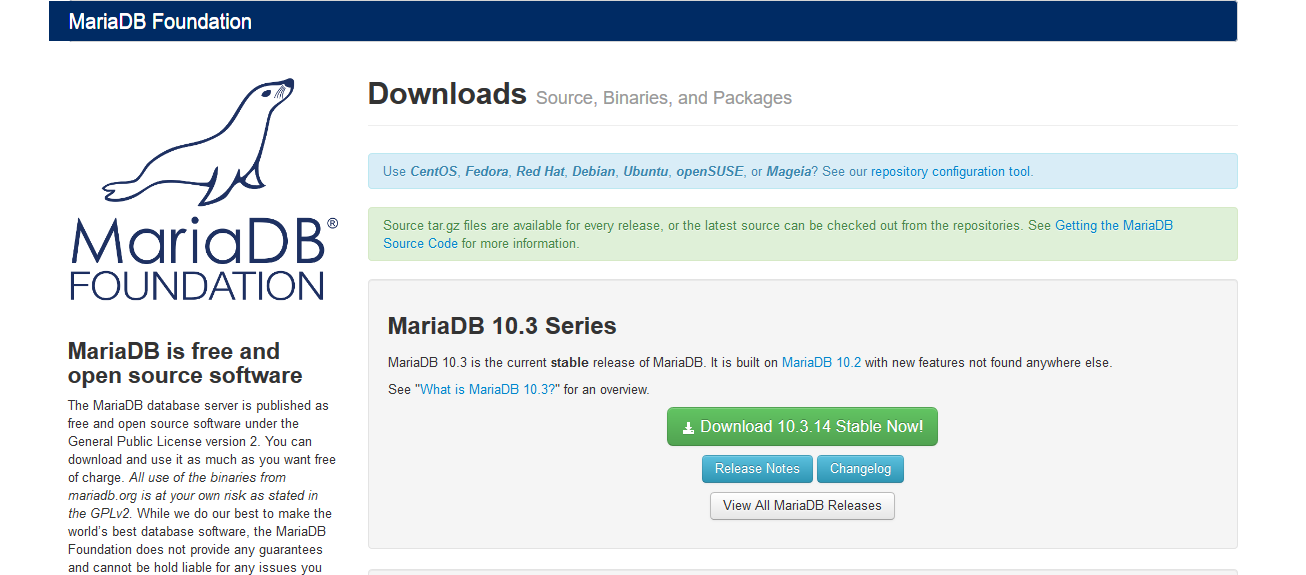
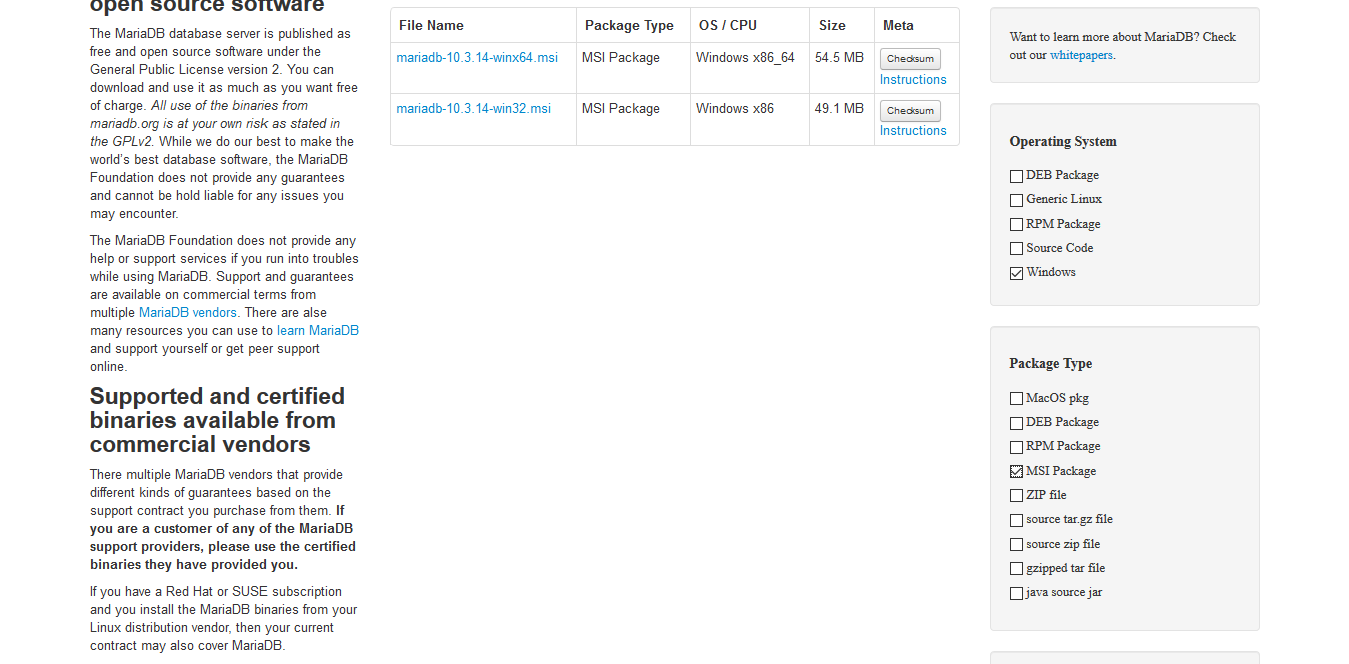
# MySQL/MariaDB Install Instructions

## Download MariaDB 10.3

In your web browser, go to https://downloads.mariadb.org/ and click the green download button below “MariaDB 10.3 Series”.



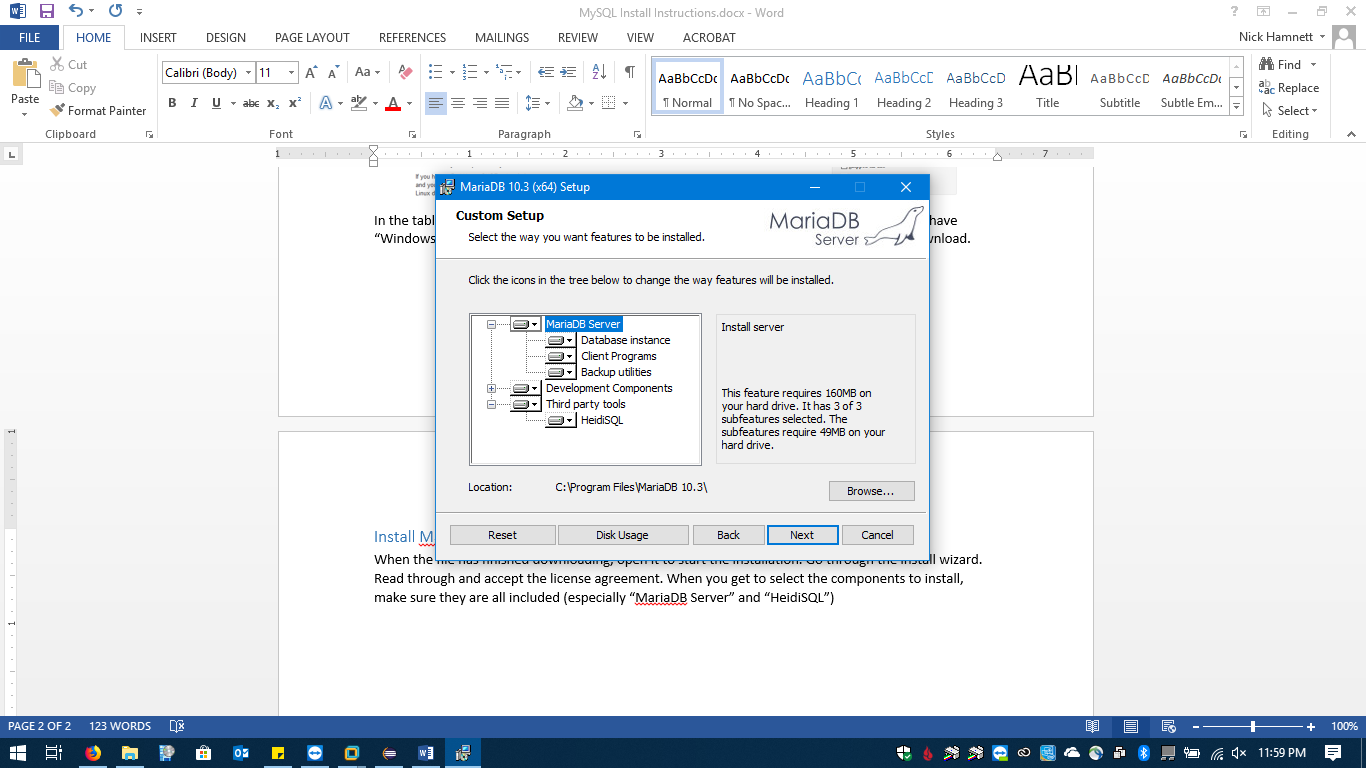
On the right side of the following page, check “Windows” below Operating System and “MSI Package” below Package Type.



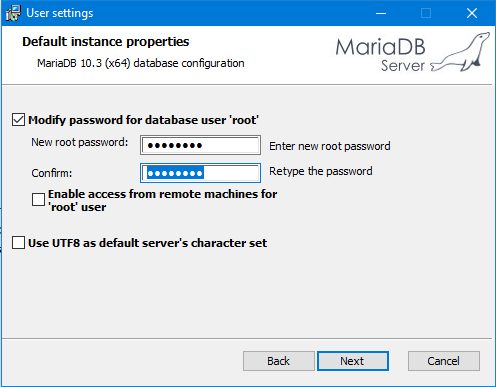
In the table in the middle of page, click the “mariadb-10.3.XX-winx64.msi” link (which should have “Windows x86\_64” as the OS / CPU). It will bring you to another page which will start the download.

## Install MariaDB 10.3

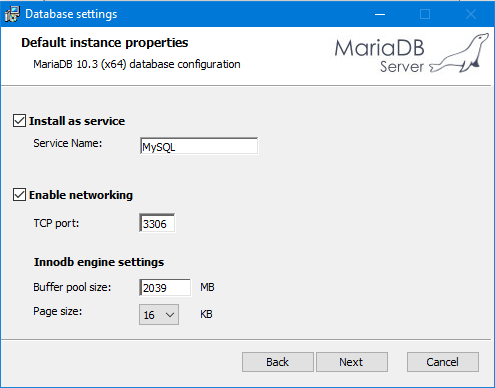
When the file has finished downloading, open it to start the installation. Go through the install wizard. Read through and accept the license agreement. When you get to select the components to install, make sure they are all included (especially “MariaDB Server” and “HeidiSQL”).



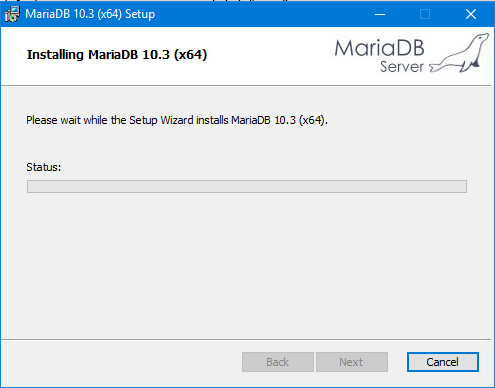
Click “Next” and you will be prompted for the root account password. It’s recommend you set it to “password”. If you set it to something else, make a note of it somewhere. Leave “Enable access from remote machines for root user” and “Use UTF8 as default server character set” unchecked. Click Next.



Leave the default instance properties as is and click “Next”.



Do not opt-in to submit usage information and go to the next page, where you click “Install” to begin the installation process.

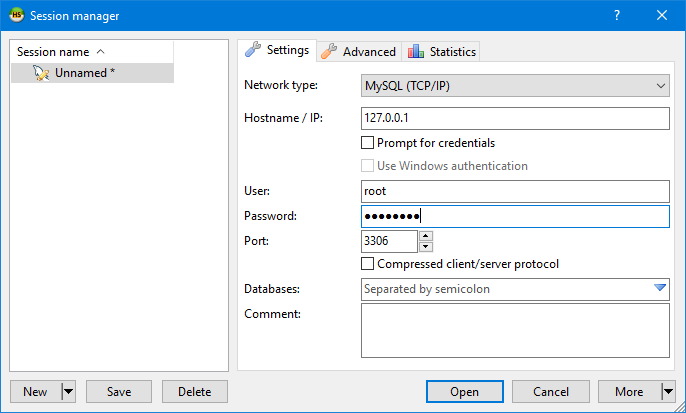


When the install is finished, click “Finish” to exit the install wizard.

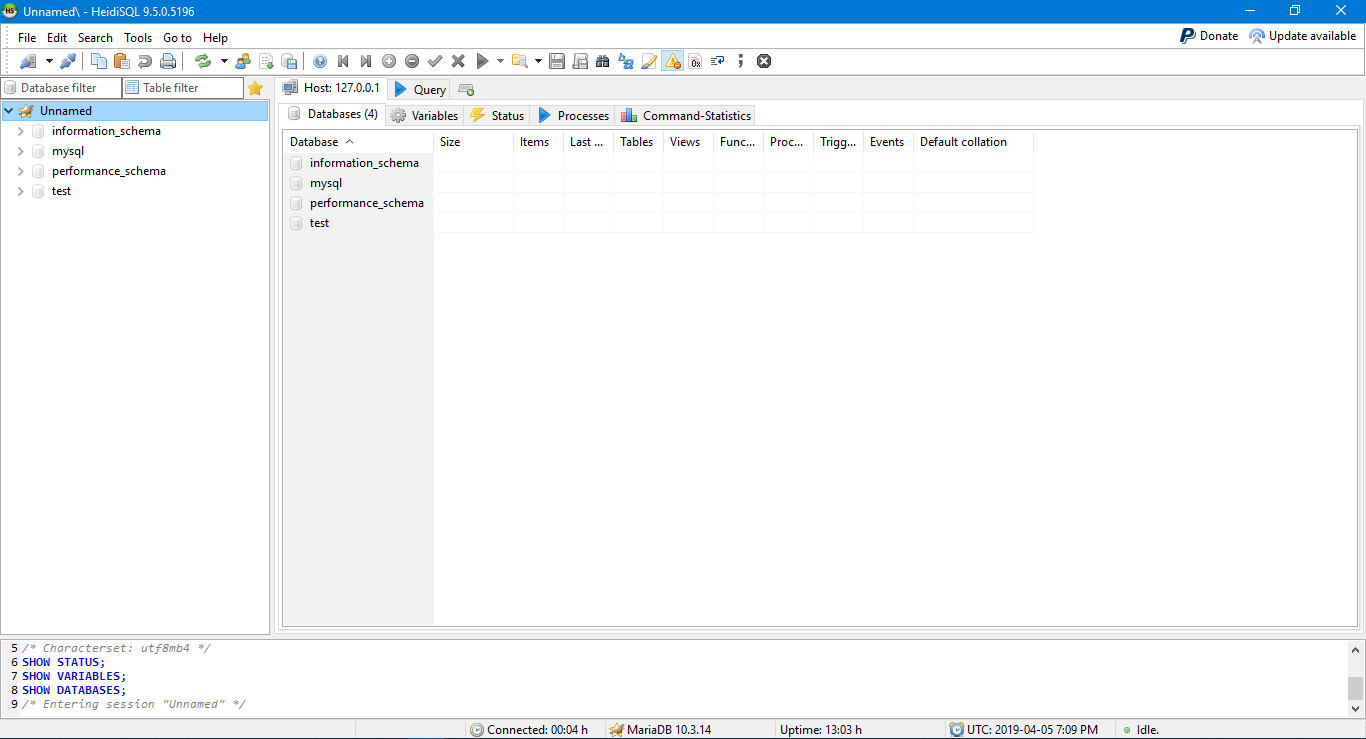
## Create Database

You will need to create a database that will hold any tables for your assignment. Open HeidiSQL using the shortcut on either your desktop or start menu. When opening it, you may be asked to update it but click “Skip” to ignore the update.

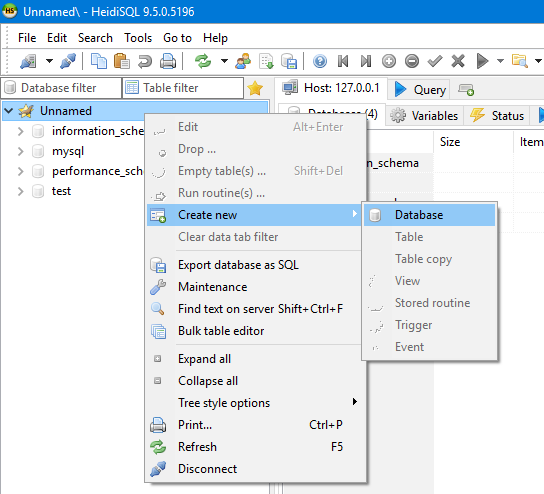
In the Session manager, leave the Network type as “MySQL (TCP/IP)” and Hostname/IP as “127.0.0.1”. The username should be “root” and password should be “password” (assuming that is what you set it previously). Leave the rest of the settings as they are and click “Open”.



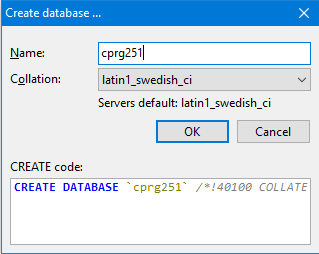
You should be brought to a new window that shows a list of databases.



On the left sidebar, right click on “Unnamed”. Go to “Create new” and left click on “Database”.

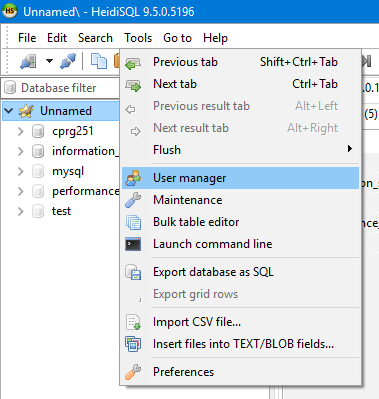


A window should come up asking for the new databases information. For the Name, enter “cprg251”, leave the collation as “latin1\_swedish\_ci” and then click OK.



The database should be created and listed in the left sidebar.

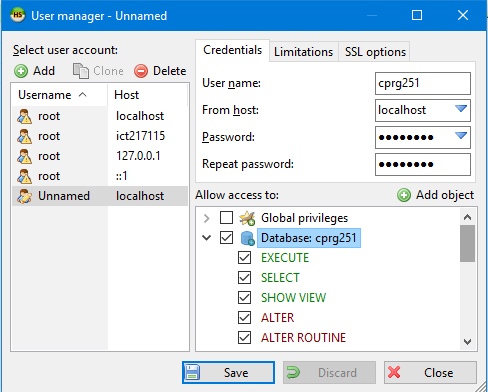
A user will need to be created that just has access to the cprg251 database. Under the “Tools” menu, open the “User manager”.



In the window that pops up, click the “Add” button on the top left. A new user should be added to the list and selected.

Enter “cprg251” for the user name. Leave the host as “localhost”. Enter “password” into both the password fields.

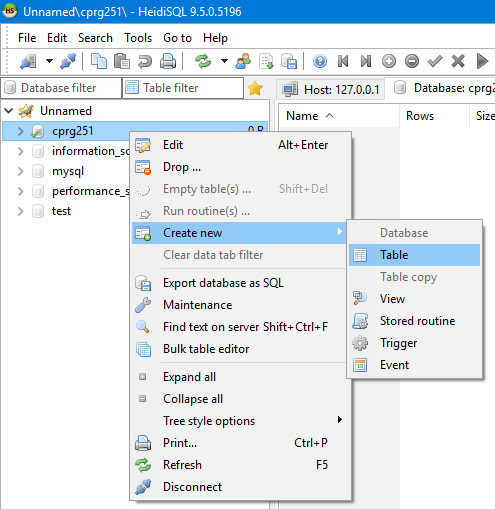
Beside “Allow access to”, click the “Add object” button. A window showing a list of the databases will be displayed. Select “cprg251” and click OK. In the tree list, check “Database: cprg251” so all of the privileges are given to the user.



Finally, click the “Save” and then “Close” button.

Now that you have the user and database created, you can begin creating and populating tables.

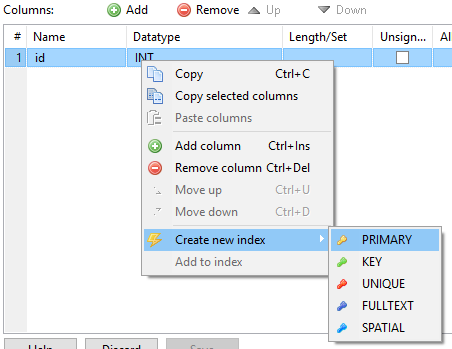
You can create a table in HeidiSQL by right clicking the “cprg251” database from the left sidebar and going to “Create new” and then “Table”.



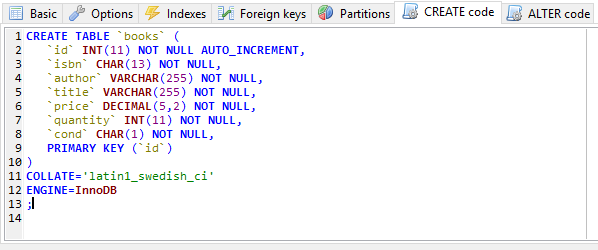
You will need to enter the name of the table and then add the columns in the table. The first column should be set with the following:

* Name: id
* Datatype: INT
* Allow Null: No
* Default: AUTO\_INCREMENT

Right click on the row that represents the column and go to “Create new index” and then select “PRIMARY”. This sets the id column has the primary key for the table.



When you have created the necessary columns in your table, you can view the SQL for creating the table by looking in the “CREATE code” tab.



It is useful to put the CREATE TABLE query into your Java code. Then use it with the driver to create the table.

## Configure Your Eclipse Project

The Eclipse project will need to have the required files so you can it can interact with the database. With your assignment 6 project created in Eclipse (possibly by creating a copy of assignment 5):

1. Download the included “mariadb-java-client-2.4.1.jar” file and place it in the “lib” folder of your assignment 6 directory. Refresh your Eclipse project files by pressing F5.
2. Add the JAR file to the build path:
   1. Right click on the project in Eclipse and go to “Build Path” -> “Configure Build Path…”.
   2. Click the “Add JARs…” button on the right.
   3. In the tree list, expand the assignment 6 project, then under “lib” select “mariadb-java-client-2.4.1.jar”.
   4. Click “OK” and then “Apply and Close”.
3. Add the “MariaDBDriver.java” and “DatabaseDriver.java” files:
   1. Open the project in Windows Explorer (right click on the project in Eclipse and go to “Show In” -> “System Explorer”).
   2. Inside the “src” folder, create a new directory called “drivers”.
   3. Copy the “MariaDBDriver.java” and “DatabaseDriver.java” into the created “drivers” directory.
   4. Back in Eclipse, refresh the file system by pressing F5.

## Configure the MariaDB Driver

If you followed the previous steps exactly then nothing should be needing to be changed.

1. Open the “MariaDBDriver.java” file in your Eclipse.
2. Modify the constants so they include the right information.
   1. The value of SERVER should be set to the IP address of the MariaDB installation. In your case, it should “localhost”.
   2. The PORT value should remain unchanged from “3306”.
   3. The DATABASE needs to be set to the name of the database you created on page 5. It should be “cprg251”. Do not use “mysql” or “information\_schema’ as the database name!
   4. The USERNAME and PASSWORD values are the login credentials created on page 6. They should be “cprg251” and “password”. Do not use “root” as the username!

## Testing

You can test the MariaDB driver is working by creating a file called “ConnectTest.java” with the following code.

package application;

import java.sql.\*;

import drivers.\*;

/\*\*

\* Class description: Simply connects and disconnects from the database.

\*

\* @author Joe Blow (123456)

\*

\*/

public class ConnectTest {

/\*\*

\* Tests connecting to the database.

\* @param args

\*/

public static void main(String[] args) {

DatabaseDriver driver = new MariaDBDriver();

try {

driver.connect();

System.out.println("Connected!");

ResultSet rs = driver.get("SELECT 'Hello from the other side!'");

boolean hasRow = rs.next();

if (hasRow) {

String column1 = rs.getString(1);

System.out.println(column1);

} else {

System.out.println("Ooops! No rows were found.");

}

driver.disconnect();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

When you run it, you should get the following output:

Connected! Hello from the other side!

## Notes

* MariaDB is built off of and is backwards compatible with MySQL. There are some differences in things like performance and storage.
* Accounts in MariaDB and MySQL consist of both a username and hostname. This means users are restricted on where they can connect to the database from. In our case, we used “localhost” as the hostname so the database can only be accessed by the same computer and cannot be accessed remotely.
* There are different storage engines available in MariaDB. Storage engines allow the data in the database to be stored on the hard drive in a way that it can be accessed faster depending on what type of data is being stored. It is recommended that you use the “InnoDB” storage engine.

*Written by: Nick Hamnett*

*Last Revised: April 6, 2019*