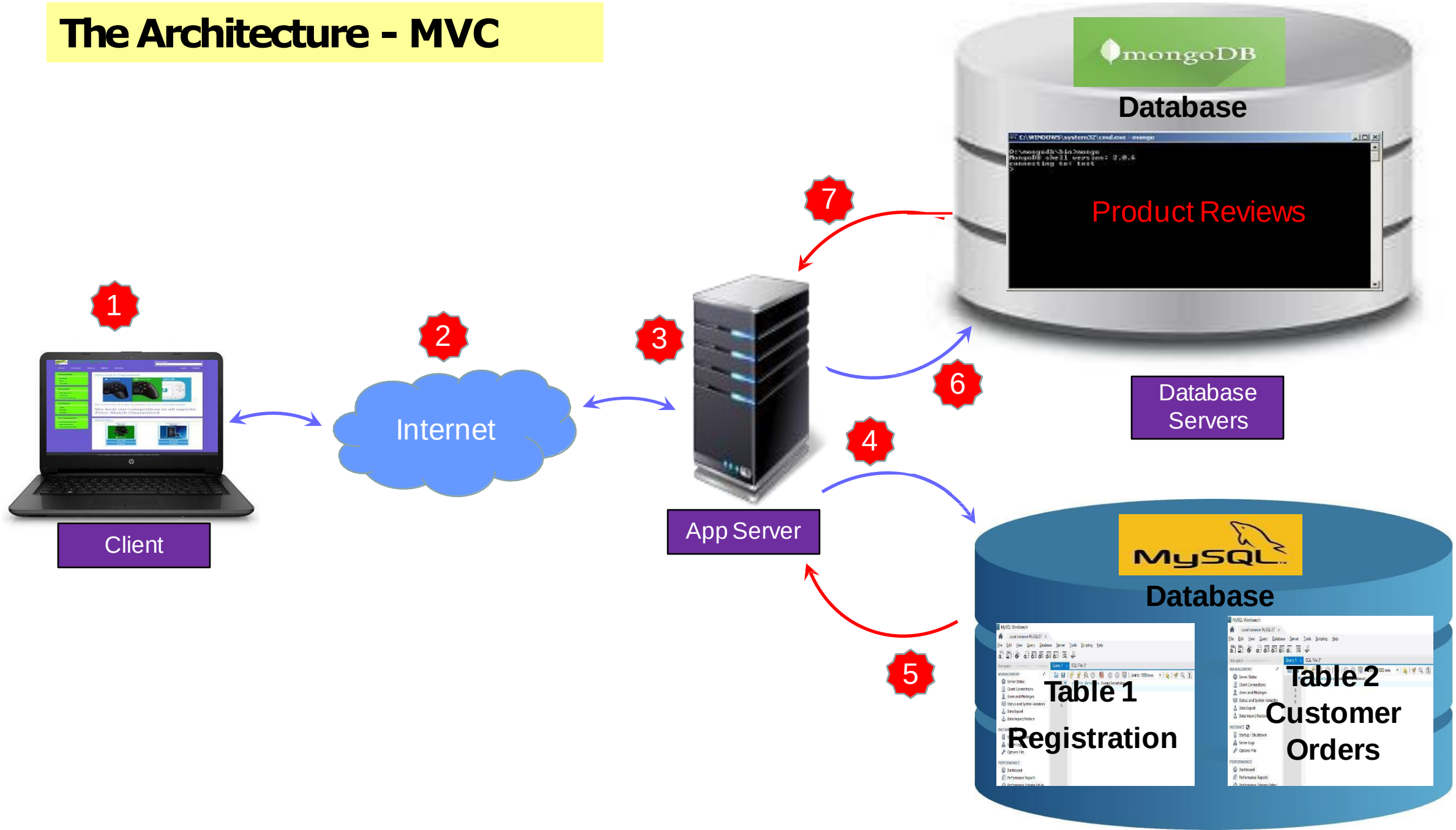
Tutorial #2

MySQL Datastore

CSP 584 – Enterprise Web Application
Dr. Atef Bader
Illinois Institute of Technology

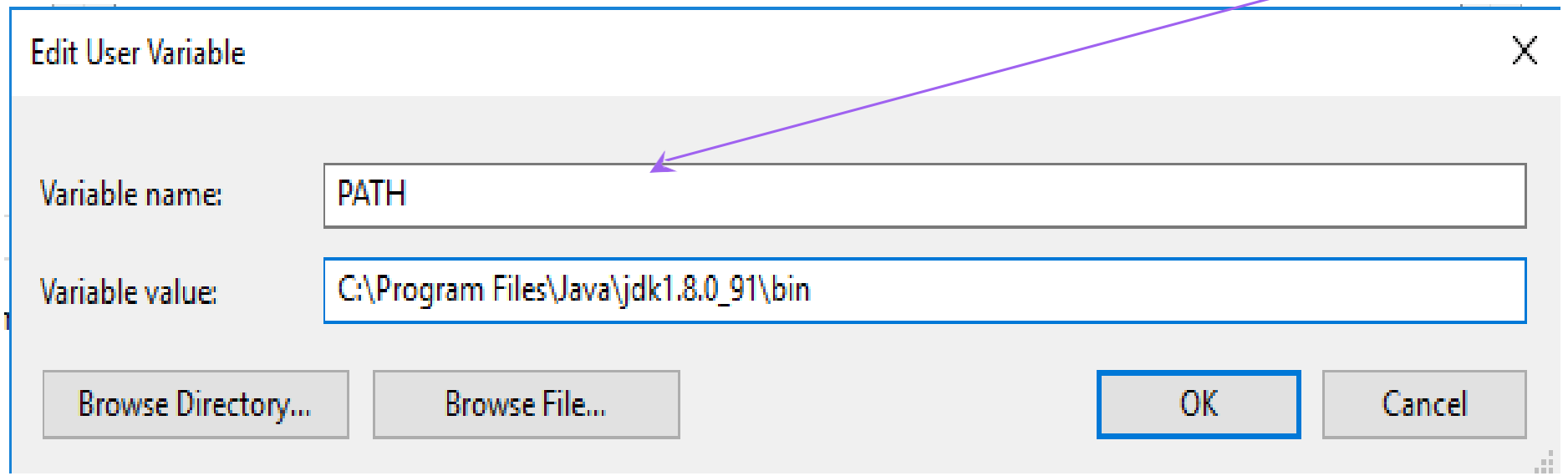
TA : Prakhar Nag
pnag@hawk.iit.edu

The Architecture – MVC



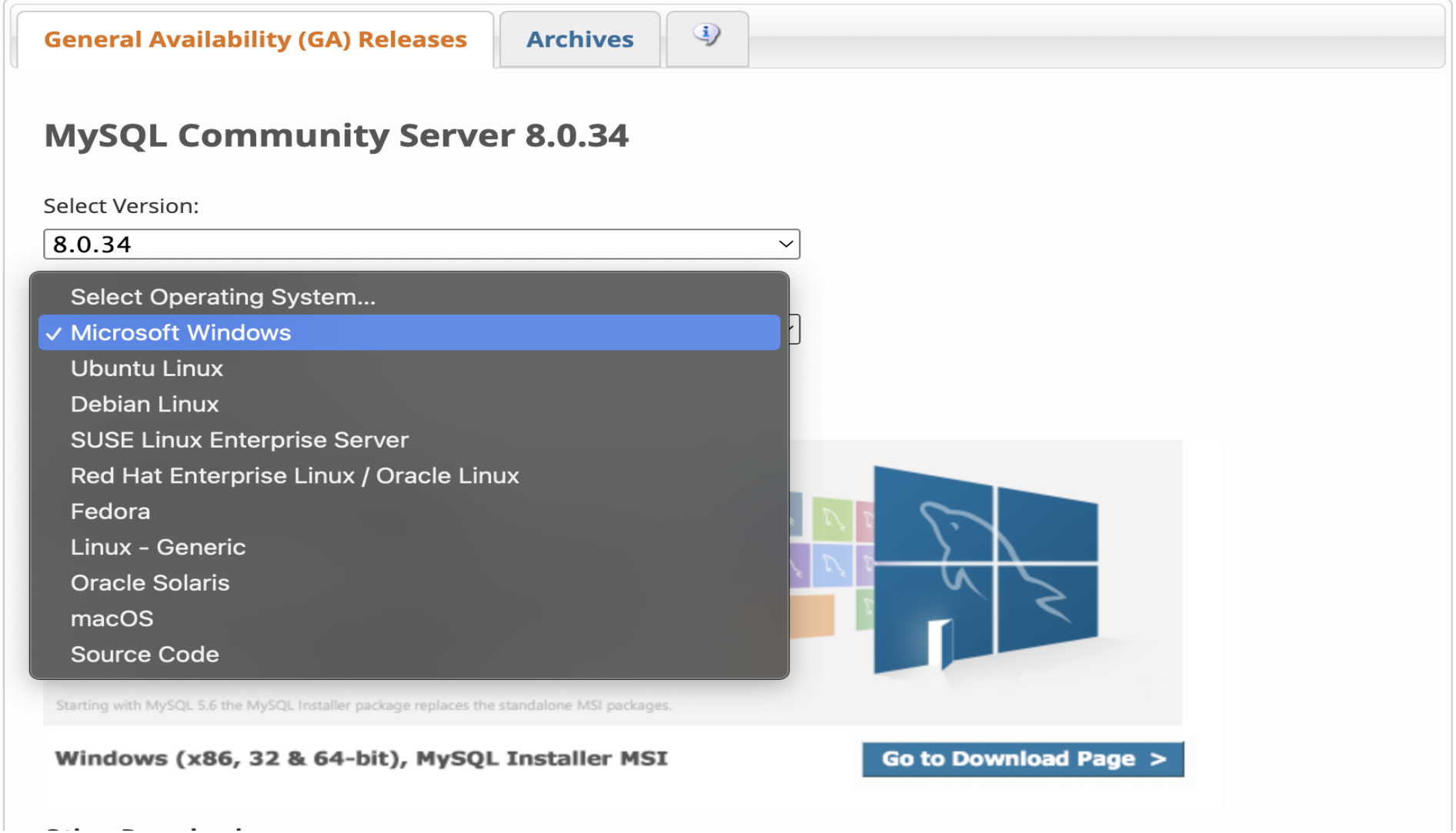
Pre-Requisites:

- Install Java latest version into your system.
- Set the PATH system variable in your local system under Control Panel –> system-> Advanced system settings -> Click on Advanced Tab ->Environment variables.



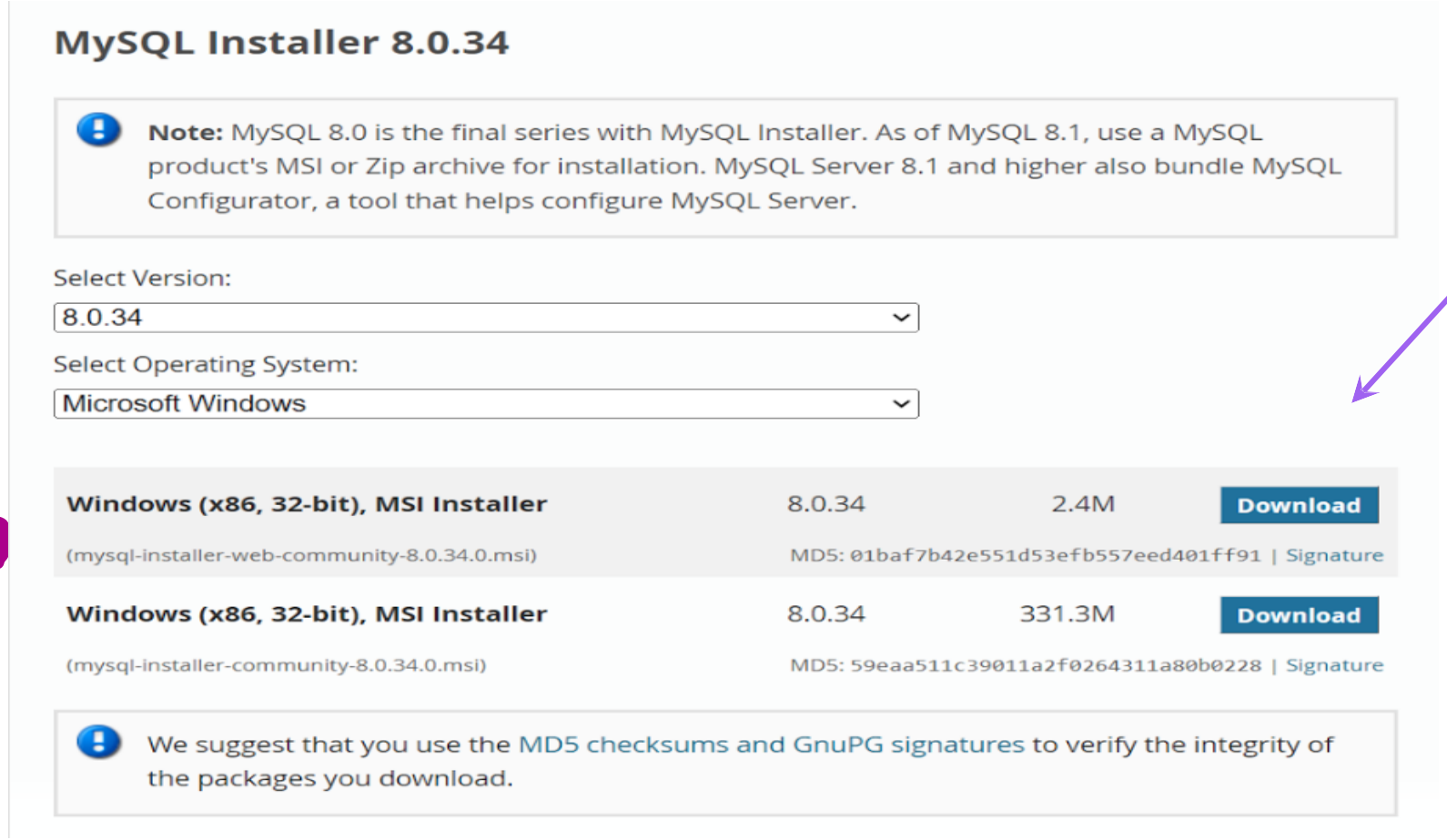
Example PATH variable

Platforms supported in MySQL

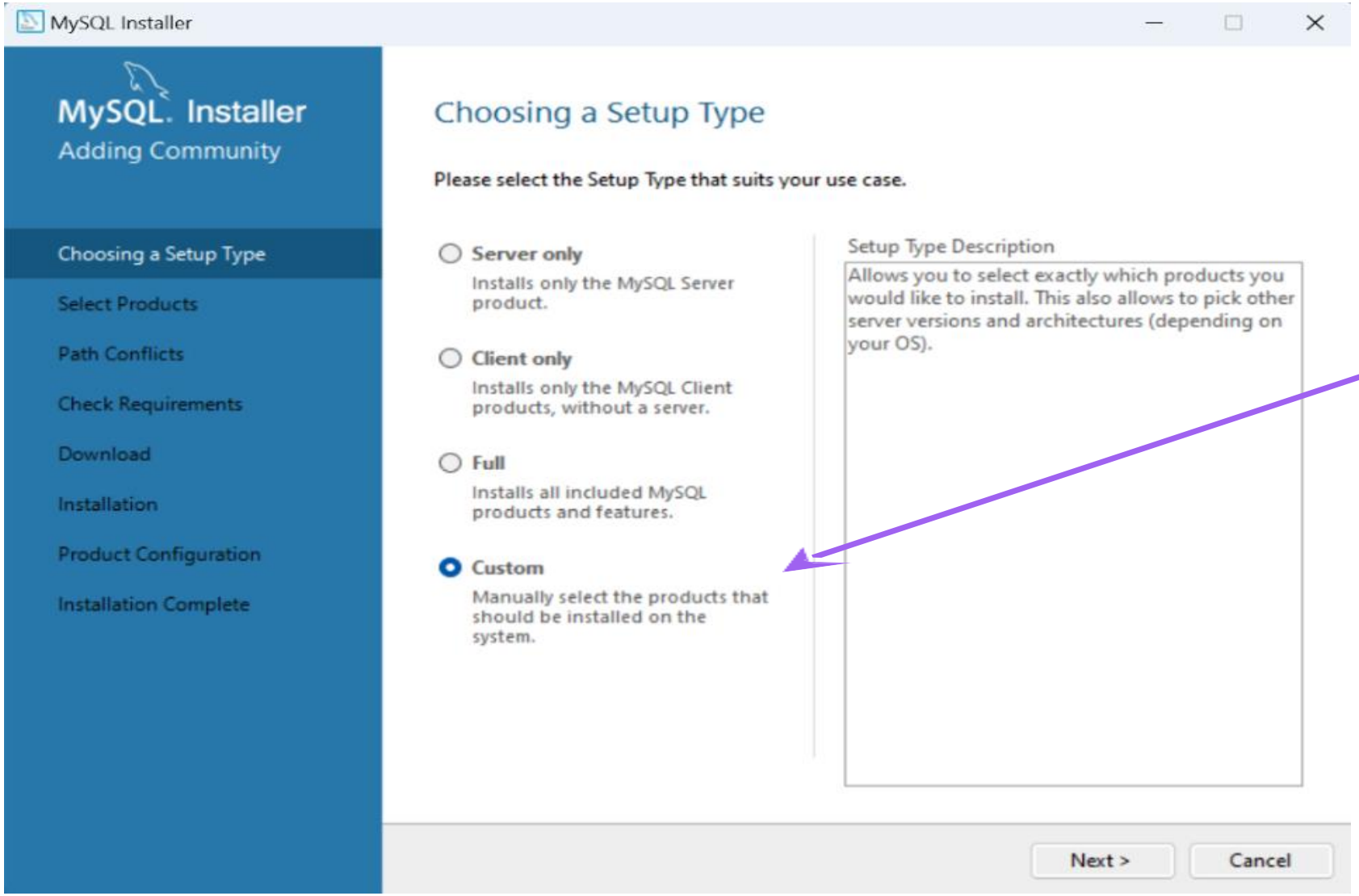


Download and Install MySQL Server

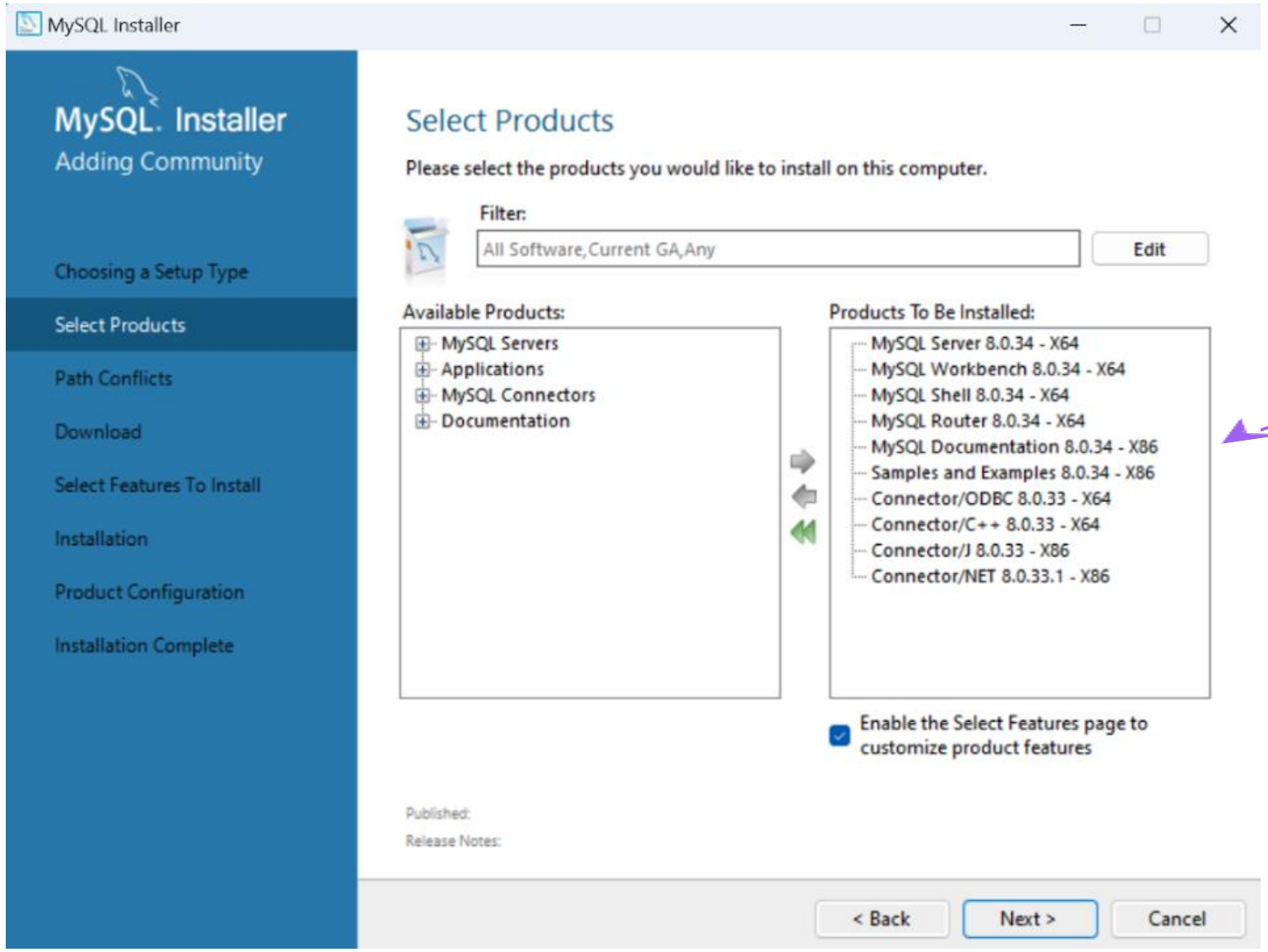
- Download the MySQL installer from https://dev.mysql.com/downloads/installer/
- (Choose the my-installer-web- community file)



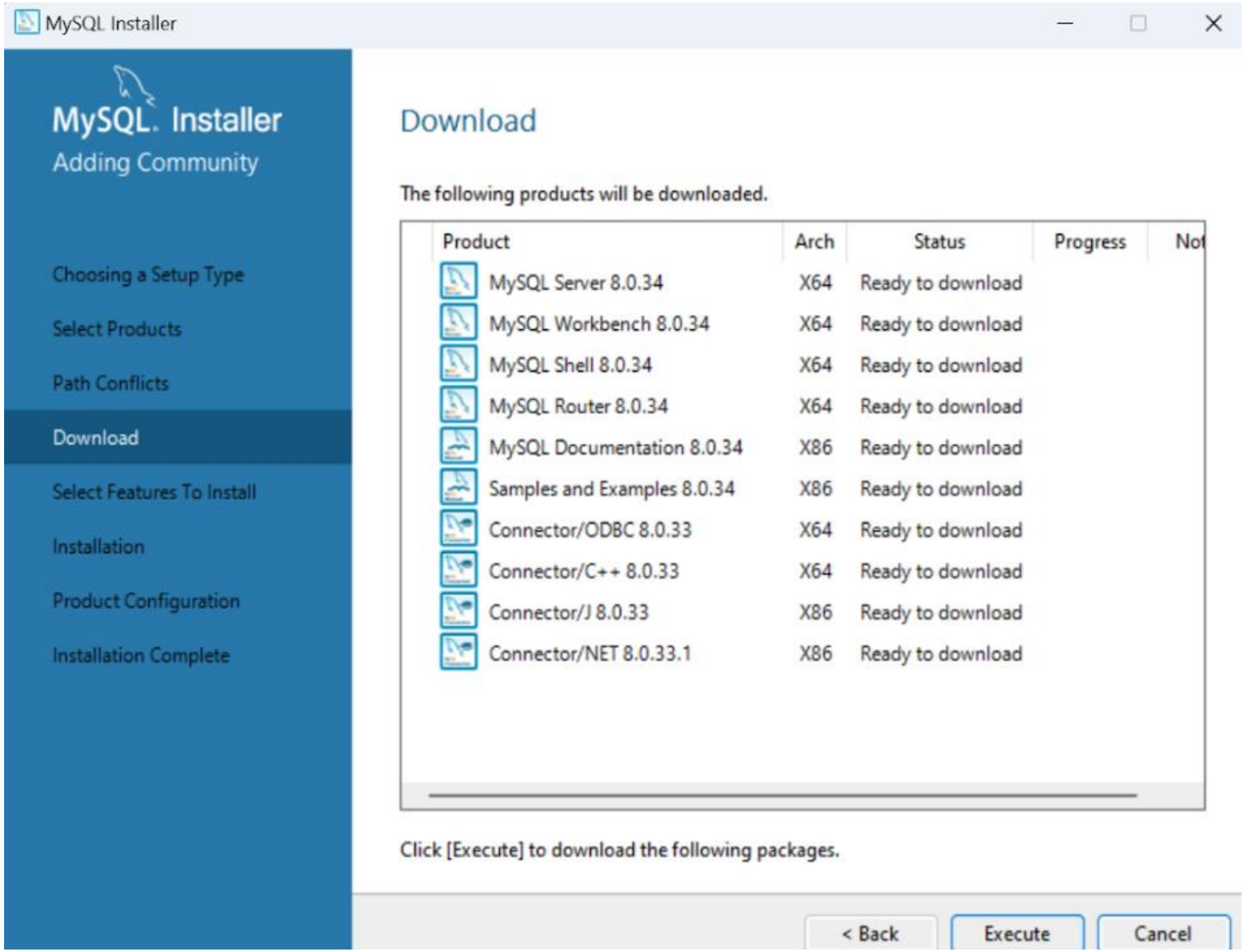
Select this file to Download and run it.



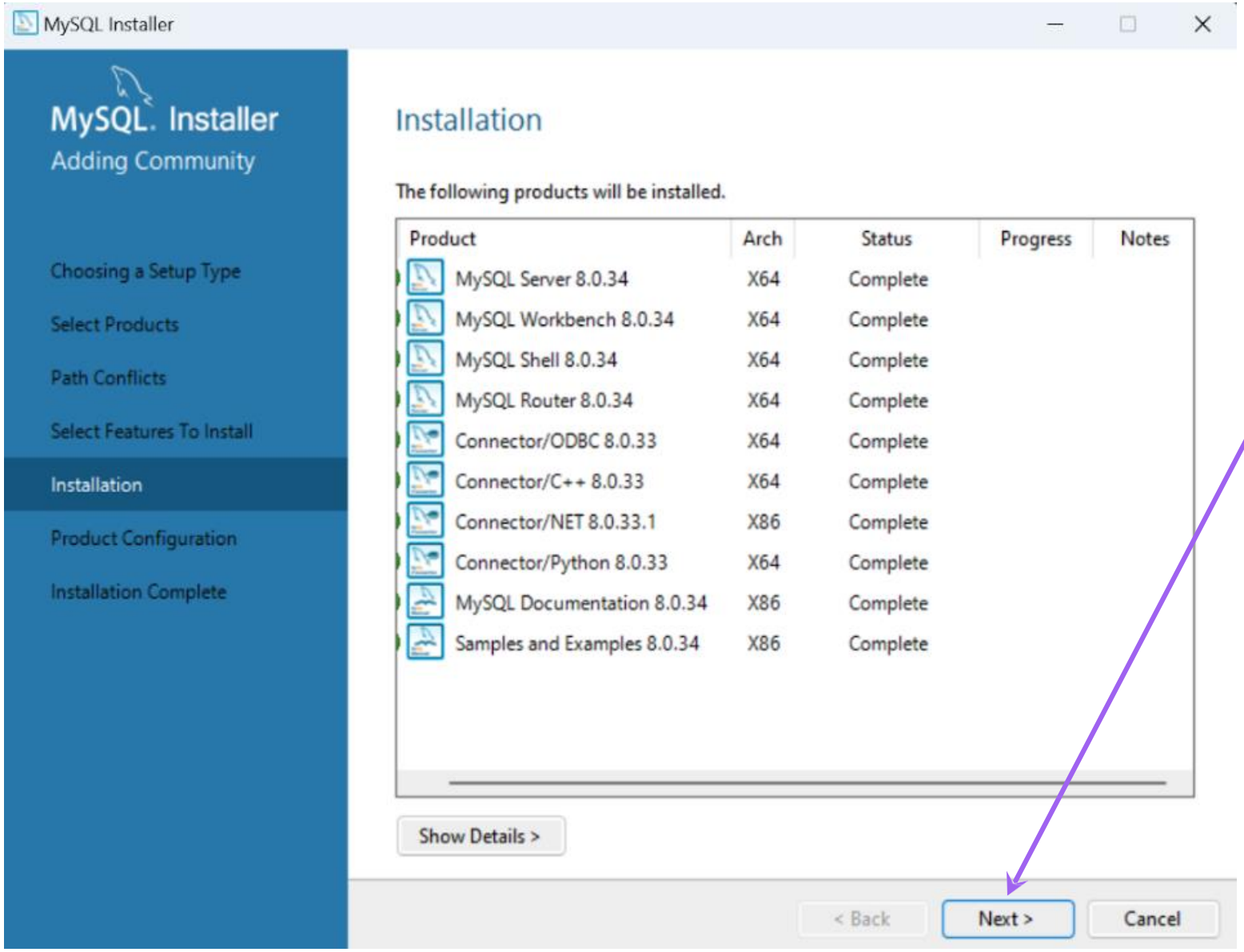
Select "Custom" as the setup type and click Next.



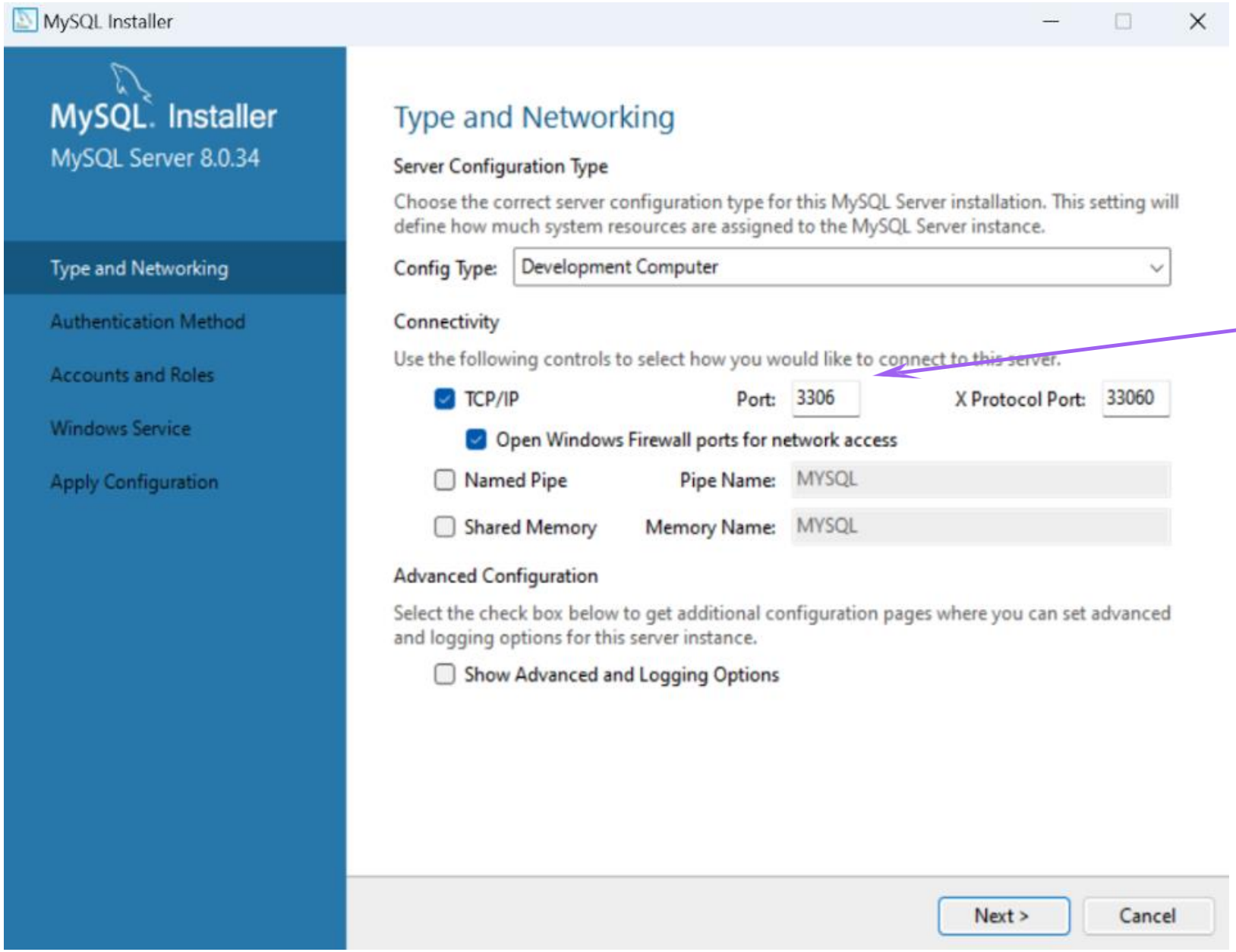
Select the products to be installed and click Next.



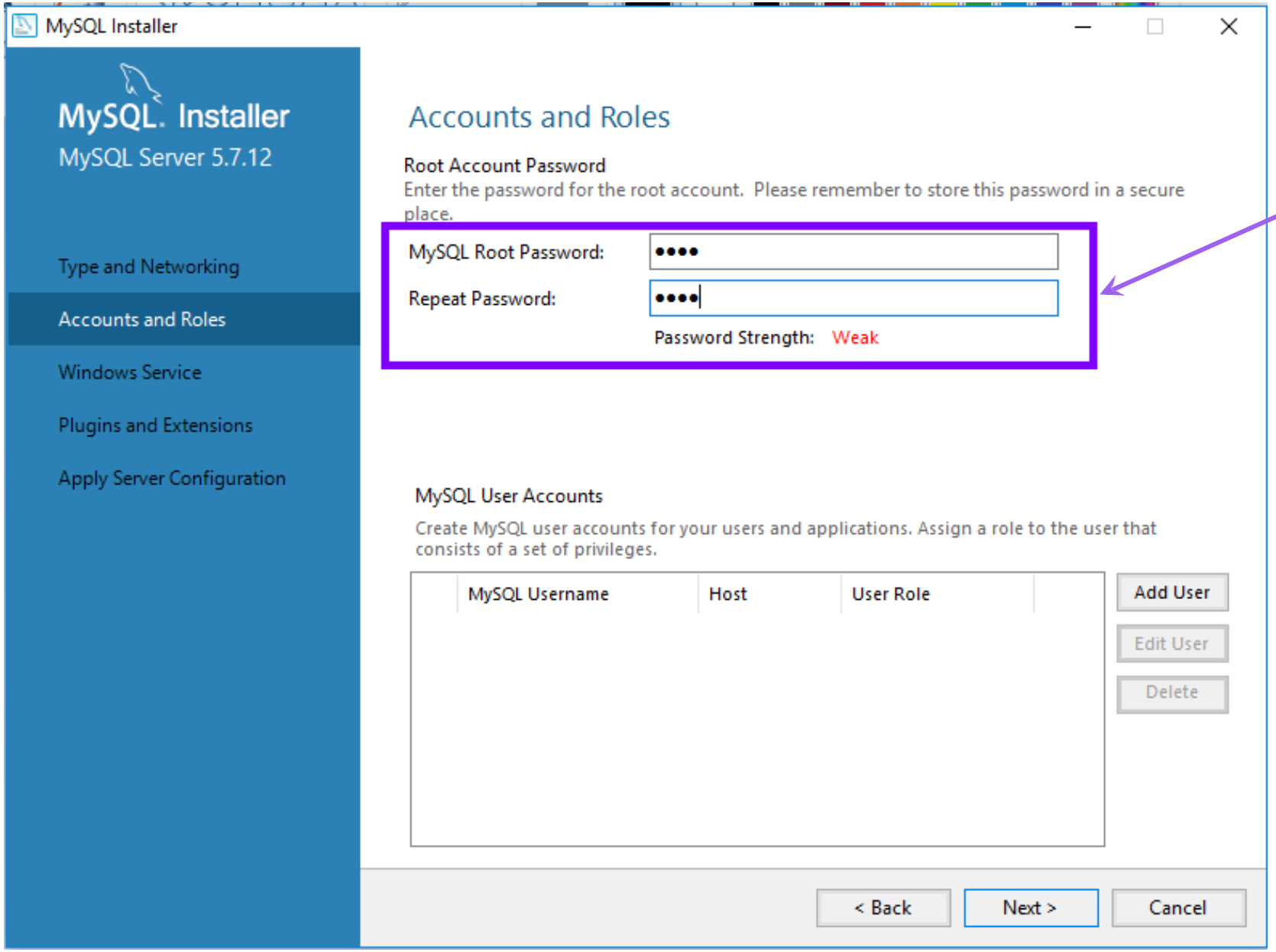
If you get a “Try Again “ option here instead, then click on the try again option to download the products.



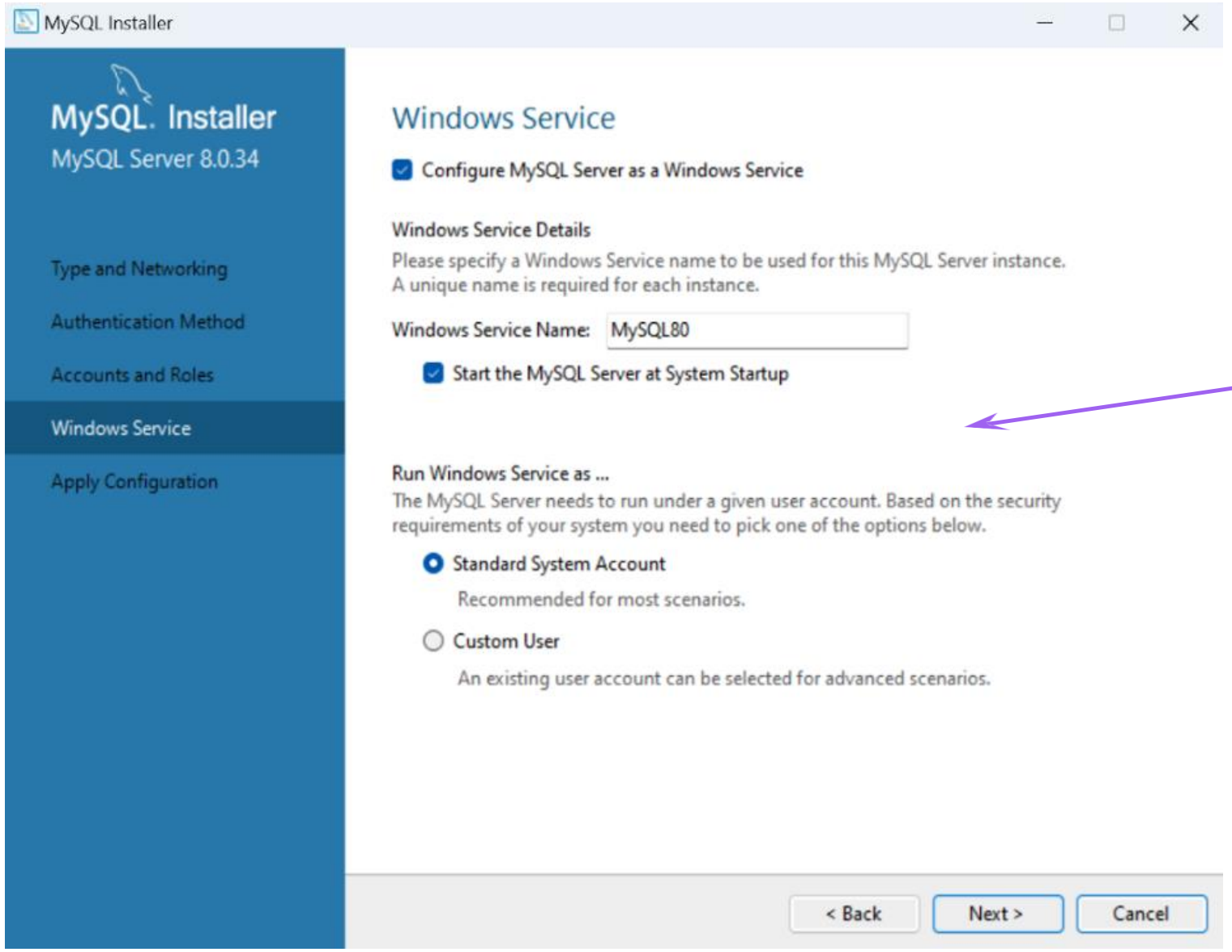
Once all the products are installed and the status is shown as “complete” proceed with the next option



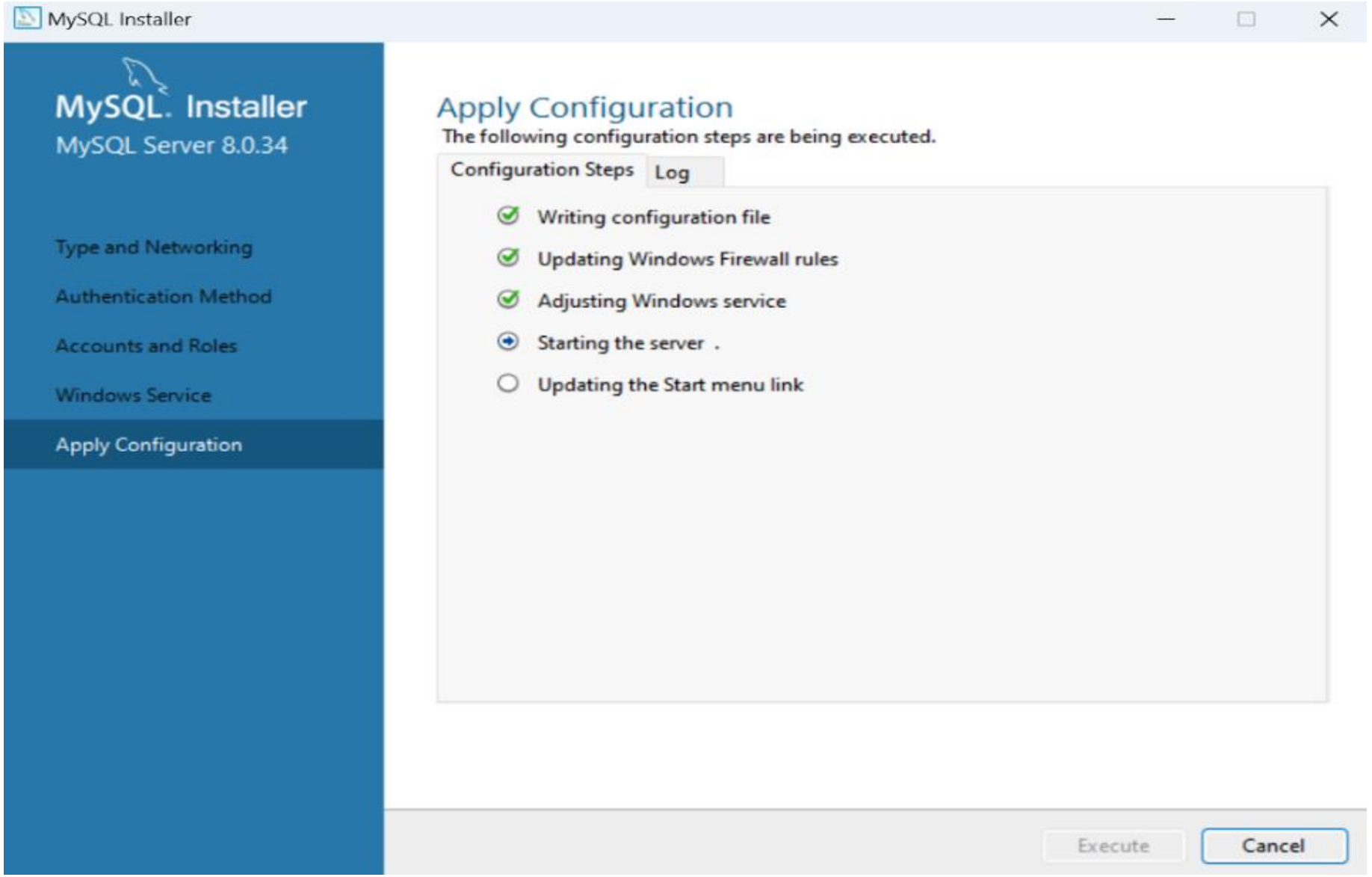
Port to connect to MYSQL

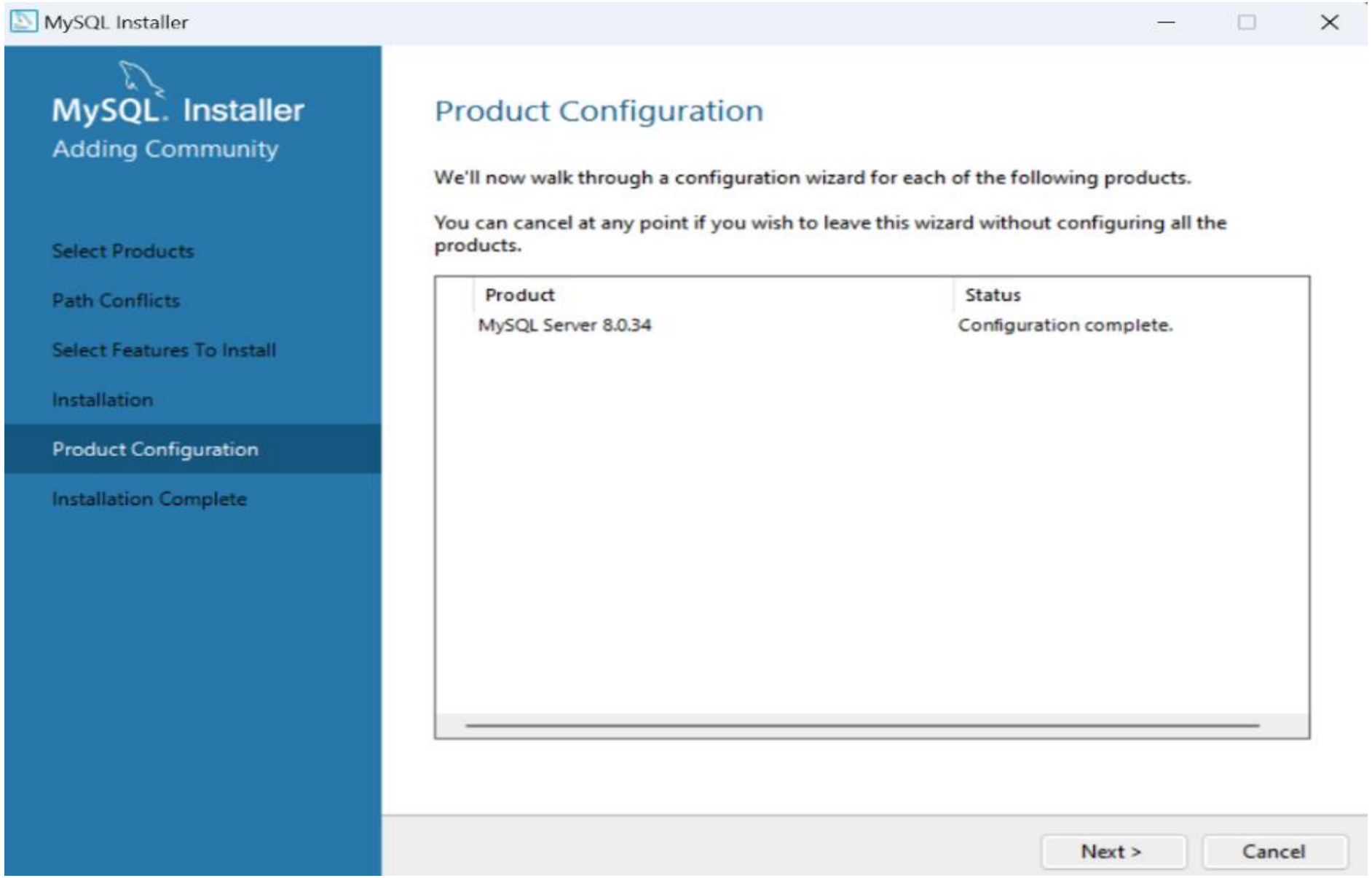


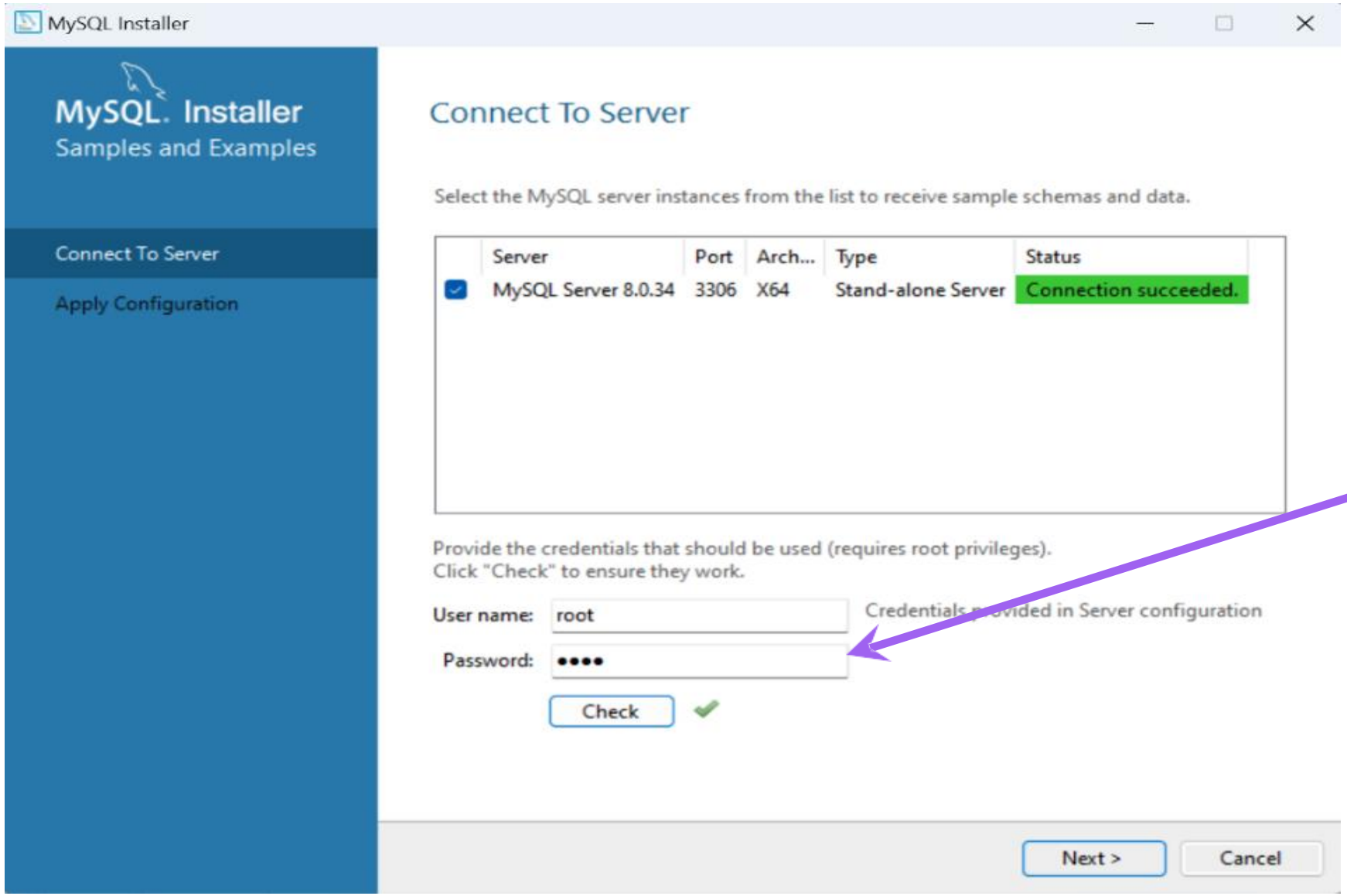
Enter the password as root. You will need to store these credentials as you need them to access MySQL.



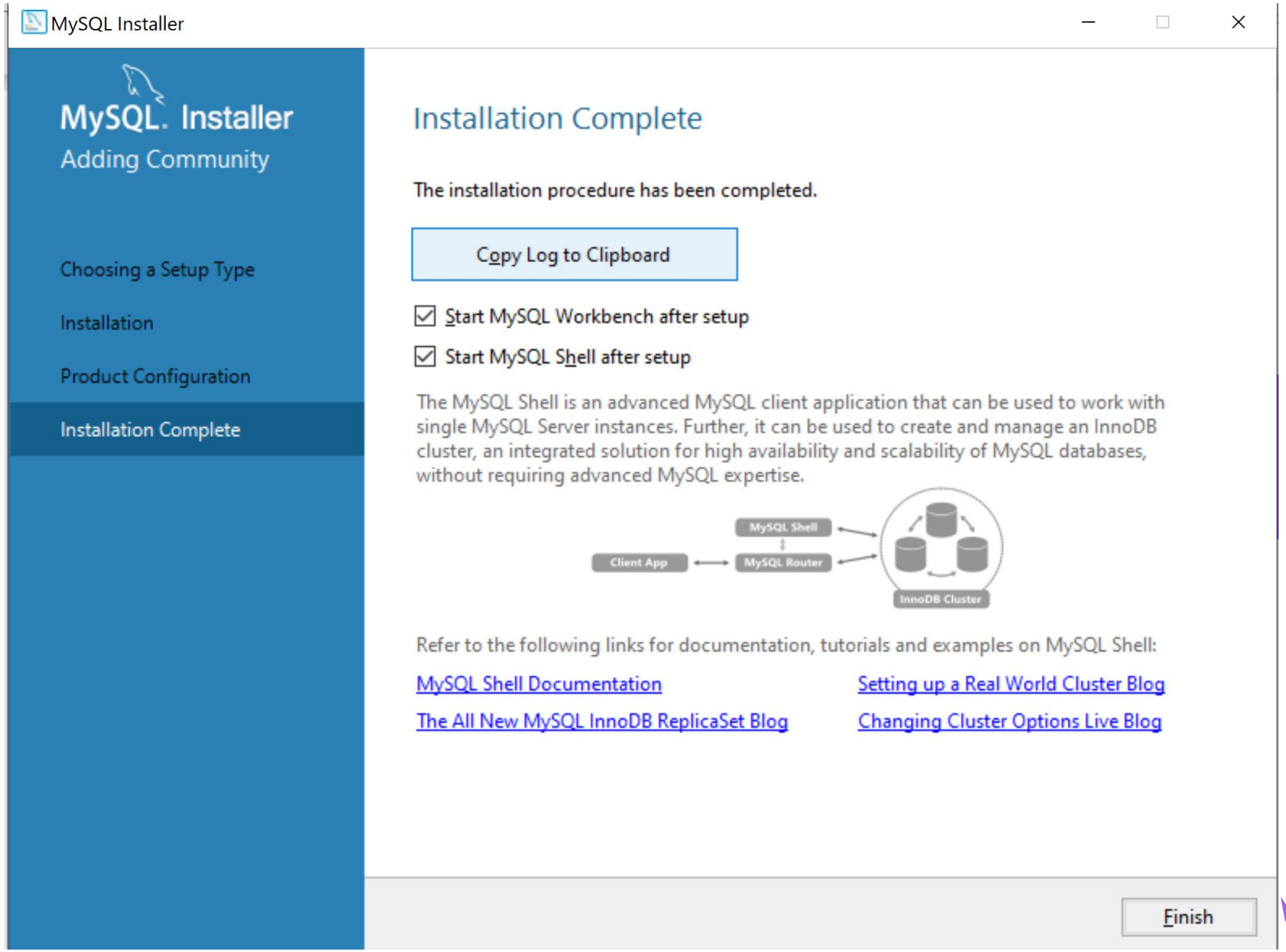
Windows Service name is MYSQL80, click next


MySQL Installer
MySQL. Installer
MySQL Server 8.0.34
Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Apply Configuration
Apply Configuration
The following configuration steps are being executed.
Configuration Steps
Log
Writing configuration file
Updating Windows Firewall rules
Adjusting Windows service
Starting the server .
Updating the Start menu link
Execute
Cancel


MySQL Installer
MySQL. Installer
Adding Community
Select Products
Path Conflicts
Select Features To Install
Installation
Product Configuration
Installation Complete
Product Configuration
We'll now walk through a configuration wizard for each of the following products.
You can cancel at any point if you wish to leave this wizard without configuring all the products.
Product
Status
MySQL Server 8.0.34
Configuration complete.
Next >
Cancel



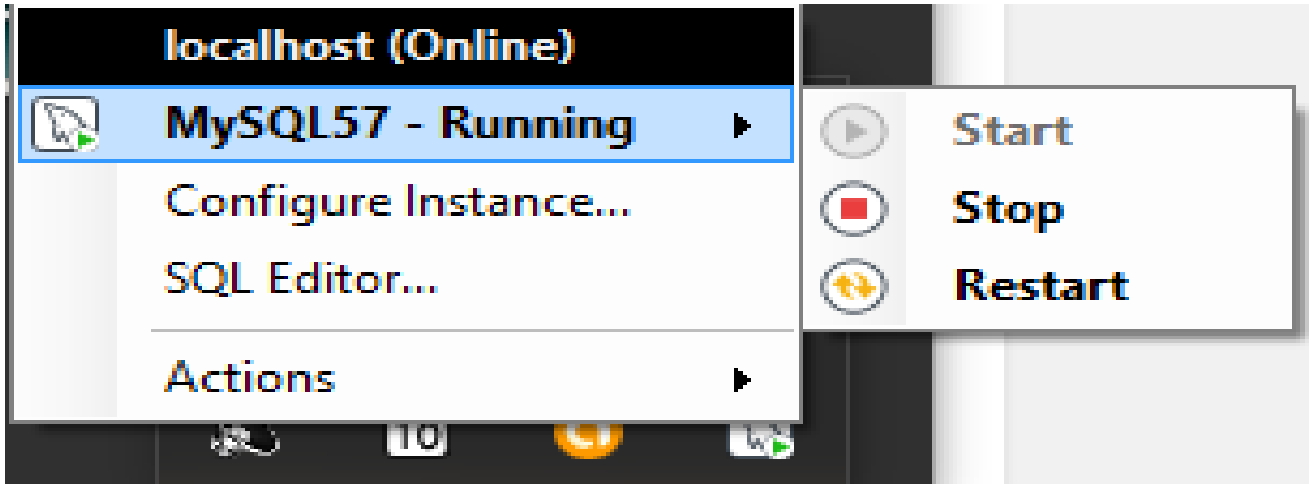
Enter the password you have set in previous step and click check to connect with MySQL server.



Now, the installation is completed.
.

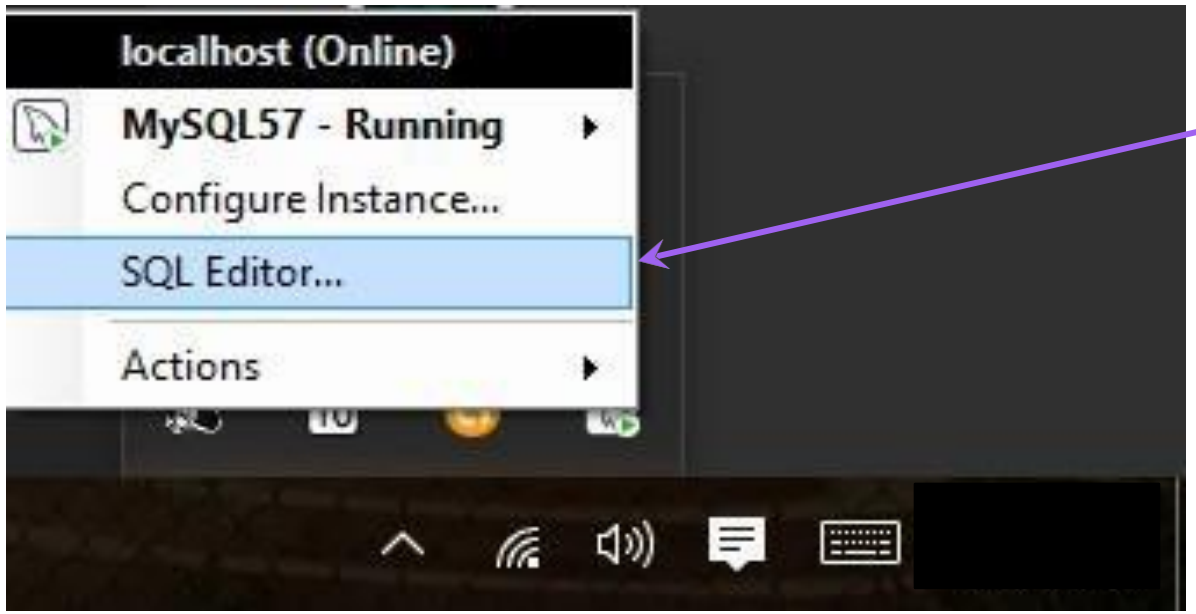
Post Installation…

- After the installation, we can monitor MySQL server by accessing MySQL notifier from the task bar.
- If MySQL notifier is not installed, download it by clicking on the link https://downloads.mysql.com/archives/notifier/



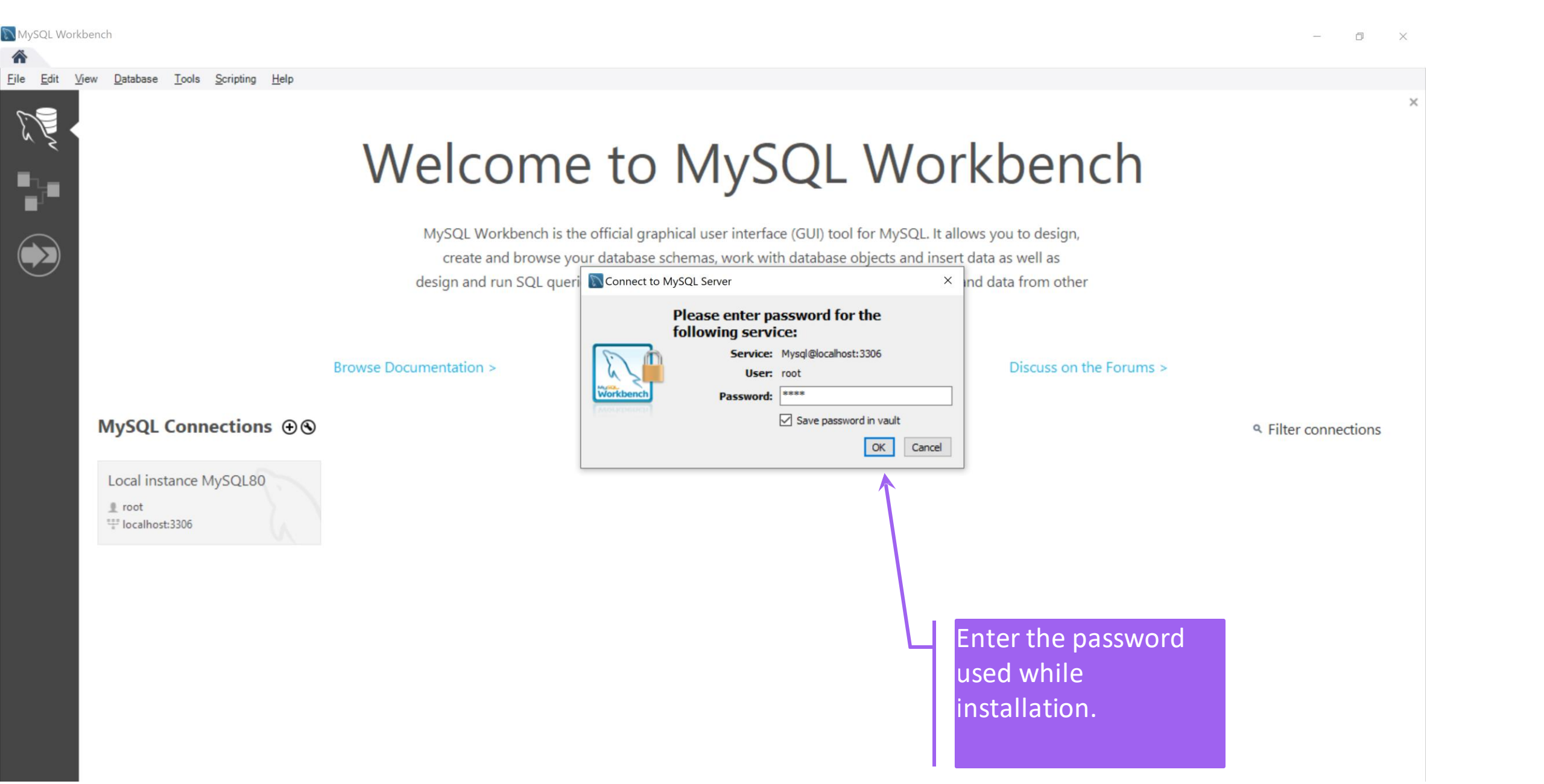
- Through the MySQL notifier, we could start, stop or restart MySQL components.

- To open the Workbench, where we write and execute several queries, we should click on “SQL Editor”.



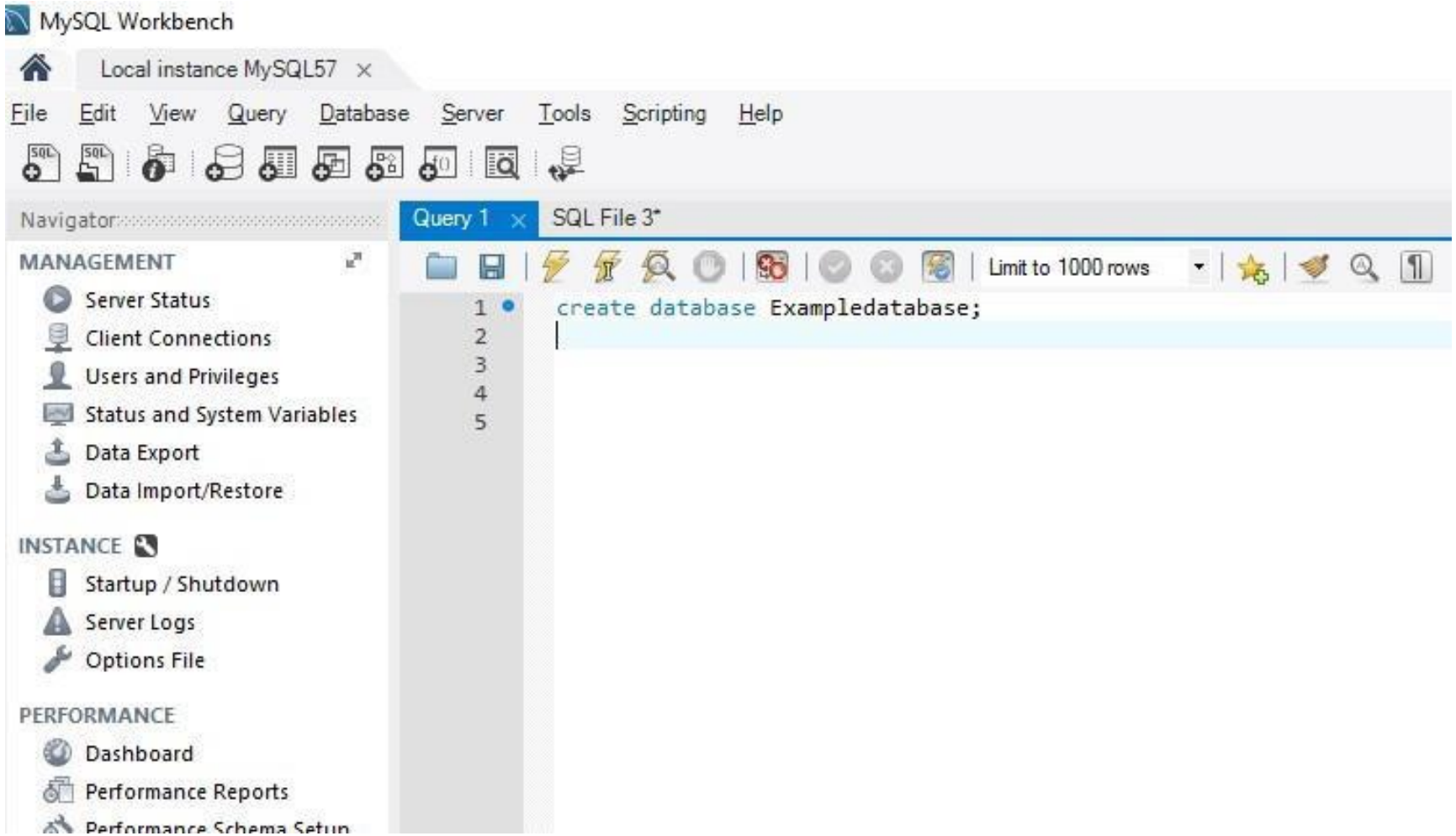
Now, the installation is completed.
.

- Once the workbench is opened, we need to enter the credentials (used during installation) to access workbench. Following screenshot exhibits the same.


MySQL Workbench
File Edit View Database Tools Scripting Help
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 queri
nd data from other
Connect to MySQL Server
Please enter password for the following service:
Service: Mysql@localhost:3306
User: root
Password: ****
Save password in vault
OK
Cancel
Browse Documentation >
Discuss on the Forums >
MySQL Connections
Filter connections
Local instance MySQL80
root
localhost:3306
Enter the password used while installation.

Executing queries in MySQL Workbench

Goto File and select “New Query Tab “ to write and execute the queries.

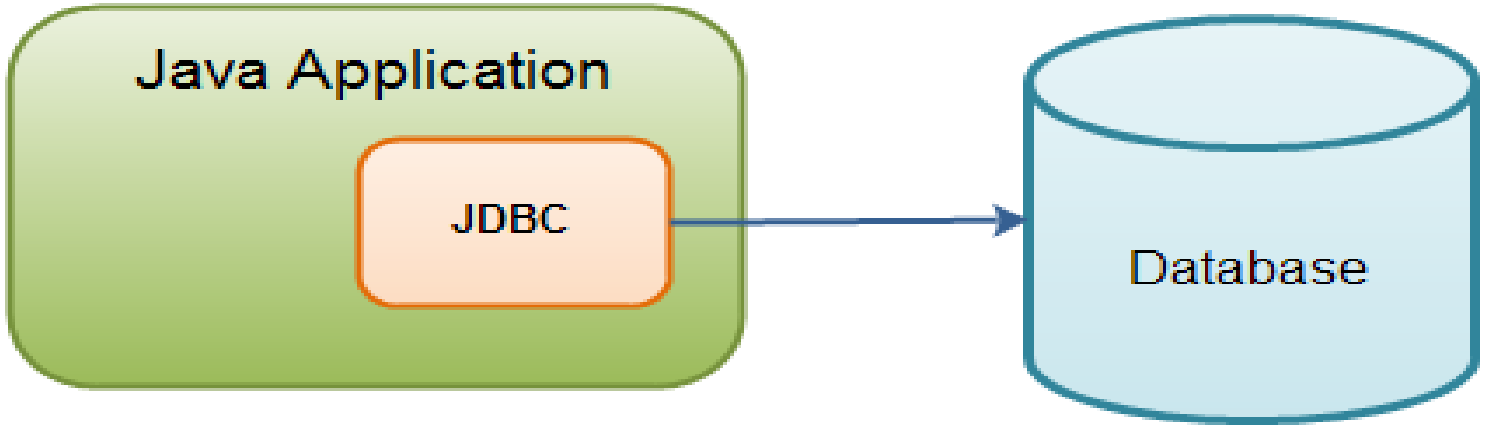


Executing queries using a Java Application

Now let us write and execute the SQL queries using a Java Application.

How do we do it?

-> By using a JDBC Driver. JDBC stands for Java Database Connectivity.



JDBC Driver Types

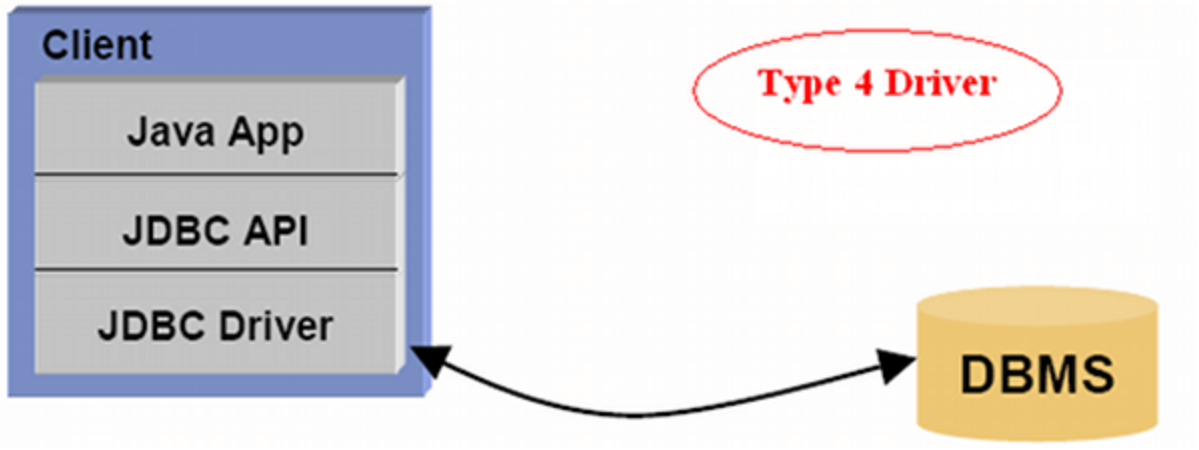
Type 1: JDBC-ODBC Bridge driver (Bridge)

Type 2: Native-API/partly Java driver (Native)

Type 3: All Java/Net-protocol driver (Middleware)

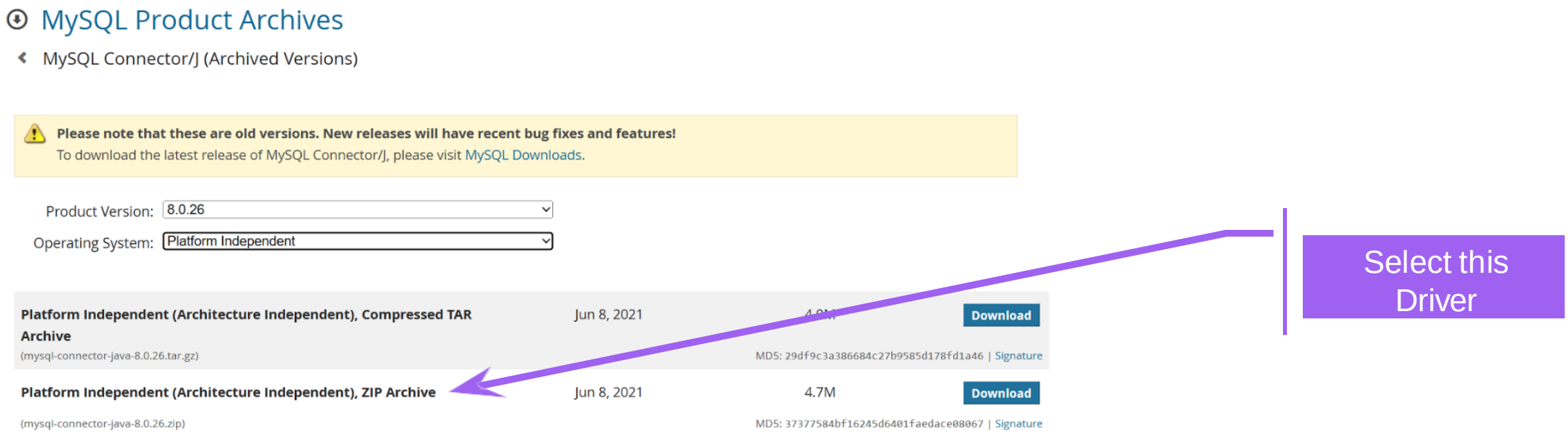
Type 4: All Java/Native-protocol driver (Pure).

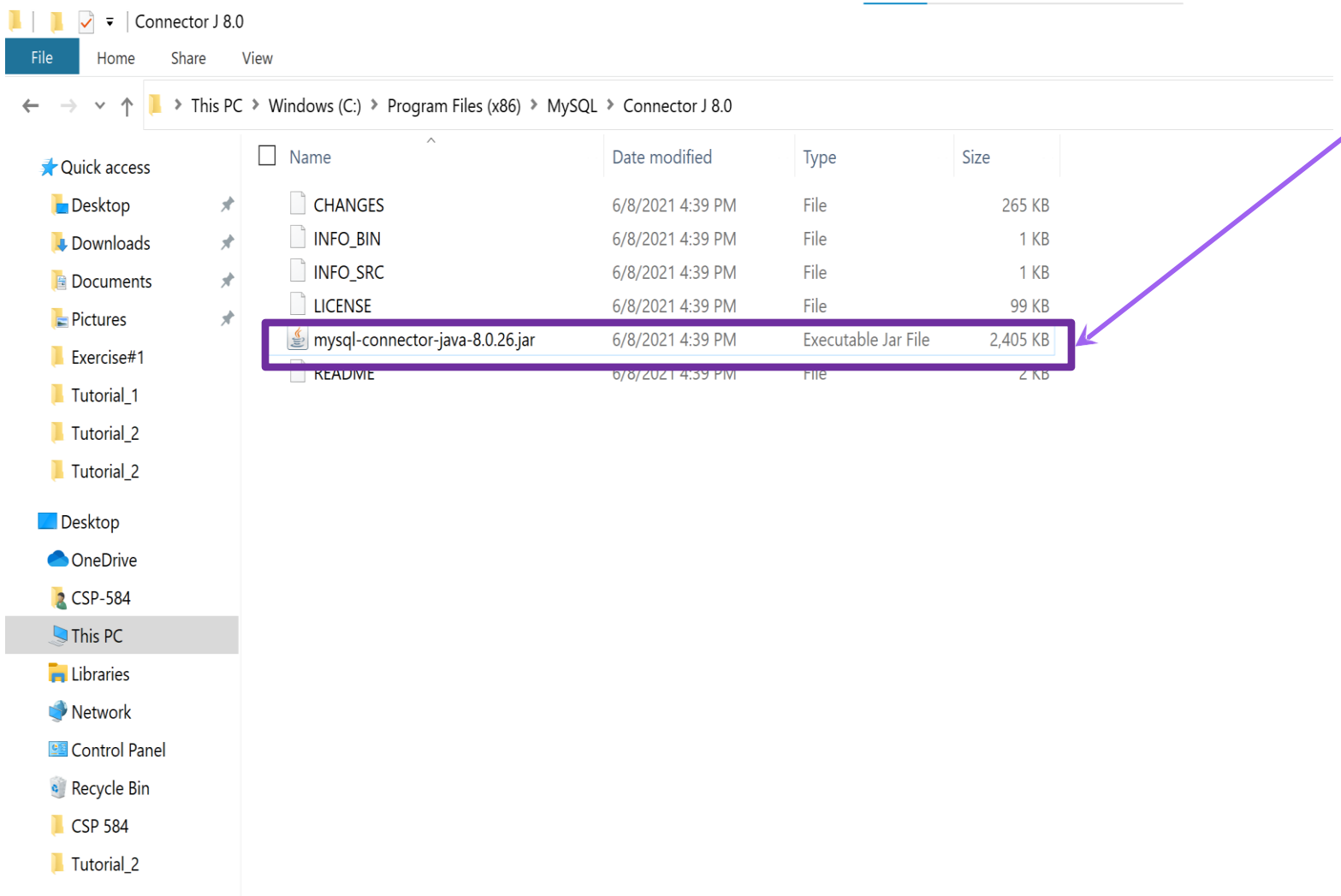
We will be using the Type 4 driver , as we use libraries to communicate directly with the database server.



Download the Jar file:

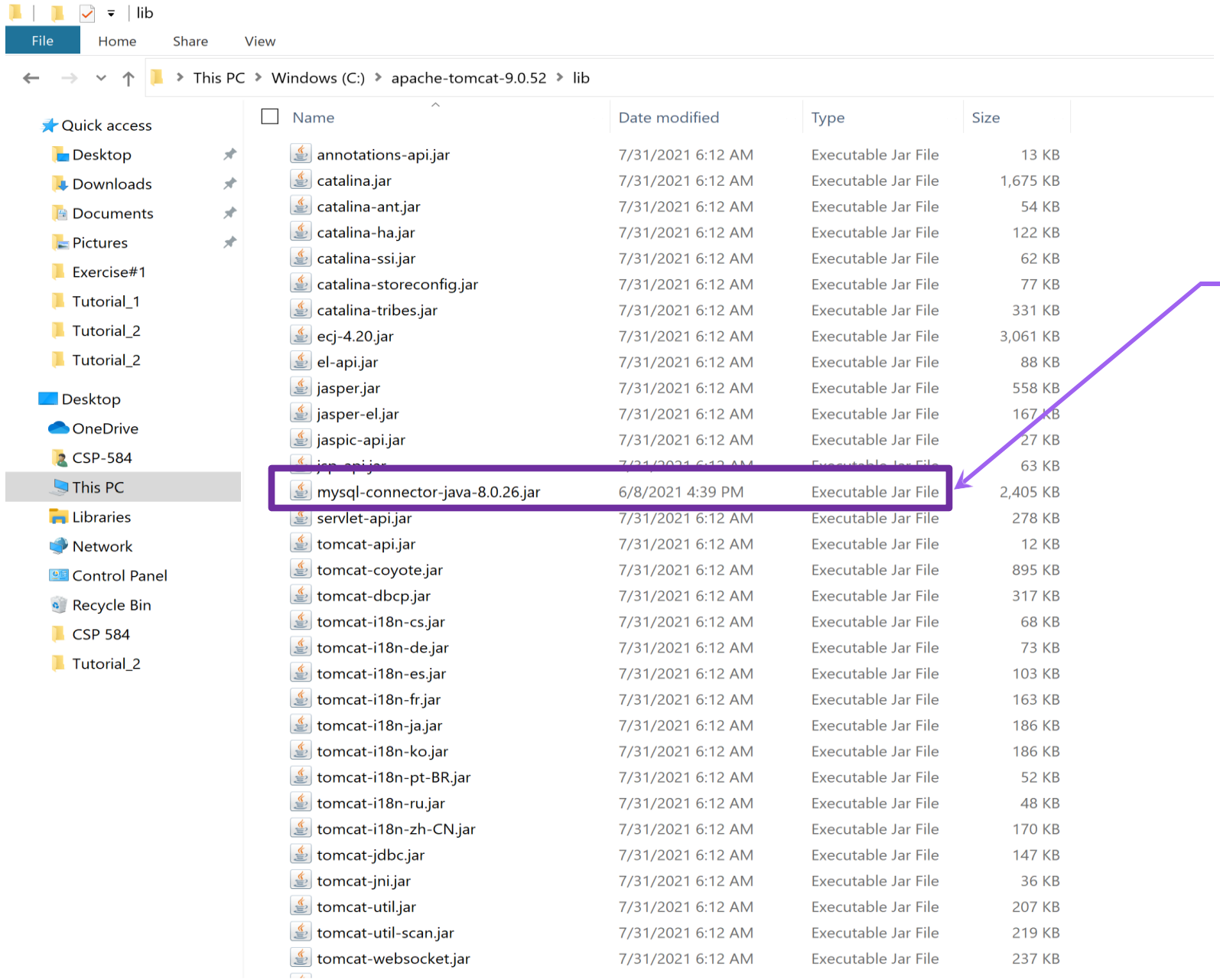
We need a jar file mysql-connector-java-8.0.26.jar to compile and execute a application. If you want to compile the examples from the command line, go to the site https://downloads.mysql.com/archives/c-j/ and download the MySQL connector.





By default the mysql connector jar file is downloaded along with Mysql Server, Workbench etc. It is located in the following directory:

C:\Program Files (x86)\MySQL\Connector J 8.0



Copy this mysql connector jar file and paste it inside C:\apache-tomcat-9.0.52\lib

Steps to Execute the application

Option 1: Executing the Program by adding the jar file path in a bat file .

Add the whole path for jar file in CLASSPATH inside your env-setup.bat file

The location of the JAR files highlighted will differ based on where they are present on your computer
Please make sure you do the changes accordingly

env-setup-for-tomcat.bat - Notepad

File Edit Format View Help

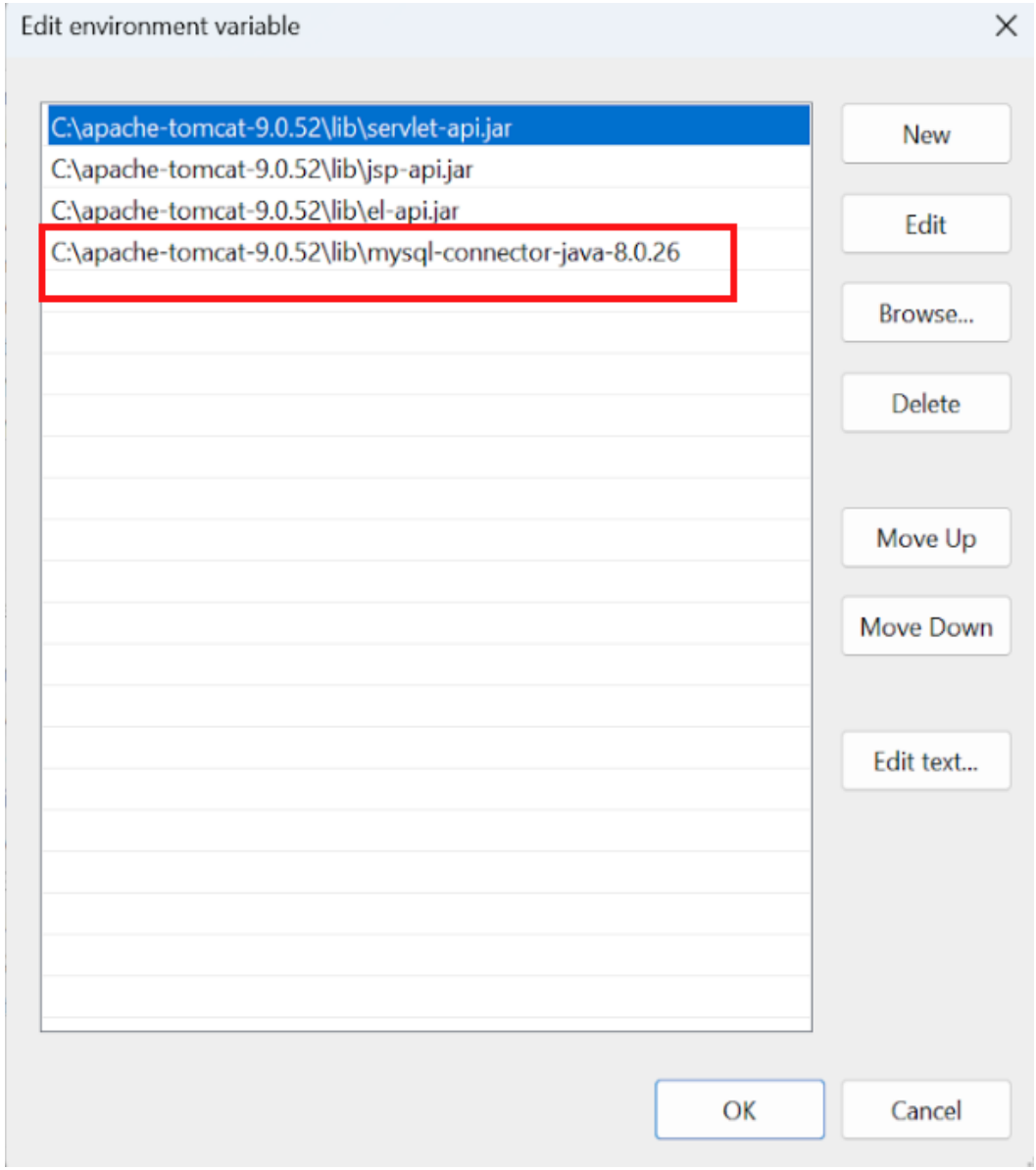
```
set JAVA_HOME=C:\Program Files\Java\jdk-14.0.2

set PATH="C:\Program Files\Java\jdk-14.0.2\bin";%PATH%

set CLASSPATH=.;C:\apache-tomcat-9.0.52\lib\servlet-api.jar;
               C:\apache-tomcat-9.0.52\lib\jsp-api.jar;
               C:\apache-tomcat-9.0.52\lib\el-api.jar;
               C:\apache-tomcat-9.0.52\lib\commons-beanutils-1.8.3.jar;
               C:\apache-tomcat-9.0.52\lib\mysql-connector-java-8.0.26.jar;

set ANT_HOME=C:\apache-tomcat-9.0.52

set TOMCAT_HOME=C:\apache-tomcat-9.0.52

set CATALINA_HOME=C:\apache-tomcat-9.0.52
```

Executing by adding jar file in class path in env-setup.bat file

Option 2 : By setting a classpath variable.

Steps:

- Goto Control Panel -> system -> Advanced system settings -> Environment Variables.
- Under User variables, choose new, and create a new variable called Classpath = “The full pathname where jar file is present in the system “
- Click on save.
- Then OK.



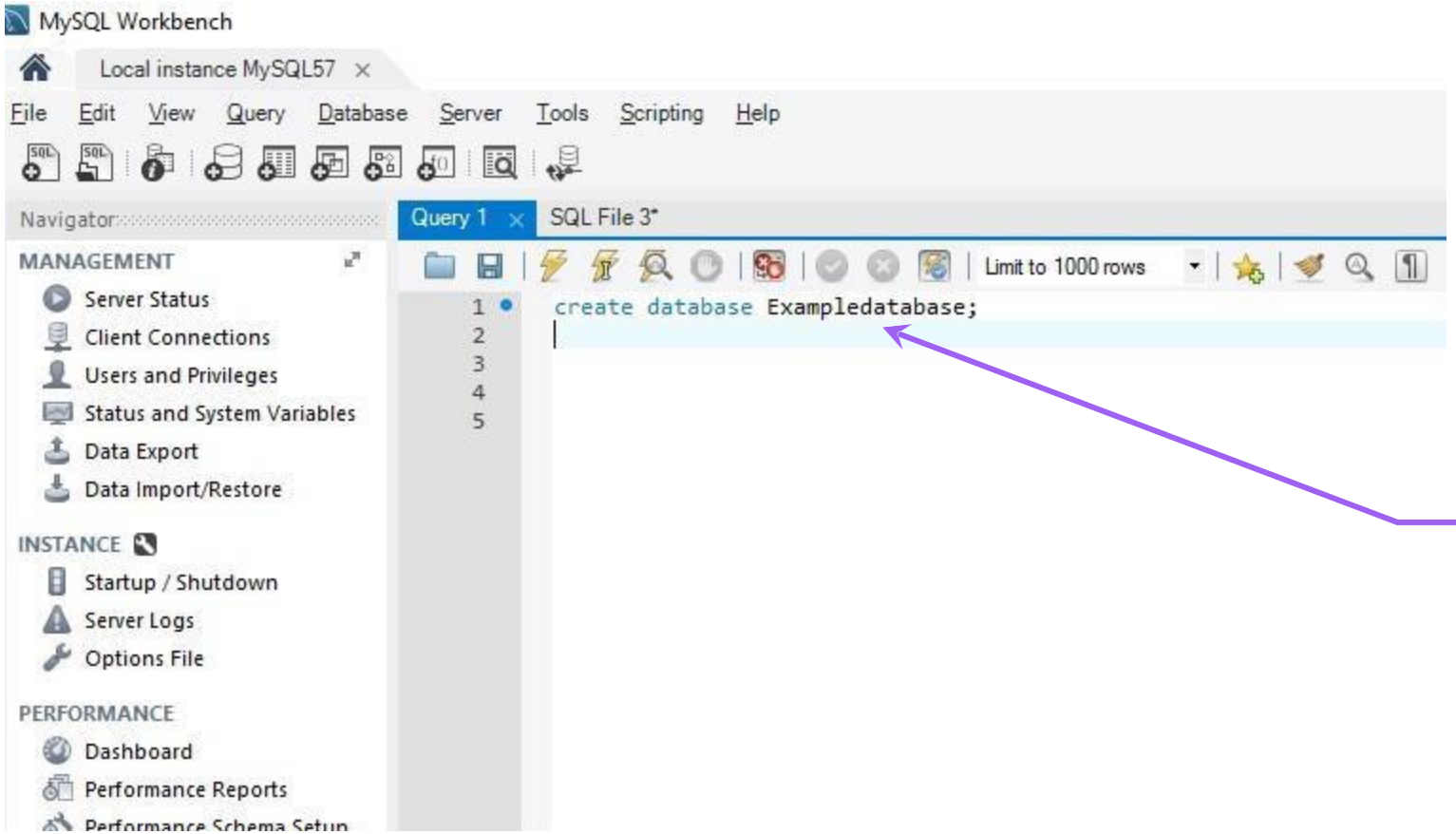
A walkthrough example

In this example,

- First, we will create a database called “exampledatabase” using MySQL workbench.
- Secondly, we will create a table “Registration” within the exampledatabase and store customer login details
- Third, we will create a table “CustomerOrders” within the exampledatabase and store order details for game speed application.

A walkthrough example

- ***Step 1***: Create a database called “exampledatabase” in the workbench space, and execute the SQL command.

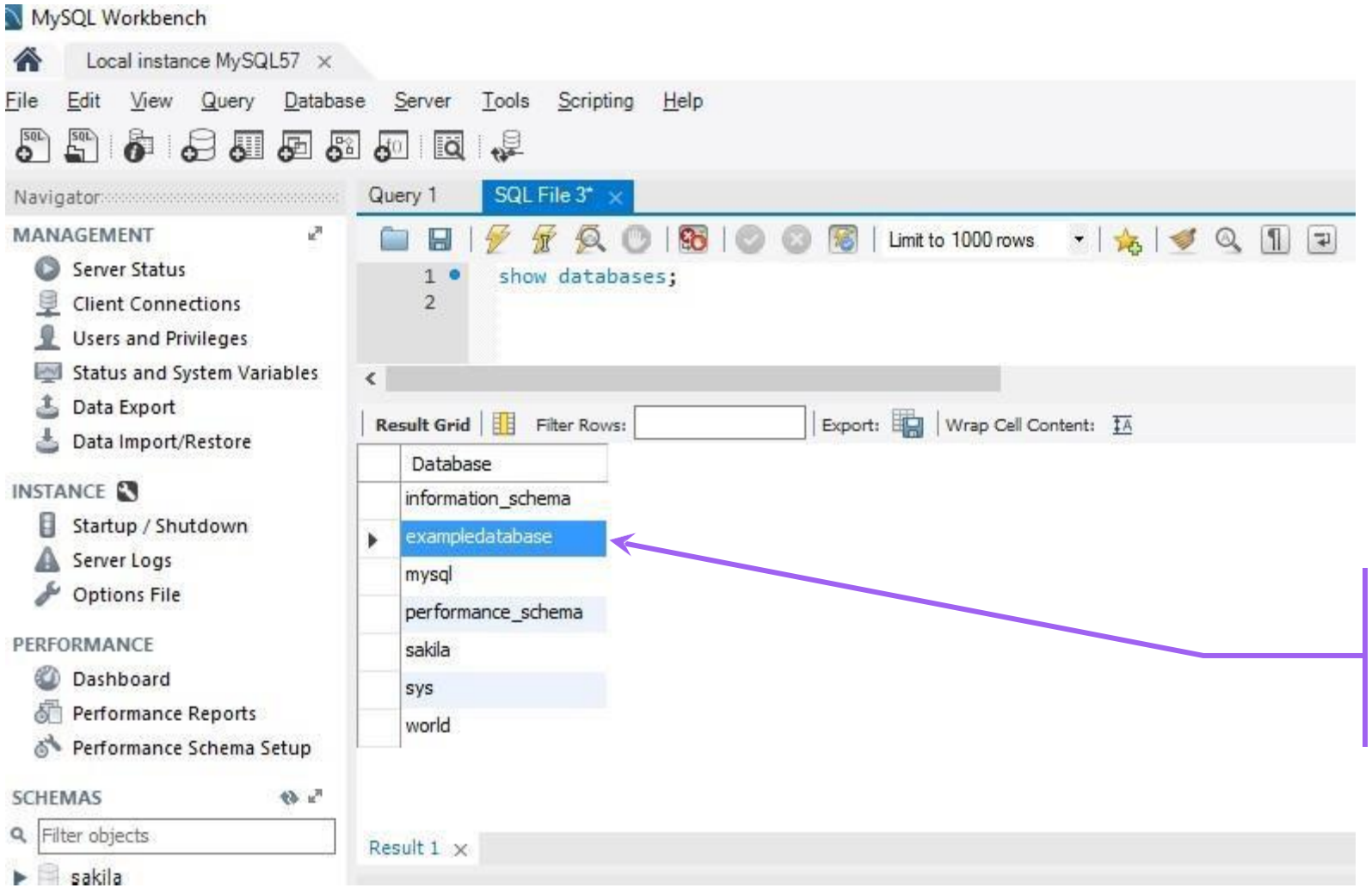


Query to create a new database

A walkthrough example

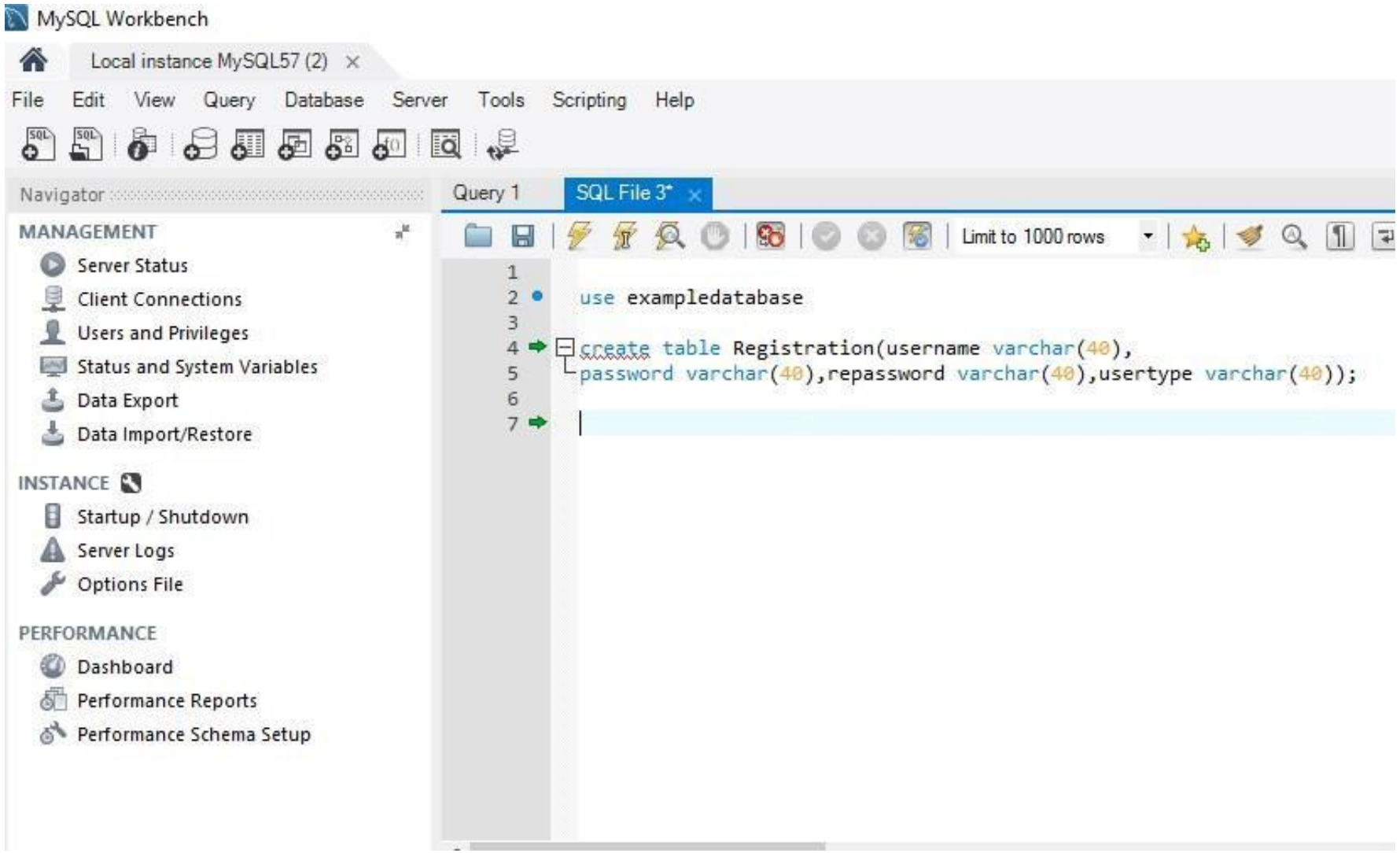
Write and execute the following commands to check if the database is created.

-> show databases;



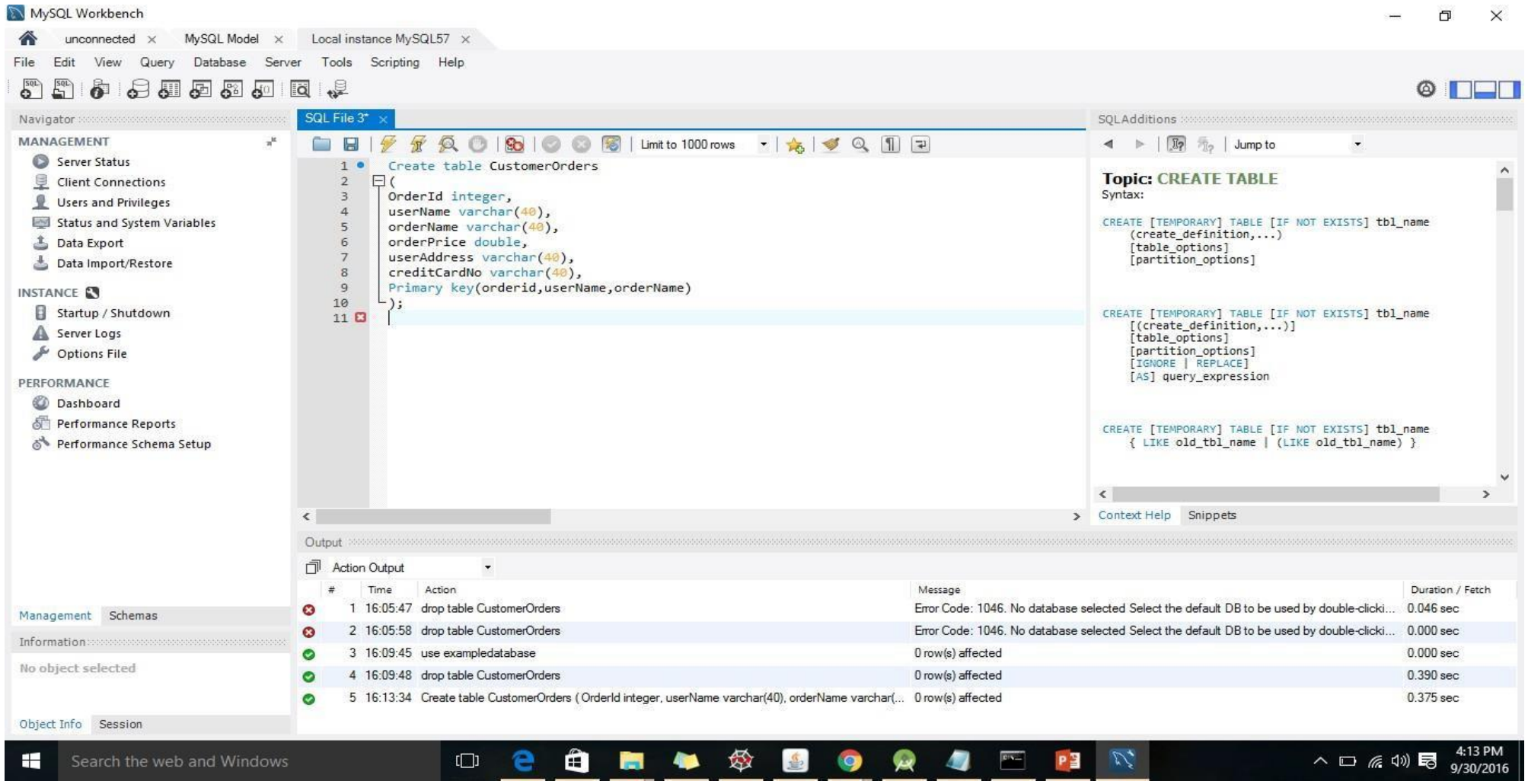
Our Exampledatabase is created

Step 2: Create a table "Registration" in the workbench space, and execute the SQL command.



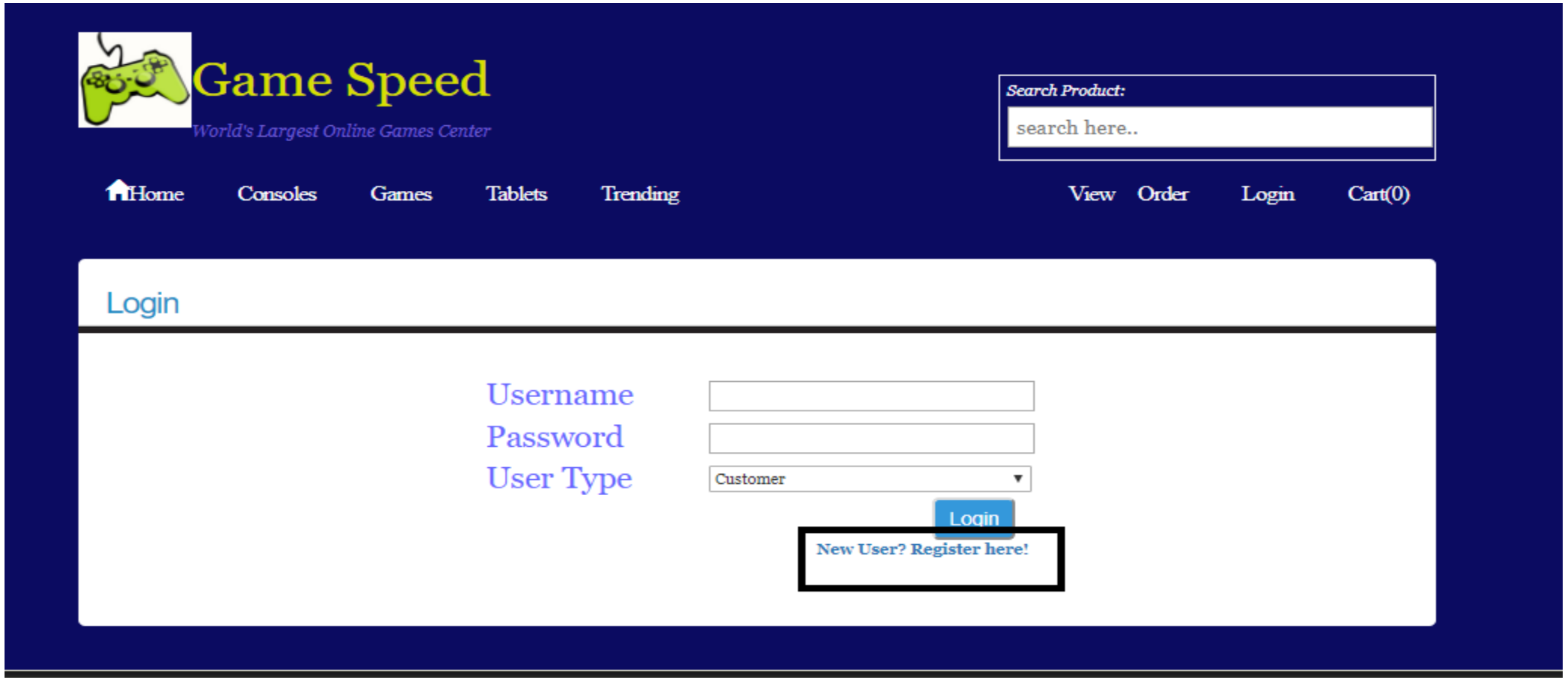
Step 3: Create a table called “CustomerOrder” in the workbench space, and execute the SQL command.

Specify the id as primary key for table

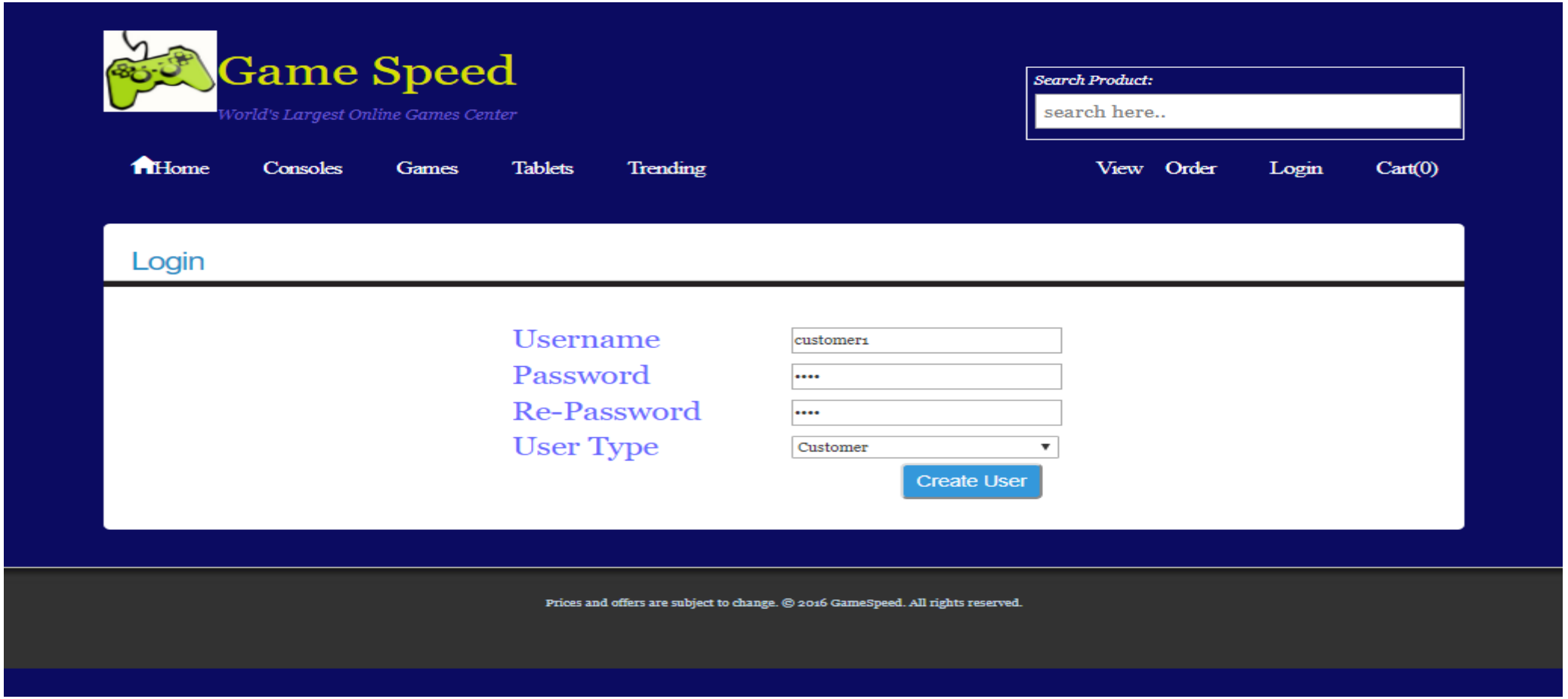


Example –Registration

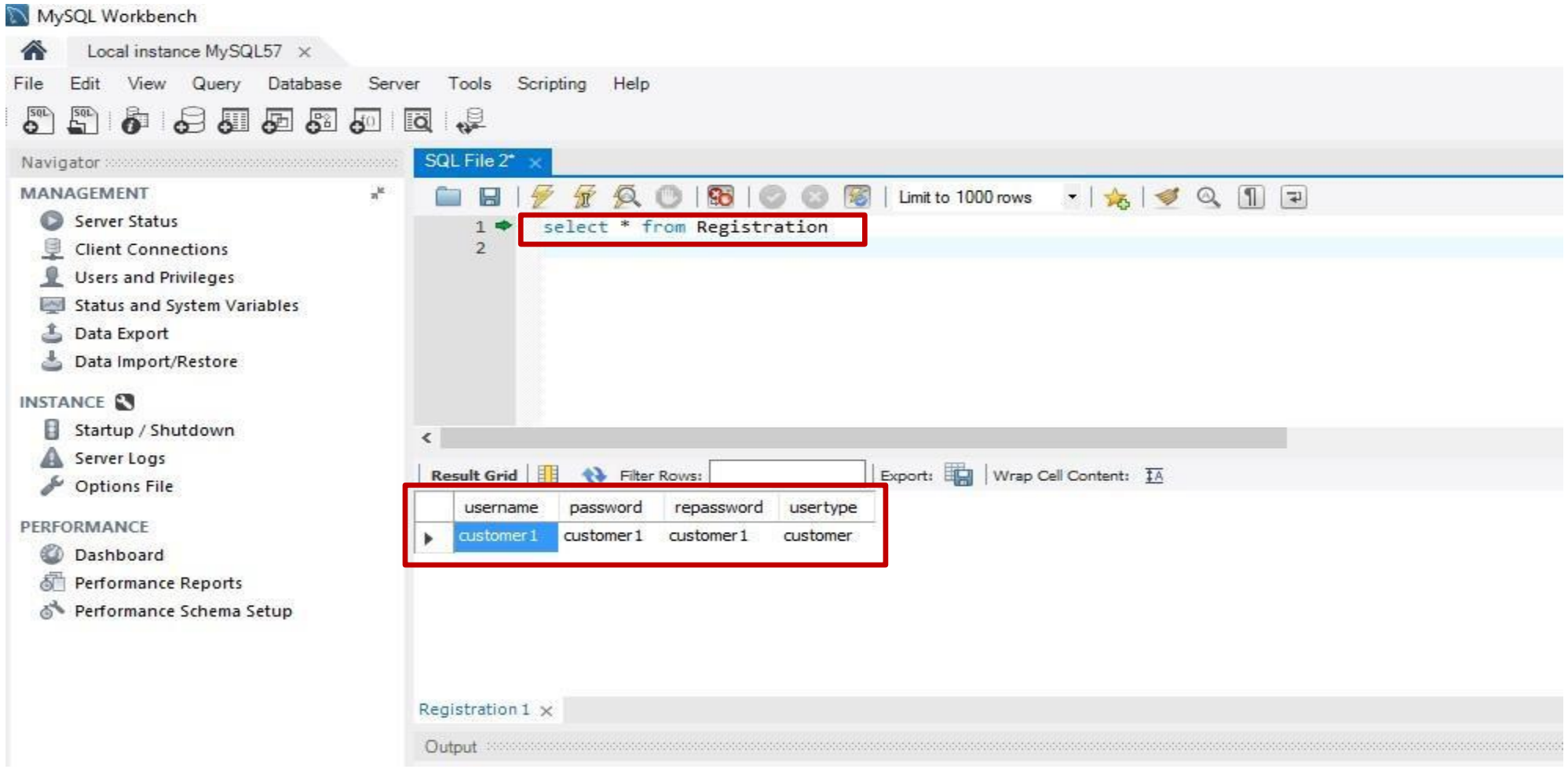
- New User can Register into Website by Clicking on the Register here link On clicking the button user is directed to Registration page



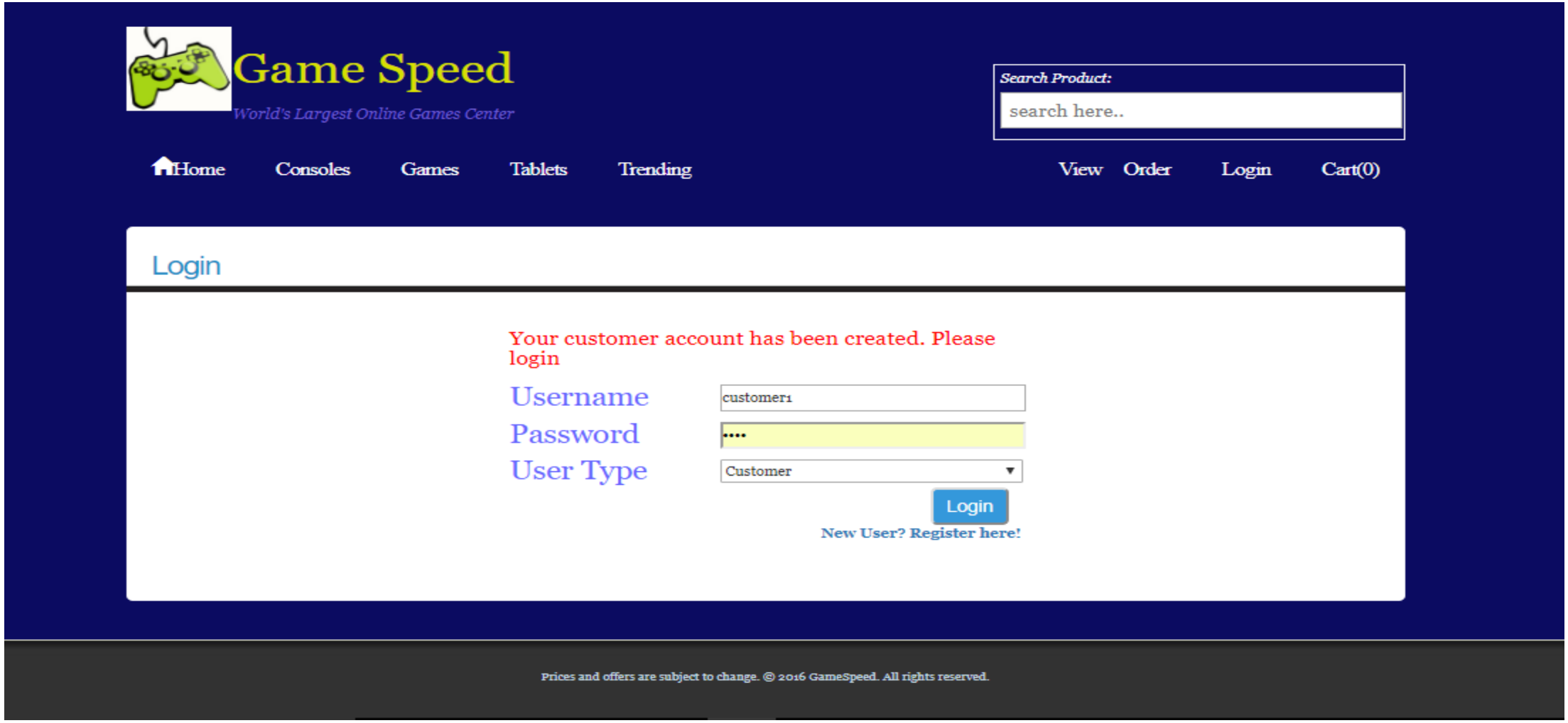
- User provides the login information
- On clicking create user button data is stored in Registration Table of My sql



- User data is stored in Registration table you can check using the select query in workbench to check if all the column values are stored properly



- After an Account is created for user in Registration table user can login into website with the credentials

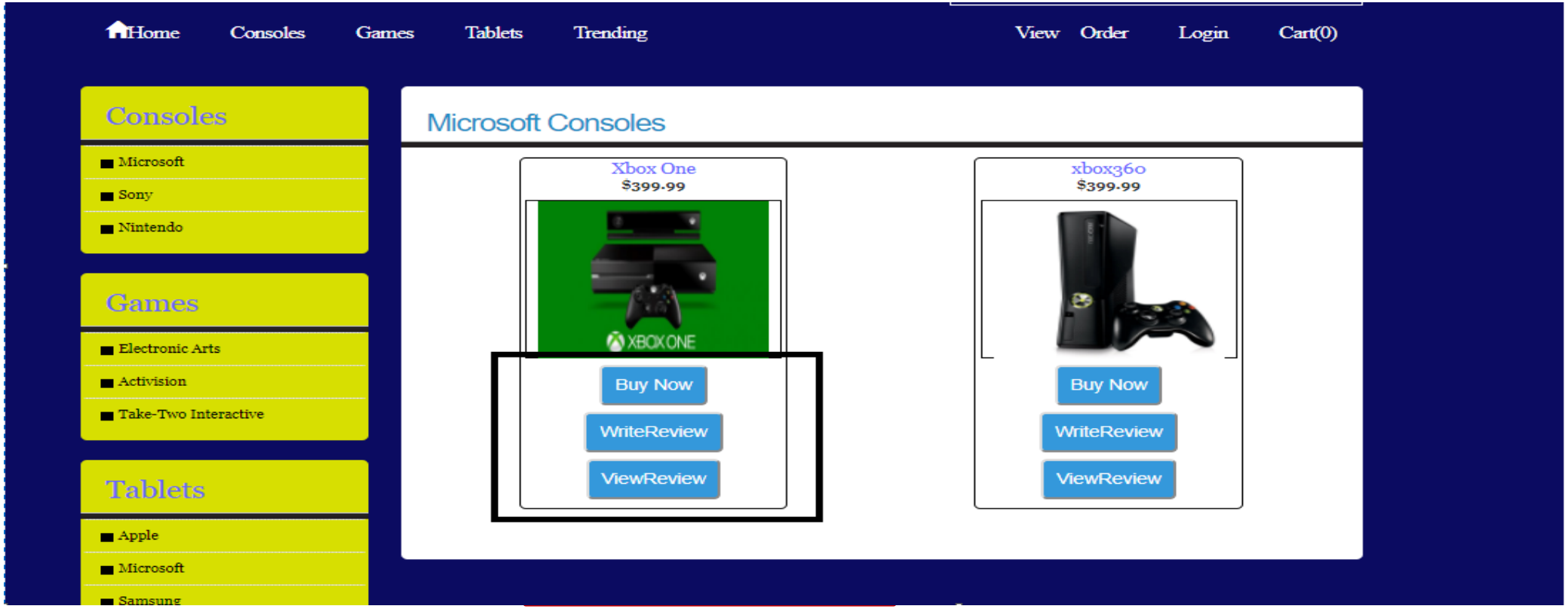


Example –Place Order

- Click on the products available in the navigation bar
- You can also select the products from the left navigation bar

Example – Place Order:

- Clicking on a product type will take you to the product page
- You have different options available such as buy a product , write reviews.

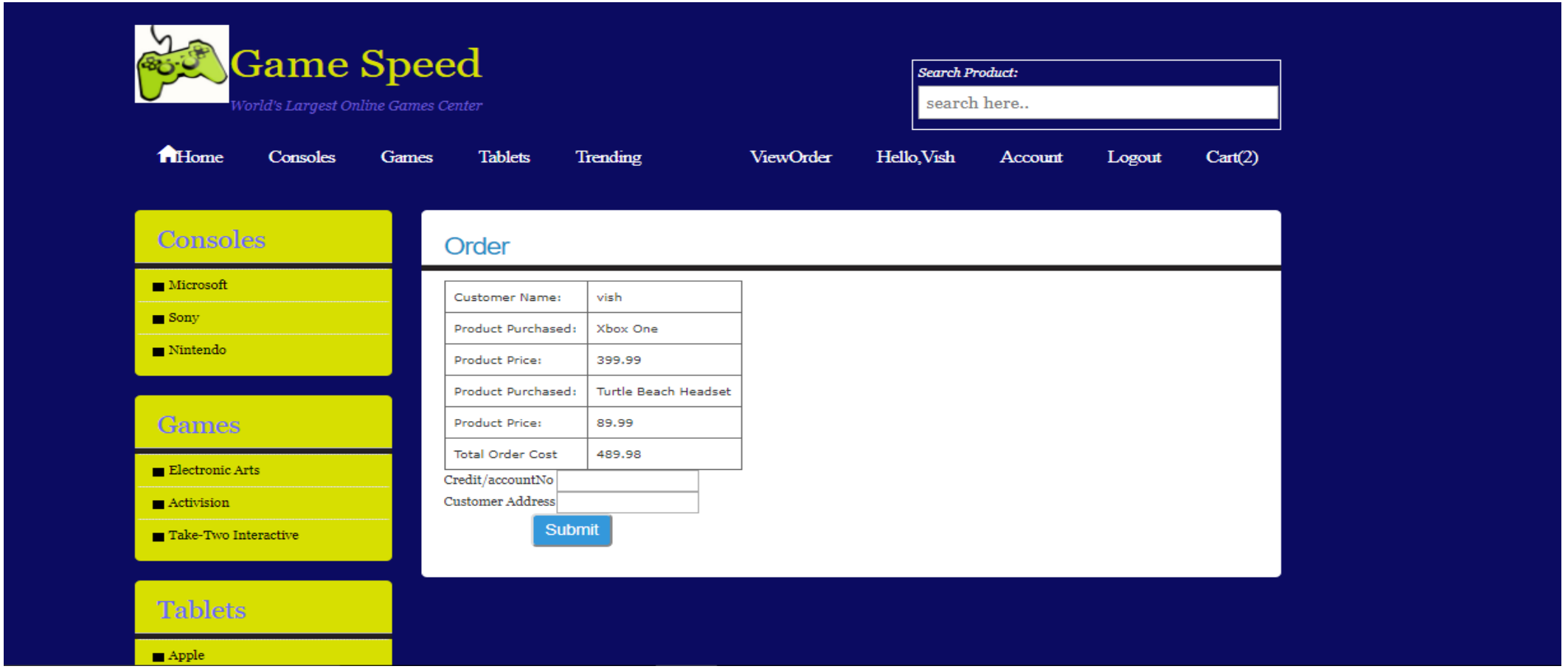


Example – Place Order:

- Click on ‘Buy’ button on the products page to purchase the product
- This should take you to a new page (Cart Servlet) where you can purchase the product
- Click on ‘Check Out’ to place the order for the selected product.

Example – Place Order:

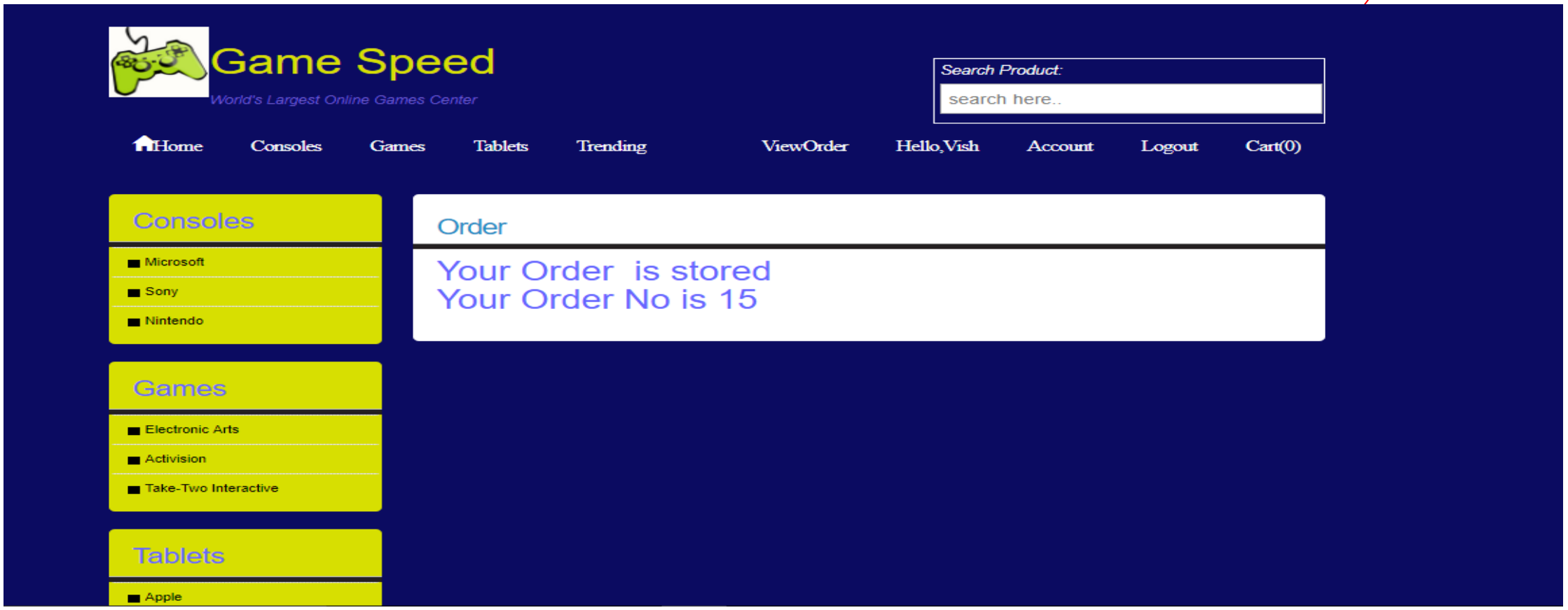
- Clicking on CheckOut Button will take you to the CheckOut webpage where you have to provide your credit card no and address information.



Example – Place Order:

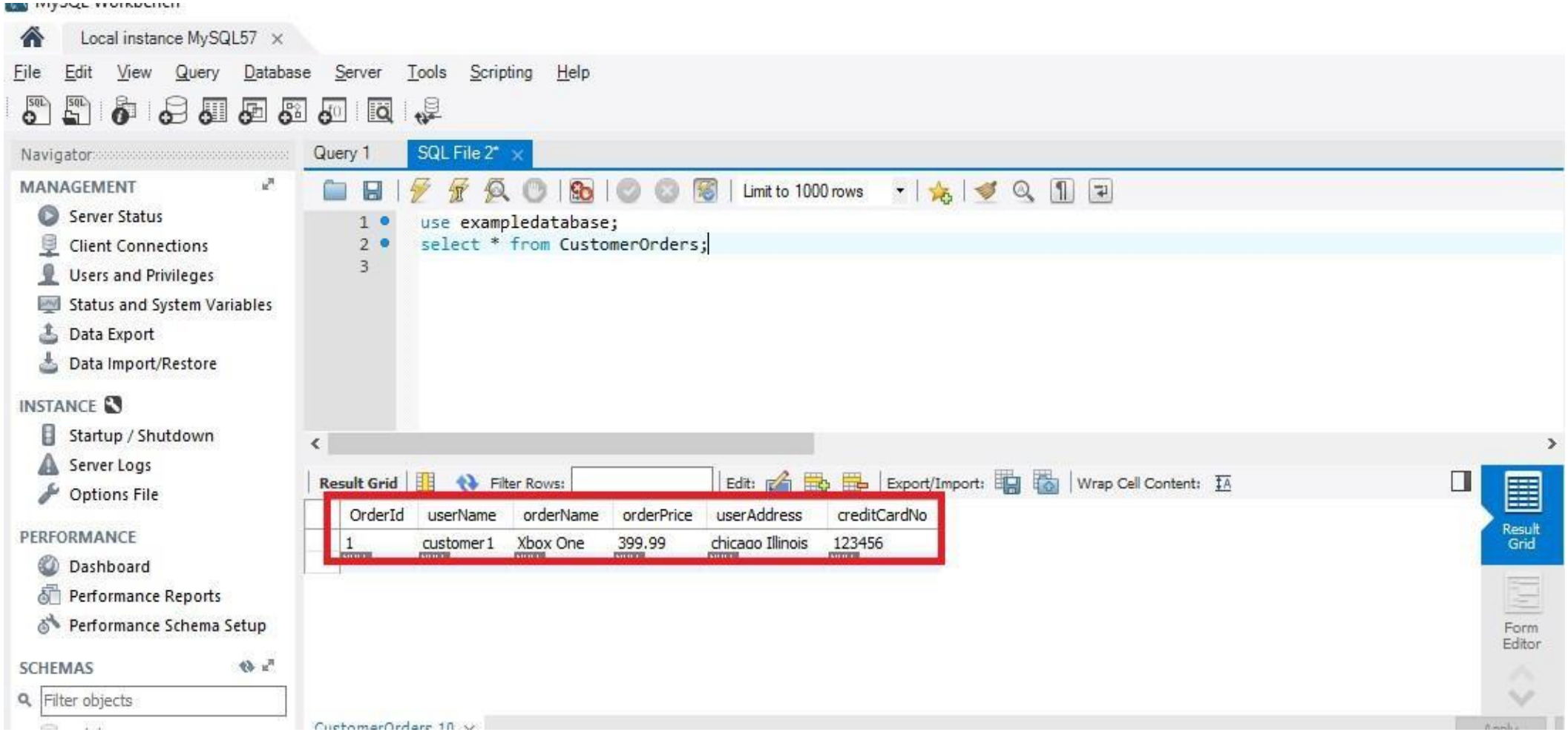
- On clicking the submit button from check out page order will be stored in My Sql database and order no will be generated

Order is Deleted from the cart



Example – Place Order:

You can Check if the order Stored by executing select Query in database using sql workbench



Servlets MySql Connection

- We will be using com.mysql.jdbc.Driver for connecting mysql from servlets

Syntax:

Connection conn=
Class.forName("com.mysql.jdbc.Driver").newInstance();

DriverManager.*getConnection() method is used to connect to my sql database*

Specify the database url , user name and password as parameter to the getConnection() method

```
conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/exampleda
tabase","root",“root");
```

Prepared Statement Execution

- Prepared Statement are used to generate Sql statement for a Query String in java
- Syntax:

  Prepared Statement ps=conn.prepareStatement(“select * from Registration where username=? And usertype=?”)

- Specify the Query String as parameter inside conn.prepareStatement() to perform insert or select into database from java
- ? Are place holder where we need to provide the value for a particular query
- In the next line ? We will replace with actual parameter value As

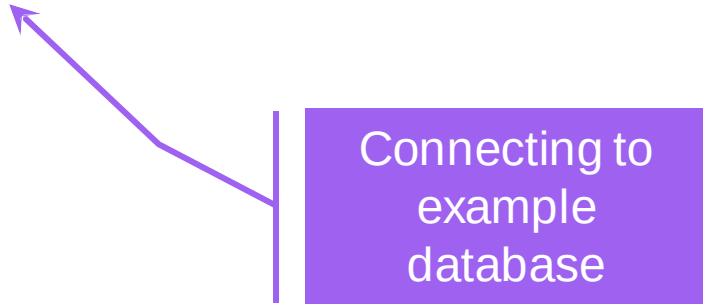
  ps.setString(1,”customer1”) –1 denotes the first ? Place

  ps.setString(2,”customer”) – 2 denotes the second ? Place

Walkthrough to get connect to Database from Servlet

MySqlDataStoreUtilities class to connect Database from Servlet

```
public class MySqlDataStoreUtilities
{
Connection conn = null;
public void getConnection()
{
    try
    {
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    conn=
DriverManager.getConnection("jdbc:mysql://localhost:3306/exampledatabase“
              ,"root",“root");
    }
    catch(Exception e)
    {}
}
```



Walkthrough for User Registration Code Snippet

User Registration Sample Code

```
HashMap<String, User> hm=new HashMap<String, User>();
        try
        {
        hm=MySqlDataStoreUtilities.selectUser();
        }
        catch(Exception e){}
        if(hm.containsKey(username))
        error_msg = "Username already exist as " + usertype;
        else
        {
        User user = new User(username,password,usertype);
        hm.put(username, user);
        MySqlDataStoreUtilities.insertUser(…);
        }
```

Calling utility function to select data from database and storing orders in hashmap

Calling utility function to insert user details in database

Utility Function For Registration

```
public static void insertUser(String username,String password,String usertype){
try{
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exampleda
               tabase“,”root”,”root”)

        //getConnection();
        String insertIntoCustomerRegisterQuery = "INSERT INTO
        Registration(username,password,usertype) "
               + "VALUES (?,?,?);";
        PreparedStatement pst =
        conn.prepareStatement(insertIntoCustomerRegisterQuery);
               pst.setString(1,username);
               pst.setString(2,password);
               pst.setString(3,usertype);
               pst.execute();
    }
    catch(Exception e){}
}
```

Connecting to example database

Query to insert data to table

Setting Value for Each Parameter

Execute method will insert data into database

Walkthrough for Inserting Order Code Snippet

Storing Order Payments

```
public void storePayment(int orderId,String orderName,double orderPrice,String
userAddress,String creditCardNo){
HashMap<Integer, ArrayList<OrderPayment>> orderPayments= new HashMap<Integer,
ArrayList<OrderPayment>>(); try
{orderPayments= MySqlDataStoreUtilities.selectOrder();
}
catch(Exception e){}
if(!orderPayments.containsKey(orderId)){
ArrayList<OrderPayment> arr = new ArrayList<OrderPayment>();
orderPayments.put(orderId, arr);
}
ArrayList<OrderPayment> listOrderPayment = orderPayments.get(orderId)
OrderPayment orderpayment = new OrderPayment (…);
listOrderPayment.add(orderpayment);
try
{MySqlDataStoreUtilities.insertOrder(…);
}}
```

Calling utility function to select data from database and storing orders in hashmap

Calling utility function to inserting orders in database

Utility Function for Select Order into hashmap

```
public static HashMap<Integer, ArrayList<OrderPayment>> selectOrder()
{
 HashMap<Integer,ArrayList<OrderPayment>> orderPayments=new HashMap<Integer,ArrayList<OrderPayment>>();
 try{
     getConnection();
     String selectOrderQuery ="select * from customerorders";
     PreparedStatement pst = conn.prepareStatement(selectOrderQuery);
     ResultSet rs = pst.executeQuery();
     ArrayList<OrderPayment> orderList=new ArrayList<OrderPayment>();
     while(rs.next())
     {
        if(!orderPayments.containsKey(rs.getInt("OrderId")))
        {
                ArrayList<OrderPayment> arr = new ArrayList<OrderPayment>();
                orderPayments.put(rs.getInt("orderId"), arr);
        }
        ArrayList<OrderPayment> listOrderPayment =orderPayments.get(rs.getInt("OrderId"));
        OrderPayment order= new
        OrderPayment(rs.getInt("OrderId"),rs.getString("userName"),rs.getString("orderName"));
        listOrderPayment.add(order);
    }
   }catch(…){}
 return orderPayments;}
```

ResultSet used to store table data obtained from database in servlet

Iterate through ResultSet and Store each order into class object

Utility Function for storing orders

```
public static void insertOrder(int orderId,String userName,String orderName)
{       try
        {
                Class.forName("com.mysql.jdbc.Driver").newInstance();
        conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/exampledatabase“
                ,”root”,”root”);
        String insertIntoCustomerOrderQuery = "INSERT INTO
        customerOrders(OrderId,UserName,OrderName) “ + "VALUES (?,?,?);";
        PreparedStatement pst =
        conn.prepareStatement(insertIntoCustomerOrderQuery);
                pst.setInt(1,orderId);
                pst.setString(2,userName);
                pst.setString(3,orderName);
                pst.execute();
        }
                catch(Exception e){}
}
```

Connecting to example database

Query to insert data to table

Setting Value for Each Parameter

Execute method will insert data into database

QUESTIONS?