# Exercises: Introduction to Databases

This document defines the **exercise assignments** for the ["Databases Basics - MySQL" course @ Software University.](https://softuni.bg/trainings/1443/databases-basics-mysql-september-2016)

## Download and Install MySQL Community Server

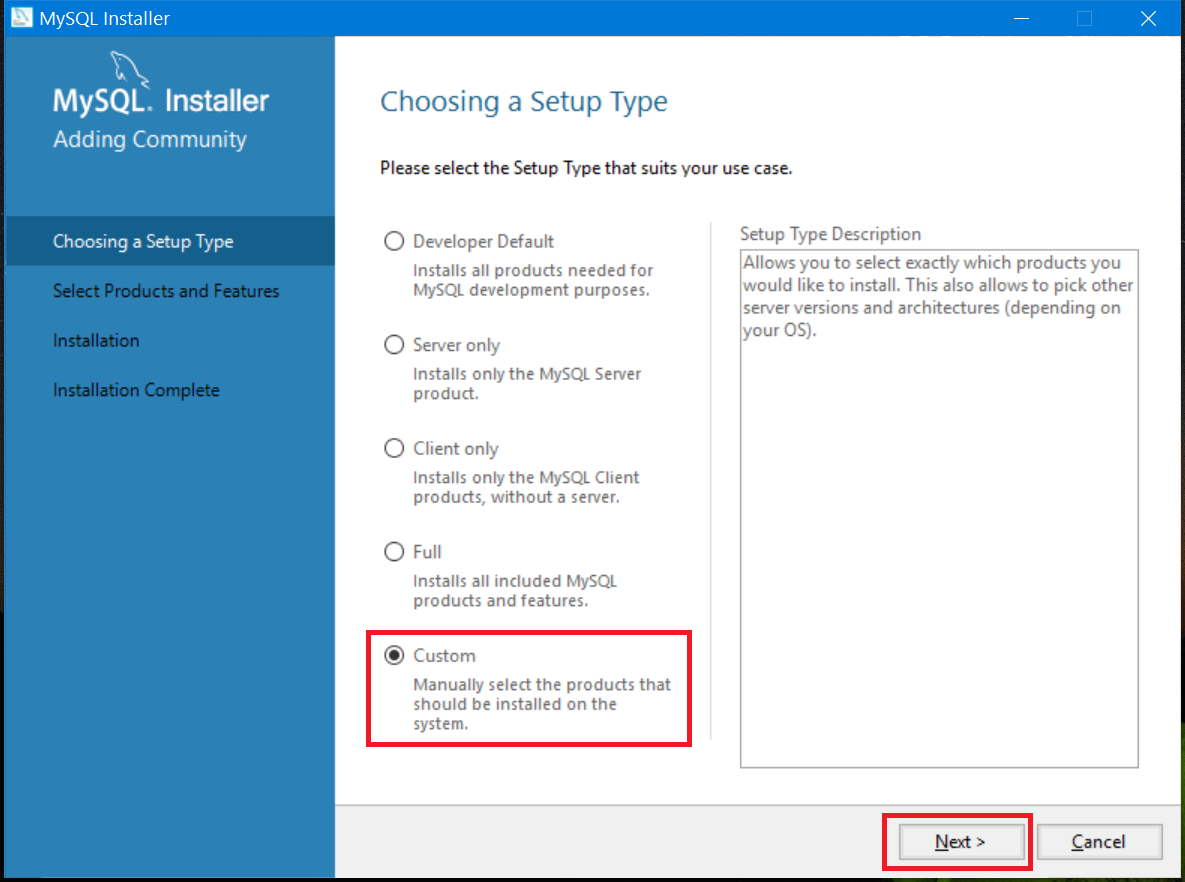
Download and install MySQL Community Server. Do not install unneeded features of MySQL Community Server.

### Task 1. Download MySQL Community Server

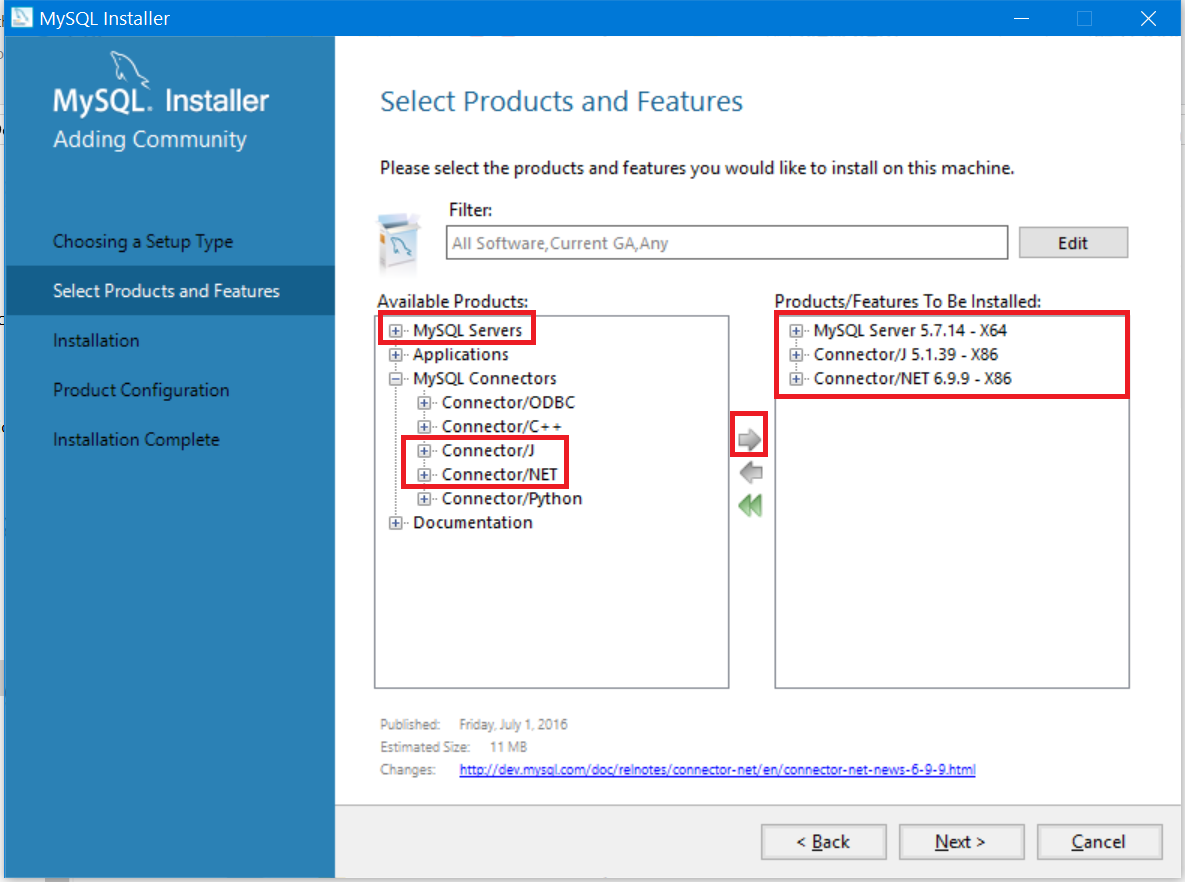
Go to the MySQL web site and download MySQL Community Server <http://dev.mysql.com/downloads/mysql/>

### Task 2. Install the MySQL Community Server

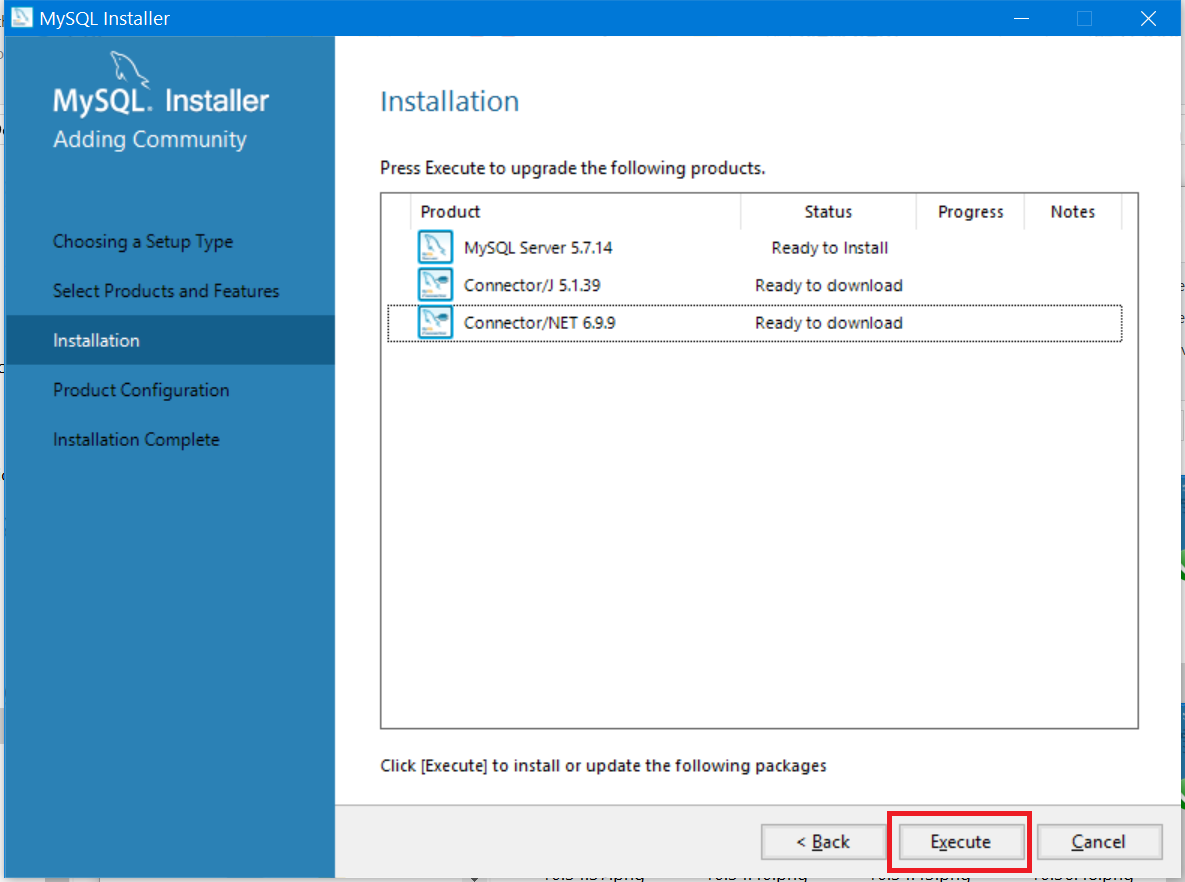
1. Open installation file and choose the **Setup type** to be **Custom**. Then click **Next.**



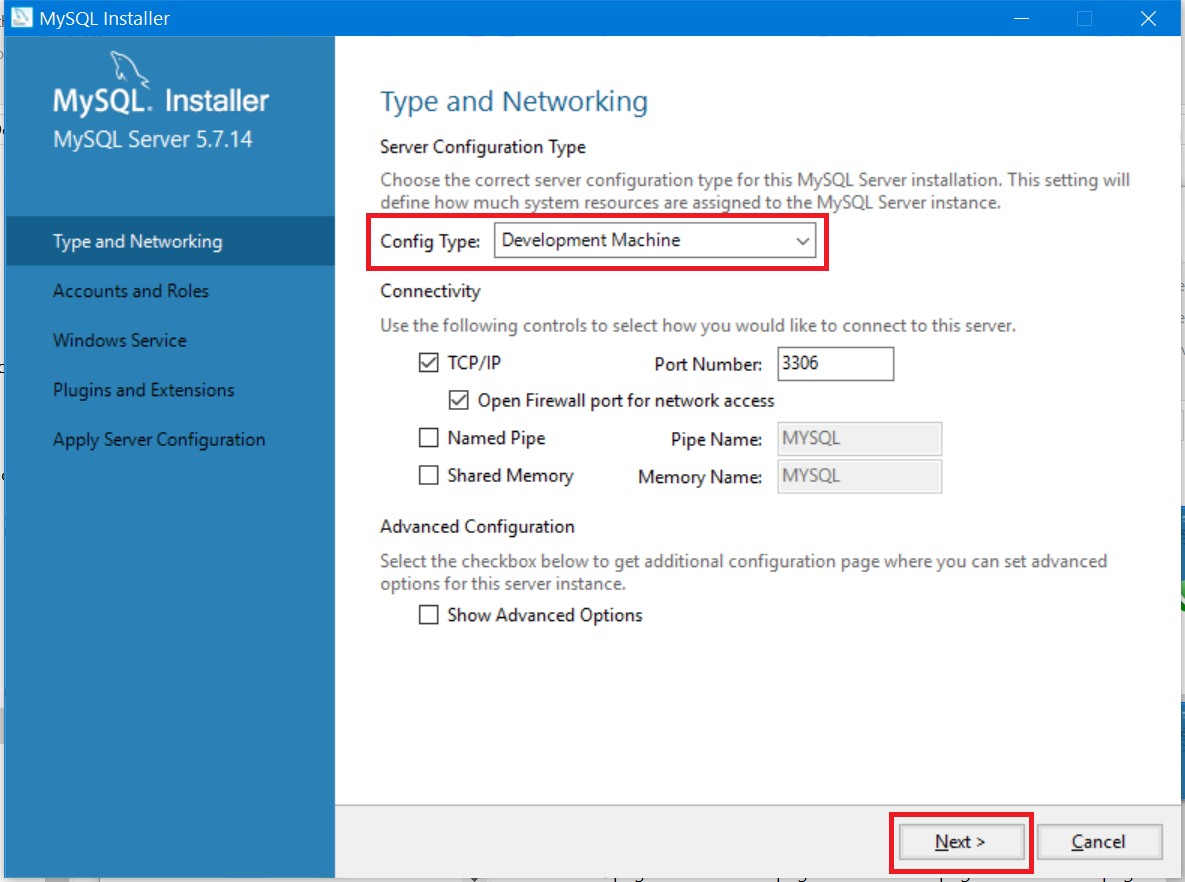
1. All the features we need are **MySQL Server**, **Connector/J** and **Connector/NET**. All other features are optional and won’t be needed for that course.



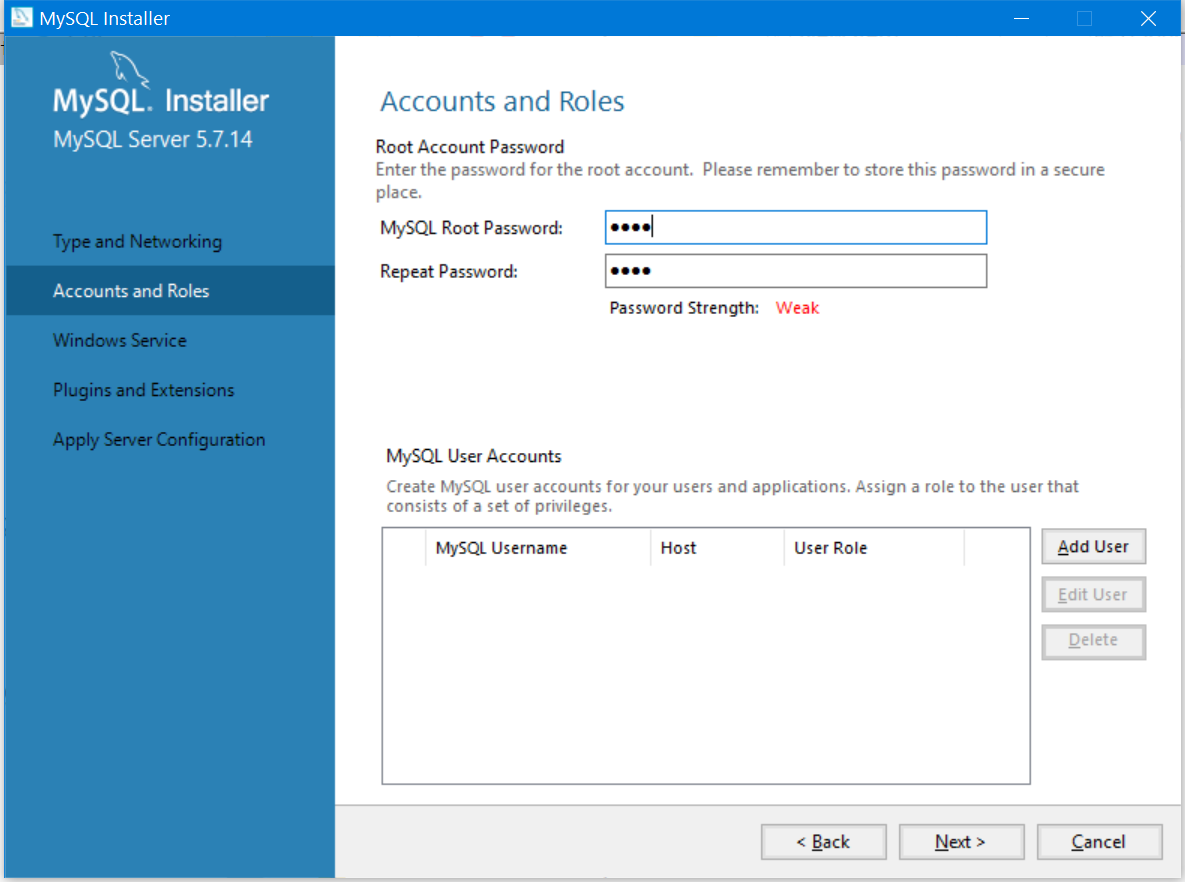
1. Just click **Execute** and the setup will install these features



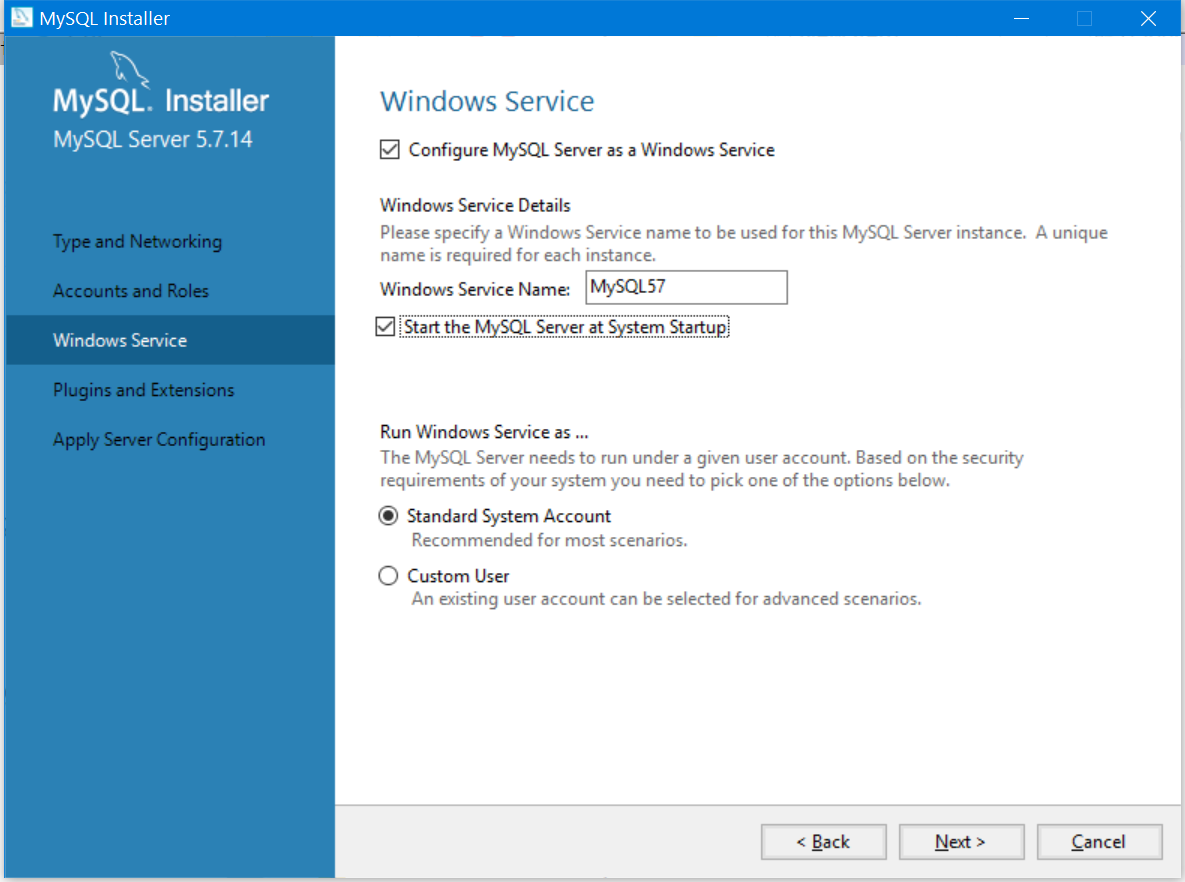
1. Choose the configuration type to be a **Development Machine** and click on Next.



1. Set password to the Root account.



1. Here you can set the **MySQL Server** to run as **Windows Service** and to start automatically at Windows start up. This is the recommended way. Otherwise you must start up MySQL every time before working with the database.



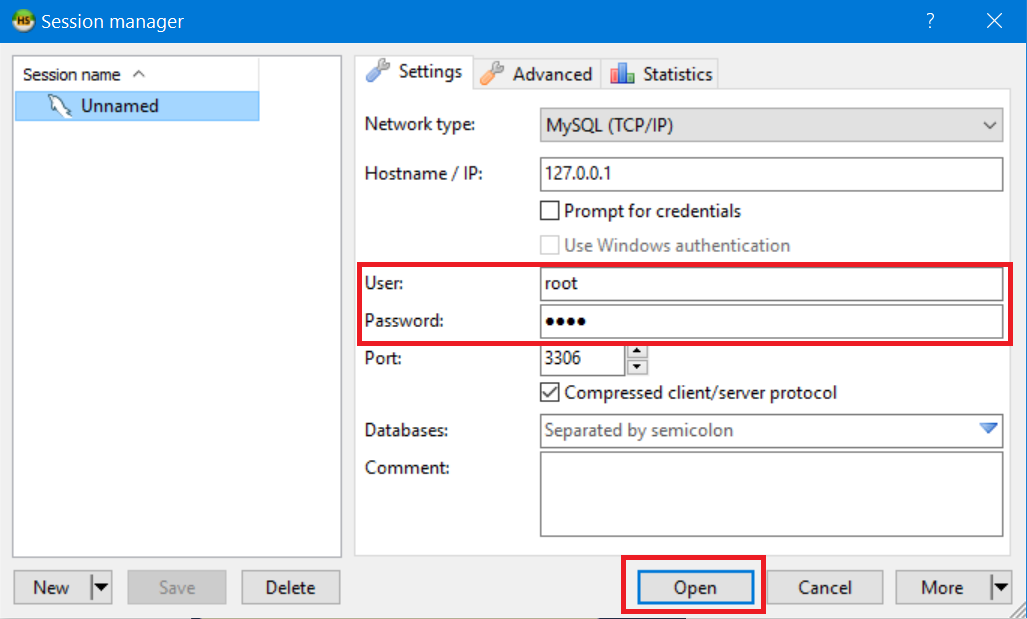
1. Click **Next >** to start the installation and wait until it finishes installing

## Download and Install HeidiSQL

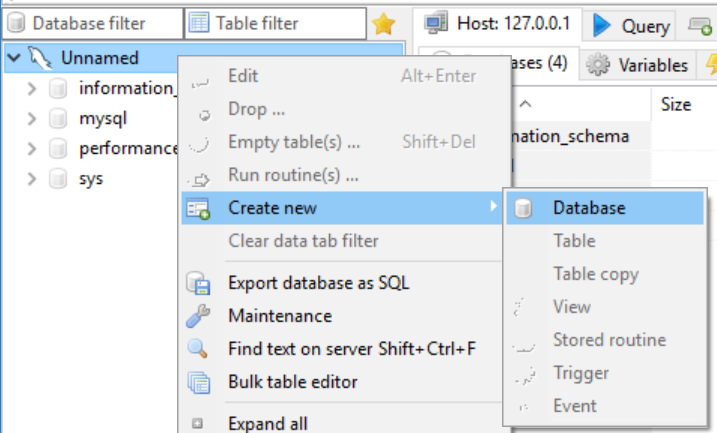
Go to the HeidiSQL web site and **download HeidiSQL** <http://www.heidisql.com/download.php>. Then **install** it.

## Create New Database

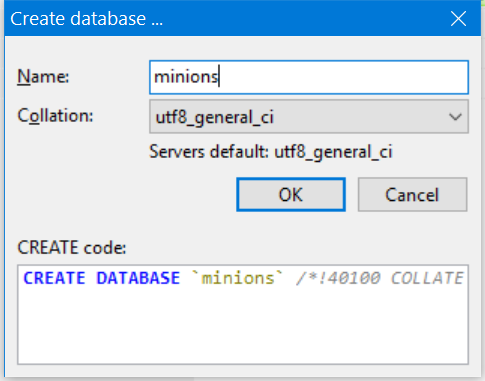
* Connect to MySQL Server with user **root** and the password you set up earlier.



* Create new database

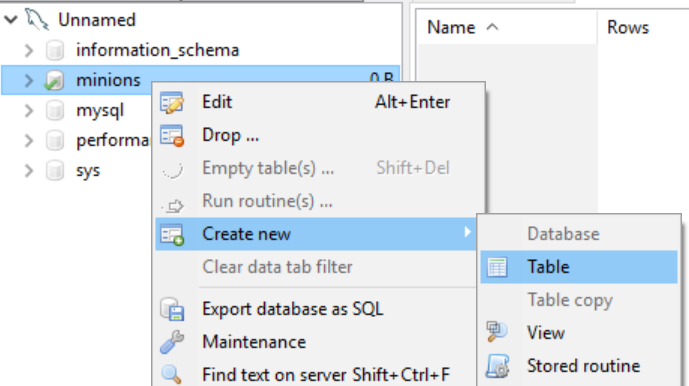


* Type the name of the database and click OK. This will create your database.

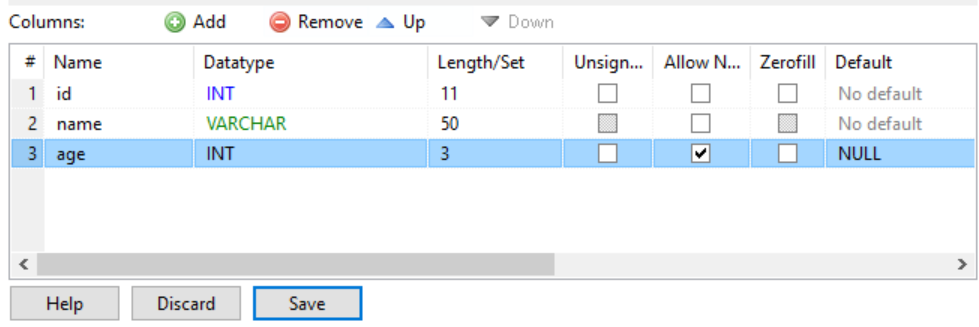


## Create Table

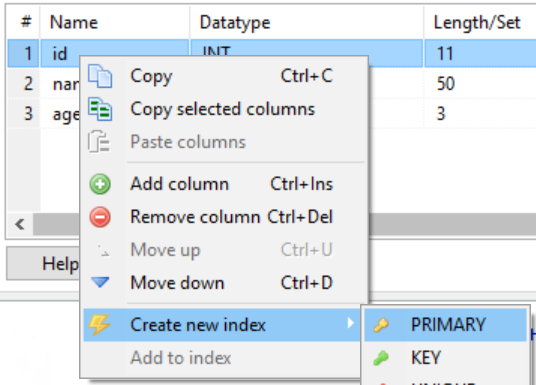
* Create **table** **Minions**



* Create columns **id, name, age**. Id and name are **required**; Age should **allow null values**.

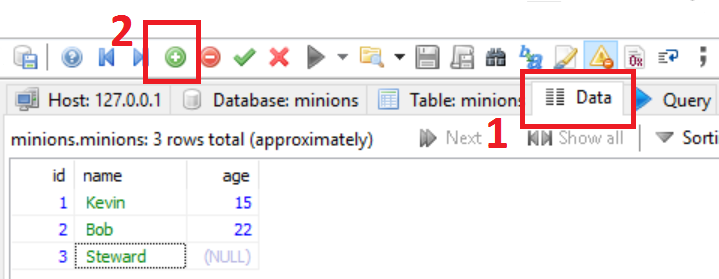


* Set the **id** as **primary key**.



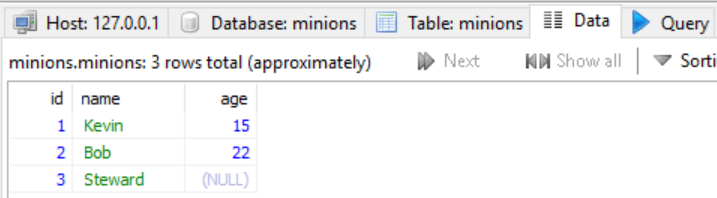
## Insert Data in the Table

Insert data in the table as it’s shown on the picture

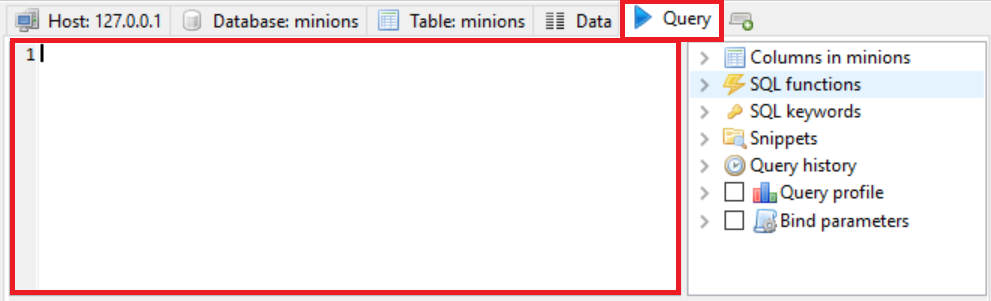


## Select Data from Table

* Select **all columns** from the Minions table.



* Open Query tab, then write the SQL.
  + \* Select **only names** from Minions table.
  + \*\* **Order** them **ascending by name**



## Update One Record

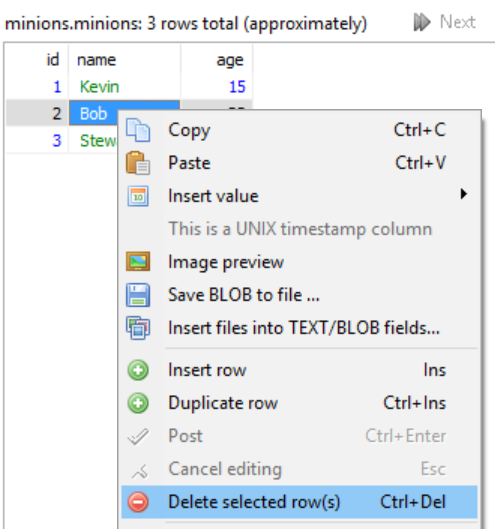
Change **Stuart’s age** from **NULL** to **10**

## Update All Records

Change all of the Minions age to be + 1 years.

## Delete Record

Open Data tab for the table, **right click** on the row where **Bob** is situated and delete it.



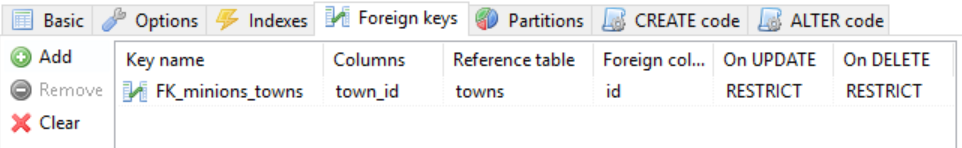
## Create New Table

Create new table **towns**. Every town has **id (int)** and **name (text)**. Make the **id** column **primary key**.

## \*\*Connect Tables

Now let’s make a connection (or relationship) between our two tables. First we need modify our minions table. Add column **town\_id** in it **(IMPORTANT: The type of the column must be the same as the type of the column id of the towns table)**.

### Hint



## Create New Database

Now on your own create a new database **school**. Add a few tables to the database: **students (id, name, age, phone\_number)**, **classes (id, name, max\_students), teachers(id, name, class)**. Add columns for the tables. Populate the tables with random content. Then delete and make changes in some records.

## \*Generate SQL Script

Generate SQL script from the **school** database. View the script file and try to understand different commands. Execute the script.