

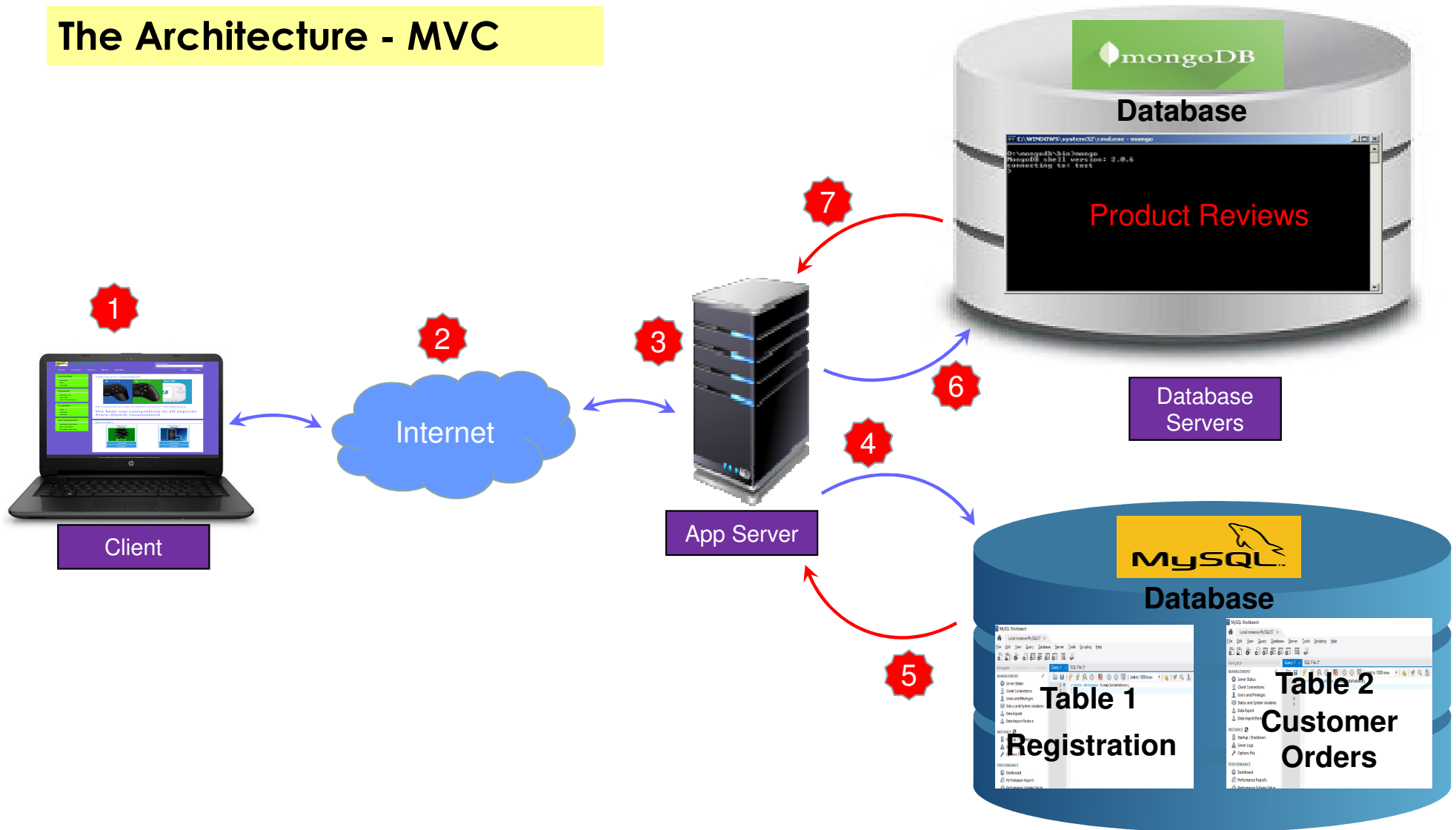
---

# Tutorial #2

MySQL

Datastore

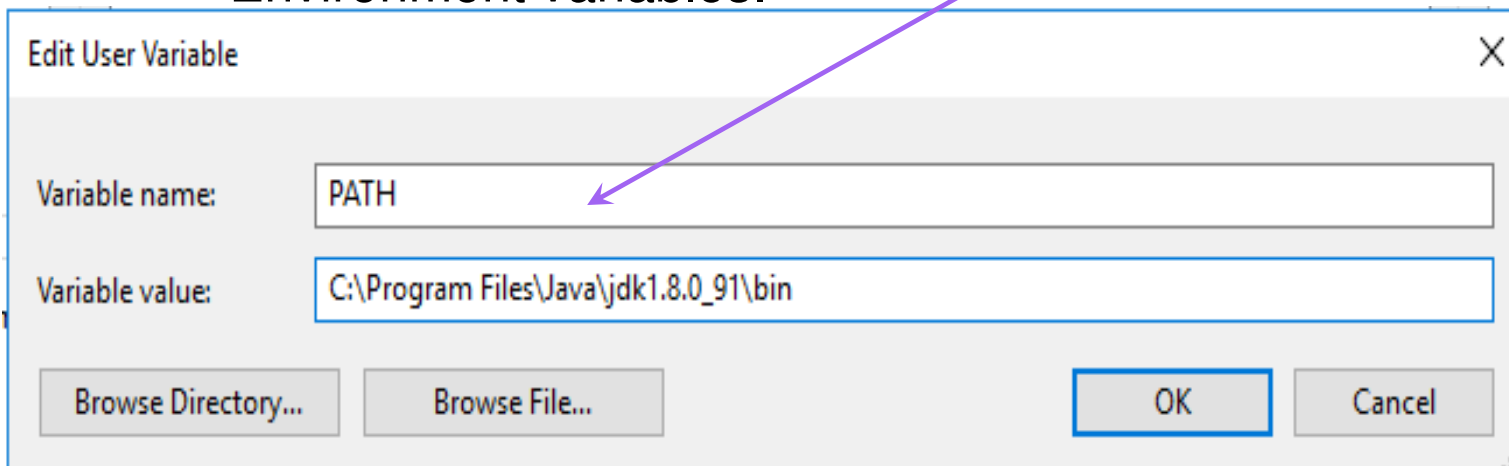
# The Architecture - MVC



## 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 →  
Environment variables.

Example PATH  
variable



# Download and Install MySQL Server

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

**Generally Available (GA) Releases**

## MySQL Installer 5.7.12

Select Platform:  
Microsoft Windows

Looking for previous GA versions?

<b>Windows (x86, 32-bit), MSI Installer</b> (mysql-installer-web-community-5.7.12.0.msi)	5.7.12	1.6M	<a href="#">Download</a>
<b>Windows (x86, 32-bit), MSI Installer</b> (mysql-installer-community-5.7.12.0.msi)	5.7.12	384.7M	<a href="#">Download</a>

MD5: 41f91cb06fee869876806e52f21d6e29 | [Signature](#)

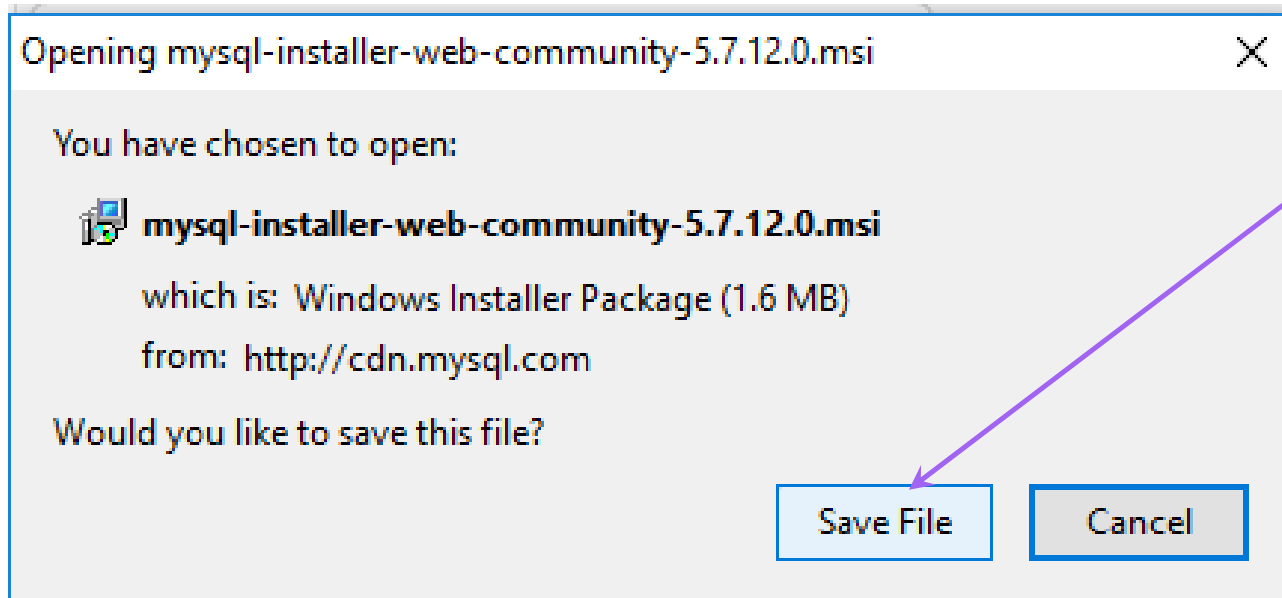
MD5: d1b17bfde2b1b89aedc4a93d701f9ed3 | [Signature](#)

 We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

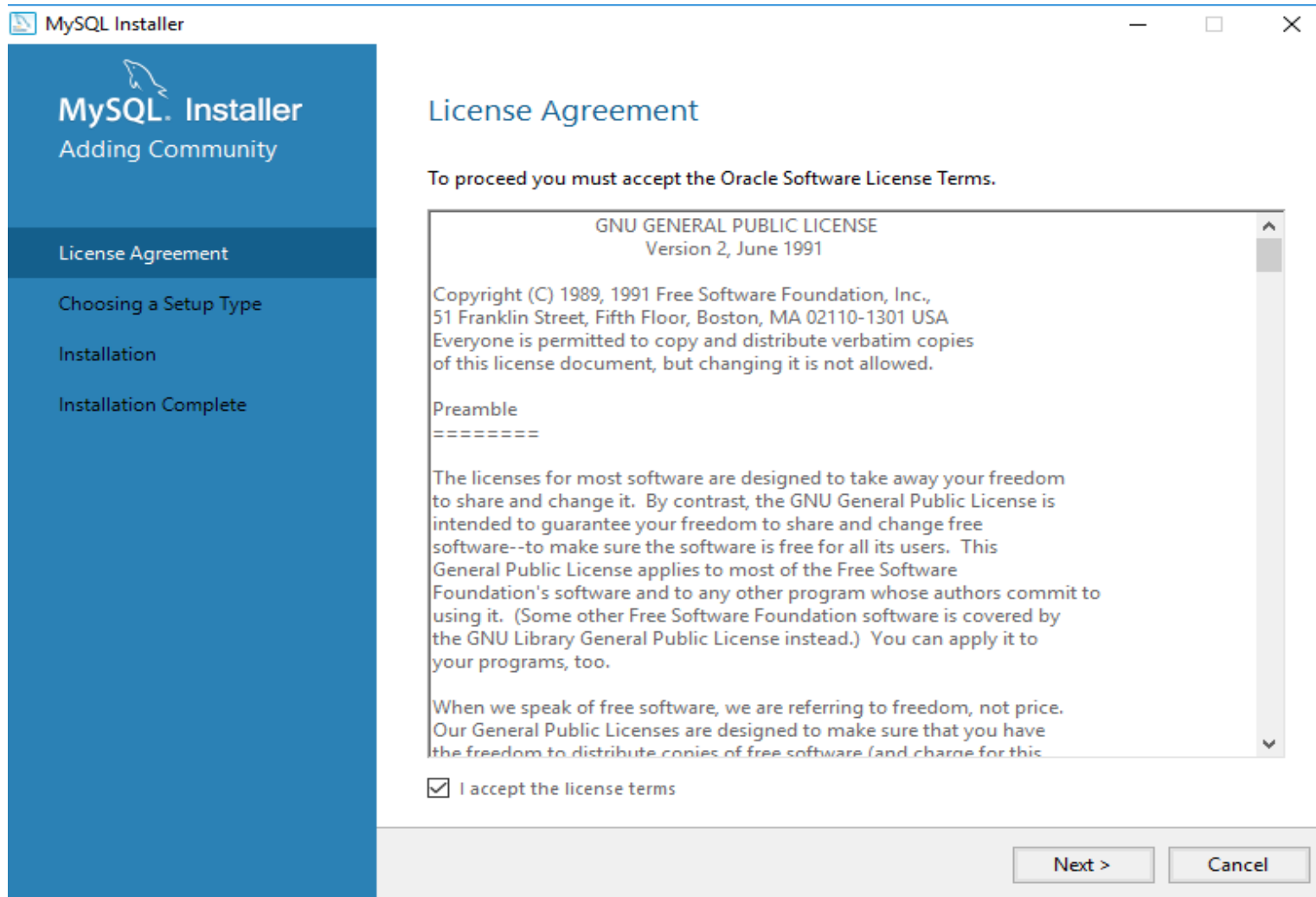
Select this file to download

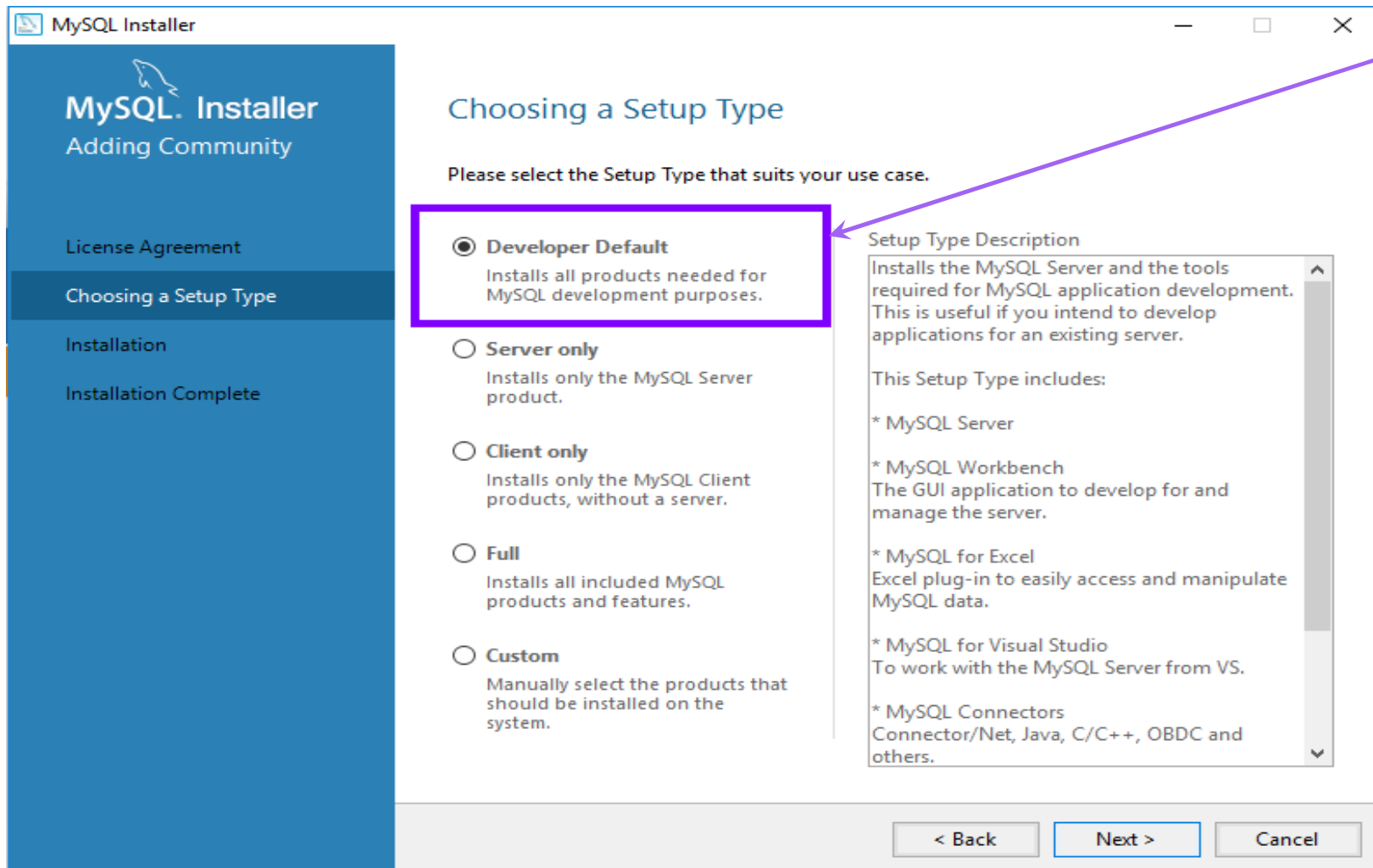
**Open file and follow the steps :**

**Step 1: Save file to download the installer.**

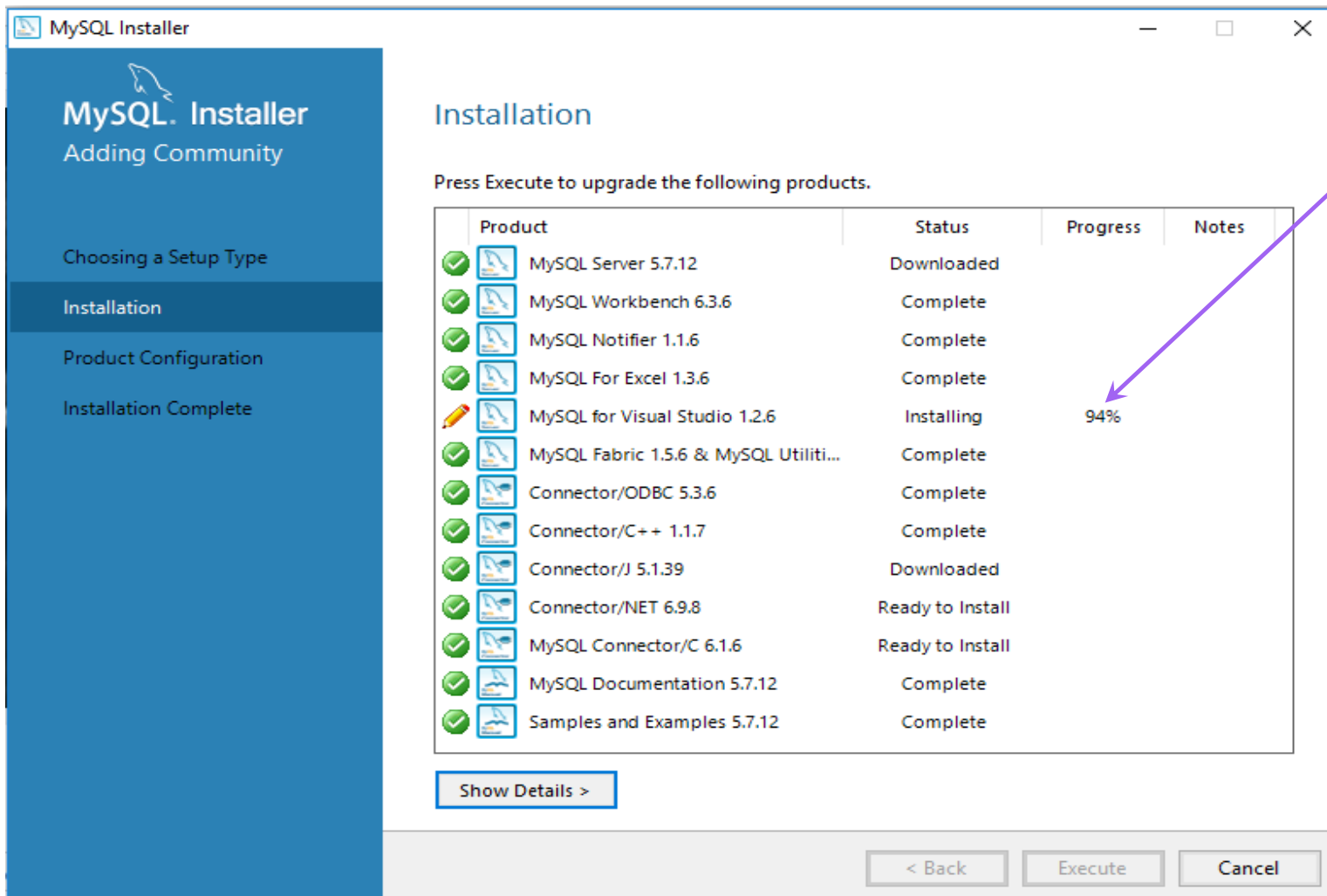


Select Save file  
option





Select the setup type as "Developer Default"



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



# MySQL Installer

Adding Community

Choosing a Setup Type

Installation

Product Configuration

Installation Complete

## Installation

Press Execute to upgrade the following products.

	Product	Status	Progress	Notes
✓	MySQL Server 5.7.12	Complete		
✓	MySQL Workbench 6.3.6	Complete		
✓	MySQL Notifier 1.1.6	Complete		
✓	MySQL For Excel 1.3.6	Complete		
✓	MySQL for Visual Studio 1.2.6	Complete		
✓	MySQL Fabric 1.5.6 & MySQL Utiliti...	Complete		
✓	Connector/ODBC 5.3.6	Complete		
✓	Connector/C++ 1.1.7	Complete		
✓	Connector/J 5.1.39	Complete		
✓	Connector/NET 6.9.8	Complete		
✓	MySQL Connector/C 6.1.6	Complete		
✓	MySQL Documentation 5.7.12	Complete		
✓	Samples and Examples 5.7.12	Complete		

Show Details >

< Back

Next >

Cancel

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

MySQL Installer

MySQL Server 5.7.12

Type and Networking

Accounts and Roles

Windows Service

Plugins and Extensions

Apply Server Configuration

### 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:

**Connectivity**

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

☒ TCP/IP

☒ Open Firewall port for network access

☐ Named Pipe Pipe Name:

☐ Shared Memory Memory Name:

**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

Port to  
connect to  
MYSQL

MySQL Installer

MySQL Server 5.7.12

Type and Networking

Accounts and Roles

Windows Service

Plugins and Extensions

Apply Server Configuration

### 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: **Weak**

**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

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

MySQL Installer

MySQL. Installer  
MySQL Server 5.7.12

Type and Networking

Accounts and Roles

Windows Service

Plugins and Extensions

Apply Server Configuration

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

# MySQL. Installer

MySQL Server 5.7.12

Type and Networking

Accounts and Roles

Windows Service

Plugins and Extensions

Apply Server Configuration

## Plugins and Extensions

### MySQL as a Document Store

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

☐ Enable X Protocol / MySQL as a Document Store

Port Number:

☐ Open Firewall port for network access

Starting with MySQL Server 5.7, MySQL supports document store development. In order to provide a complete document store/NoSQL experience there is a new communications protocol called the X Protocol. The expanded capabilities of the X Protocol enable us to provide modern developer APIs with features such as asynchronous calls, pipelining, and more. In addition to implementing document collections, the new X DevAPI also supports relational and combined document store/relational capabilities. Now developers, designers and DBAs can deploy MySQL databases that implement document store, relational, or hybrid document/relation models.

[Click here to view MySQL as a Document Store online documentation](#)

< Back

Next >

Cancel

# MySQL Installer

MySQL Server 5.7.12

Type and Networking

Accounts and Roles

Windows Service

Plugins and Extensions

Apply Server Configuration

## Apply Server Configuration

The following configuration steps are being executed.

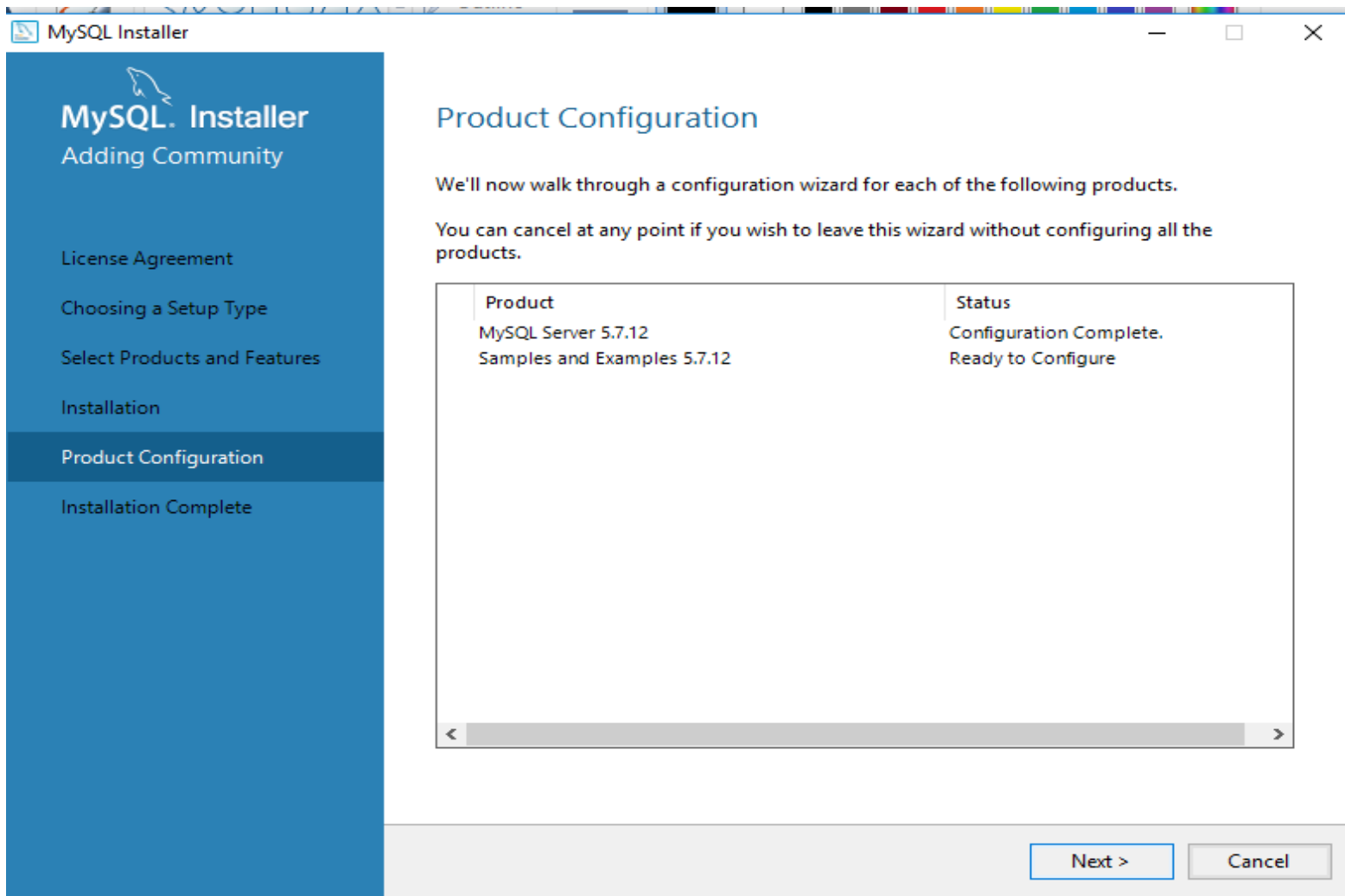
Configuration Steps

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
- ☐ Updating Firewall for MySQL Document Data Feature Set

Execute

Cancel



MySQL Installer

MySQL. Installer  
Samples and Examples

Connect To Server

Apply Server Configuration

### Connect To Server


Here are the compatible servers installed. If more than one, please select one.

Server	Architecture	Status
MySQL Server 5.7.12	X64	Running

Now give us the credentials we should use (needs to have root privileges). Click check to make sure they work.

User:  Credentials provided in Server configuration

Password:

 Connection successful.

Next > Cancel

Before proceeding, make sure you check the connection by providing required credentials.



# MySQL Installer

Samples and Examples

Connect To Server

Apply Server Configuration

## Apply Server Configuration

The following configuration steps are being executed.

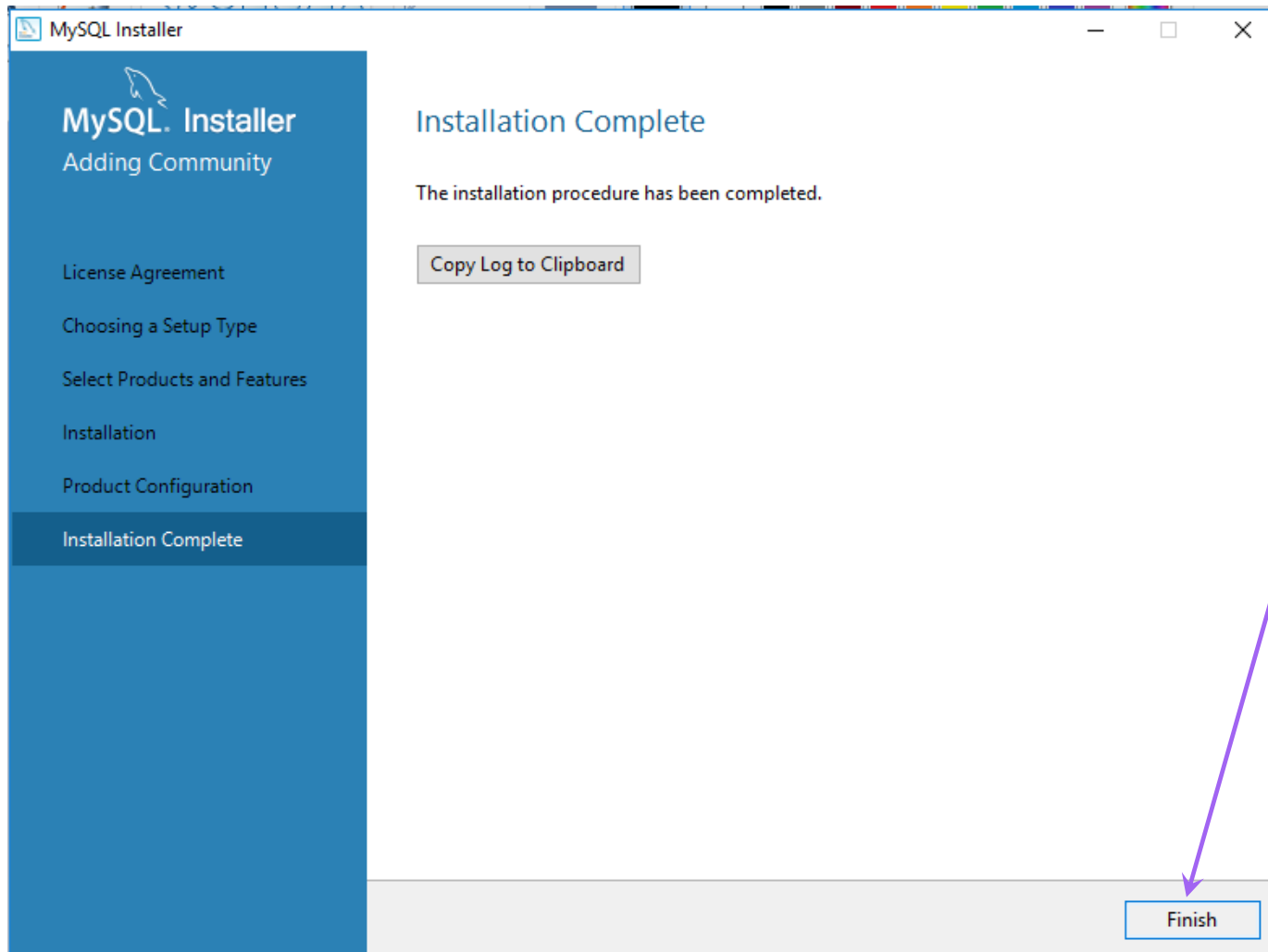
Configuration Steps

Log

- ☒ Checking if there are any features installed that need configuration.
- ☒ Starting Server.
- ☒ Getting Connection information (Provide credentials if it is needed).
- ☒ Running Scripts. ..
- ☐ Stopping Server.

Execute

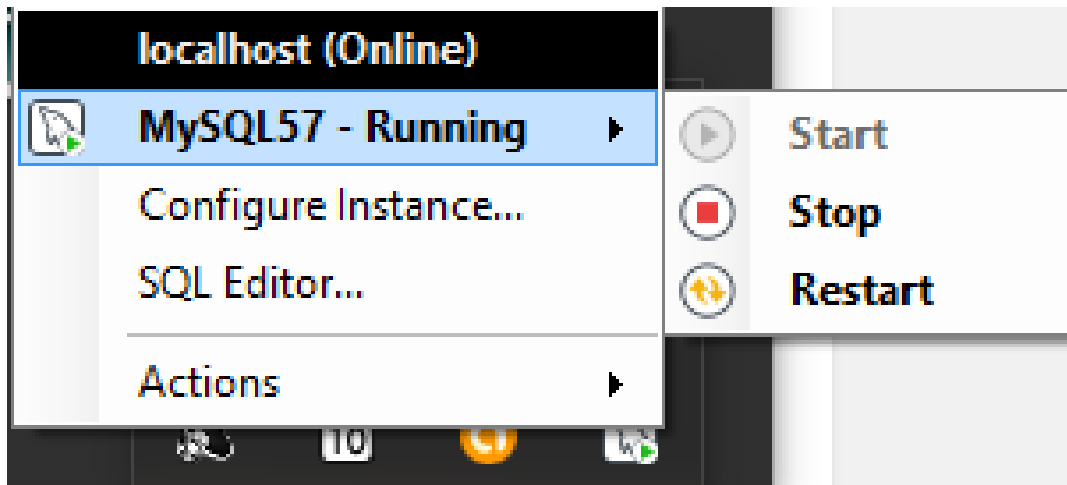
Cancel



Now, the  
installation is  
completed.

## Post Installation...

After the installation, we can monitor MySQL server by accessing MySQL notifier from the task bar.



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 Connections +

Filter connections

Local instance MySQL57

root  
localhost:3306

Connect to MySQL Server

Please enter password for the following service:

Service: Mysql@localhost:3306  
User: root  
Password:

☐ Save password in vault

OK Cancel

Enter the  
Password used  
while installation

## Models +

sakila\_full

MySQL Workbench 6.3 CE/extras  
sakila  
10 Dec 15, 13:46

## Shortcuts



MySQL Utilities



Database Migration



MySQL Bug Reporter



Workbench Blogs



Planet MySQL



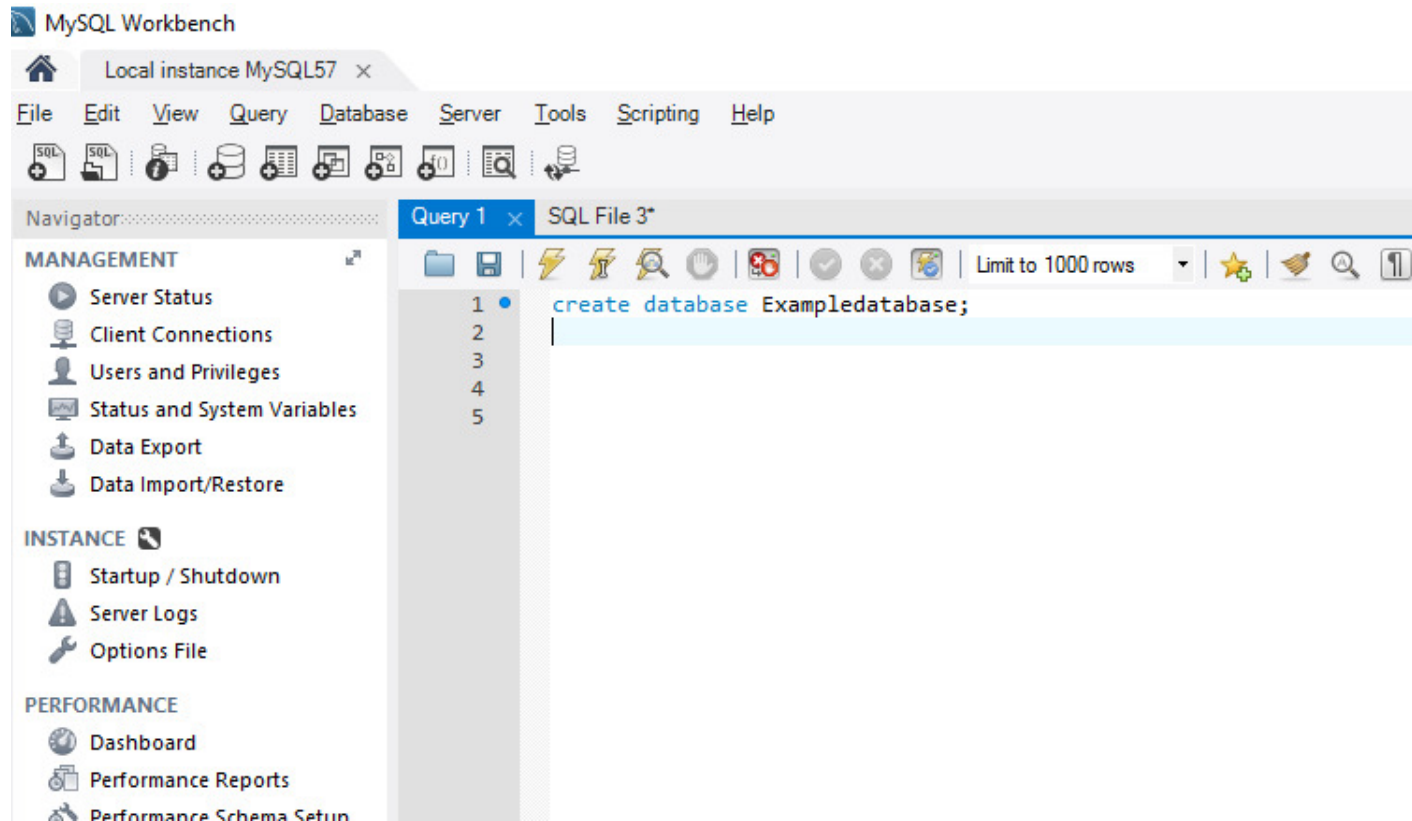
Workbench Forum



Scripting Shell

# Executing queries in MySQL Workbench

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

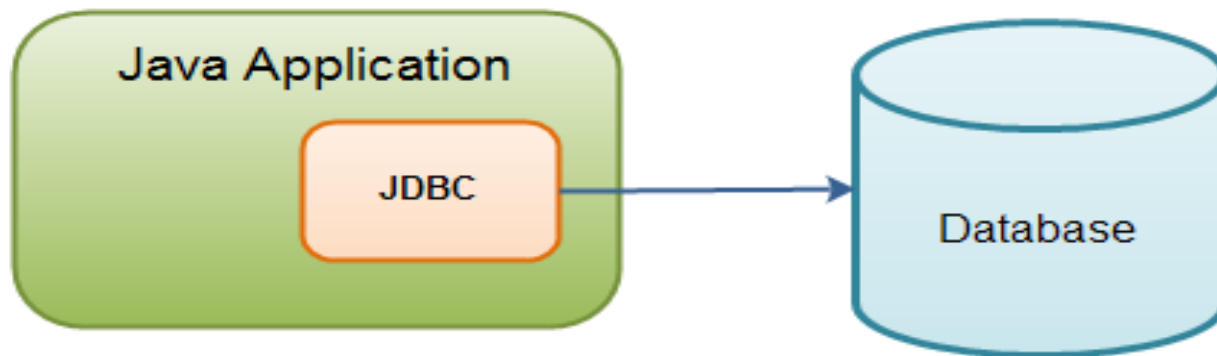


## Executing queries using a Java Application

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

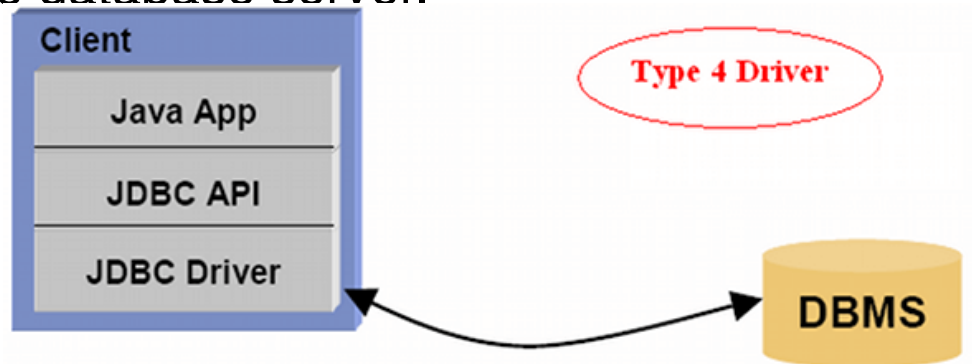
**Type 1:** JDBC-ODBC Bridge driver (Bridge)

**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-5.1.39-bin.jar` to compile and execute a application. If you want to compile the examples from the command line, go to the site <http://www.mysql.com/products/connector/> and download the MySQL connector for the Java language.

### MySQL Connectors

MySQL provides standards-based drivers for JDBC, ODBC, and .Net enabling developers to build database applications in their language, allows developers to embed MySQL directly into their applications.

#### Developed by MySQL

ADO.NET Driver for MySQL (Connector/NET)	<a href="#">Download</a>
ODBC Driver for MySQL (Connector/ODBC)	<a href="#">Download</a>
JDBC Driver for MySQL (Connector/J)	<a href="#">Download</a>
Python Driver for MySQL (Connector/Python)	<a href="#">Download</a>
C++ Driver for MySQL (Connector/C++)	<a href="#">Download</a>
C Driver for MySQL (Connector/C)	<a href="#">Download</a>
C API for MySQL (mysqlclient)	<a href="#">Download</a>

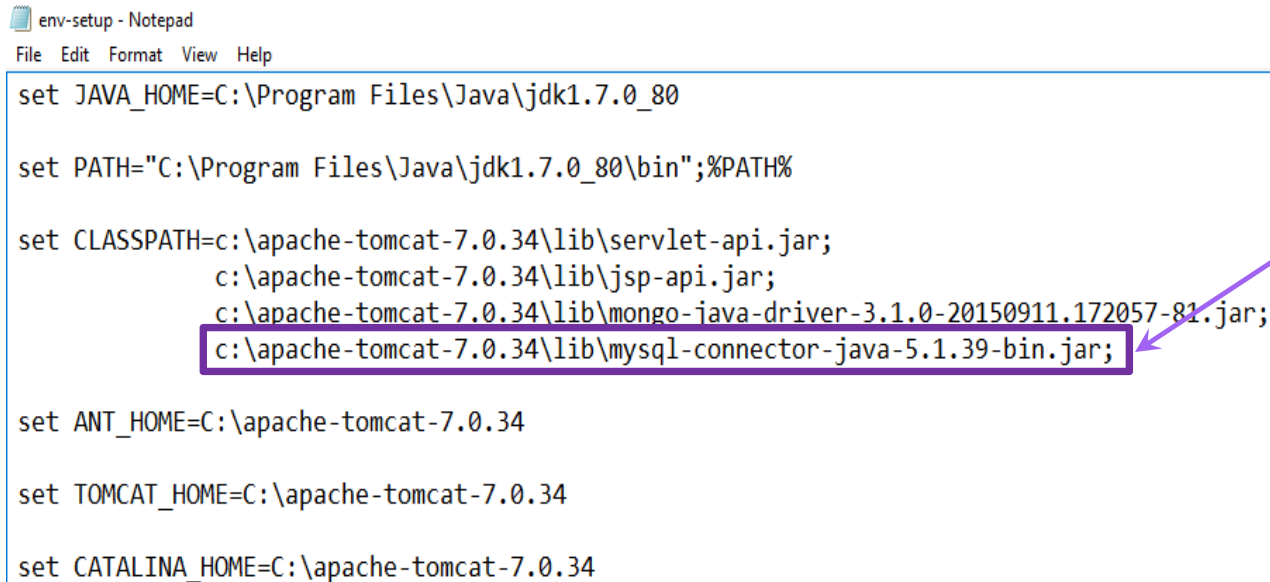
Select JDBC Driver  
as we are using  
Java.

# 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 - Notepad
File Edit Format View Help

set JAVA_HOME=C:\Program Files\Java\jdk1.7.0_80

set PATH="C:\Program Files\Java\jdk1.7.0_80\bin";%PATH%

set CLASSPATH=c:\apache-tomcat-7.0.34\lib\servlet-api.jar;
               c:\apache-tomcat-7.0.34\lib\jsp-api.jar;
               c:\apache-tomcat-7.0.34\lib\mongo-java-driver-3.1.0-20150911.172057-81.jar;
               c:\apache-tomcat-7.0.34\lib\mysql-connector-java-5.1.39-bin.jar;

set ANT_HOME=C:\apache-tomcat-7.0.34

set TOMCAT_HOME=C:\apache-tomcat-7.0.34

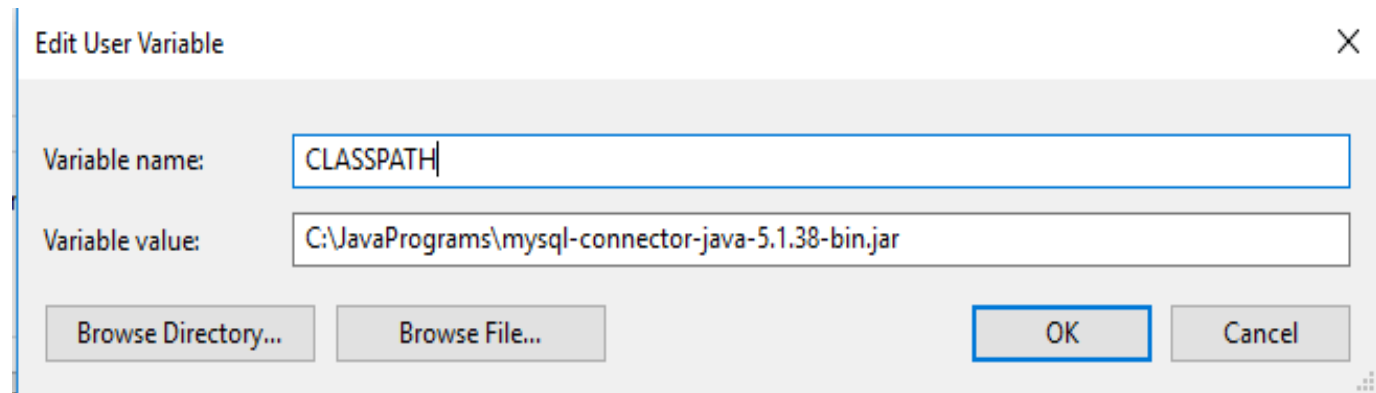
set CATALINA_HOME=C:\apache-tomcat-7.0.34
```

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.



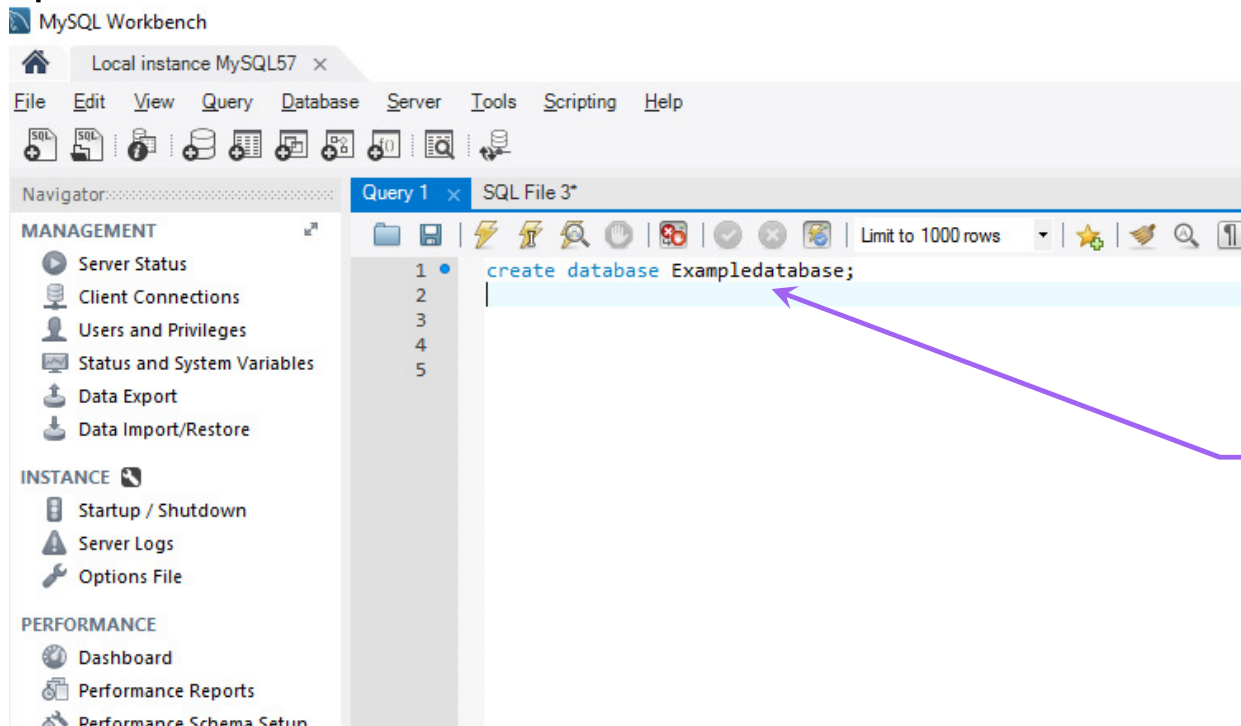
## A walkthrough example

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 “exempldatabase” in the workbench space, and execute the SQL command.

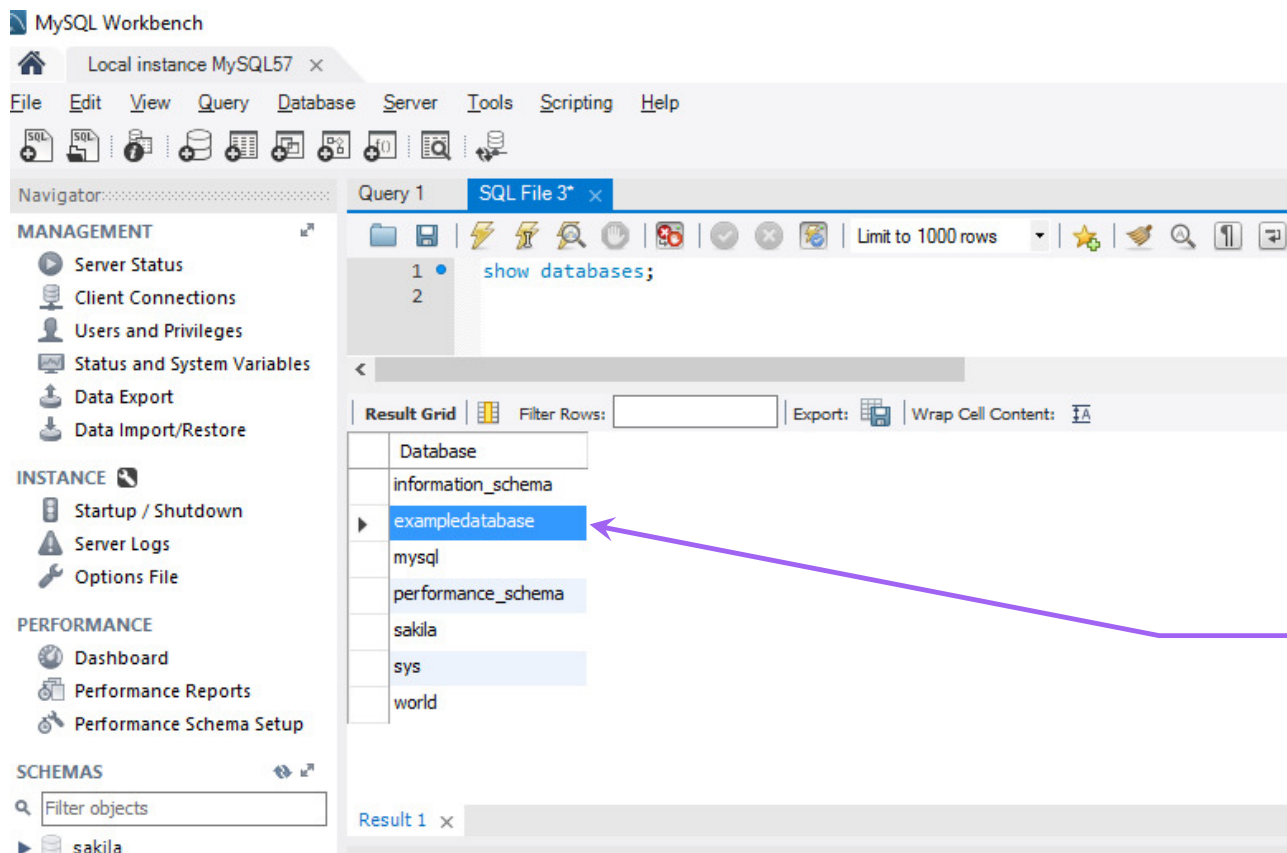


Query to create a new database

# A walkthrough example

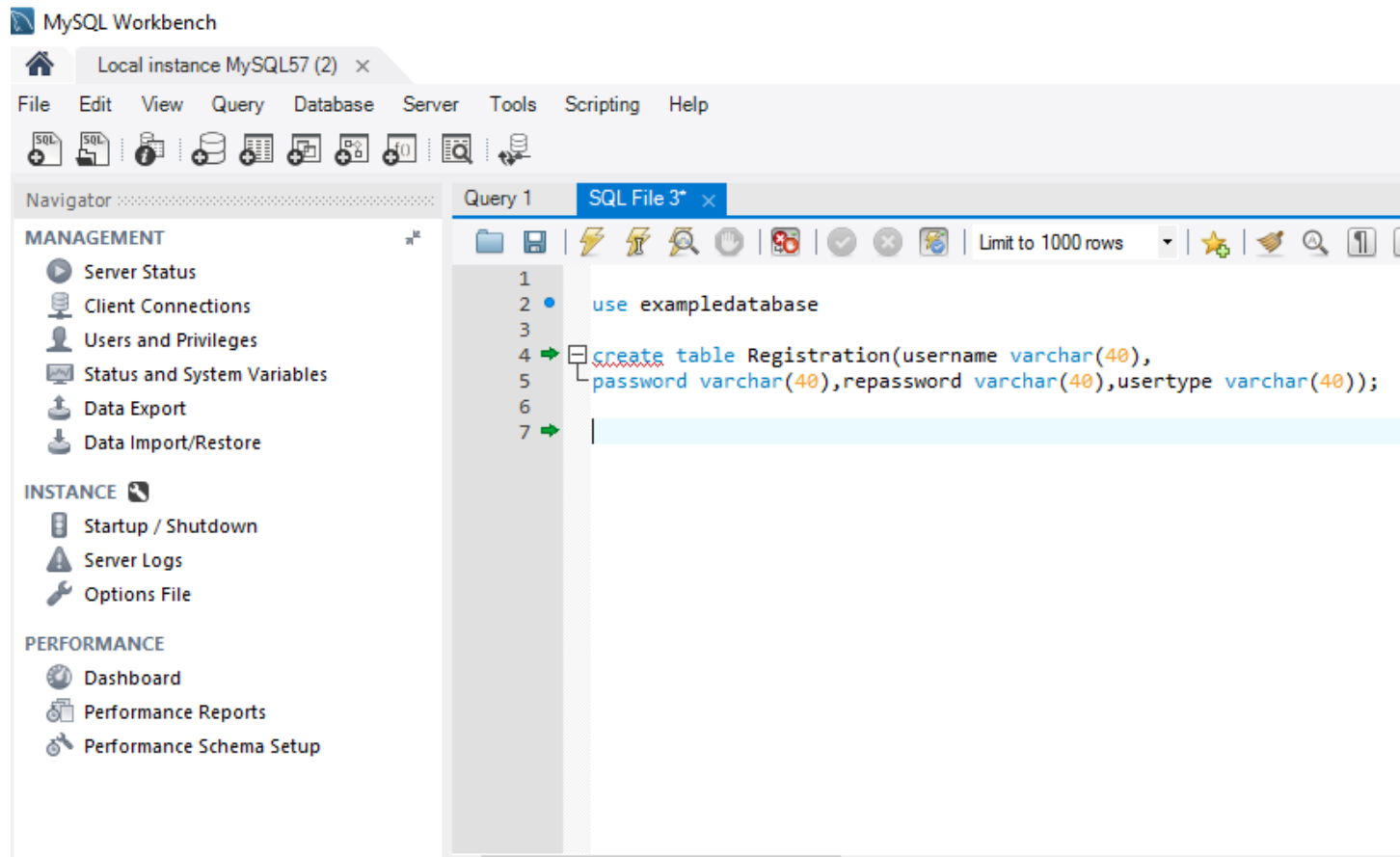
Write and execute the following commands to check if the database is created.

-> 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

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'MANAGEMENT' and 'PERFORMANCE' sections. The central editor window shows the following SQL command:

```
1 Create table CustomerOrders
2 (
3   OrderId integer,
4   userName varchar(40),
5   orderName varchar(40),
6   orderPrice double,
7   userAddress varchar(40),
8   creditCardNo varchar(40),
9   Primary key(orderid,userName,orderName)
10 );
11
```

The right sidebar shows the 'SQLAdditions' section with the following syntax:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
(create_definition,...)
[table_options]
[partition_options]
```

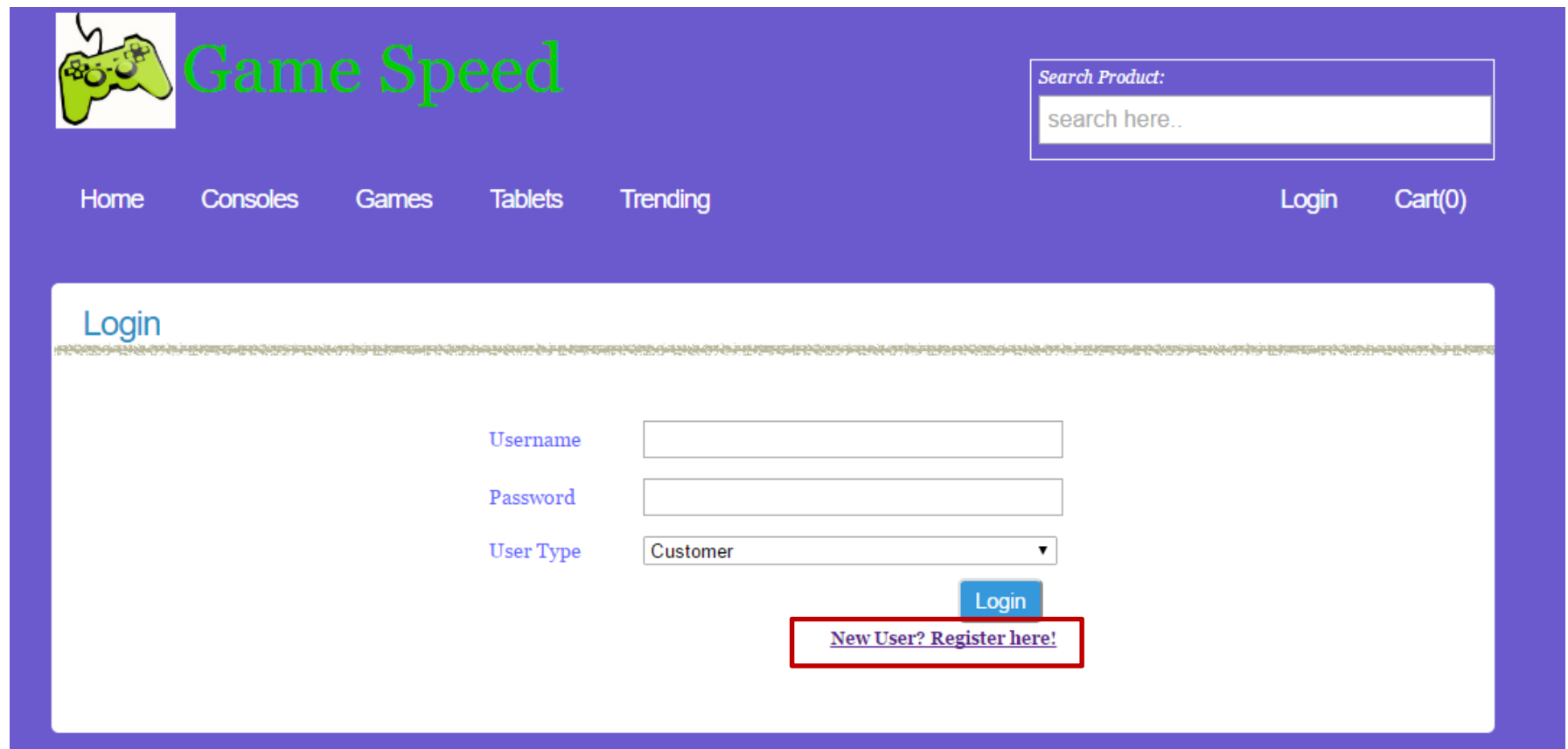
The bottom section shows the 'Output' window with the following table:

#	Time	Action	Message	Duration / Fetch
1	16:05:47	drop table CustomerOrders	Error Code: 1046. No database selected Select the default DB to be used by double-clicki...	0.046 sec
2	16:05:58	drop table CustomerOrders	Error Code: 1046. No database selected Select the default DB to be used by double-clicki...	0.000 sec
3	16:09:45	use exampledatabase	0 row(s) affected	0.000 sec
4	16:09:48	drop table CustomerOrders	0 row(s) affected	0.390 sec
5	16:13:34	Create table CustomerOrders ( OrderId integer, userName varchar(40), orderName varchar(...	0 row(s) affected	0.375 sec



## 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



The image shows a website header for "Game Speed" with a green game controller icon. The header includes a search bar labeled "Search Product:" with the placeholder text "search here..". Below the header is a navigation menu with links: Home, Consoles, Games, Tablets, Trending, Login, and Cart(0). The main content area is titled "Login" and contains a form with three fields: "Username" (text input), "Password" (text input), and "User Type" (dropdown menu with "Customer" selected). A blue "Login" button is positioned below the form. A red-bordered box highlights the text "[New User? Register here!](\"#\")".

**Game Speed**

Search Product:  
search here..

Home Consoles Games Tablets Trending Login Cart(0)

Login

Username

Password


User Type

Login

[New User? Register here!](#)

User provides the login information

On clicking create user button data is stored in Registration Table of My sql



# Game Speed

Search Product:  
search here..

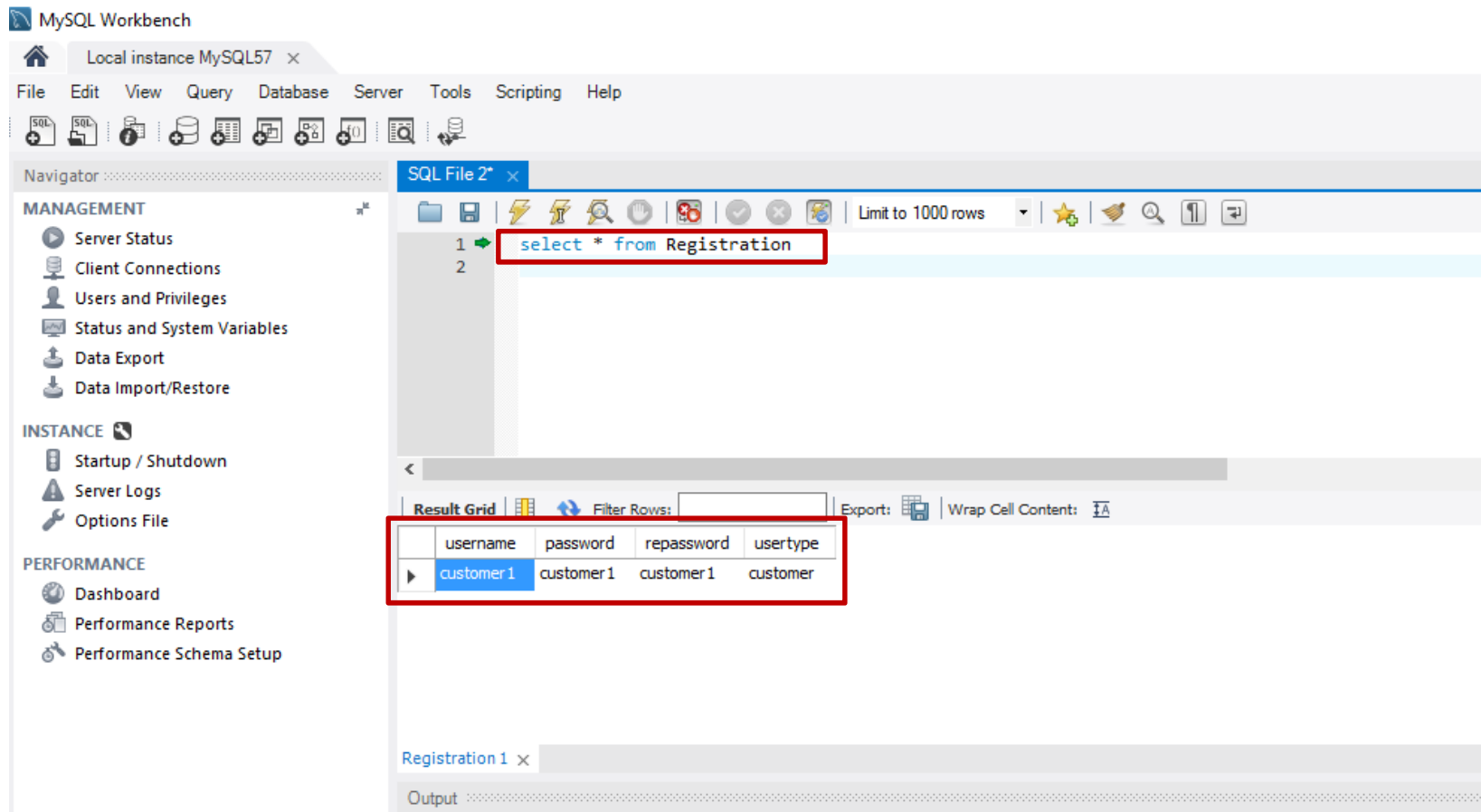
Home Consoles Games Tablets Trending Login Cart(0)

## Login

Username	<input type="text" value="customer1"/>
Password	<input type="password" value="....."/>
Re-Password	<input type="password" value="....."/>
User Type	<input type="text" value="Customer"/>

Create User


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



The screenshot shows the MySQL Workbench interface. The left sidebar contains a 'Navigator' pane with sections for 'MANAGEMENT' (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), 'INSTANCE' (Startup / Shutdown, Server Logs, Options File), and 'PERFORMANCE' (Dashboard, Performance Reports, Performance Schema Setup). The main area is titled 'SQL File 2\*' and contains a SQL query: `select * from Registration`. Below the query editor is a 'Result Grid' showing the output of the query. The grid has four columns: 'username', 'password', 'repassword', and 'usertype'. The first row of data shows 'customer 1' for all four columns. The 'Output' pane at the bottom is empty.

	username	password	repassword	usertype
▶	customer 1	customer 1	customer 1	customer

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



# Game Speed

[Home](#) [Consoles](#) [Games](#) [Tablets](#) [Trending](#) [Login](#) [Cart\(0\)](#)

## Login

**Your customer account has been created. Please login**

Username

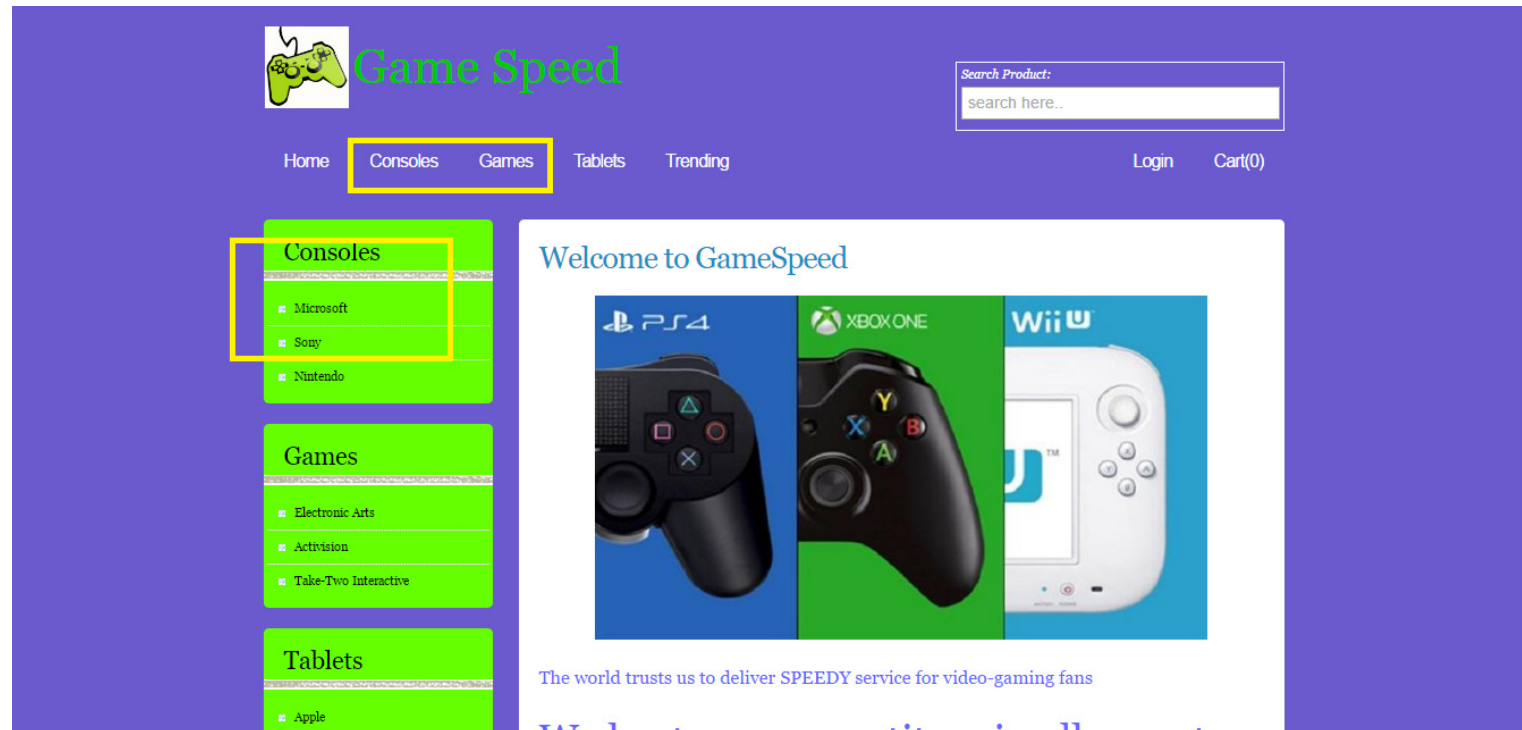
Password

User Type

[New User? Register here!](#)

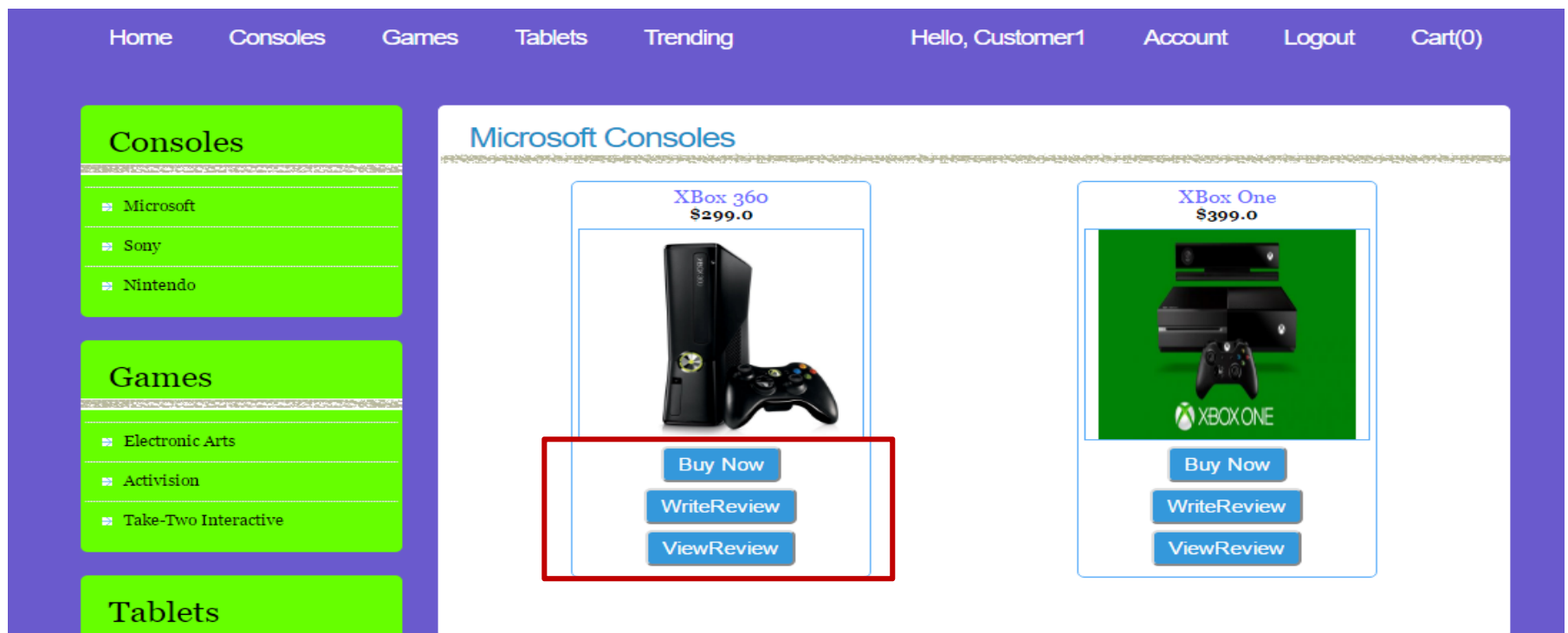
## Example –Place Order

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



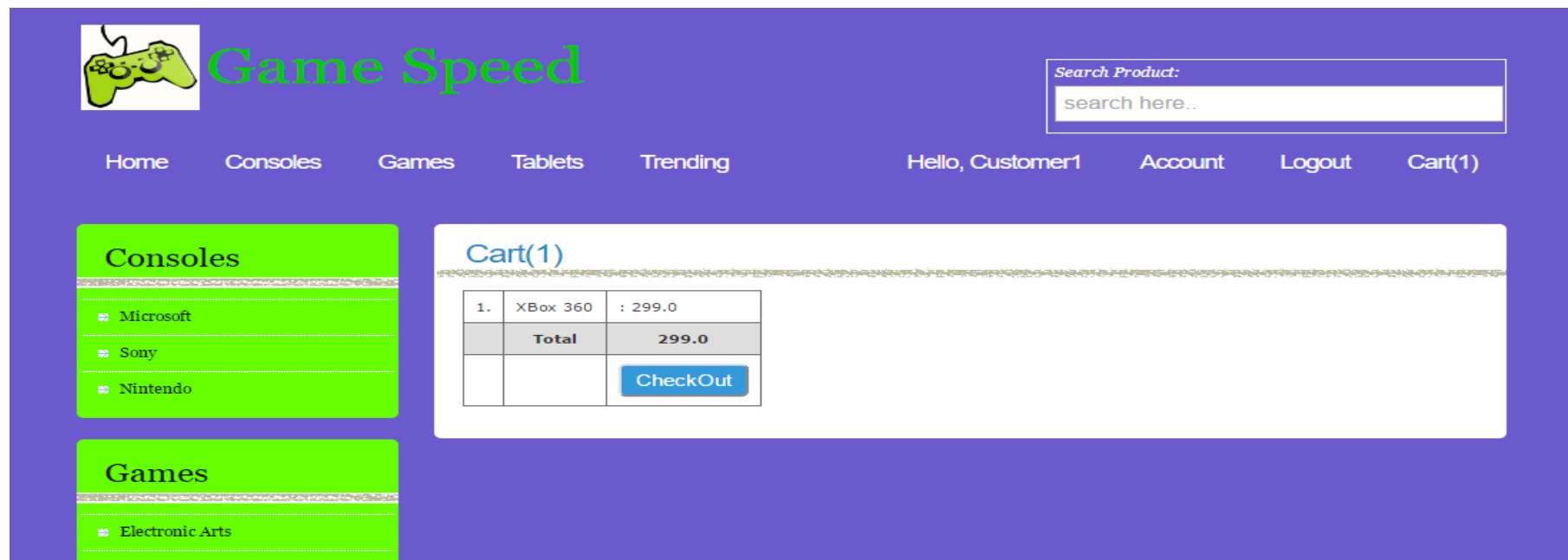
## Example – Place Order:

- 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.



The screenshot shows the 'Game Speed' website interface. The header features a green game controller icon and the site name 'Game Speed' in green. A search bar is located in the top right corner. The navigation menu includes links for Home, Consoles, Games, Tablets, and Trending. User information 'Hello, Customer1' and links for Account, Logout, and Cart(1) are displayed on the right. The left sidebar contains expandable categories for Consoles (listing Microsoft, Sony, and Nintendo) and Games (listing Electronic Arts). The main content area displays the 'Cart(1)' page with a table showing one item: XBox 360 priced at 299.0. A 'CheckOut' button is positioned below the table.

**Game Speed**

Search Product:  
search here..

Home Consoles Games Tablets Trending Hello, Customer1 Account Logout Cart(1)

**Consoles**

- Microsoft
- Sony
- Nintendo

**Games**


- Electronic Arts

**Cart(1)**

1.	XBox 360	: 299.0
	<b>Total</b>	<b>299.0</b>
		<a href="#">CheckOut</a>

## 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.



# Game Speed

[Home](#) [Consoles](#) [Games](#) [Tablets](#) [Trending](#) [Hello, Customer1](#) [Account](#) [Logout](#) [Cart\(1\)](#)

### Consoles

- Microsoft
- Sony
- Nintendo

### Games

- Electronic Arts
- Activision

### Order

Customer Name:	customer1
Product Purchased:	XBox 360
Product Price:	299.0
Total Order Cost	299.0

Credit/accountNo

80456666

Customer Address

Chicago Illinois

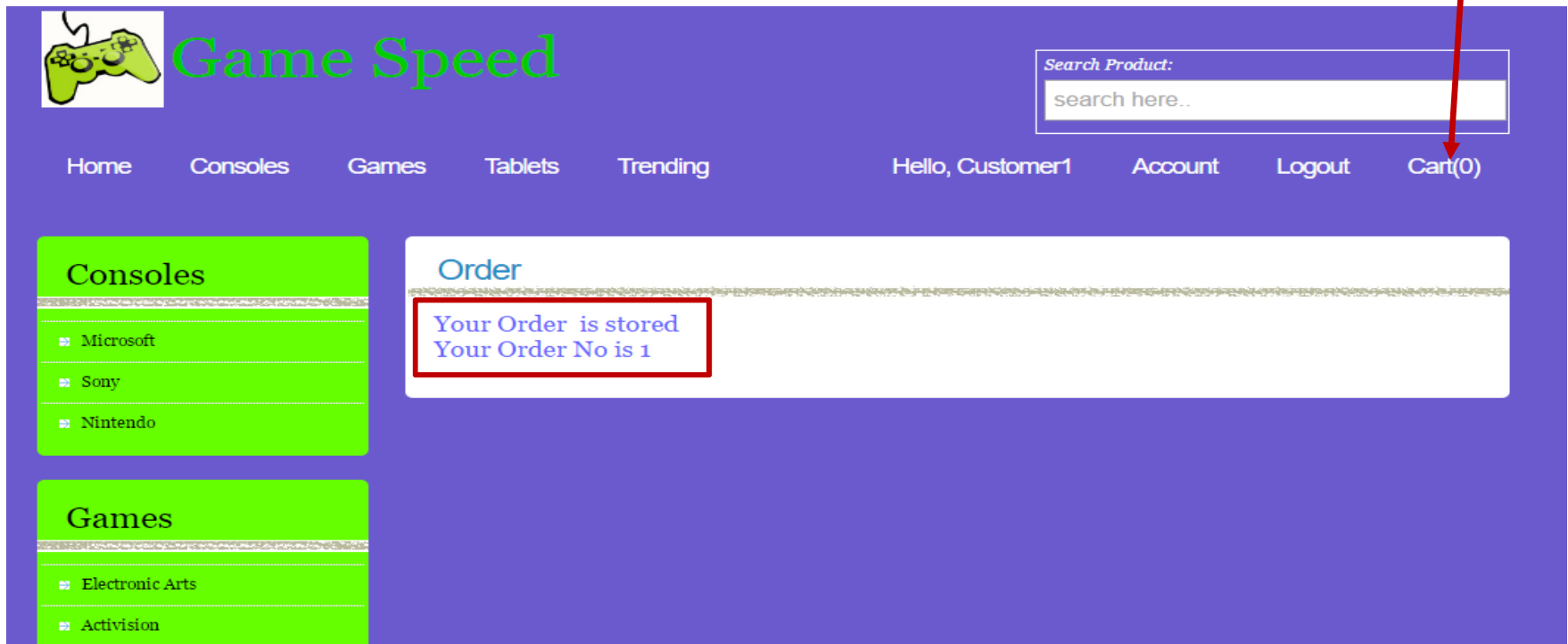
Submit



## Example – Place Order:

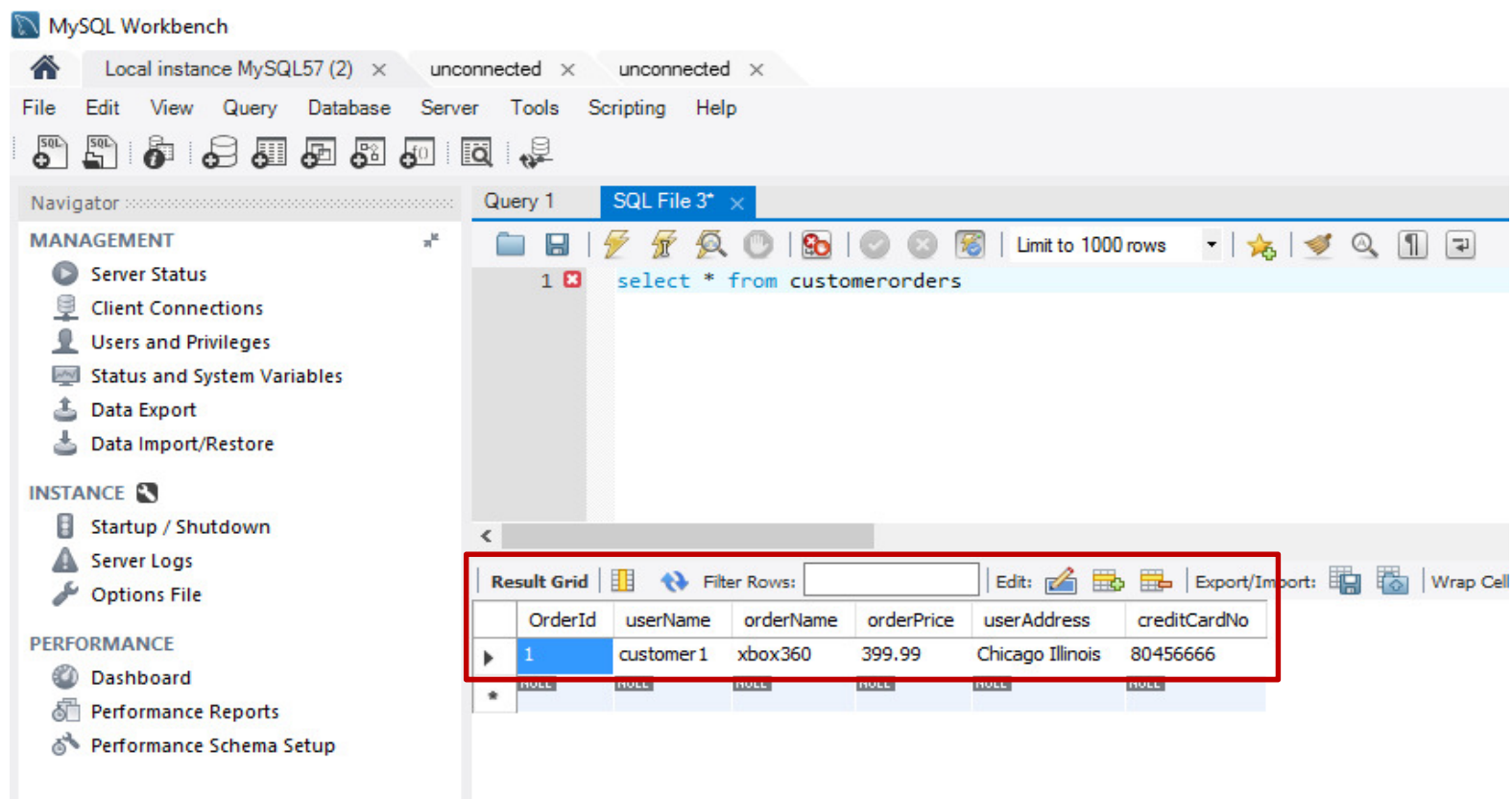
- On clicking the submit button from check out page order will be stored in My Sql database and order no will be generated

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



MySQL Workbench

Local instance MySQL57 (2) × unconnected × unconnected ×

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Query 1 SQL File 3\* ×

Limit to 1000 rows

```
1 select * from customerorders
```

Result Grid

	OrderId	userName	orderName	orderPrice	userAddress	creditCardNo
▶ 1	customer1	xbox360	399.99	Chicago Illinois	80456666	
*	NULL	NULL	NULL	NULL	NULL	NULL

## 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/exampledatabase", "root", "root");
```

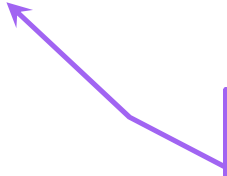
## Prepared Statement Execution

- 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)
        {}
    }
}
```

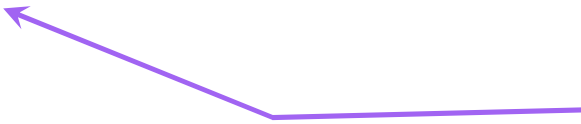


Connecting to  
example  
database

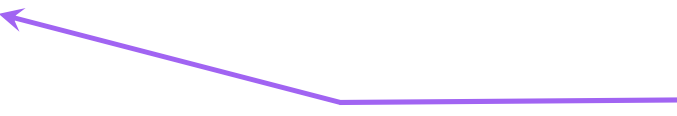
# Walkthrough for User Registration Code Snippet

# 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")

    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

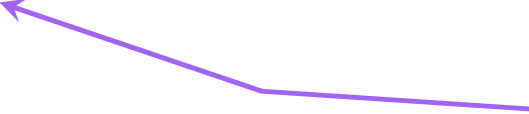
Setting Value for Each  
Parameter

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(...);
    }
    }
```



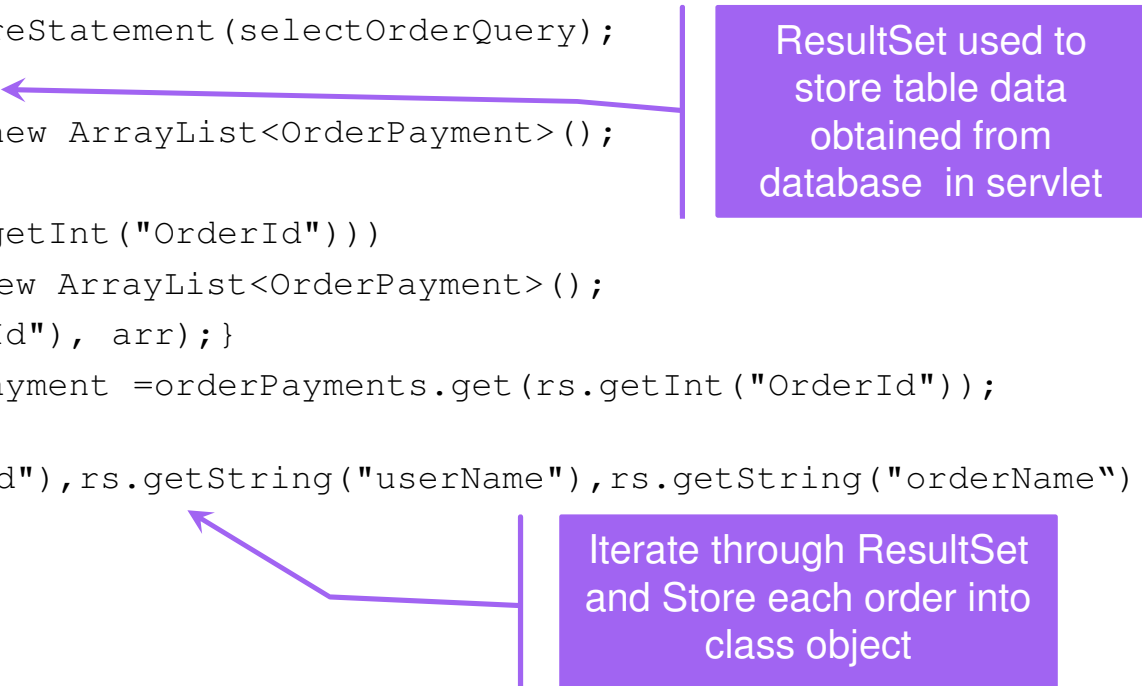
Calling utility function  
to select data from  
database and storing  
orders in hashmap



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