SQL [MySQL]

* ENVIRONMENT SETUP

MySQL is free, open-source, and cross-platform software, which can be downloaded from its official website. MySQL management system installs on Linux, macOS, and windows.

MySQL requires a framework for an interface like an eclipse, visual studio, and others. It is necessary to have a minimum of 4 GB RAM in the computer or application devices.

The following information describes the installation procedure on the Windows operating system.

**Installation for Windows operating system**

**Step1**: Download the MySQL installer on the Windows operating system.

The Windows users can download the MySQL installer using the following link. You can choose either offline or online software installer.

MySQL link: <https://dev.mysql.com/downloads/installer/>

The offline MySQL database installation can be done using the following file.

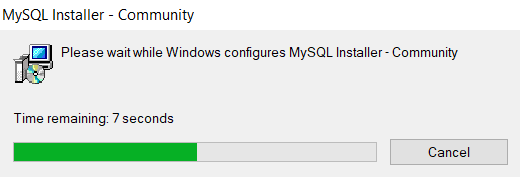
mysql-installer-community-<version>.exe

The online MySQL database installation can be done using the link given below:

mysql-installer-web-community-<version>.exe

**Step2**: Install the MySQL data management system software on your device.

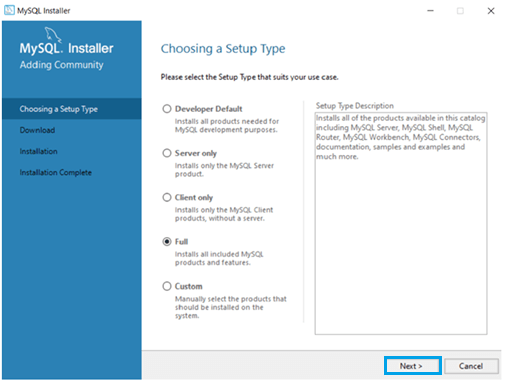
Double-click on **MySQL Installer** and start the installation process. If Windows operating system asks for installation permission, then select “**YES**.” The MySQL software starts the installation process.



The MySQL installer waits for configuration with the windows operating system. Once the configuration is complete, you set up the software as per your requirement.

**Step3:**Choose the setup type.

Here, you choose the setup type for MySQL installation. There are four setup types for installation. Each setup type has its function.

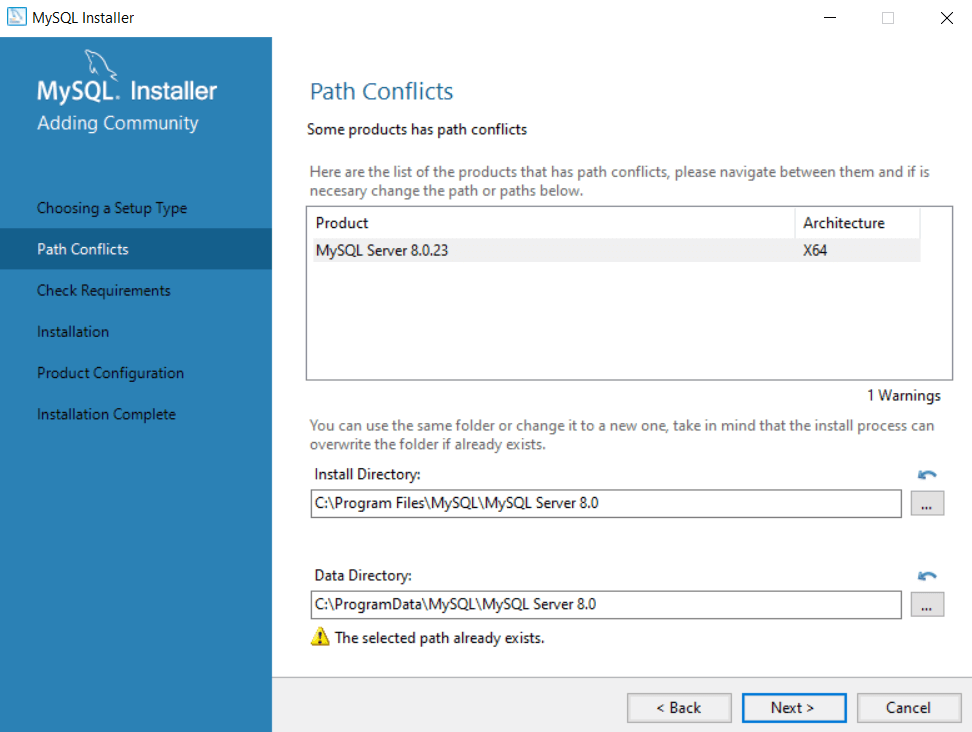


Mostly, the developer prefers to select the “**Full**” or “**Developer Default**” setup type. However, a student can pick the “**Full**” or “**Custom**” setup type.

After you choose the”**Full**” setup type, proceed by clicking on “**Next**” for further installation.

**Step4:**Choose the path for data and software.

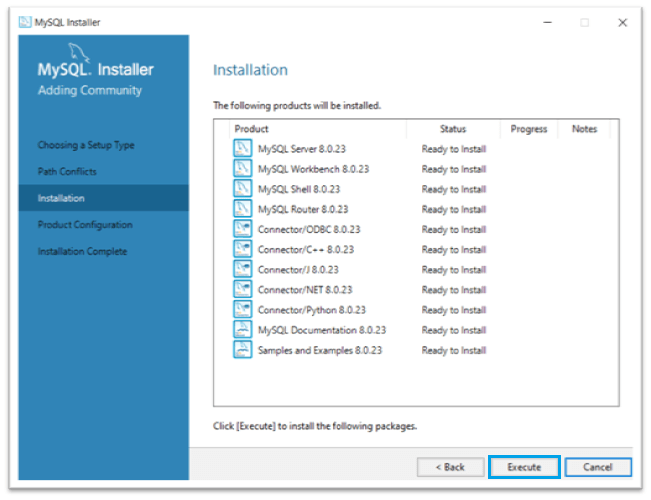
The MySQL installer provides a default path in the “C:” drive. The installer and data directory does not need to be similar but prefers to be the same directory path. If the old version of MySQL is available in the default path, the installer automatically updates the new version.



MySQL changes the directory path of software and data. The path is set by default based on available memory. After this, click on “Next” for further installation.

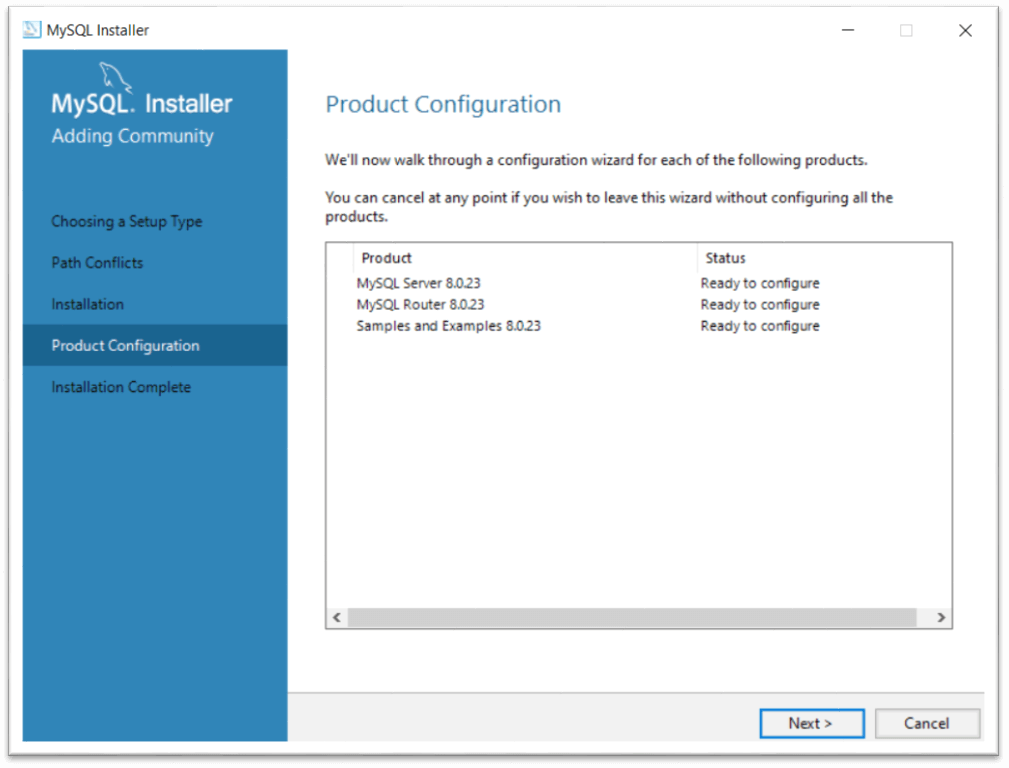
**Step5:** Installation for the MySQL server.

You must check the availability of the MySQL installer and manually execute it one by one. Click on “**Next**” for further installation.



Click on the “**Execute**” button to install MySQL interfaces or connectors. Now you can proceed further with product configuration.

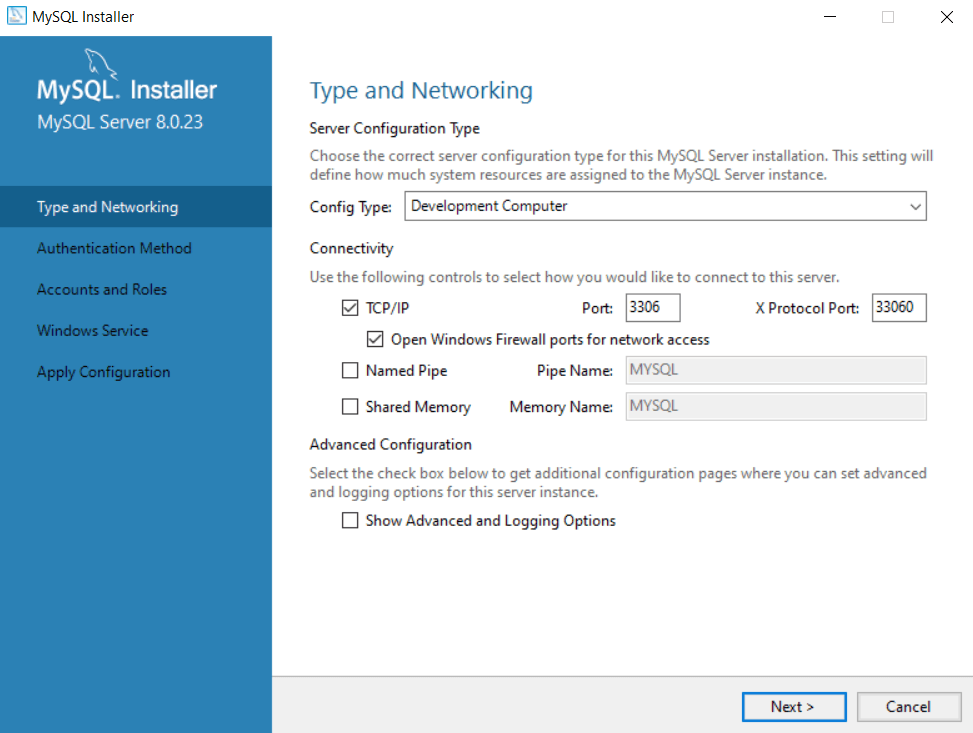
**Step6**:Start the product configuration.



Click on the “**Next**” button to complete the installation. Now select the server configuration.

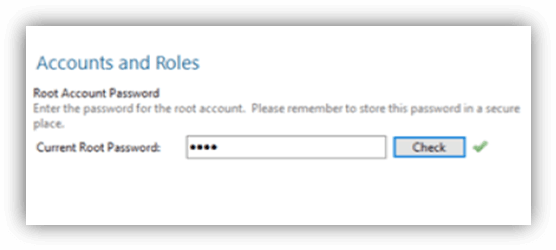
**Step6**: MySQL completes the installation.

Installation needs “server configuration,” MySQL password, port as per the availability.

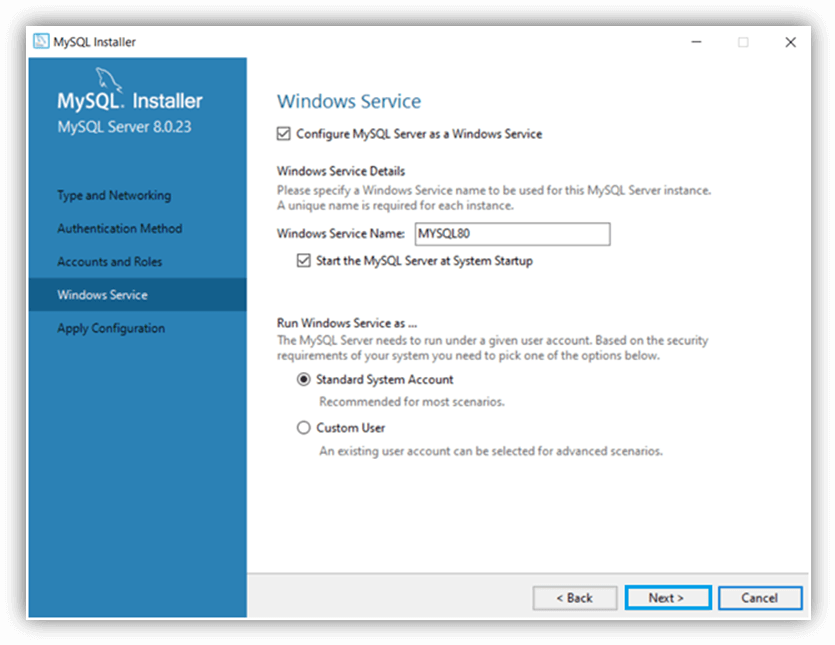


You select the server configuration type to connect the server. The popular configuration type is a “Development Machine.” Select the “**TCP/IP**” network for MySQL.

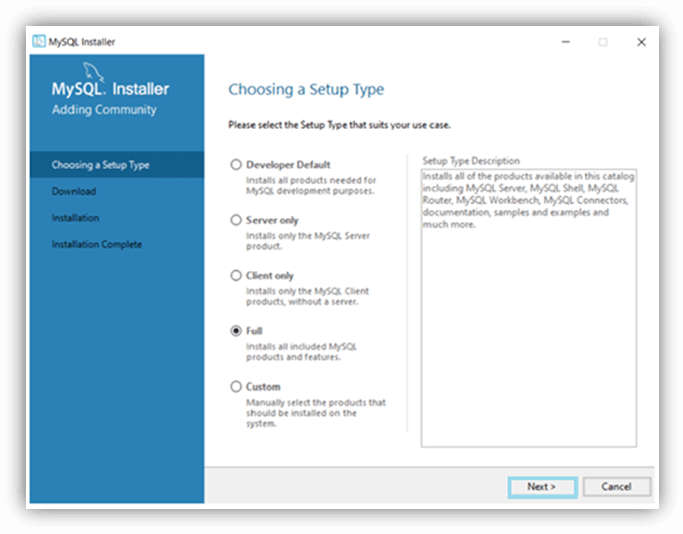
Write an available port number of your virtual device. Commonly, the MySQL server chooses the **3306**port number of your computer. After, click on the “**Next**” button for MySQL installation.



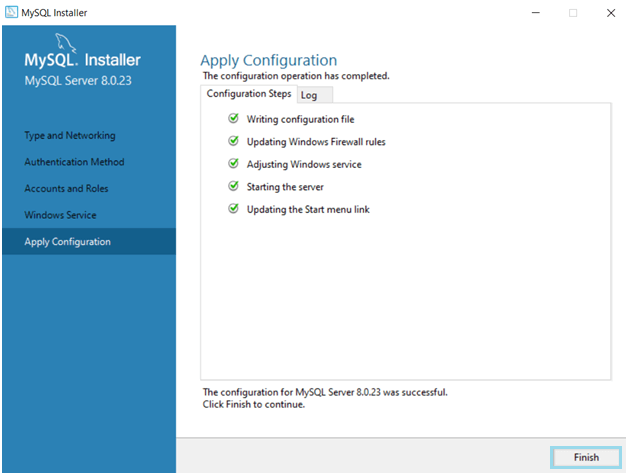
Create MySQL Root password for privacy and safety. It would be best if you can create a strong password. This password is necessary for every use of MySQL software.



MySQL installer chooses the required windows service and clicks on the “**Next**” button. MySQL is ready for the windows operating system.



Write the user name and password and check the authentication. If MySQL server displays “connection succeeded,” then click on the “**Next**” button.



Click on the “**Finish**” button. After, the installation process of the MySQL installer is complete. Here, MySQL 8.0.23 is successfully installed.

You mostly get the MySQL Command line client, MySQL workbench, MySQL shell interfaces, and other connectors.MySQL supports cross-platform. These are popular operating systems for students and developers. If you want to work on a different operating system, then install MySQL as per requirement.

**MySQL User Interface**

MySQL server has two popular ways to connect with a virtual device. These interfaces are easy, fast, and user-friendly.

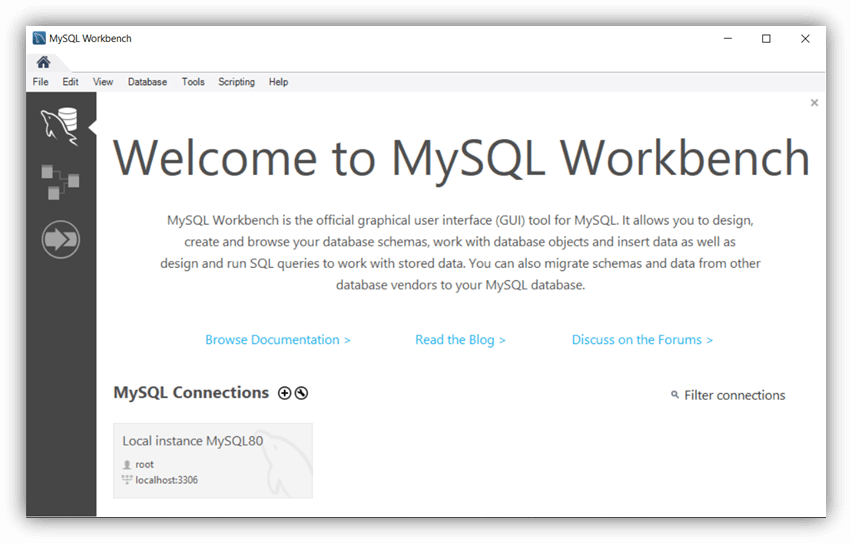
MySQL Command line client

MySQL workbench

**MySQL workbench**

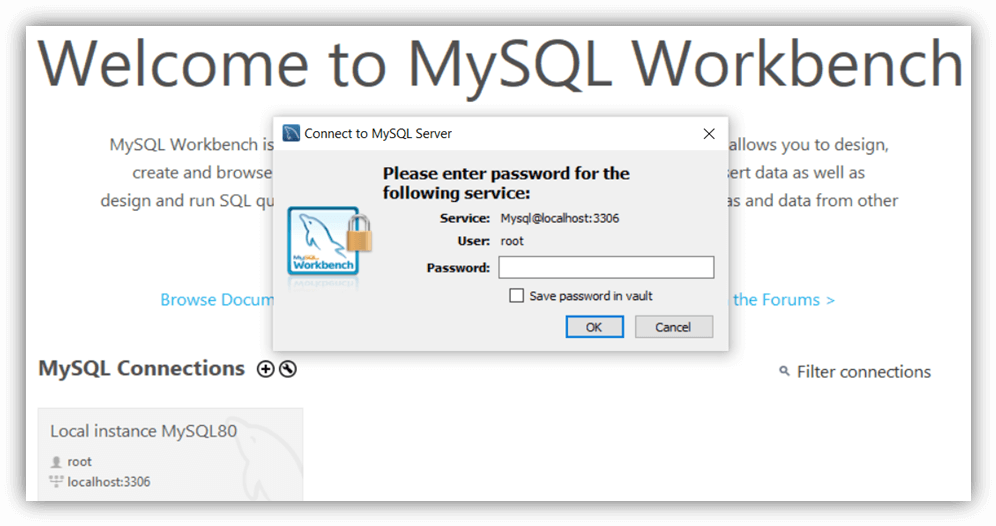
MySQL workbench is an application to operate MySQL server without command. Now, see the initial procedure of the workbench interface.

**Step1**: Open the MySQL workbench application. The first scene of the workbench is given below.



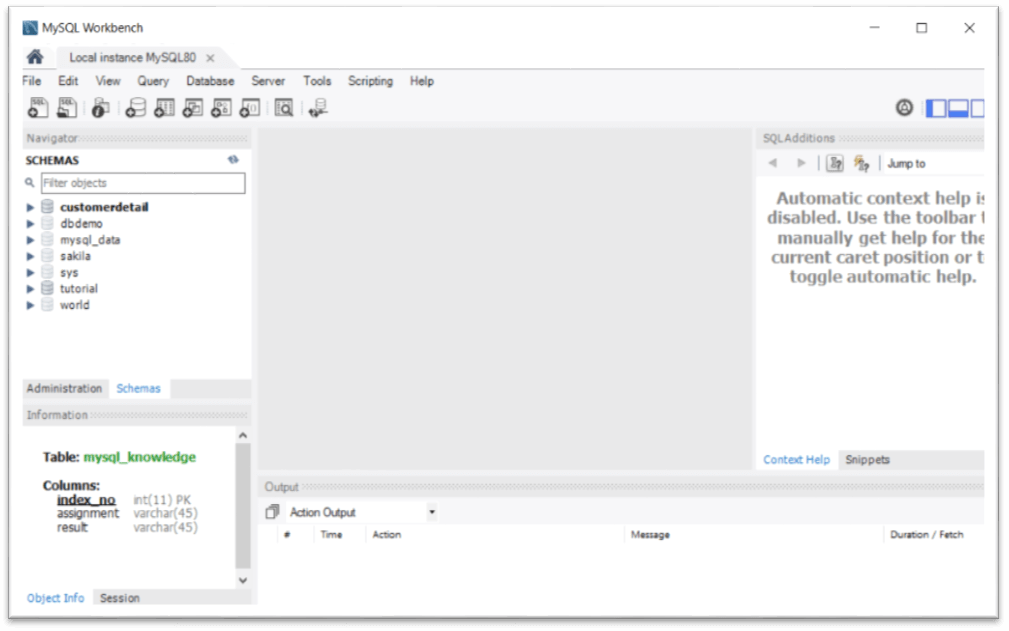
This interface creates more than one MySQL connection. The connection needs to set a unique password.

**Step2**:Create a MySQL connection using username and root. Every time user needs to enter a password to open MySQL workbench. The image displays the connection of the MySQL server.



Enter the password and press the “**OK**” button. If a password is correct, then the workbench will display the home page.

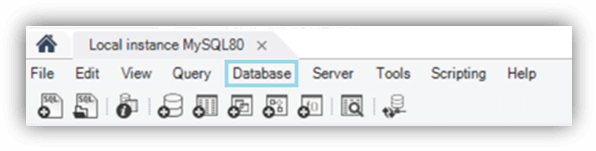
**Step3**: The MySQL workbench displays all databases and tables in the left side column. You can see the menu bar, navigator column, command column on the home page.



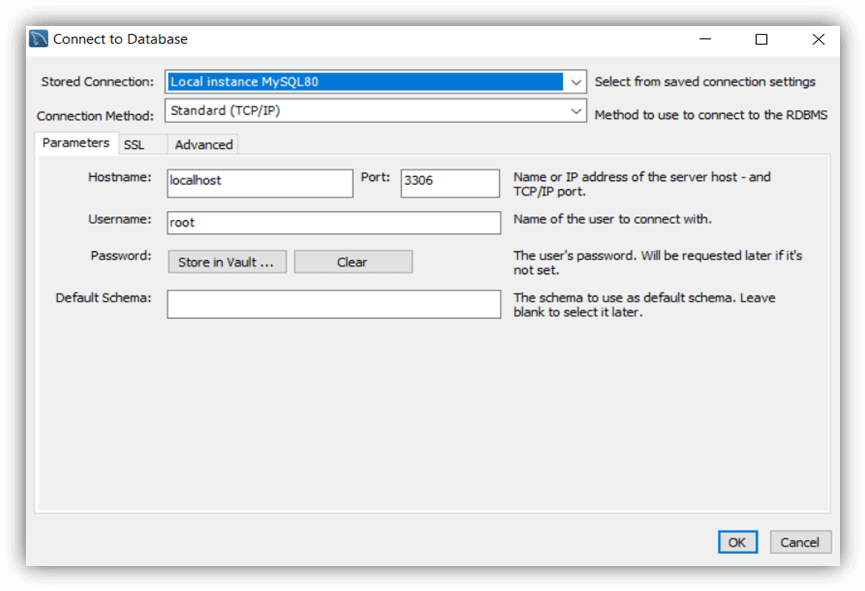
The above image shows the home page of the MySQL workbench interface. You can start the execution of the data operation.

**Step4**:Connect and manages database and connection.

The interface manages a database using the “**Database**” icon on the menu bar. Pressthe **CTRL+U** button.Next, click on the “**Connect to Database**” icon to display the required database.



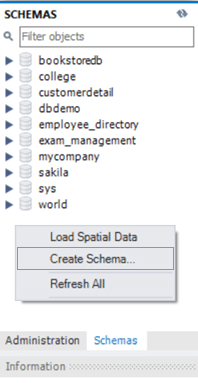
The image displays the menu bar of the workbench interface. This menu bar provides data operation quickly. It helps to manage critical operations and connections.



Here you are required to use the username and password every time you want to connect the database. Set up the MYSQL connections using the connection name, username, and required inputs. Press the “**OK**” button.

**Step5**: Create a new database or schema.

You can create a schema of MySQL as per project and requirement. The schema is similar to a command-line client database. You can create multiple schemas in one workbench interface.



The above image shows schemas and how to create a schema. You can click on “**Schemas**” and choose “**Create Schema…**” to contain multiple tables.

**Step8**:You can choose the required schema and create a table. The table is used to store data in row and column format.

**Step9**:You are ready to start a data management system. You can create, operate, and remove data using MySQL workbench.

* The schema / structure / other representation used in Mysql

-Schema

Before you can store anything in MySQL, you need to **clearly define tables and columns** and also, every row in the table should have the same column.

And because of this, there isn’t much space for flexibility in the manner of storing data if you follow normalization. **Development and deployment process is slowed down** as well due to the fact that even a little modification in data model mandates the change in schema design.

When it comes to choosing your database, one of the things you have to think about is the shape of your data, what model it will follow, and how the relationships formed will help us as we develop a schema.

A database schema is a **blueprint** or **architecture** of how our data will look. It doesn’t hold data itself, but instead describes the shape of the data and how it might relate to other tables or models. An entry in our database will be an instance of the database schema. It will contain all of the properties described in the schema.

Schema diagram from given data is as follows.



If the schema is then modified to accommodate new application requirements, the table is locked for some operations until existing data is copied into the new schema, requiring applications to be quiesced during schema migration.

This is how MySQL stores the data. As you can see, the table design is quite rigid and it is not easily changeable. You can’t even type letters where numbers are expected.

The schema can not be changed. The inputs following the given schema are only entered.

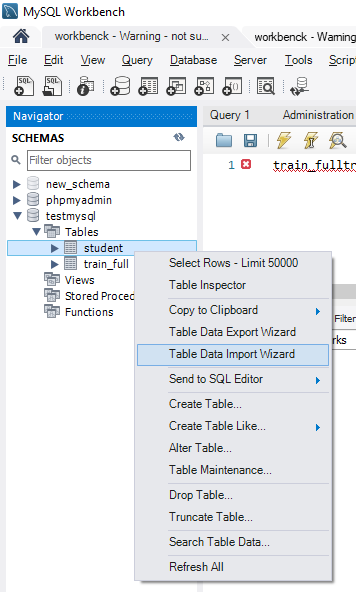
* Import csv file into mysql using GUI tool

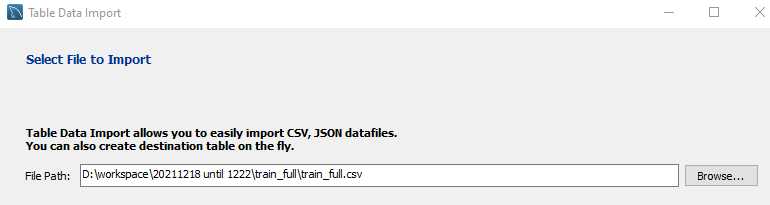
We used Mysqlworkbench as Mysql GUI tool .

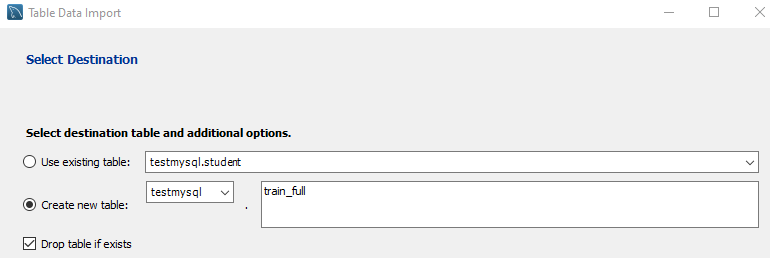
To import csv file into mysql database ,used Table Data Import Wizard.

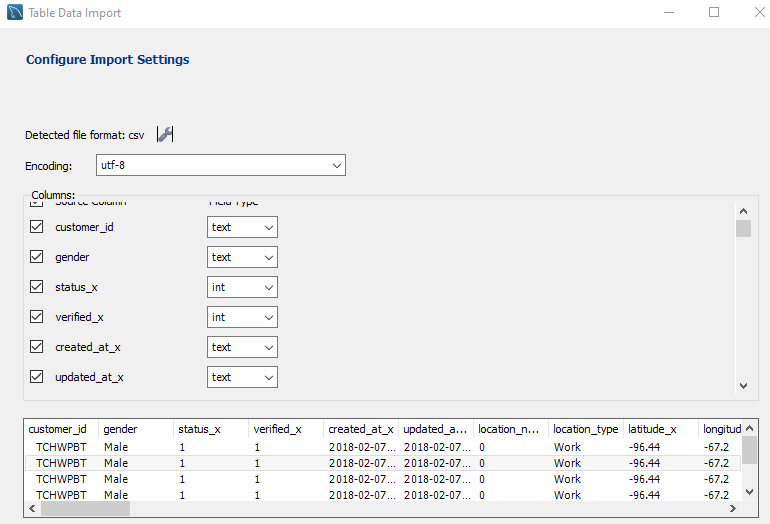
This wizard supports import operations using CSV and JSON files, and includes several configuration options (separators, column selection, encoding selection, and more). The wizard can be performed against local or remotely connected MySQL servers, and the import action includes table, column, and type mapping.

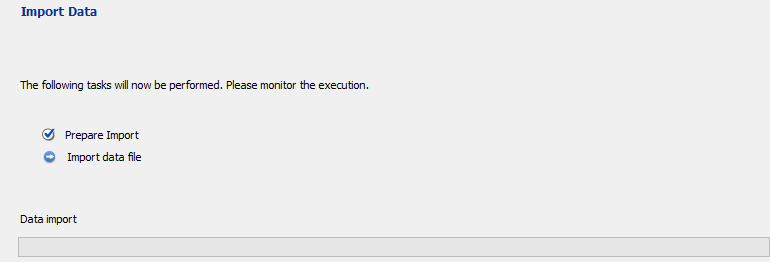
The wizard is accessible from the object browser's context menu by right-clicking on a table and choose either Table Data Export Wizard or Table Data Import Wizard, as the next figure shows.











* The exact queries [Write, Read, Delete, Group by, Order by] used for each of the operation mentioned above for Mysql database

WRITE: If a given record is not found in the storage, then the pair is added to the storage. Otherwise, it updates the value for the given key in the storage. This operation therefore combines Create and Update operations of the CRUD model.

query = "REPLACE INTO train\_full VALUES('TCHWPBT', 'Male', 1, 1, '2018-02-07 19:16:23', '2018-02-07 19:16:23', 0, 'Work', -96.44, -67.2, 4, 118597, -0.5884, 0.7544, 'Restaurants', 2, 0, 6, 1, '11:00AM-11:30PM', '-', 15, 0, 'Yes', 0, 1, 1, 11, 'EN', 4.4, '00:00:00', '00:30:00', '08:00:00', '23:59:00', '00:00:00', '00:30:00', '08:00:00', '23:59:00', '00:00:00', '00:30:00', '08:00:00', '23:59:00', '00:00:00', '00:30:00', '08:00:00', '23:59:00', '00:00:00', '00:30:00', '08:00:00', '23:59:00', '00:00:00', '00:30:00', '10:00:00', '23:59:00', '00:00:00', '00:30:00', '10:00:00', '23:59:00', '{\"primary\_tags\":\"4\"}', 1, '2,4,5,8,91,22,12,24,16,23', 'Arabic,Breakfast,Burgers,Desserts,Free Delivery,Grills,Lebanese,Salads,Sandwiches,Shawarma', 'Y', 1, 1, '2018-01-30 14:42:04', '2020-04-07 15:12:43', 3, 1, 0, 4, 'TCHWPBT X 0 X 4', 0)

Read: SELECT name, birth FROM train\_full ORDER BY id;

Delete: DELETE FROM train\_full CID\_X\_LOC\_NUM\_X\_VENDOR =' TCHWPBT X 0 X 4 '

Group by, order by: select id, customer\_id, gender, location\_type, vendor\_tag\_name from train\_full group by special order by id desc

* Various performance parameters considered (with reasons) while executing the   
  queries and their analysis

Query : insert, select, update

Mysql

MySQL is a relational database that stores data in a tabular structure and is grouped by tables. It uses SQL as language for communication with the database, which allows for a very powerful syntax to create both simple and complex queries to retrieve and structure data as you want. A relational database is, as the name implies, useful when the data you want to store in different tables has a relation between each other. MySQL comes with with 8 different database engines which can be chosen based on different criteria that one might have. By default, MySQL uses InnoDB if no other engine is defined when creating a new database.

The measurement of performance between the database technologies was made by looking how fast each technology could finish the execution of four different types of queries:

Inserting data

Updating data

Removing data

Selecting data

These operations were chosen because they are the four main ones when operating on databases.

* Visualizations based on the performance parameters used for the queries and their interpretation

Selecting data test, Removing data test, Inserting data test, update data test

Our first experiment measures the time taken to select data from database. See Figure 1 which summarizes the results of this experiment.

For each of these tests, we run the chosen operation (such as Read or Write) five times and take the average time.

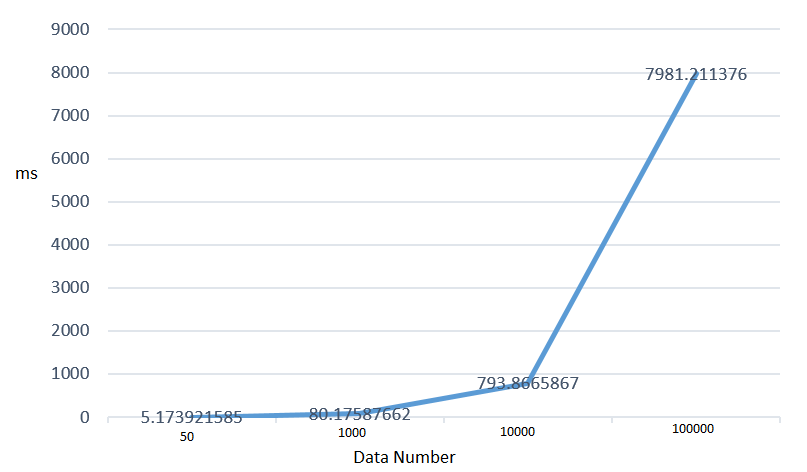


Fig. 1. Time for selecting data form database (ms)

Note that the times are averaged over five runs. The absolute time values are not significant; what is significant is the time values relative to one another.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Number of operations | | | |
| 50 | 1000 | 10000 | 100000 |
| 1 | 0.004927397 | 0.07976532 | 0.816764116 | 8.040037155 |
| 2 | 0.004983425 | 0.079787016 | 0.787860394 | 7.956768751 |
| 3 | 0.005979776 | 0.079758167 | 0.788890839 | 7.97282052 |
| 4 | 0.004991531 | 0.081809282 | 0.786921263 | 7.958940268 |
| 5 | 0.004987478 | 0.079759598 | 0.788896322 | 7.977490187 |

Table 1. Time for selecting (ms)

Our second experiment measures the time taken to remove from the database. Figure and Table II summarizes the result.

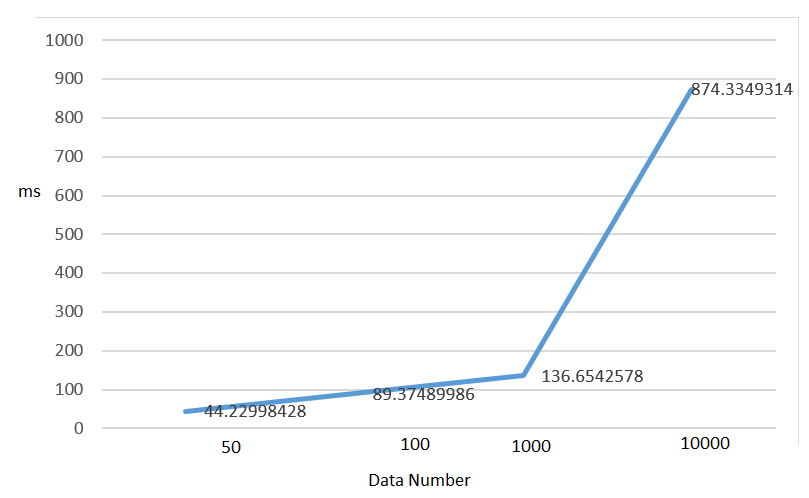


Fig. 2. Time for removing data form database (ms)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Number of operations | | | |
| 50 | 100 | 1000 | 10000 |
| 1 | 0.043557882 | 0.156238079 | 0.157536507 | 0.248025894 |
| 2 | 0.026753187 | 0.019946337 | 0.033630848 | 1.709678411 |
| 3 | 0.021312714 | 0.038923502 | 0.041273355 | 3.008111 |
| 4 | 0.185551405 | 0.037049055 | 0.067420959 | 2.009045362 |
| 5 | 0.044902086 | 0.022511721 | 0.115772009 | 1.500643969 |

Table 2. Time for removing (ms)

Our third experiment measures the time taken to insert to the database. Figure and Table II summarizes the result.

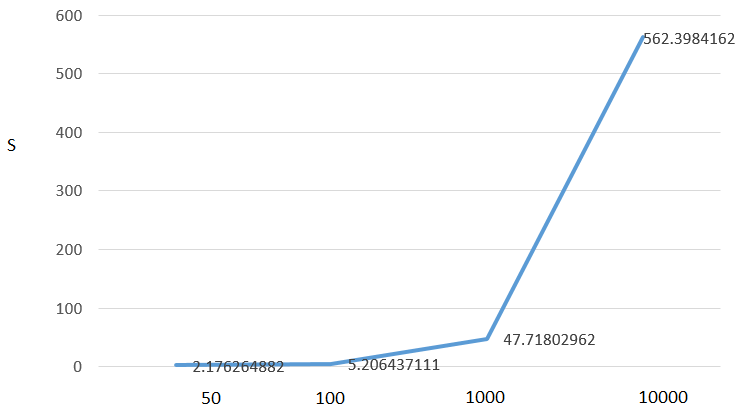


Fig. 3. Time for inserting data form database (ms)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Number of operations | | | |
| 50 | 100 | 1000 | 10000 |
| 1 | 2.368009329 | 5.073161125 | 43.22420478 | 553.4642324 |
| 2 | 1.791949749 | 4.92201829 | 47.57787848 | 561.32564 |
| 3 | 2.105541706 | 4.780855417 | 50.0625596 | 542.9563 |
| 4 | 1.912190437 | 4.831614256 | 50.39170504 | 555.2356 |
| 5 | 1.984520435 | 5.339713097 | 52.21185446 | 571.3326 |

Table 3. Time for inserting (ms)

Our final experiment measures the time taken to update to the database. Figure and Table II summarizes the result.

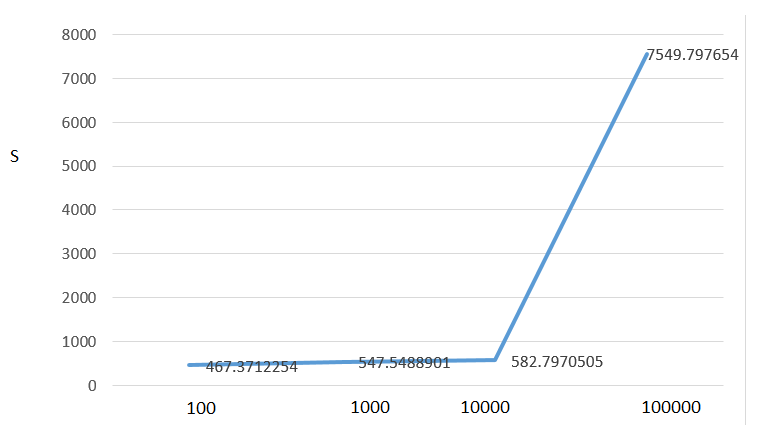


Fig. 4. Time for updating data form database (ms)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Number of operations | | | |
| 100 | 1000 | 10000 | 100000 |
| 1 | 0.518040419 | 0.562349319 | 0.700570345 | 13.86146951 |
| 2 | 0.417593241 | 0.672270298 | 0.507528067 | 1.317472696 |
| 3 | 0.416009188 | 0.883544922 | 0.513138056 | 1.166902065 |
| 4 | 0.417318344 | 0.634755373 | 0.509124041 | 0.962557316 |
| 5 | 0.416702032 | 0.532748461 | 0.465023756 | 1.238125801 |

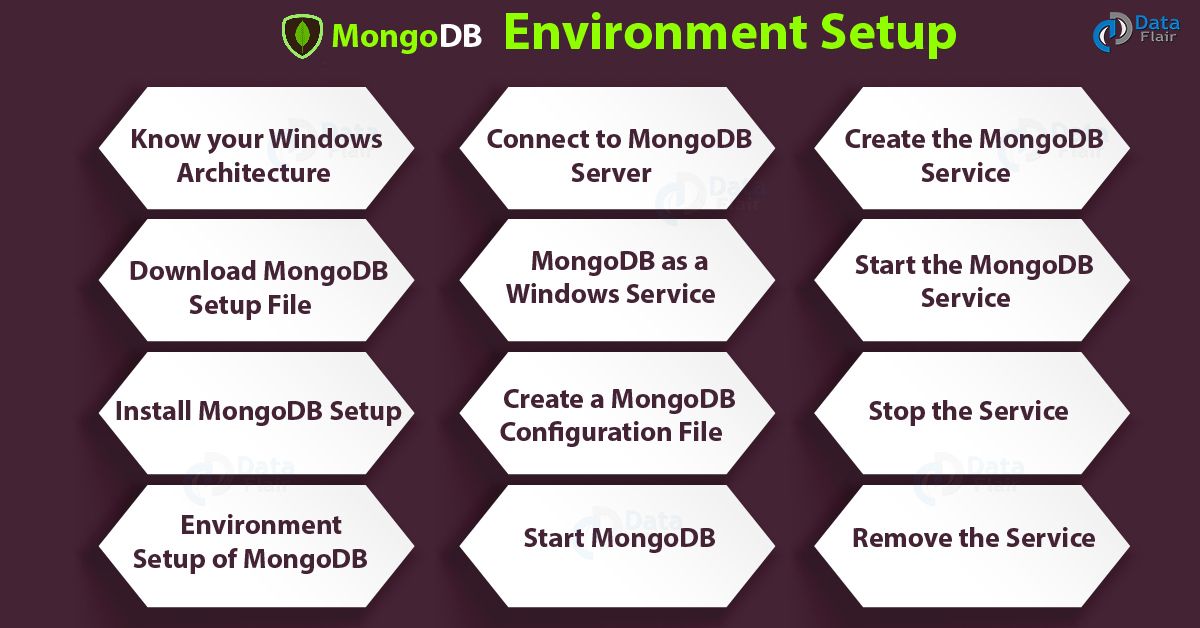
Table 4. Time for updating (ms)

Selecting all documents in the database performed almost equally when it contained 103 and 104 documents. When it contained 105 documents MySQL Document Store increased more in time than MongoDB did, and at 106 documents the difference was even larger. See figure 9.

NO SQL [ MongoDB]

* ENVIRONMENT SETUP

The following figure is showed Mongodb environment setup.



MongoDB Environment Setup is very easy for Windows OS. To do environment setup, you will have to follow few simple steps.

Know your Windows architecture

Download MongoDB setup file

Install MongoDB setup

Set up MongoDB Environment

Connect to the MongoDB server

MongoDB as a Windows service

Create configuration file

Run MongoDB Environment setup

1. Know your Windows Architecture

Before downloading MongoDB setup, you must know which Windows version you are using. To know about your system architecture, open the command prompt and execute some commands given below.

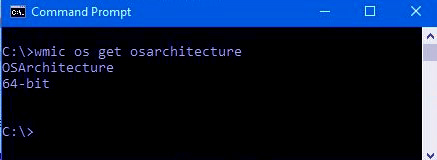
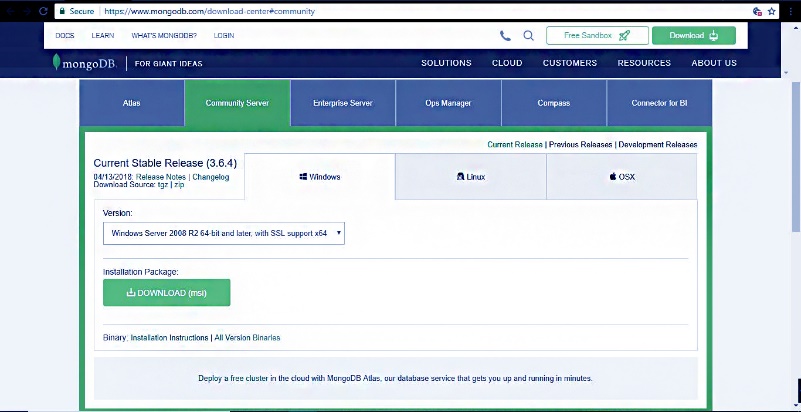
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/cmd.jpg)

Image.1 MongoDB Environment Setup – Know your Windows Architecture

After executing these commands, you will know if your system is running on 32-bit or 64-bit architecture. Then you have to download the MongoDB setup accordingly.

2. Download MongoDB Setup File

After knowing the architecture, you have to download the latest version of MongoDB from the official website.

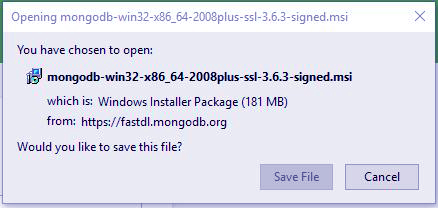
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/Capture-2-1.jpg)

*Image.2 MongoDB Environment Setup – Download MongoDB Setup File*

You can follow this link to download from community server of MongoDB.

[ttps://www.mongodb.org/downloads](https://www.mongodb.org/downloads)

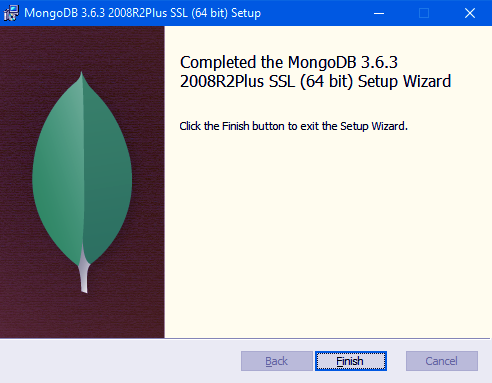
Download the MSI file from this server.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/INSTALLATION-PACKAGE.jpg)

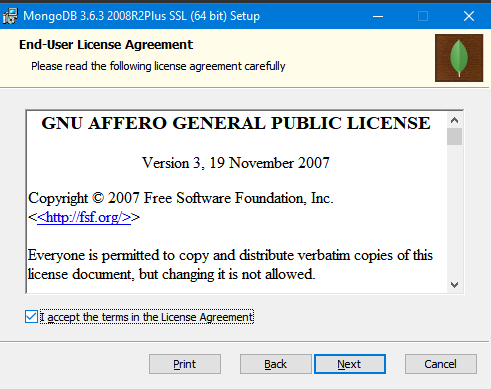
*Image 3. Download MongoDB Setup File*

Install MongoDB Setup

To install MongoDB setup, you need to run the file as administrator. After doing that you need to follow the setup guide that appears in the installation process.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/Screenshot-47-1.png)

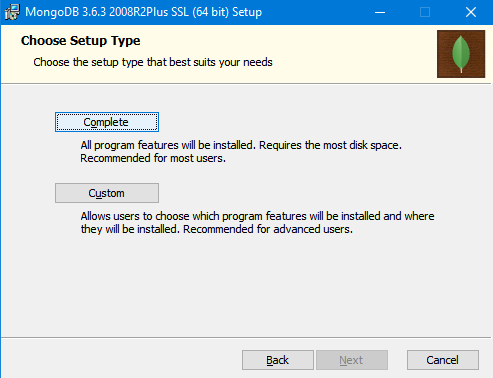
*Image. 4 MongoDB Environment Setup – Install MongoDB Setup*

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/Screenshot-36-2.png)

*Image. 5 MongoDB Environment Setup – Install MongoDB Setup*

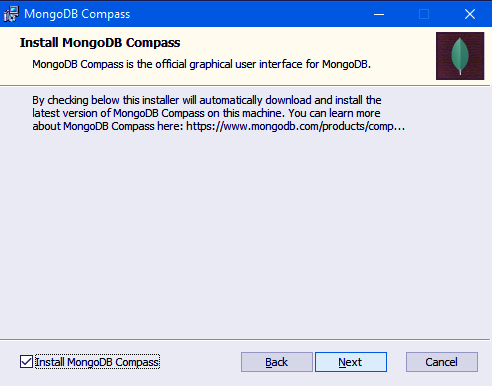
You can choose a custom directory for the MongoDB data and log files. Otherwise, by default, it will install MongoDB to C:\Program Files\MongoDB\Server\3.6\.

MongoDB does not have any system dependency, so it is not problematic to choose any folder in the directory. You can choose any directory in the system.

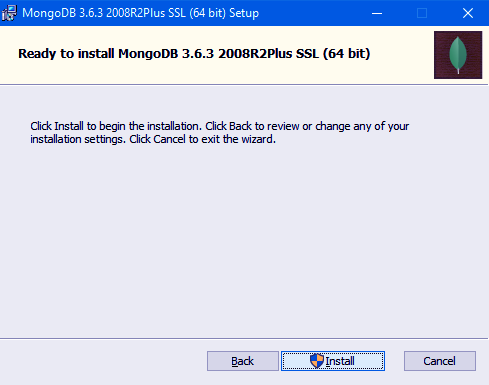
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/Screenshot-38.png)

*Image. 6 MongoDB Environment Setup – Install MongoDB Setup*

You will see the option to install MongoDB compass in addition to MongoDB serve and you can check the box if you want it to install MongoDB. If not, leave it blank.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/Screenshot-39-1.png)

*Image. 7 MongoDB Environment Setup – Install MongoDB Setup*

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/Screenshot-40-1.png)

*Image. 8 MongoDB Environment Setup – Install MongoDB Setup*

4. MongoDB Environment Setup

All of MongoDB data is stored in a specified data directory. You need to create it manually in the MongoDB Folder on the C drive. For this, run the following command at the command prompt.

 md \data\db

You can install MongoDB at any location of your choice.

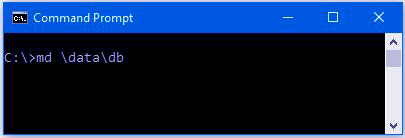
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/path.jpg)

Image. 9 MongoDB Environment Setup

5. Start MongoDB

To start MongoDB you need to execute this command. This will start the main MongoDB process. You will see at the bottom of the command prompt a message as “waiting for a connection”. This means that the process has started successfully.

“C:\Program Files\MongoDB\Server\3.6\bin\mongod.exe”

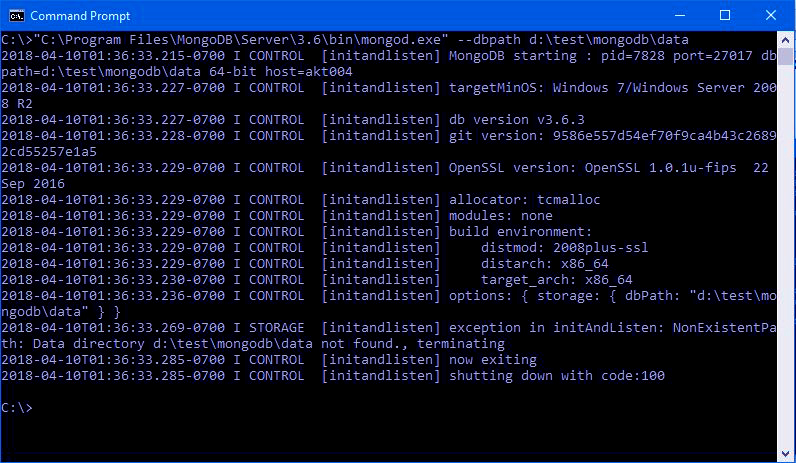
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/pathh2.jpg)

Image. 10 MongoDB Environment Setup – Start MongoDB

6. Connect to MongoDB Server

After executing the MongoDB.exe file, the process has begun. Now we need to connect it to the MongoDB server. We will connect it through Mongo.exe shell by opening another command prompt. Now you need to execute a command to connect with the shell.

“C:\Program Files\MongoDB\Server\3.6\bin\mongo.exe”

Now MongoDB is ready to use. You can terminate the running Process of MongoDB by pressing “ctrl+c”.

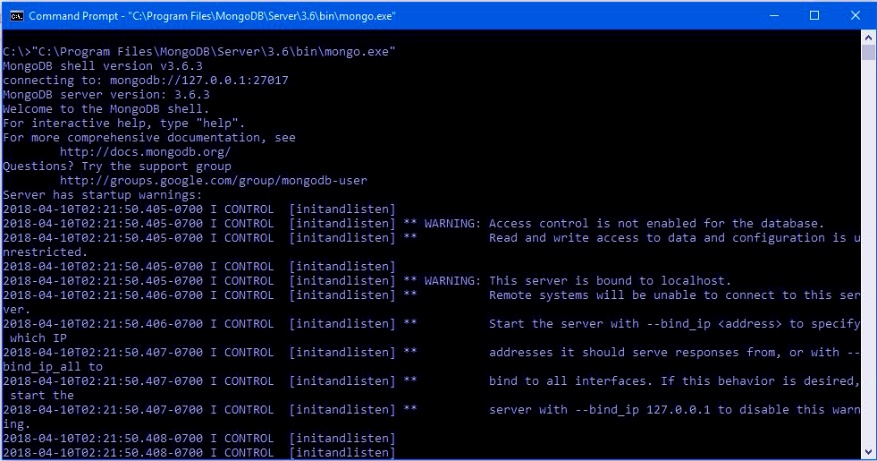
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/ongoDBSHell.jpg)

Image 11. MongoDB Environment Setup – Connect to MongoDB Server

7. MongoDB as a Windows Service

Now you need to create some directories using command prompt & executing the following commands:

mkdir c:\data\db

mkdir c:\data\log

Also, you can directly create new folders on the c drive. First, create a folder and name it “data” then create two folders as “db” and “log” inside the data folder. All of your database and the log files will be stored in these two folders.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/04/datalog.jpg)

Image 11. MongoDB Environment Setup – MongoDB as a Windows Service

8. Create a MongoDB Configuration File

After creating directories, you need to create a configuration file. The file must specify both the systemLog.path and storage.dbPath. Otherwise, an error will be shown on the display and service will not be able to start.

systemLog:  
    destination: file  
    path: c:\data\log\mongod.log  
storage:  
    dbPath: c:\data\db

9  Create the MongoDB Service

You need to execute this command for creating the MongoDB service

sc.exe create MongoDB binPath= “\”C:\Program Files\MongoDB\Server\3.6\bin\mongod.exe\” –service –config=\”C:\Program Files\MongoDB\Server\3.6\mongod.cfg\”” DisplayName= “MongoDB” start= “auto”

If the service is successfully created the following message will display

[SC] CreateService SUCCESS

10. Start the MongoDB Service

If you want to start the service, you have to write a single sentence in a command prompt.  
net start MongoDB.

11. Stop the Service

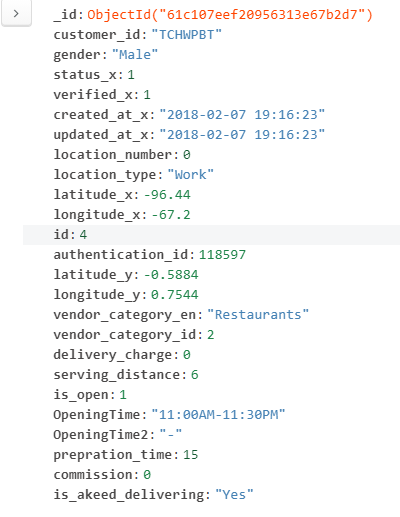
Similarly, to stop the service you have to execute the following command.  
net stop MongoDB

12. Remove the Service

You can delete the service whenever you want by executing this simple code. net stop MongoDB   
This will first stop the service then delete it from the system.

This was all about MongoDB Environmental Setup Tutorial. Hope you like our explanation of How to Install MongoDB on Windows 10.

* The schema / structure / other representation used in each of the databases



MongoDB is amazing for modern-day applications as it offers a flexible schema design that allows you to meet the ever-changing conditions of Big Data applications. With MongoDB, you can easily store and combine any type of data and dynamically modify schema without experiencing application downtime.

So MongoDB is a schema-less database (written in C++) which makes it flexible.

The fact that there are no limitations on schema design is the greatest advantage to **MongoDB**. Within a collection, you may drop a few documents and it is not important for these documents to have any relations. The only limitation to this is the supporting [data structures](https://codersera.com/blog/top-15-data-structures-interview-questions-answers/). You might need to consistently optimize the schema depending on how the application can access the data because of the lack of Joins and transactions.

However, you need to properly identify tables and columns before you can store something in **MySQL**, and every row in your table needs to have the same column. And because of this, if you pursue normalization, there’s not much room for flexibility in the way of storing data.

* The exact queries [Write, Read, Delete, Group by, Order by] used for each of the operation mentioned above for each database

Read: db.train\_full.find({id: 80})

Delete: db.train\_full.remove({‘customer\_id’: ‘MKOW15V’})

Group by: db.train\_full.aggregate([ {$group: {\_id: {customer\_id:"$customer\_id", gender:"$gender"} } } ])

Order by: db.train\_full.find().sort({id:1})

* Various performance parameters considered (with reasons) while executing the   
  queries and their analysis

The measurement of performance between the database technologies was made by looking how fast each technology could finish the execution of four different types of queries:

Inserting data

Updating data

Removing data

Selecting data

These operations were chosen because they are the four main ones when operating on databases.

* Visualizations based on the performance parameters used for the queries and their interpretation

Our first experiment measures the time taken to select data from database. See Figure 1 which summarizes the results of this experiment.

For each of these tests, we run the chosen operation (such as Read or Write) five times and take the average time.

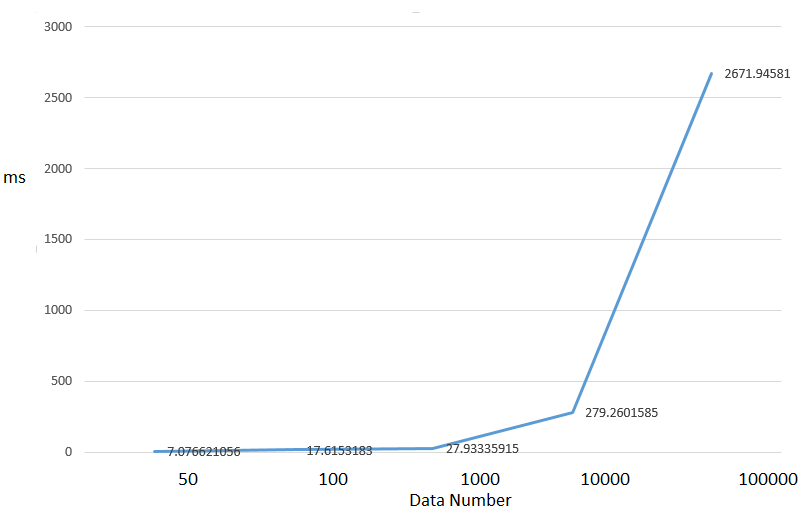


Fig. 5. Time for selecting data form database (ms)

Note that the times are averaged over five runs. The absolute time values are not significant; what is significant is the time values relative to one another.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Number of operations | | | | |
| 50 | 100 | 1000 | 10000 | 100000 |
| 1 | 0.013179064 | 0.03223753 | 0.030944347 | 0.294258118 | 2.523763657 |
| 2 | 0.002005339 | 0.003017664 | 0.02793479 | 0.267257929 | 2.847629309 |
| 3 | 0.001984358 | 0.002966404 | 0.025922775 | 0.26631546 | 3.213415384 |
| 4 | 0.001020193 | 0.003015757 | 0.024938107 | 0.261460781 | 2.765344381 |
| 5 | 0.000974178 | 0.002993107 | 0.024922371 | 0.264262199 | 2.820127964 |

Table 5. Time for selecting (ms)

Our second experiment measures the time taken to insert from the database. Figure and Table 6 summarizes the result.

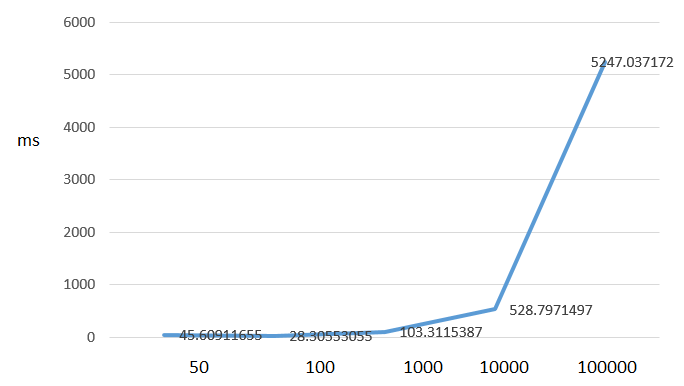


Fig. 6. Time for inserting data form database (ms)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Number of operations | | | | |
| 50 | 100 | 1000 | 10000 | 100000 |
| 1 | 0.088199139 | 0.051624775 | 0.157753944 | 0.567047358 | 5.157520294 |
| 2 | 0.00398922 | 0.005014658 | 0.062832594 | 0.504261494 | 5.171452999 |
| 3 | 0.015985966 | 0.021953344 | 0.087920666 | 0.501516819 | 5.313789606 |
| 4 | 0.00302124 | 0.034431458 | 0.081781149 | 0.462757587 | 5.120893955 |
| 5 | 0.003019094 | 0.004986286 | 0.048869133 | 0.490546942 | 5.33655405 |

Table 6. Time for inserting (ms)

Our third experiment measures the time taken to insert to the database. Figure and Table 7 summarizes the result.

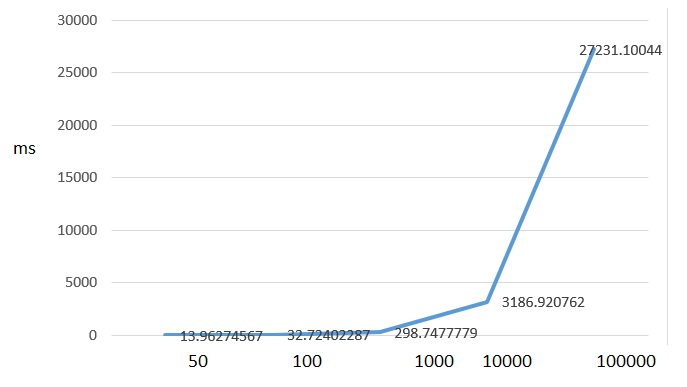


Fig. 7. Time for removing data form database (ms)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Number of operations | | | | |
| 50 | 100 | 1000 | 10000 | 100000 |
| 1 | 0.014960289 | 0.031462669 | 0.320337534 | 3.219058275 | 27.28431296 |
| 2 | 0.01396203 | 0.035155535 | 0.281220675 | 3.272027254 | 26.58778715 |
| 3 | 0.013962507 | 0.034183741 | 0.26046586 | 3.189687252 | 26.47356582 |
| 4 | 0.013962507 | 0.030883551 | 0.282449245 | 3.170702696 | 27.40680909 |
| 5 | 0.012965202 | 0.033985376 | 0.277158022 | 3.154783249 | 27.177892 |

Table 1. Time for removing (ms)

Our final experiment measures the time taken to update to the database. Figure and Table 8 summarizes the result.

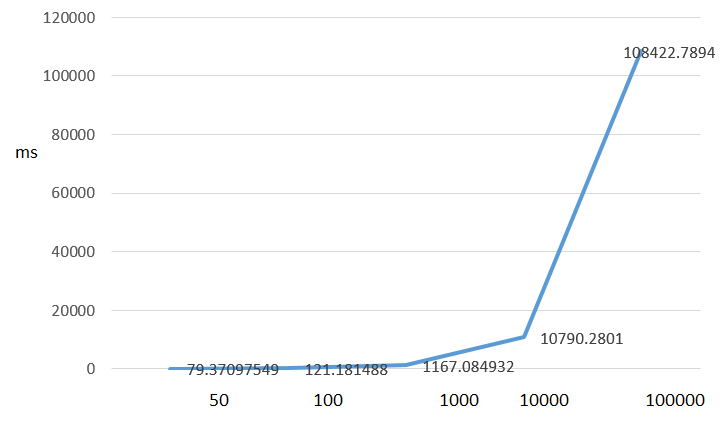


Fig. 8. Time for updating data form database (ms)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Number of operations | | | | |
| 50 | 100 | 1000 | 10000 | 100000 |
| 1 | 0.095831633 | 0.123703718 | 1.202823162 | 10.72178888 | 105.6098988 |
| 2 | 0.061834574 | 0.137634039 | 1.214331388 | 10.89841962 | 109.1234 |
| 3 | 0.079785347 | 0.117112875 | 1.186710358 | 10.78130341 | 106.2346461 |
| 4 | 0.059839725 | 0.109879017 | 1.119815111 | 10.87631011 | 110.2356 |
| 5 | 0.062910318 | 0.118659258 | 1.131346703 | 10.85877132 | 111.23568 |

Table 8. Time for updating (ms)

Summary Details:

* summary of the prototyping exercise
* First I imported package connecting Mysql and mongodb
* Declaration of class to create and read and update and remove on database table
* Application of class
* Set connection config for example server url and username and password
* Instantiate class
* Prepare the query
* Create, read, update and remove run
* Clearly provide your recommendation

In every single test we found MongoDB to perform better than MySQL DS. This was specially noticeable when operating on 105 documents or more, where MySQL Document

Store could perform multiple times worse than MongoDB.

But since MySQL Document Store was recently released you could expect some performance issues.