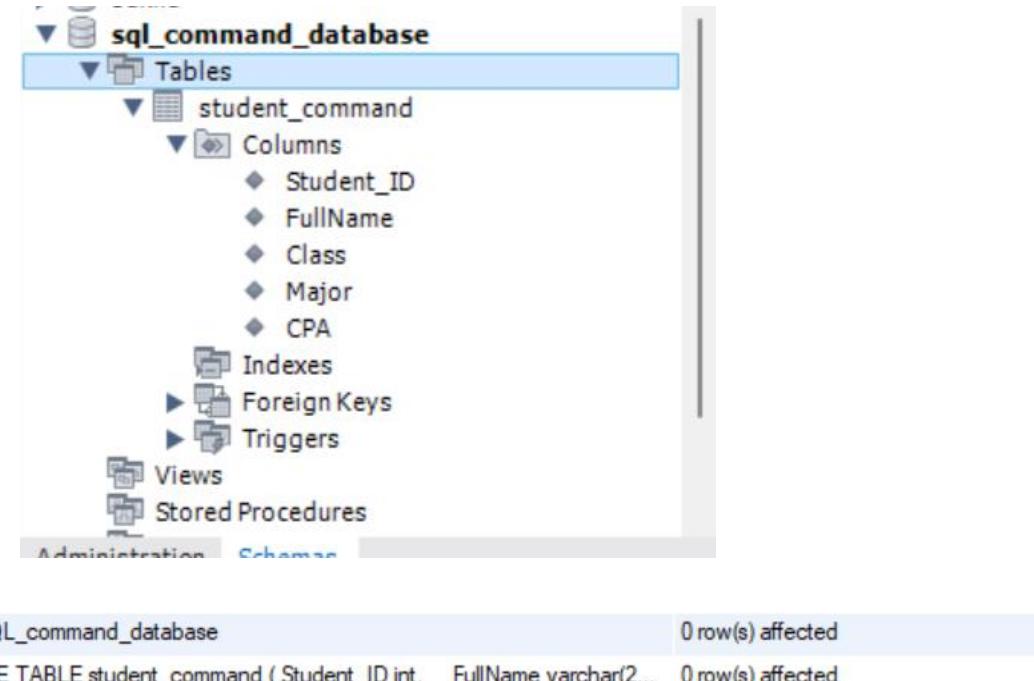
Using SQL command

❖ Type this command to **Create a new table:**

- ❑ USE SQL_command_database;
- ❑ CREATE TABLE student_command (
 Student_ID int,
 FullName varchar(255),
 Class varchar(255),
 Major varchar(255),
 CPA float);



Before run



Time	Command	Rows Affected
10 23:37:06	USE SQL_command_database	0 row(s) affected
11 23:37:06	CREATE TABLE student_command (Student_ID int, FullName varchar(255), Class varchar(255), Major varchar(255), CPA float);	0 row(s) affected

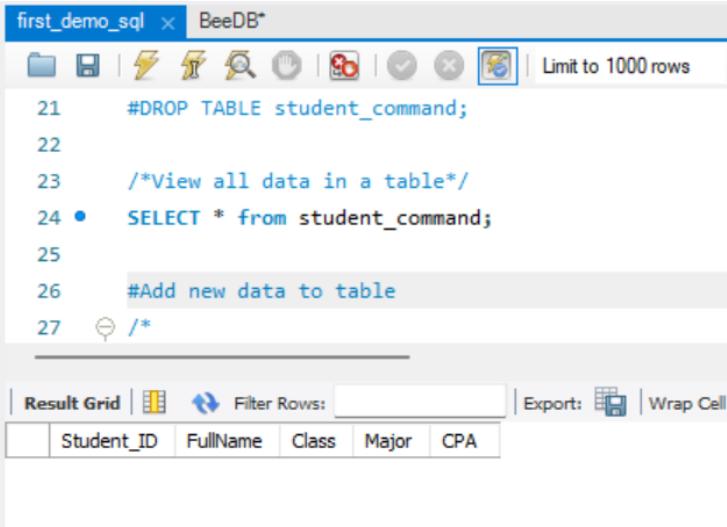
After run

Using SQL command

❖ Type this command to Insert data into a Table:

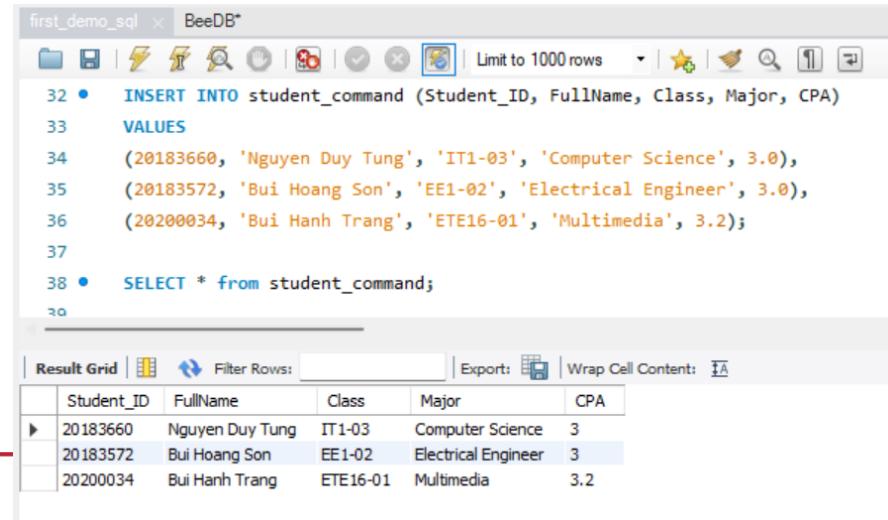
- USE SQL_command_database;
- INSERT INTO student_command (Student_ID, FullName, Class, Major, CPA)
VALUES
(20183660, 'Nguyen Duy Tung', 'IT1-03', 'Computer Science', 3.0),
(20183572, 'Bui Hoang Son', 'EE1-02', 'Electrical Engineer', 3.0),
(20200034, 'Bui Hanh Trang', 'ETE16-01', 'Multimedia', 3.2);

Before run



```
first_demo_sql x BeeDB*
21 #DROP TABLE student_command;
22
23 /*View all data in a table*/
24 • SELECT * from student_command;
25
26 #Add new data to table
27 */
```

After run

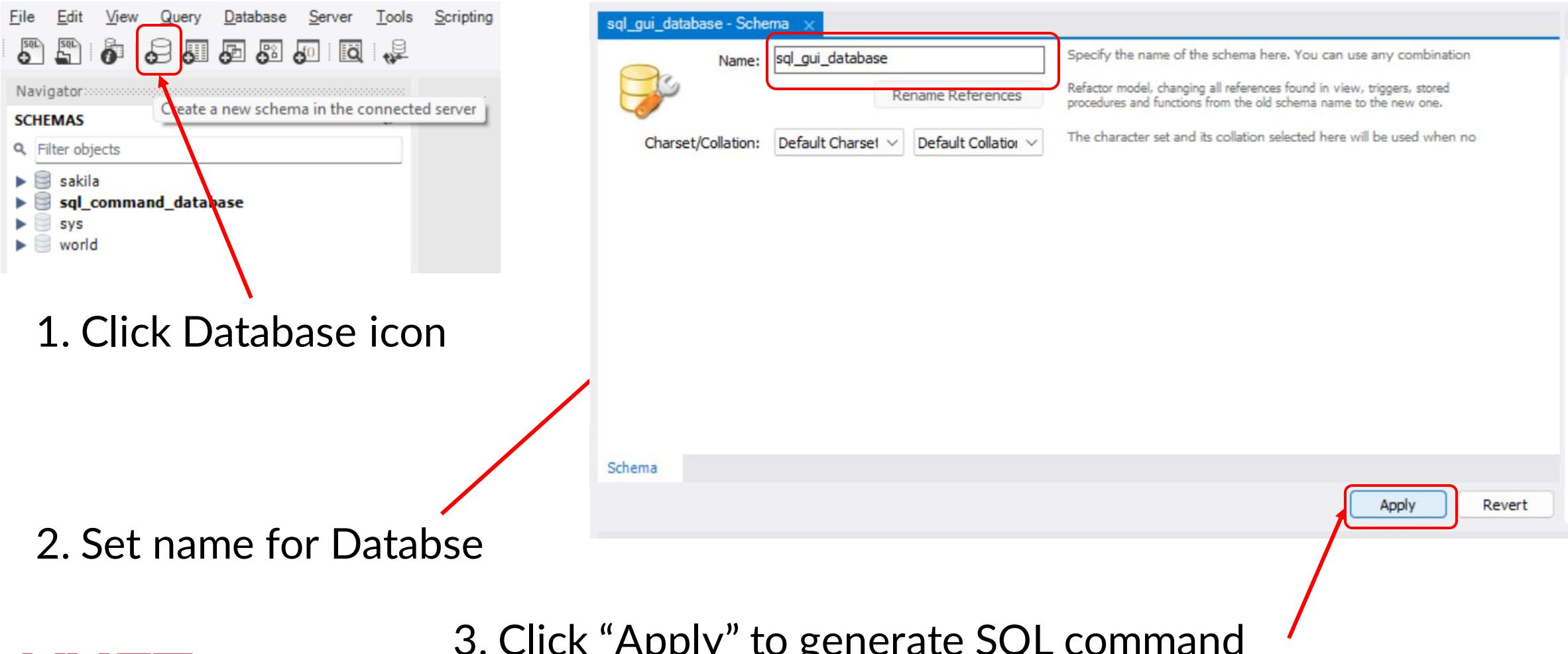


```
first_demo_sql x BeeDB*
32 • INSERT INTO student_command (Student_ID, FullName, Class, Major, CPA)
33 VALUES
34 (20183660, 'Nguyen Duy Tung', 'IT1-03', 'Computer Science', 3.0),
35 (20183572, 'Bui Hoang Son', 'EE1-02', 'Electrical Engineer', 3.0),
36 (20200034, 'Bui Hanh Trang', 'ETE16-01', 'Multimedia', 3.2);
37
38 • SELECT * from student_command;
39
```

Student_ID	FullName	Class	Major	CPA
20183660	Nguyen Duy Tung	IT1-03	Computer Science	3
20183572	Bui Hoang Son	EE1-02	Electrical Engineer	3
20200034	Bui Hanh Trang	ETE16-01	Multimedia	3.2

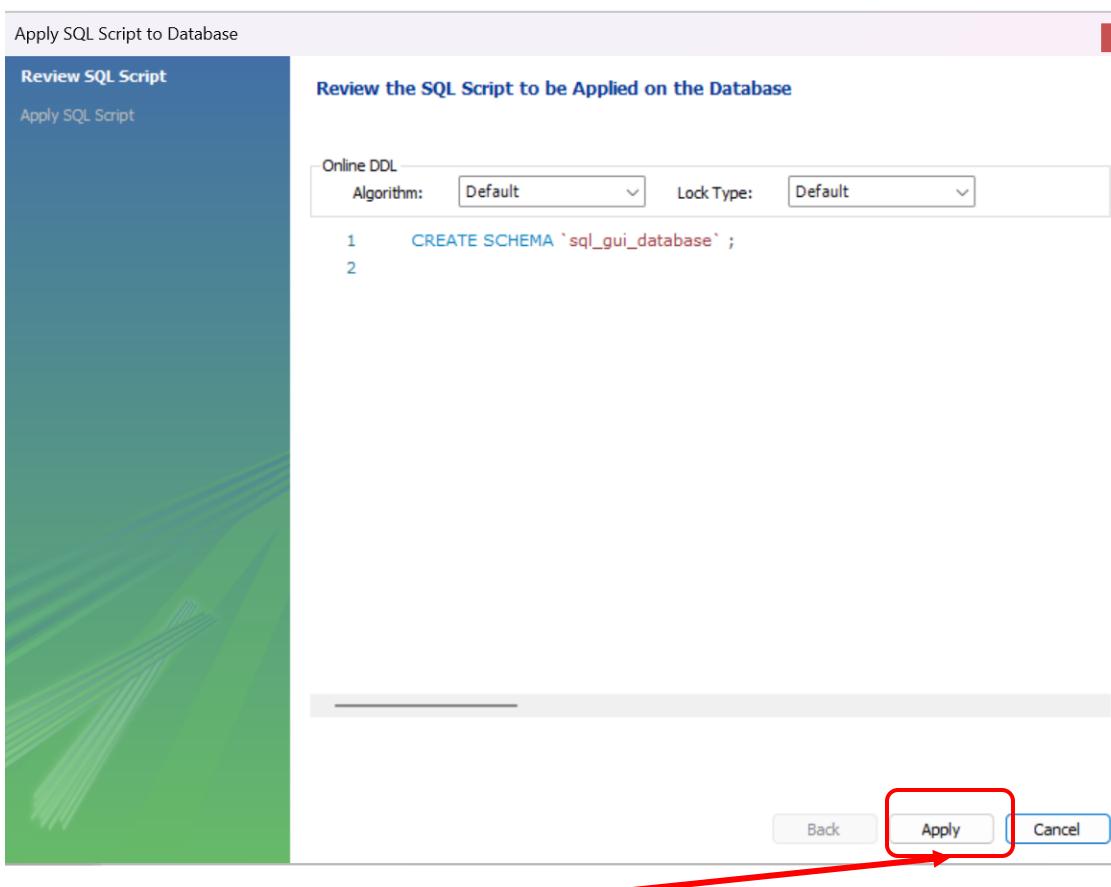
Using GUI

❖ Create a new database:

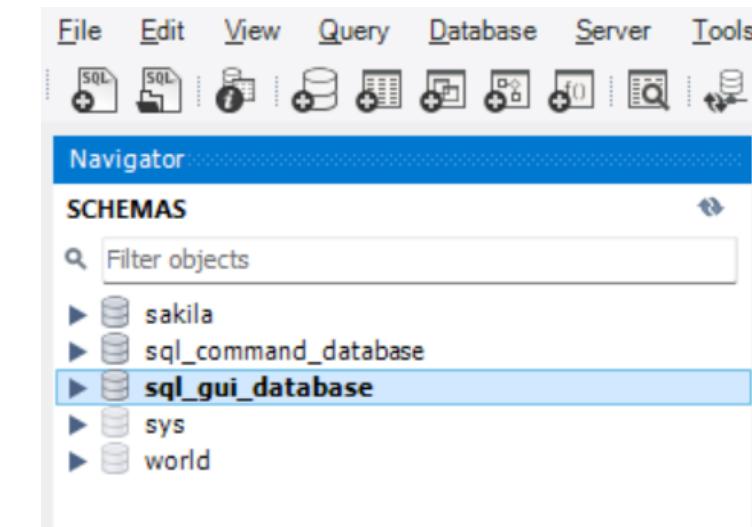


Using GUI

❖ Create a new database:



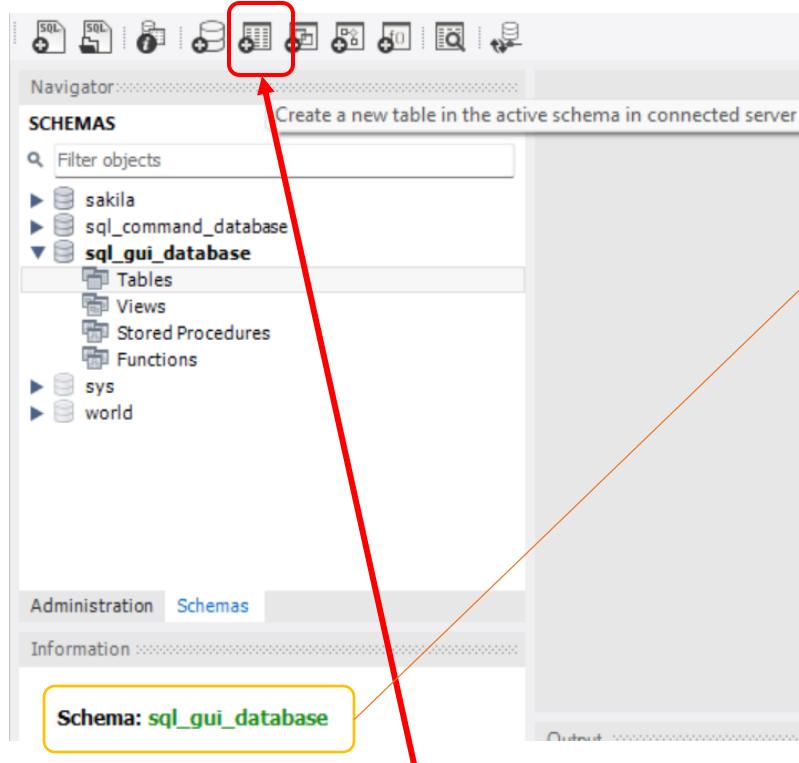
4. Click “Apply” to execute



New created Database

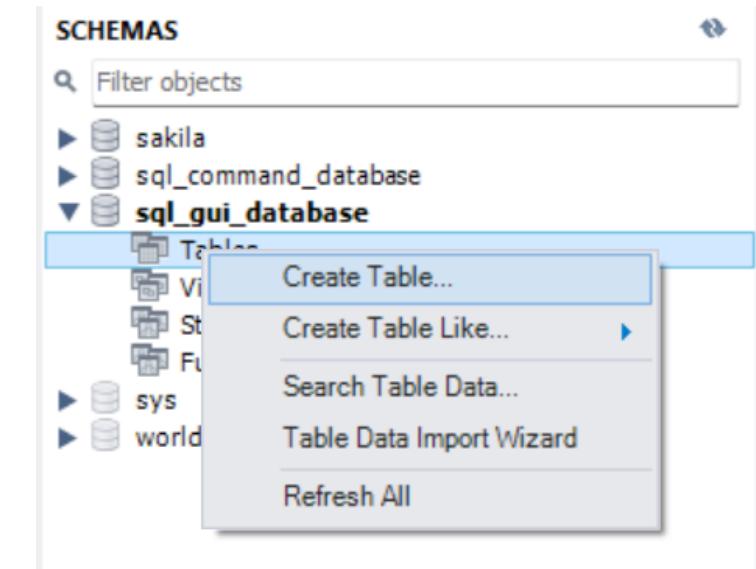
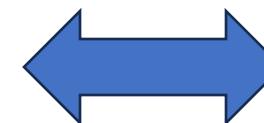
Using GUI

❖ Create a new table:



1. Click Table icon

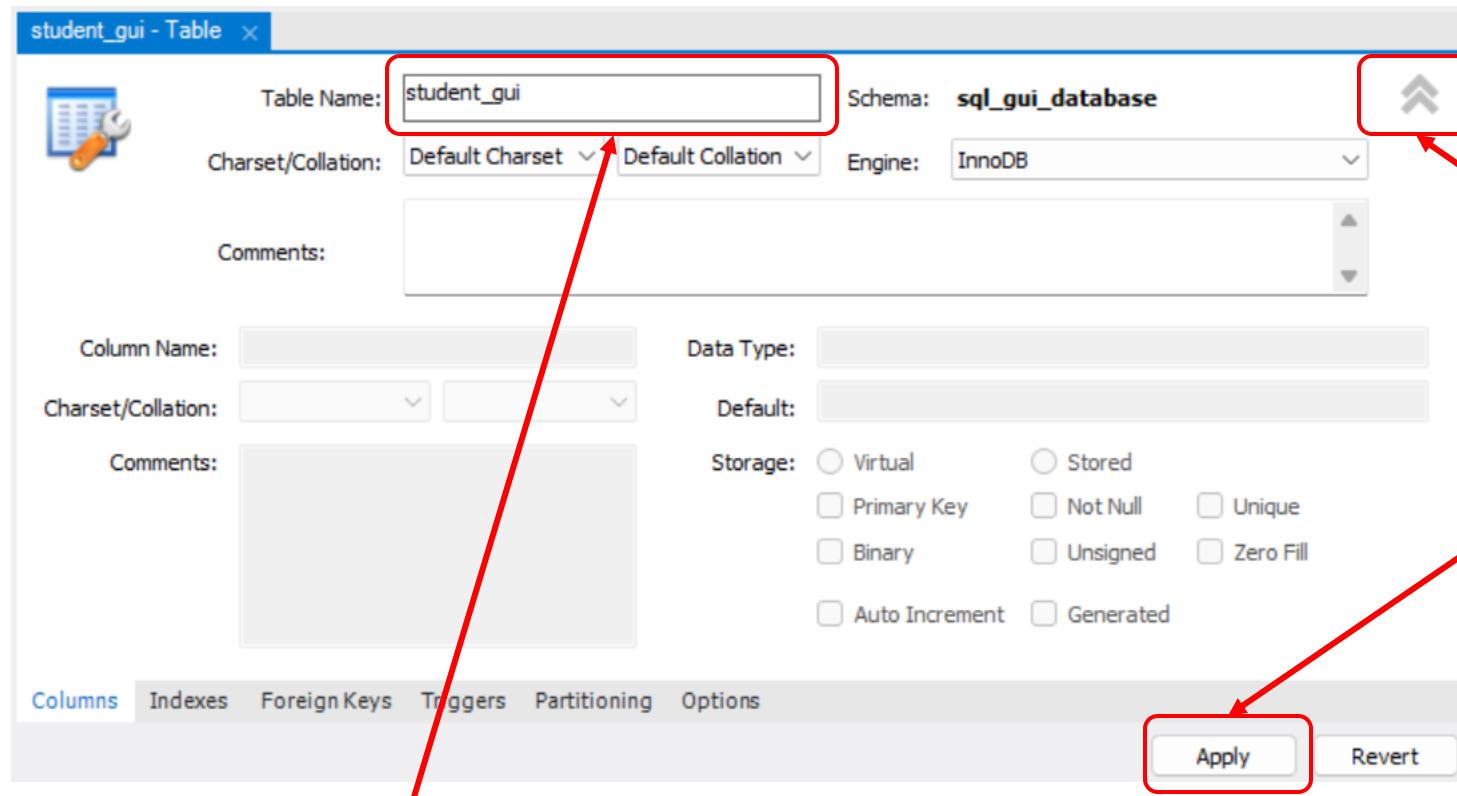
Make sure selected DB is correct



1. Right click on “Tables”, select “Create Table”

Using GUI

❖ Create a new table:



2. Set name for table

4. Click “Apply” to generate SQL command

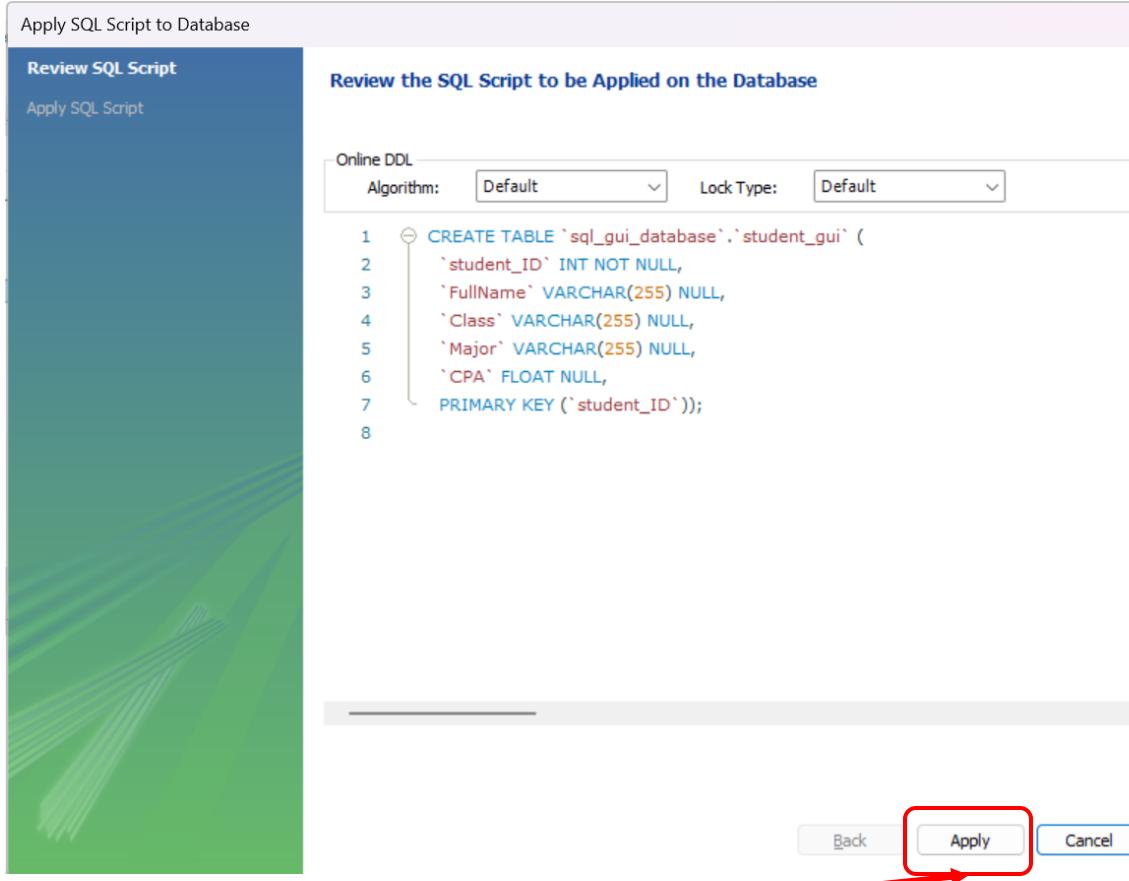
3. Click Arrow icon to set column for table

The screenshot shows the 'student_gui' table structure in MySQL Workbench. It has four columns: 'student_ID' (INT, PK, NN), 'FullName' (VARCHAR(255)), 'Class' (VARCHAR(255)), and 'Major' (VARCHAR(255)). The 'Default/Expression' column is empty. A large blue arrow points downwards from the 'Apply' button in the previous step to this table structure.

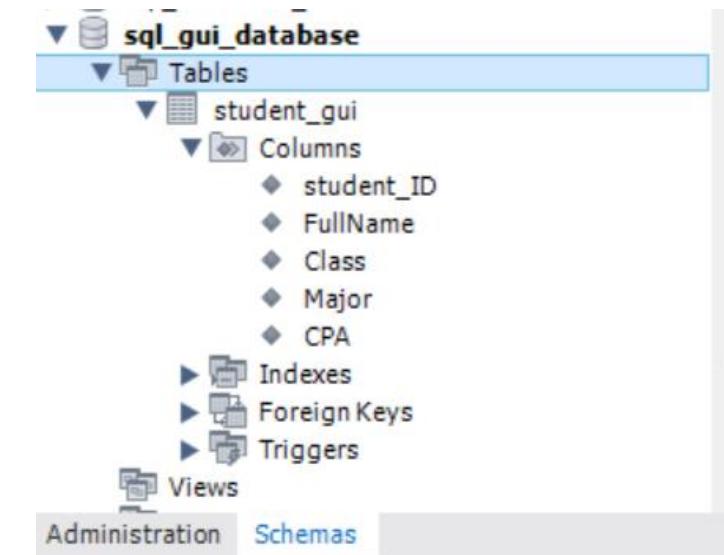
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
student_ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
FullName	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Class	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Major	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Using GUI

❖ Create a new table:



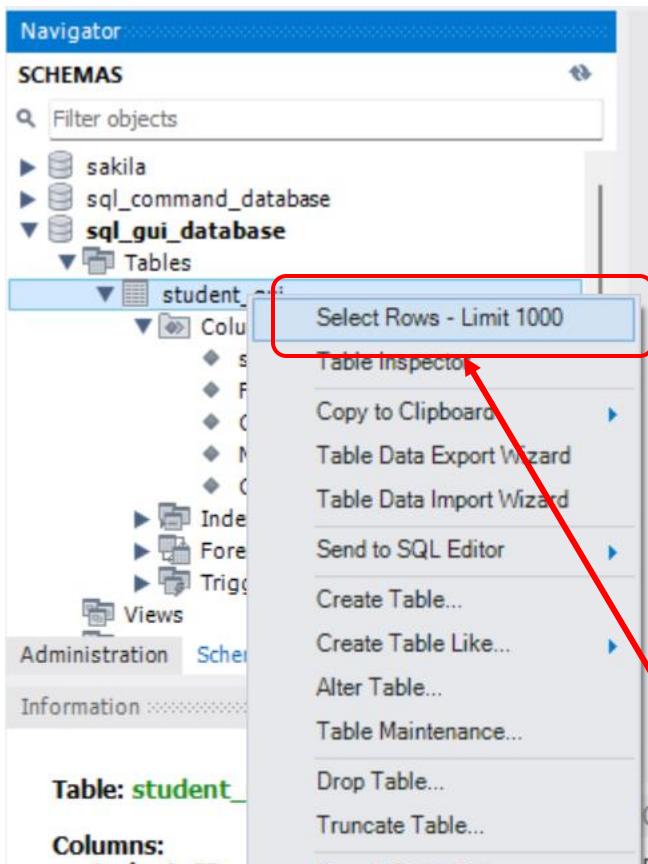
5. Click “Apply” to execute



New created Table

Using GUI

❖ Insert data into a new Table:



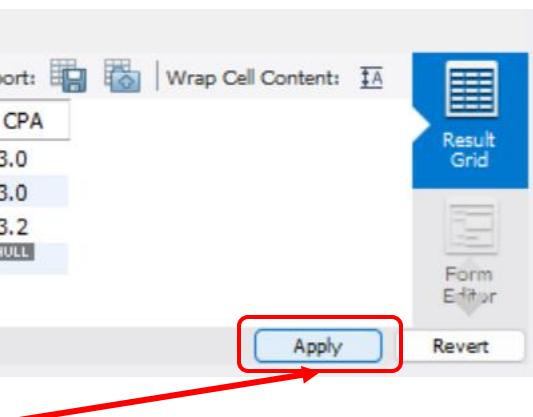
1. Right click on Table,
select “Select Rows”

	student_ID	FullName	Class	Major	CPA
▶ *	HULL	HULL	HULL	HULL	HULL

2. Double click on cell to add
data

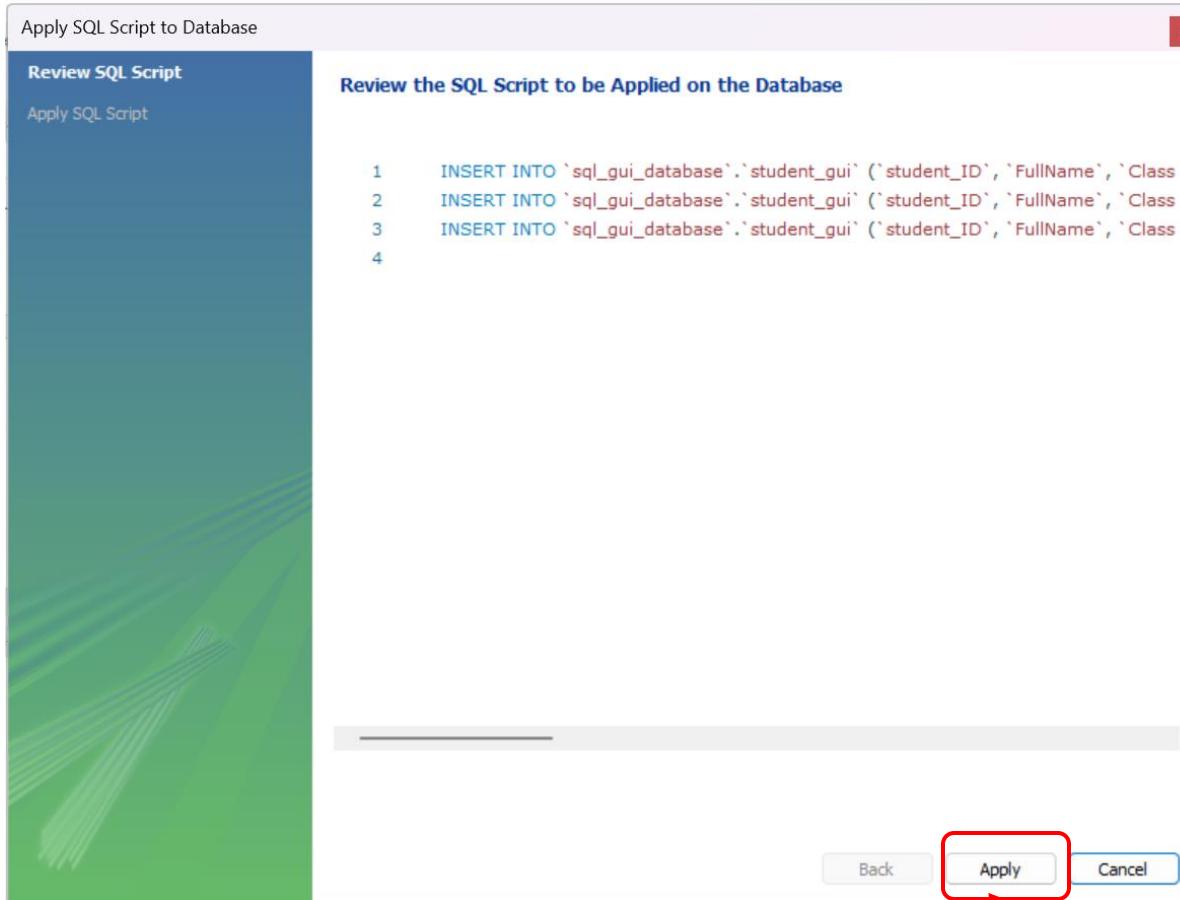
	student_ID	FullName	Class	Major	CPA
	20183660	Nguyen Duy Tung	IT1 03	Computer Science	3.0
▶	20183572	Bui Hoang Son	EE1 02	Electrical Engineer	3.0
*	20200034	Bui Hanh Trang	ETE16	Multimedia	3.2
	HULL	HULL	HULL	HULL	HULL

3. Click “Apply” to
generate SQL command



Using GUI

❖ Insert data into a new Table :



	student_ID	FullName	Class	Major	CPA
▶	20183572	Bui Hoang Son	EE1 02	Electrical Engineer	3
	20183660	Nguyen Duy Tung	IT1 03	Computer Science	3
*	20200034	Bui Hanh Trang	ETE16	Multimedia	3.2
	HULL	NULL	NULL	NULL	NULL

New added data

4. Click “Apply” to execute

Using phpMyAdmin



Overview:

- ❖ 05 components:

- Cross-platform (X)
- MariaDB (M)
- Perl (P)
- Apache (A)
- PHP (P)

- ❖ XAMPP is a web server that is free, open-source.

Install:

- ❖ Only XAMPP: [\[Xampp_1\]](#)
- ❖ Before Workbench: [Xampp_2](#)
- ❖ After Workbench: [\[Xampp_3\]](#)

XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?

XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.



Download
Click here for other versions

 XAMPP for Windows
8.0.9 (PHP 8.0.9)

 XAMPP for Linux
8.0.9 (PHP 8.0.9)

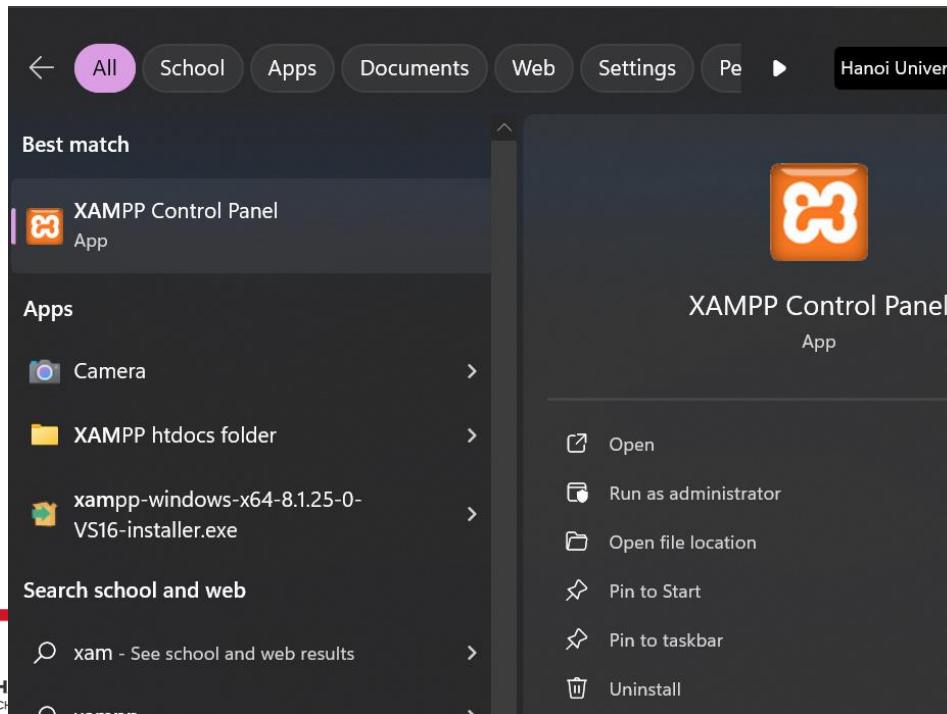
 XAMPP for OS X
8.0.9 (PHP 8.0.9)

phpMyAdmin

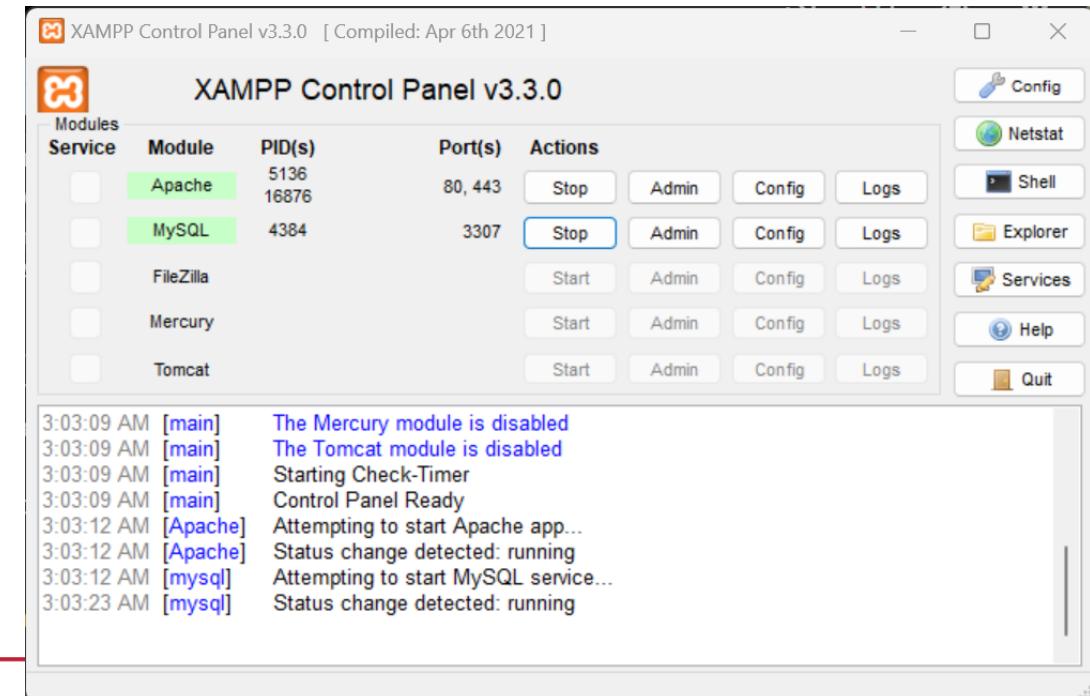
phpMyAdmin:

- ❖ is a free software tool written in PHP.
- ❖ handles the administration of MySQL over the Web.
- ❖ supports a wide range of operations on MySQL and MariaDB

Open XAMPP Control Panel

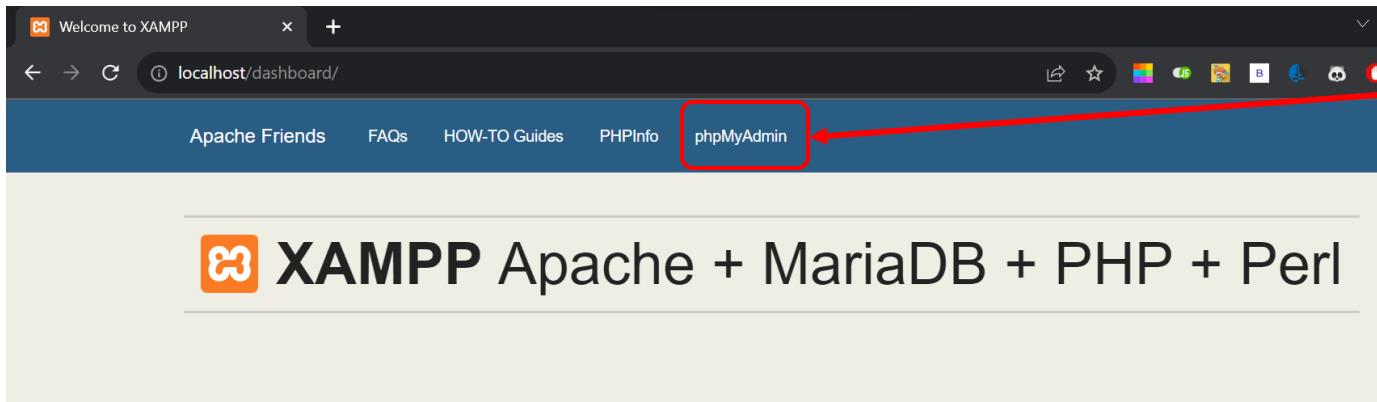


Start Apache and MySQL



phpMyAdmin

❖ Access: <http://localhost/dashboard/> or <http://localhost:80>



Click “phpMyAdmin”



Welcome to XAMPP for Windows 8.1.25

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

Dashboard site

Enter password

phpMyAdmin

localhost/phpmyadmin/index.php?route=/&route=%2F

Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Binary log Replication More

General settings

Change password Server connection collation: utf8mb4_unicode_ci More settings

Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MySQL
- Server connection: SSL is not being used
- Server version: 8.0.35 - MySQL Community Server - GPL
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Appearance settings

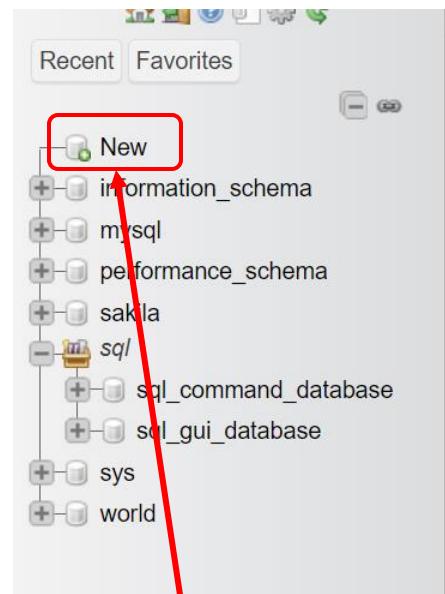
Language English Theme pmahomme View all

Web server

- Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.1.25
- Database client version: libmysql - mysqlnd 8.1.25
- PHP extension: mysqli curl mbstring
- PHP version: 8.1.25

New created Databases

❖ Create a new Database:



Click “New”

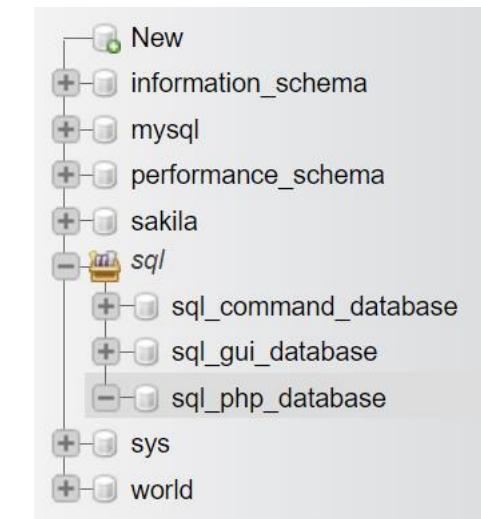
The 'Create database' dialog box shows the database name 'sql_php_database' in the input field, a dropdown for character set/collation set to 'utf8mb4_0900_ai_ci', and a 'Create' button highlighted with a red box. Below the dialog is a table of existing databases.

Database	Collation	Primary replication	Action
information_schema	utf8mb3_general_ci	Replicated	<input type="button" value="Check privileges"/>
mysql	utf8mb4_0900_ai_ci	Replicated	<input type="button" value="Check privileges"/>
performance_schema	utf8mb4_0900_ai_ci	Replicated	<input type="button" value="Check privileges"/>
sakila	utf8mb4_0900_ai_ci	Replicated	<input type="button" value="Check privileges"/>
sql_command_database	utf8mb4_0900_ai_ci	Replicated	<input type="button" value="Check privileges"/>
sql_gui_database	utf8mb4_0900_ai_ci	Replicated	<input type="button" value="Check privileges"/>
sys	utf8mb4_0900_ai_ci	Replicated	<input type="button" value="Check privileges"/>
world	utf8mb4_0900_ai_ci	Replicated	<input type="button" value="Check privileges"/>

Total: 8

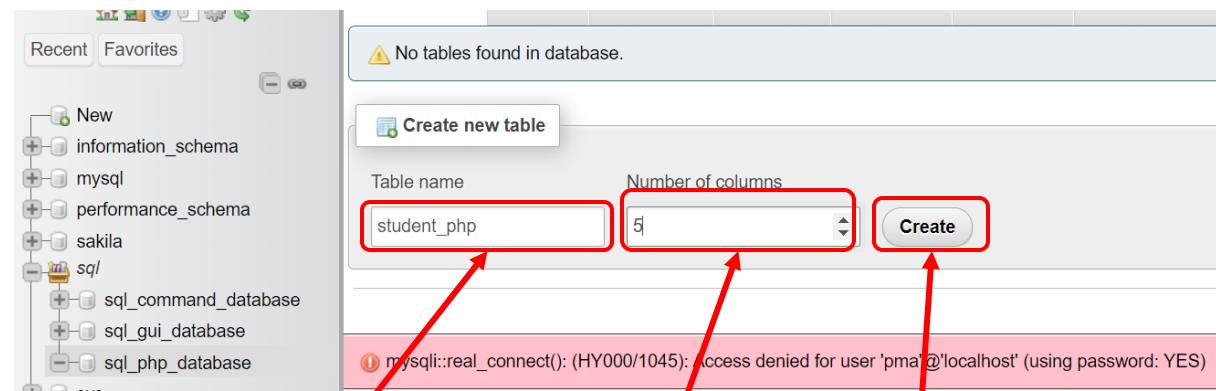
Set the name of Database

Click “Create”



New database

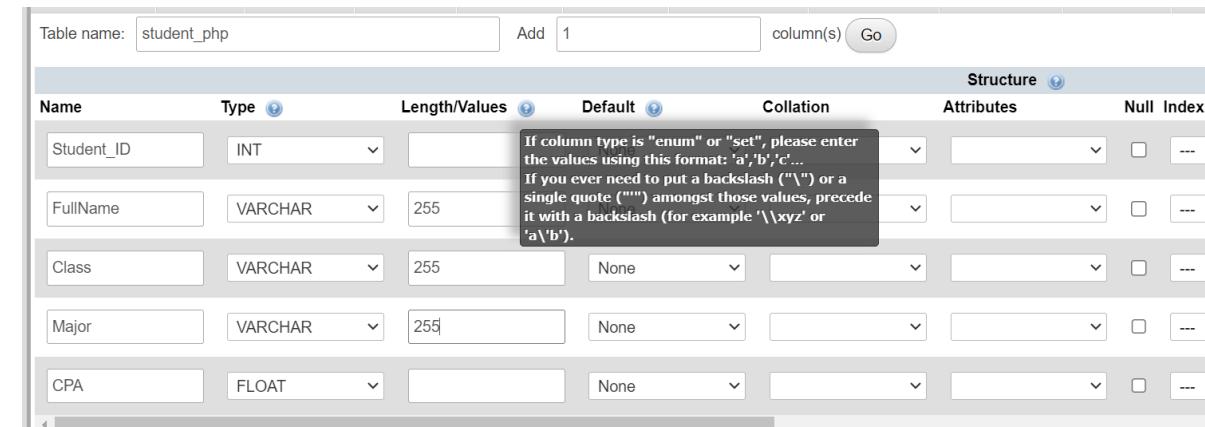
❖ Create a new Table:



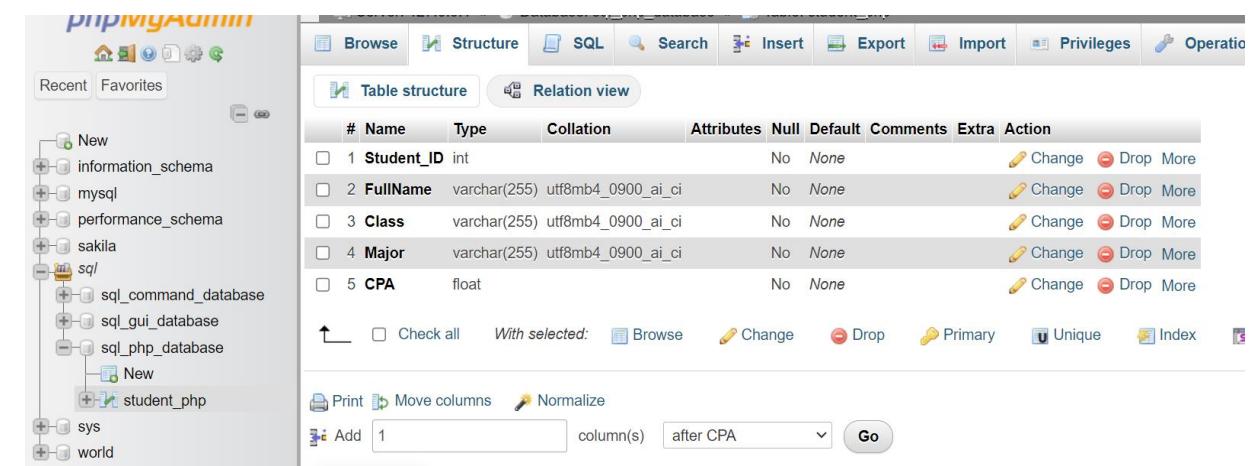
Set the name of Table

Choose number of cols

Click "Create"



Add columns



New Table

❖ Insert data into a new Table :

Browse Structure SQL Search Insert Export Import Privileges

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available

Showing rows 0 - 3 (4 total, Query took 0.0005 seconds.)

```
SELECT * FROM `student_php`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table

Click “Insert”

Click “Go”

Column	Type	Function	Null	Value
Student_ID	int			2019
FullName	varchar(255)			Tran Van B
Class	varchar(255)			CCCCC
Major	varchar(255)			ABCDEF

Console

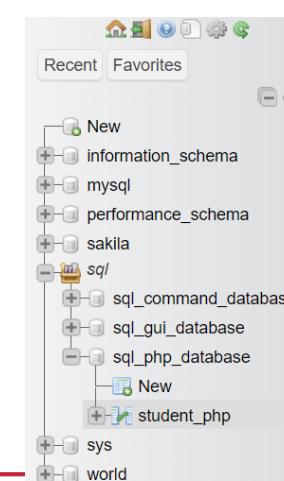
Insert Data

Run SQL query/queries on table sql_php_database.student_php: ⚡

```
1 INSERT INTO `student_php` (`Student_ID`, `FullName`, `Class`, `Major`, `CPA`) VALUES ('2019', 'Tran Van B', 'CCCCC', 'ABCDEF', '4');
```

SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query Bind parameters Delimiter : Show this query here again Retain query box Rollback when finished Enable foreign key checks

Go



Browse Structure SQL Search Insert

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available

Showing rows 0 - 11 (12 total, Query took 0.0005 seconds.)

```
SELECT * FROM `student_php`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

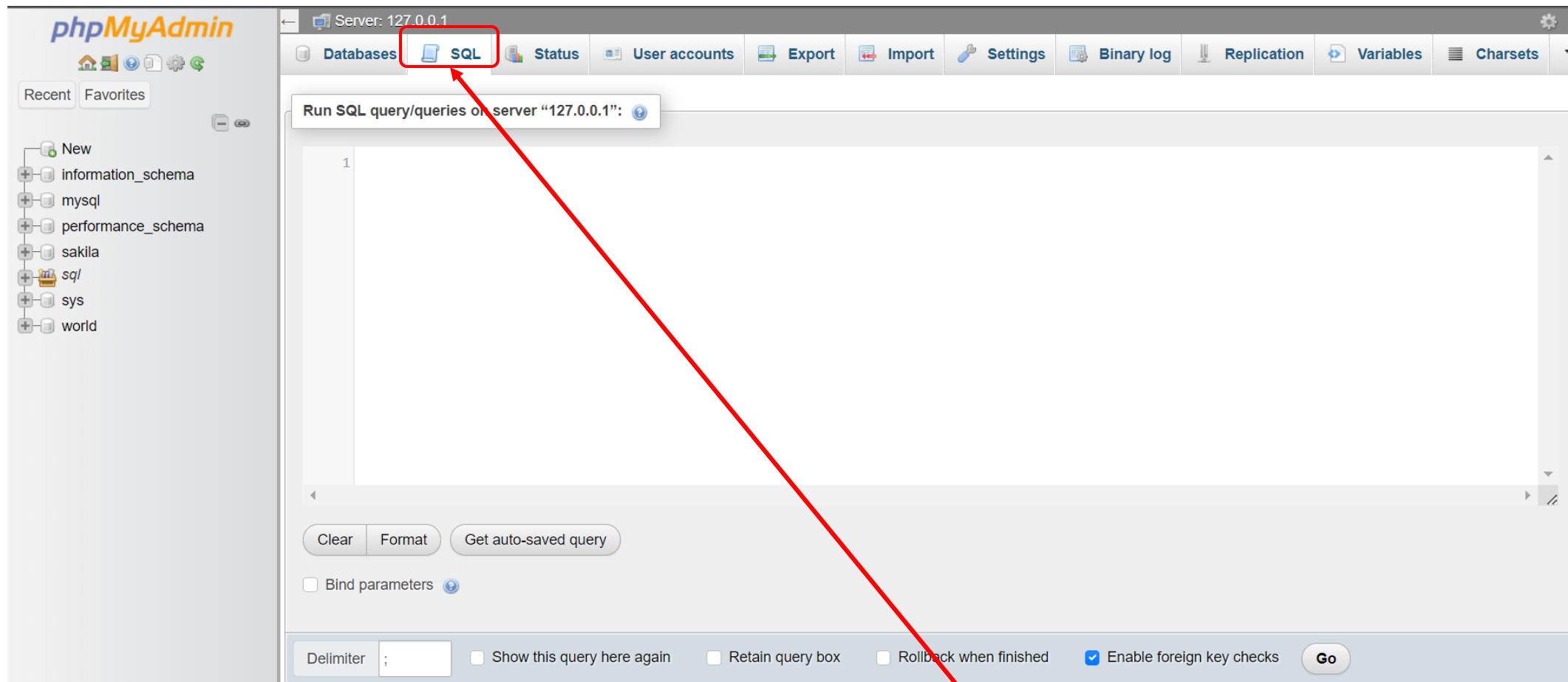
Show all Number of rows: 25 Filter rows: Search this table

Extra options

Student_ID	FullName	Class	Major	CPA
20183660	Nguyen Duy Tung	IT1-03	Computer Science	3
20183572	Bui Hoang Son	EE1-02	'Electrical Engineer	3
20200034	Bui Hanh Trang	ETE16-01	Multimedia	3.2

New Data

❖ Using SQL command:



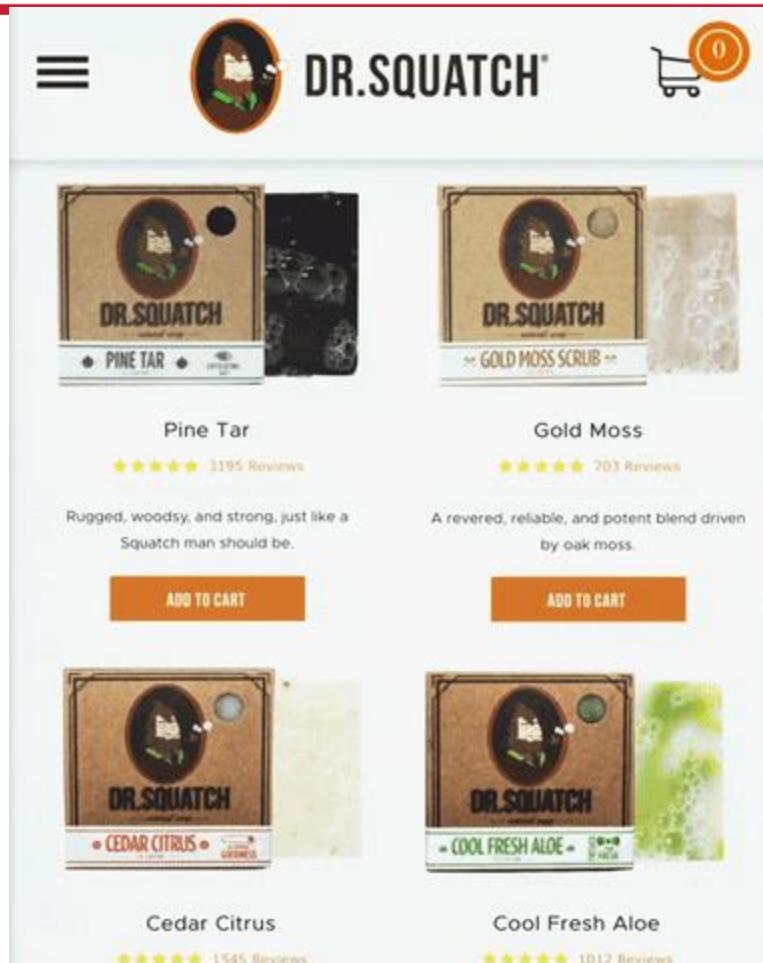
Click “SQL”

The background of the slide features a dark blue gradient with a red dotted pattern forming a stylized, winding shape that resembles a path or a river.

HUST

3. USING MYSQL WITH OTHERS PROGRAMMING LANGUAGES

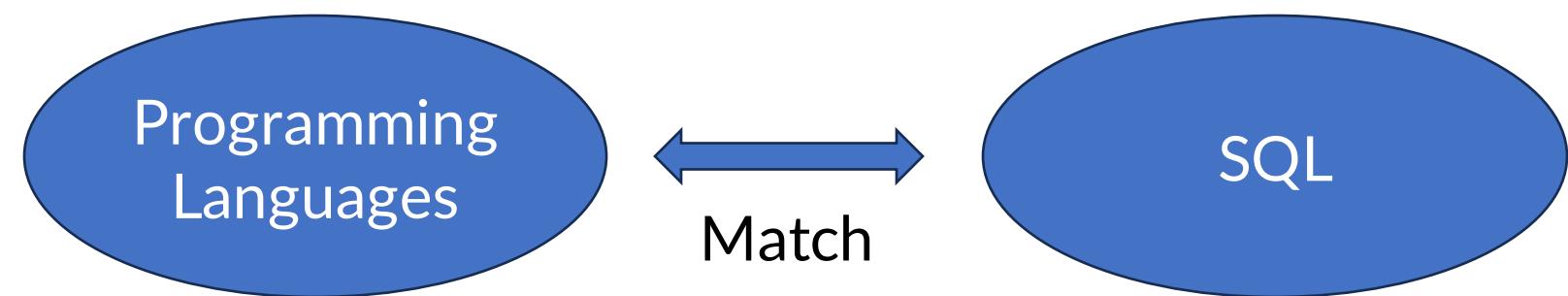
How can I connect to MySQL database?



❖ Devs use various Languages to build apps

- Python PHP Perl
- Java C# v.v.

❖ SQL only for communicating in Databases



❖ Solutions:

- Programming Languages' mechanism
- ORMs (Object-Relational Mapping)

Example of E-commerce app

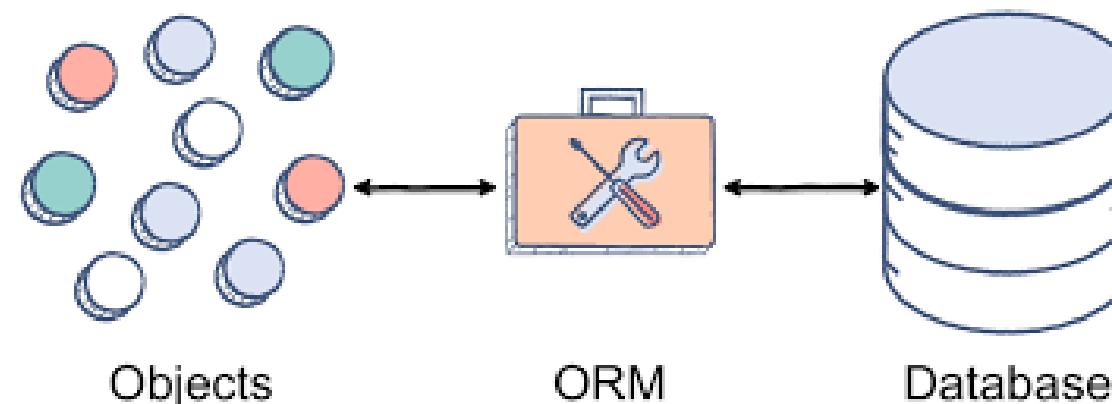
Overview of ORMs

- ❖ ORM is a technique → allows query and manipulate data from a database using an object-oriented paradigm
- ❖ It is a completely ordinary library written in your language of choice
→ Not necessary to use SQL
- ❖ Pros:

- Save time
- Flexible

- ❖ Cons:

- Difficult to learn for beginners
- Can not handle complex SQL command



Mapping Languages to SQL

❖ Python

- mysql-connector-python libraries
- SQLAlchemy
- Django

❖ Java

- JDBC Driver
- Hibernate

❖ PHP

- PHP Data Object (PDO)
- Propel
- Doctrine



django



SQLAlchemy



HIBERNATE



propel
Smart, easy object persistence



doctrine

mysql-connector-python

❖ Install: pip install mysql-connector-python

❖ Verify the installation:

```
import mysql.connector
```

```
sql_test = mysql.connector.connect(  
    host='localhost',  
    database='sql_gui_database',  
    user='root',  
    password='root')
```

```
print(sql_test)
```

Successful output

```
(sql_env) D:\TA_\Databases>python sql_demo.py  
<mysql.connector.connection_cext.CMySQLConnection object at 0x0000014E7712B850>
```

mysql-connector-python

❖ Using .sql file: **python_mysql.sql** to work

❖ Connect to a Database in MySQL:

```
def connect():
    """Connect to MySQL database
    """
    conn = None
    try:
        #establishes a connection to the "python_mysql" database and
        #returns a MySQLConnection object
        conn = mysql.connector.connect(host='localhost',
                                       database='python_mysql',
                                       user='root',
                                       password='root')
        #Check if the connection to the MySQL database has been established
        #successfully
        if conn.is_connected():
            print('Connected to MySQL database')
    except Error as e:
        if conn is not None and conn.is_connected():
            conn.close()
```

Output

```
(sql_env) D:\TA_\Databases>python sql_demo.py
Connected to MySQL database
```

mysql-connector-python

❖ Add data to a Table:

```
def insert_book(title, isbn):
    query = "INSERT INTO books(title,isbn) " \
            "VALUES(%s,%s)"
    args = (title, isbn)

    try:
        conn = mysql.connector.connect(host='localhost',
                                       database='python_mysql',
                                       user='root',
                                       password='root')

        cursor = conn.cursor()
        cursor.execute(query, args)

        # Retrieve the last insert id of the AUTO_INCREMENT column by using
        # the lastrowid property of the MySQLCursor object
        if cursor.lastrowid:
            print('last insert id', cursor.lastrowid)
        else:
            print('last insert id not found')

        # Make change to Database
        conn.commit()
    except Error as error:
        print(error)

    finally:
        # Close the cursor and connection
        cursor.close()
        conn.close()

if __name__ == '__main__':
    insert_book('A Life of Tung_just parody', '0000000000')
    # connect()
```

	id	title	isbn
81	The Stereo and God	1231316672178	
82	UFO vs. CBS	1239778693754	
83	Vietnam Victory	1237098200581	
84	A Sudden Light	9781439187036	
*	NULL	NULL	NULL

Table “Book” before add

```
(sql_env) D:\TA\_Databases>python sql_demo.py
last insert id 85
```

Output

	id	title	isbn
82	UFO vs. CBS	1239778693754	
83	Vietnam Victory	1237098200581	
84	A Sudden Light	9781439187036	
85	A Life of Tung_just parody	0000000000	
*	NULL	NULL	NULL

Table “Book” after add

- ❖ Install: `pip install mysql-connector-python`
`pip install sqlalchemy`

- ❖ Verify the installation:

```
create_engine("mysql+mysqlconnector://<user>:<password>@<host>/<database>")
```

```
# Connect to the database
```

```
engine = create_engine(
```

```
    "mysql+mysqlconnector://root:root@localhost/python_mysql")
```

```
# Test the connection
```

```
connection = engine.connect()
```

```
print(connection)
```

Successful output

```
(sql_env) D:\TA_\Databases>python sqlalchemy_demo.py
<sqlalchemy.engine.base.Connection object at 0x00000247A0461750>
```

SQLAlchemy

❖ Add data to the table:

```
#Create base class
Base = declarative_base()

#Define table class
class Books(Base):
    __tablename__ = 'books'
    id = Column('id', Integer, primary_key=True, autoincrement=True,
    nullable=False)
    title = Column('title', String(255), nullable=False)
    isbn = Column('isbn', String(13), nullable=False)

#Create table
Base.metadata.create_all(engine)

#Create books objects
book1 = Books(title='Demo_1', isbn='1111111111')
book2 = Books(title='Demo_2', isbn='2222222222')
book3 = Books(title='Demo_3', isbn='3333333333')

#Add books to session
session.add(book1)
session.add(book2)
session.add(book3)

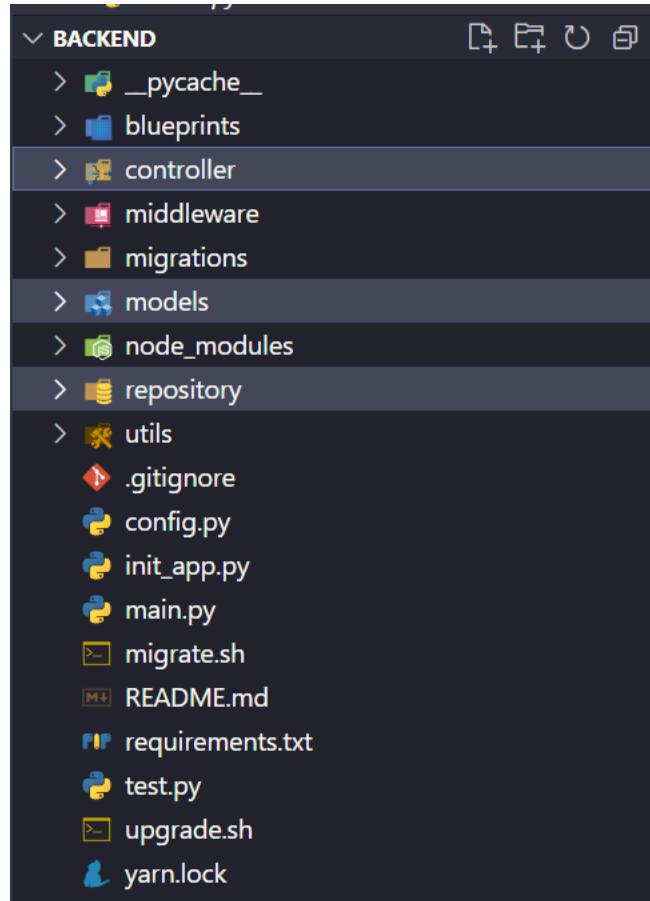
session.commit()
```

	id	title	isbn
	85	A Life of Tung_just parody	0000000000
	86	Demo_1	1111111111
	87	Demo_2	2222222222
▶	88	Demo_3	3333333333
*	NULL	NULL	NULL

Successful output

SQLAlchemy

❖ Project examples:



```
from flask_migrate import Migrate
from config import SQLALCHEMY_DATABASE_URI, SECRET_KEY, DATA_PATH

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = SQLALCHEMY_DATABASE_URI
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True
app.config['UPLOAD_FOLDER'] = DATA_PATH
app.config['SQLALCHEMY_ENGINE_OPTIONS'] = {
    'max_overflow': 30,
    'pool_size': 100
}
app.secret_key = SECRET_KEY

CORS(app, resources={r"/": {"origins": "*", "send_wildcard": "False"}})

db = SQLAlchemy(app=app, session_options={'autocommit': False, 'autoflush': False})

migrate = Migrate(app, db)
with app.app_context():
    #     db.init_app(app)
    migrate.init_app(app, db)
```

❖ Project examples:

```
> 🐍 brand.py > 📁 Brand
truong-st, 8 months ago | 2 authors (You and others)
from init_app import db

truong-st, 8 months ago | 2 authors (You and others)
└ class Brand(db.Model):
    __tablename__ = 'brand'

    id = db.Column('id', db.Integer, autoincrement=True, primary_key=True)
    name = db.Column('name', db.VARCHAR(50))
    detail = db.Column('detail', db.Text)
    image = db.Column('image', db.Text)
    is_activated = db.Column('is_activated', db.Integer, nullable=False)

    def __init__(self, **kwargs):
        super(Brand, self).__init__(**kwargs)

    def to_full_json(self):
        json_token = {
            'id': self.id,
            'name': self.name,
```

Create Table “brand” object in Databases

❖ Project examples:

```
repository > 📁 brand_repo.py > ✏️ get_order_by_name
          truong-st, 9 months ago | 2 authors (You and others)
1   from models.brand import Brand, db
2
3   # Get all Brands
4   def find_all():
5       return Brand.query.all()
6
7   # Get Brands by filtering
8   # By id
9   def find_by_id(id):
10      return Brand.query.filter_by(id=id).first()
11
12  # By name
13  def find_by_name(name):
14      return Brand.query.filter_by(name=name).first()
15
16  # Get Brands in order
17  # By name (1: ascending, 2:descending)
18  def get_order_by_name(type=1):
19      if type == 1:
20          return Brand.query.order_by(Brand.name.asc())
21      else:
22          return Brand.query.order_by(Brand.name.desc())
```

Operating with “brand” object (mapping to SQL)

❖ Project examples:

```
controller > 🐍 comment_controller.py > ...
    truong-st, 8 months ago | 1 author (truong-st)
1  from flask import request, g
2  from repository.product_review_repo import *
3
4
5  def get_by_id(id=None):
6      comment = find_by_id(id).to_full_json()
7      return comment
8
9
10 def get_comment_product(product_id):
11     comments = find_by_product_id(product_id)
12     comments = list(map(lambda x: x.to_full_json(), comments))
13     return comments
14
15
16 def add(data):
17     try:
18         insert(data)
19         return True
20     except:
21         return False
```

Operating with “brand” object (mapping to Python)

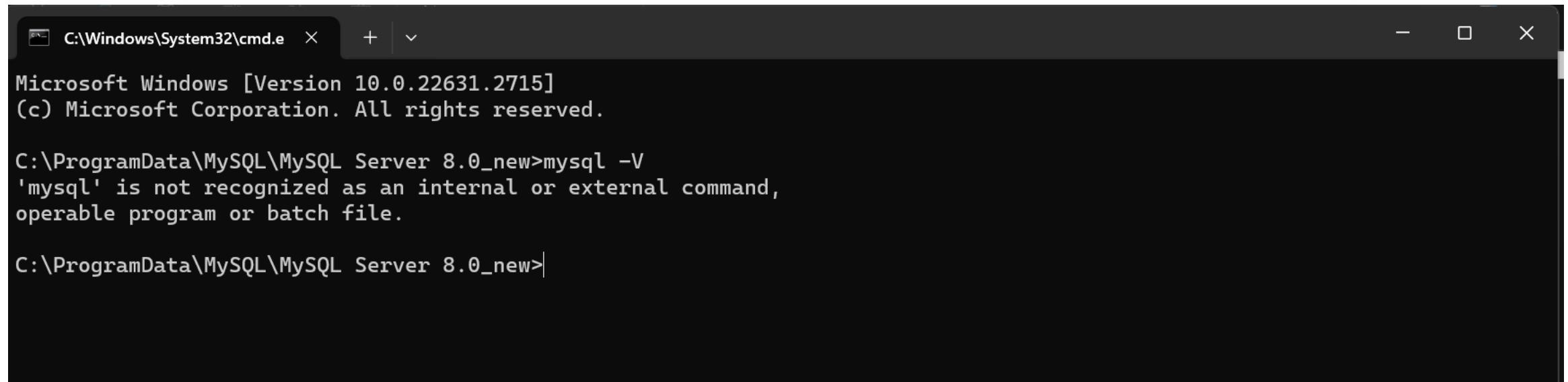


HUST

4. DEMO

Questions & Answers

Problem?



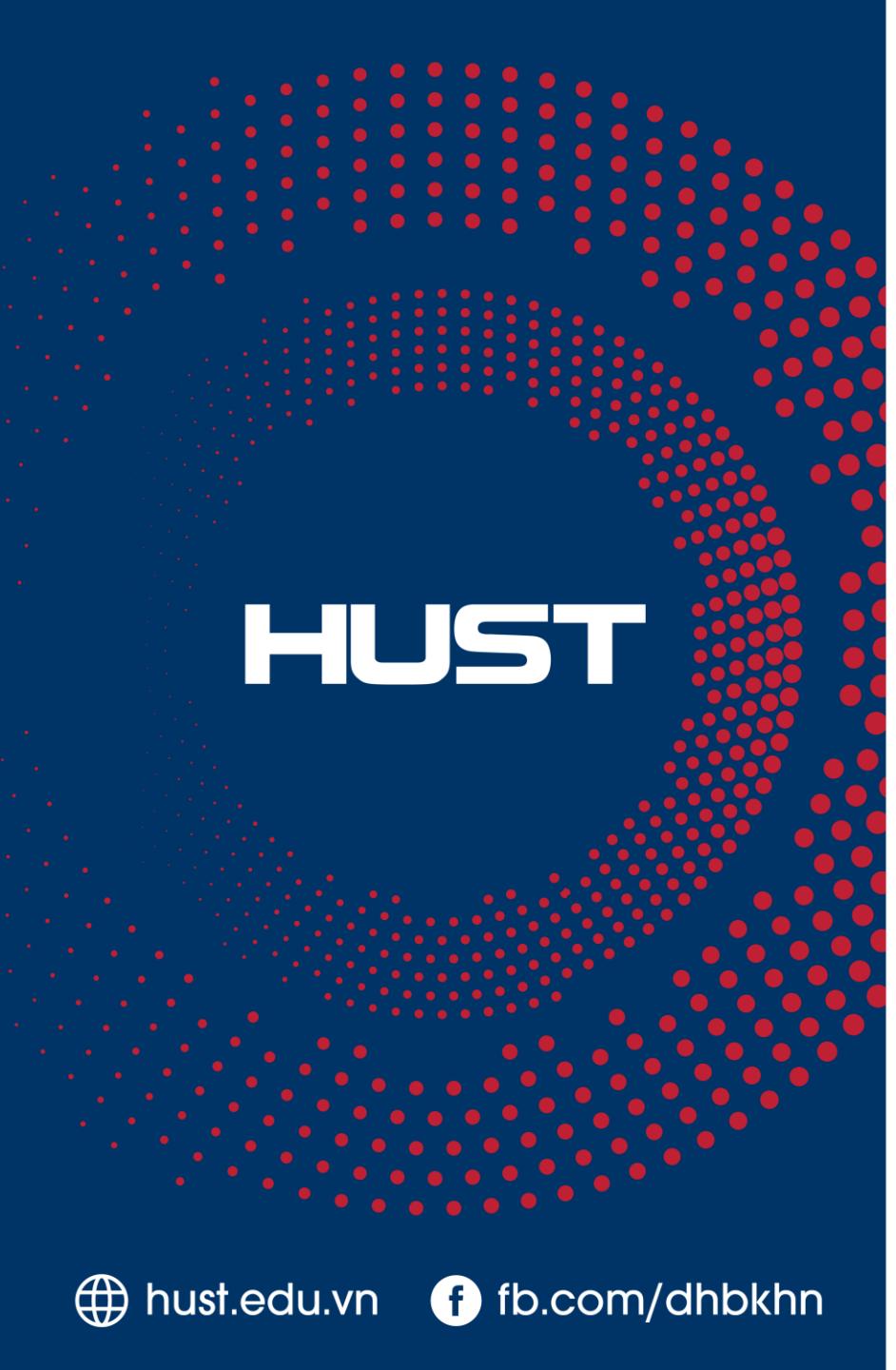
A screenshot of a Windows Command Prompt window titled "C:\Windows\System32\cmd.e". The window shows the following text:

```
Microsoft Windows [Version 10.0.22631.2715]
(c) Microsoft Corporation. All rights reserved.

C:\ProgramData\MySQL\MySQL Server 8.0_new>mysql -V
'mysql' is not recognized as an internal or external command,
operable program or batch file.

C:\ProgramData\MySQL\MySQL Server 8.0_new>
```

Q: Why after following above steps to installing MySQL, cannot check version by Command Line?



HUST

5. CONCLUSION

Conclusion

- ❖ Get basic concepts of RDBMS and technologies for application/web programming.
- ❖ Install and Set up MySQL Server, MySQL Workbench for Windows and so on
- ❖ Basic operations when working with MySQL Client (Workbench, phpMyAdmin)
- ❖ Connecting to MySQL Server via Python

Useful links

- ❖ <https://www.dataquest.io/blog/install-mysql-windows/>
- ❖ <https://www.freecodecamp.org/news/how-to-install-mysql-workbench-on-windows/>
- ❖ <https://blog.devart.com/how-to-check-mysql-version.html#find-mysql-version-command-line>
- ❖ <https://www.apachefriends.org/download.html>
- ❖ <https://roadmap.sh/>

The logo consists of a dark blue square with a large, stylized red dotted bee shape on the left side. The word "HUST" is written in white capital letters on the right side.

HUST

