CST2355 – Database Systems Lab Assignment 6

Student Name: Yanzhang Wu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID: 41056465\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

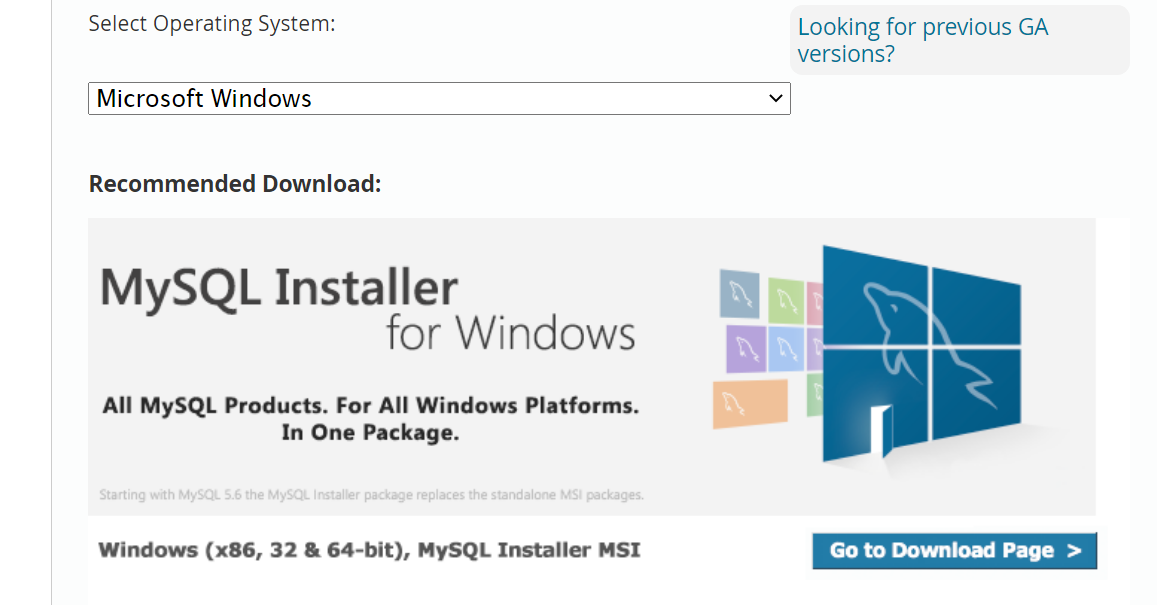
Student email: wu000298@algonquinlive.com\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Hand-in:

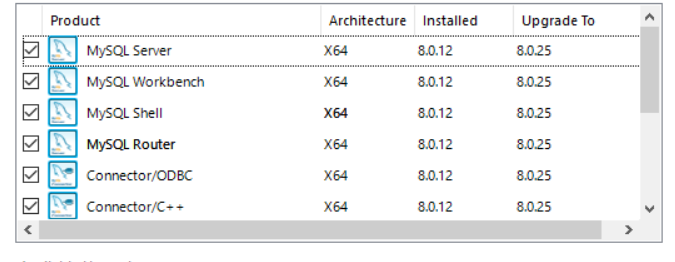
1. The lab assignment will be graded out of a maximum 4 points.
2. This template should be used to submit your lab assignment.
3. Make sure you have enough screenshots to completely document that you have completed all the steps.

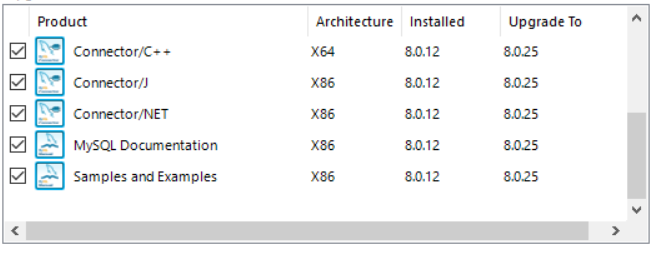
# Activities (Steps):

1. In this lab you will be using the MySQL (Version 8.0.25 or later) database management system, and MySQL Workbench (Version 18). The first step is to get the software installed.
   1. Navigate to: <https://dev.mysql.com/downloads/mysql/>

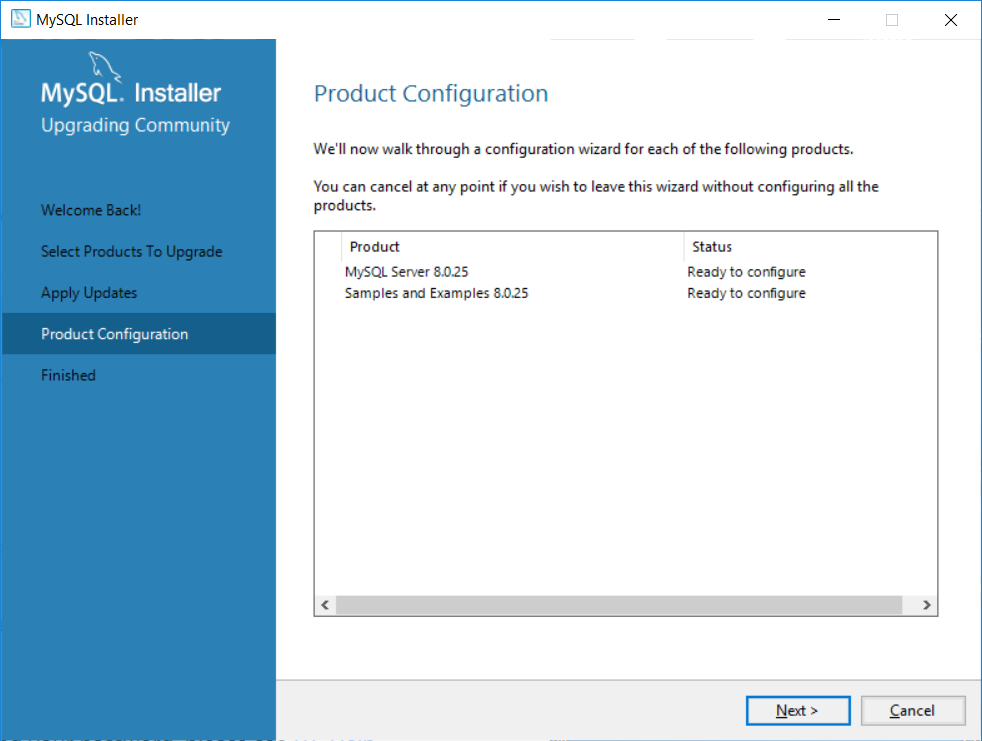


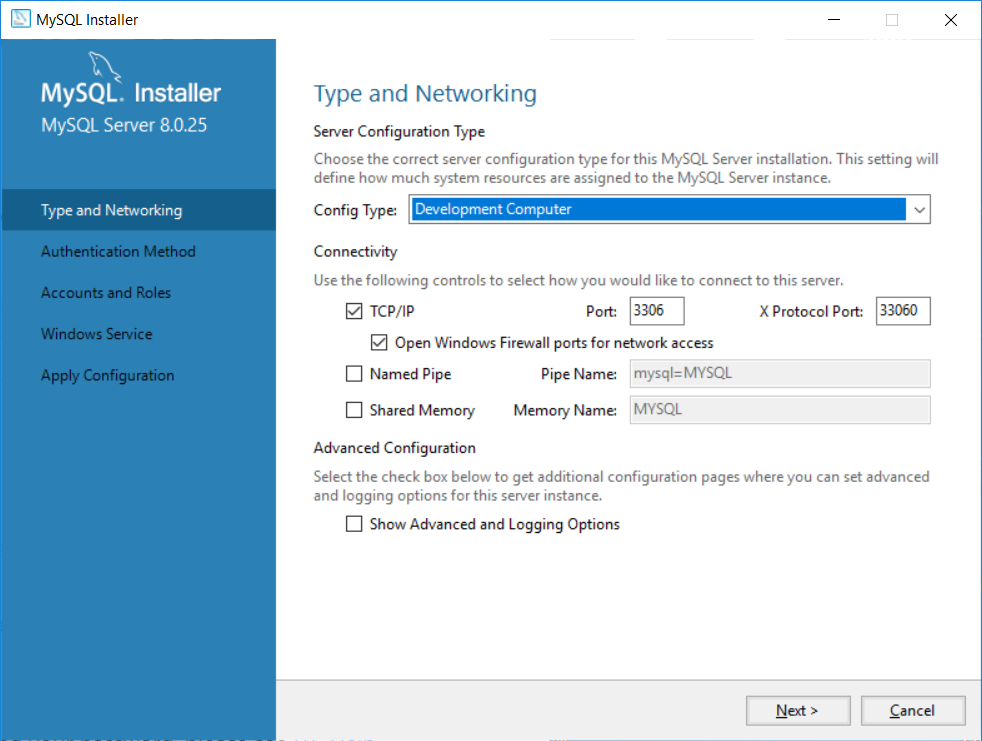
* 1. Navigate to the Download Page and click on Download for the larger installer (435.7MB). You will likely have to create an ORACLE account. (You should do so – you can use the account again to download ORACLE database server software later.) Download the installer and when downloaded, run it.
     1. When prompted, select ALL the products you would like to install/update.
        1. You should select everything.

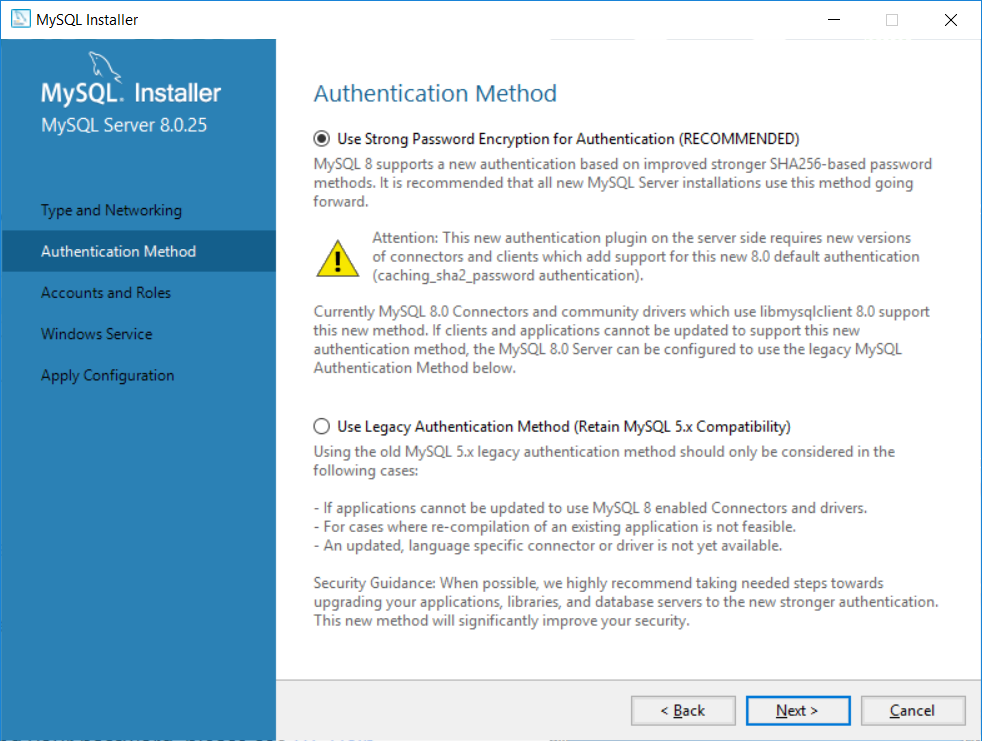


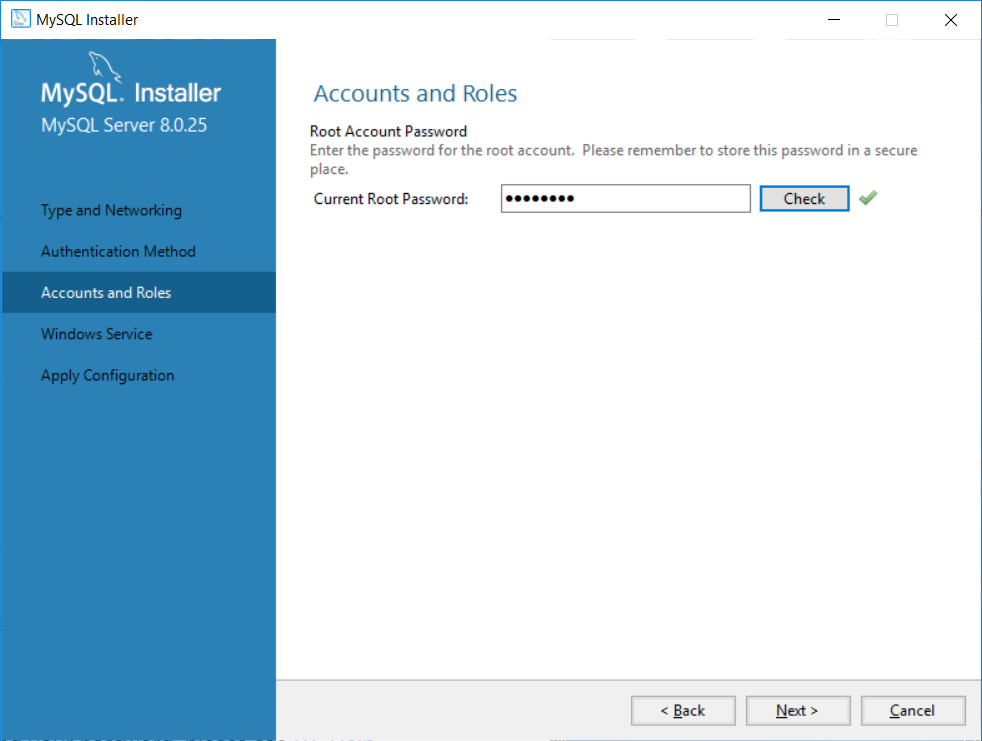


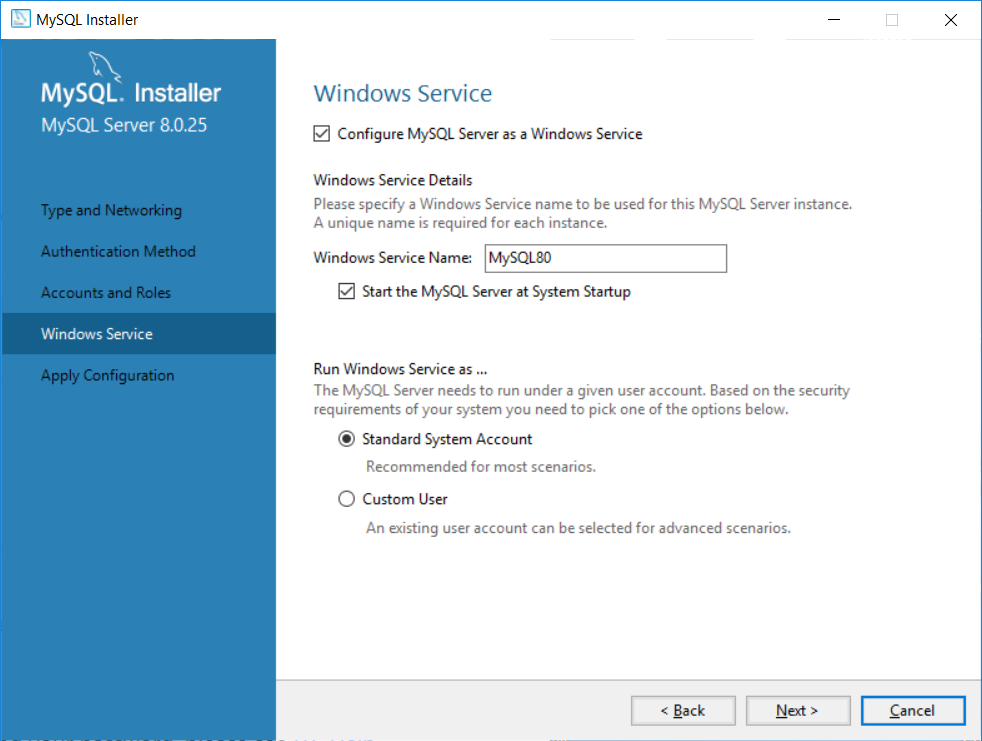
* 1. You will be prompted for some configuration parameters. The most important is the ‘Root Account Password’. **YOUR INSTALL SEQUENCE MAY BE DIFFERENT, depending upon the software already installed on your computer.** Here is my update/install sequence:



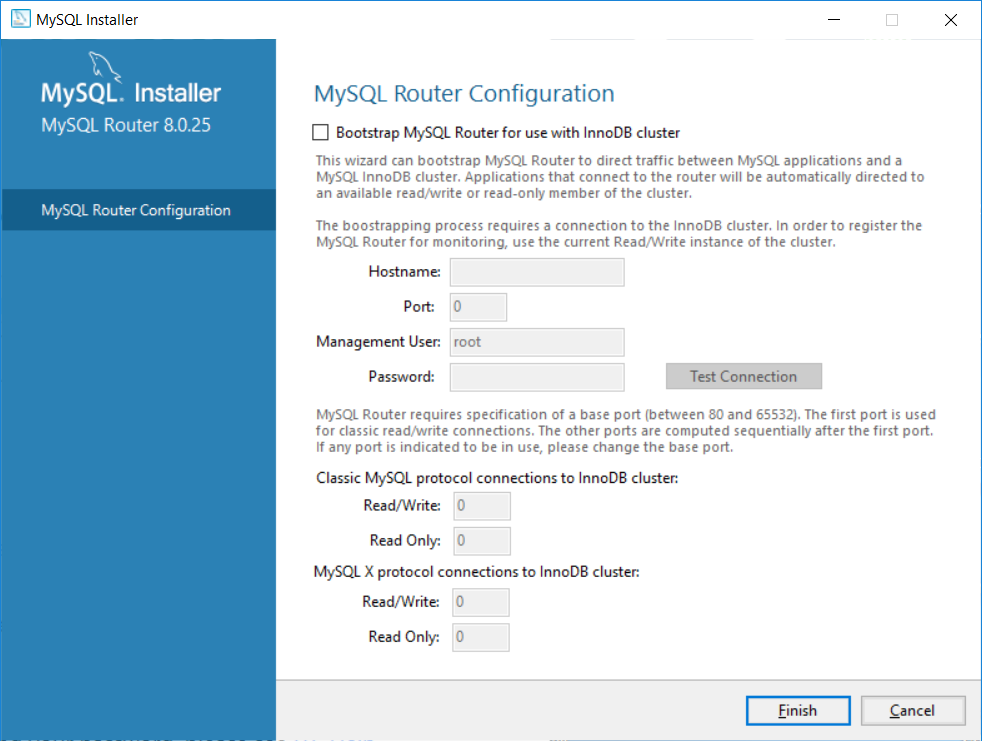




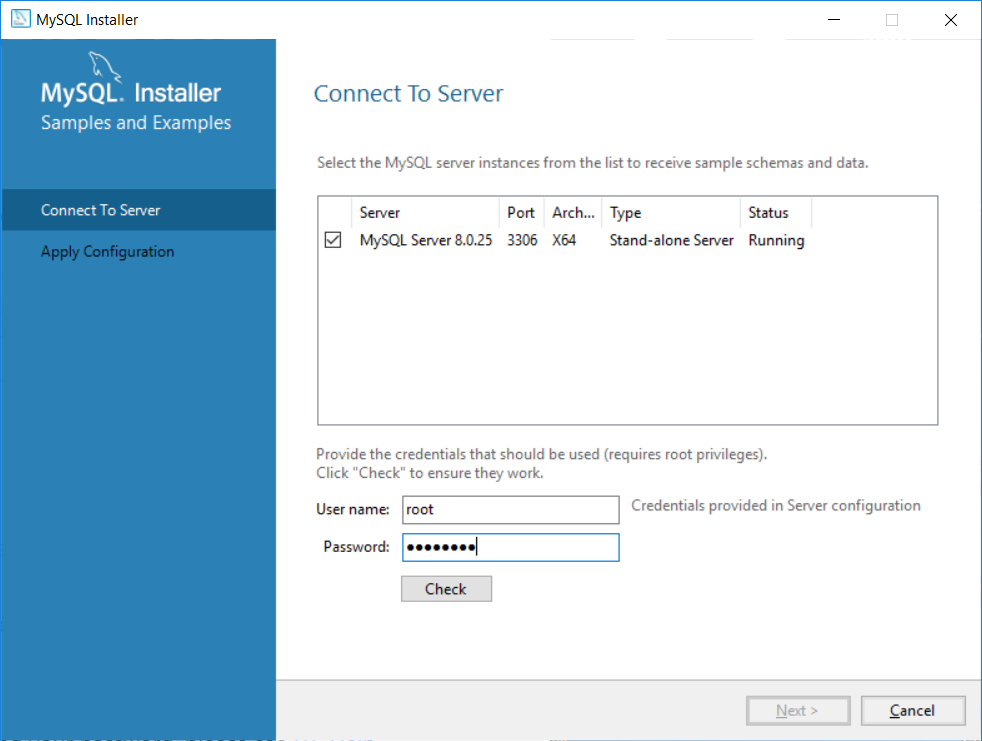




When / if prompted to configure the router:



* + 1. When (re) configuring the Examples:

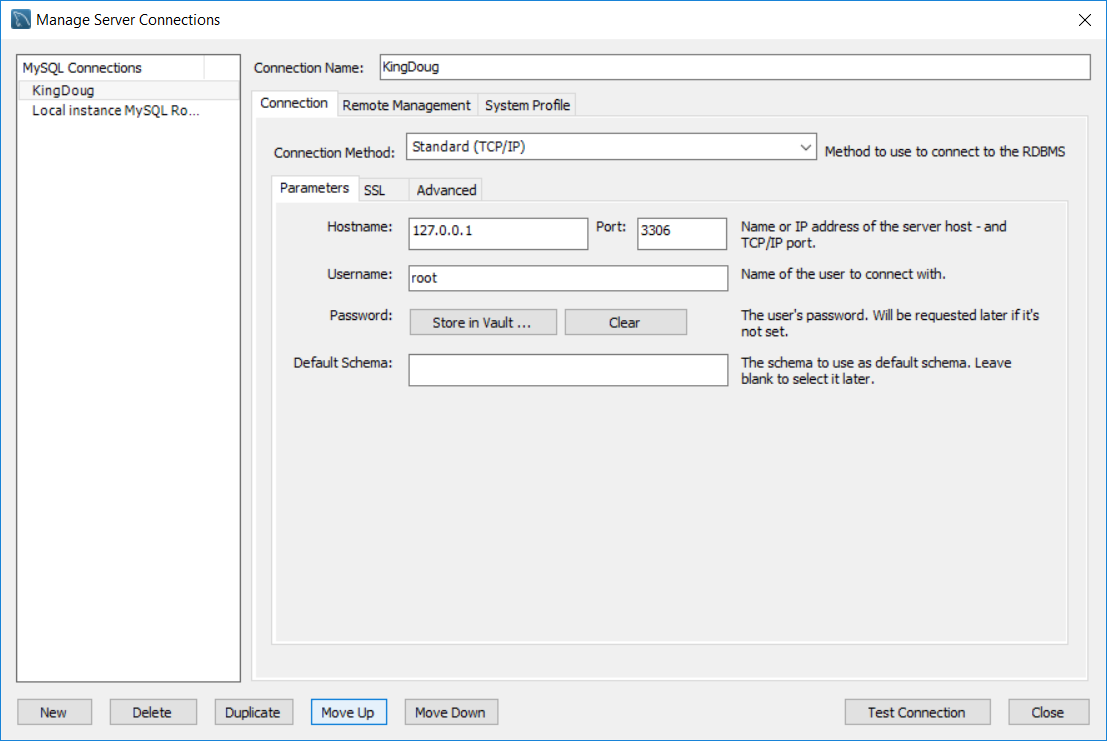


* + 1. When everything has been configured and/or reconfigured, you can exit the MySQL Installer.

1. Open MySQL Workbench 8.0 from the Windows Start Menu. You should see something like this:



1. Connect to your server by Clicking on the ‘Local instance MySQL Router’ connection.
2. Then select Database > Manage Connections from the MySQL Workbench menu.
   1. Create a “New” connection called ‘called ‘YourlastnameYourfirstname’ (e.g., ‘KingDoug’) using the localhost default configuration, and then click “Move Up” to move it to the top of the list.

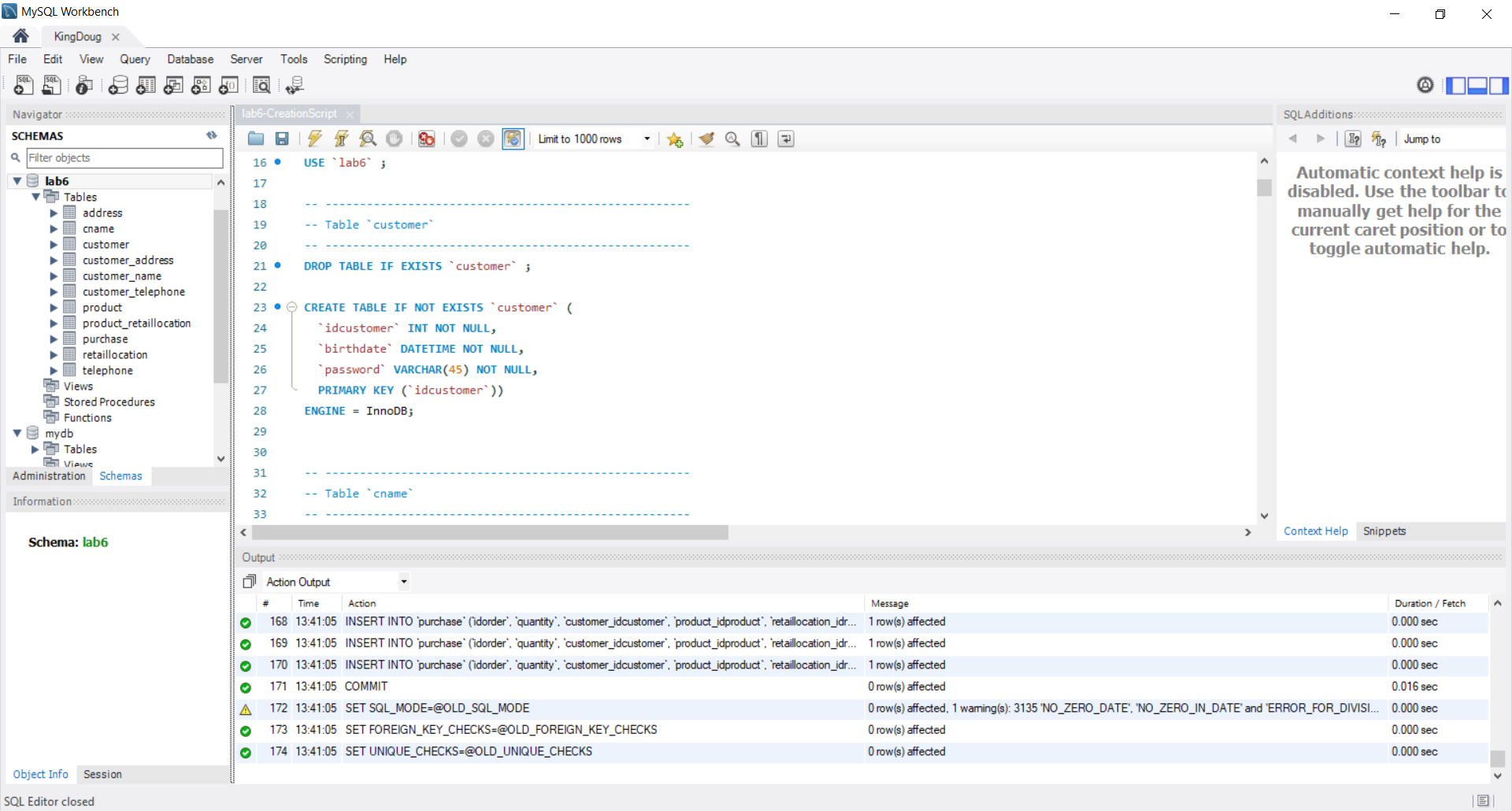


* + 1. Test the connection and paste your confirmation screen shot below:

Graphical user interface, application

Description automatically generated

* 1. Exit MySQL Workbench and restart it. You should see your new connection. Select your new connection – **this will be your default connection for the rest of this lab**.
  2. Once connected,
     1. Click on the “Schemas” tab (not the Administration tab…) and then click on the Folder icon to open the lab6-CreationScript.sql file.
     2. Once opened, execute the script. It will create a new database schema called ‘lab6’.
     3. Use the navigator to display the list of tables in the lab6 schema. Take a screenshot of the entire MySQL Workbench window that clearly shows your connection, the SQL script, and your connection name. Here is mine:

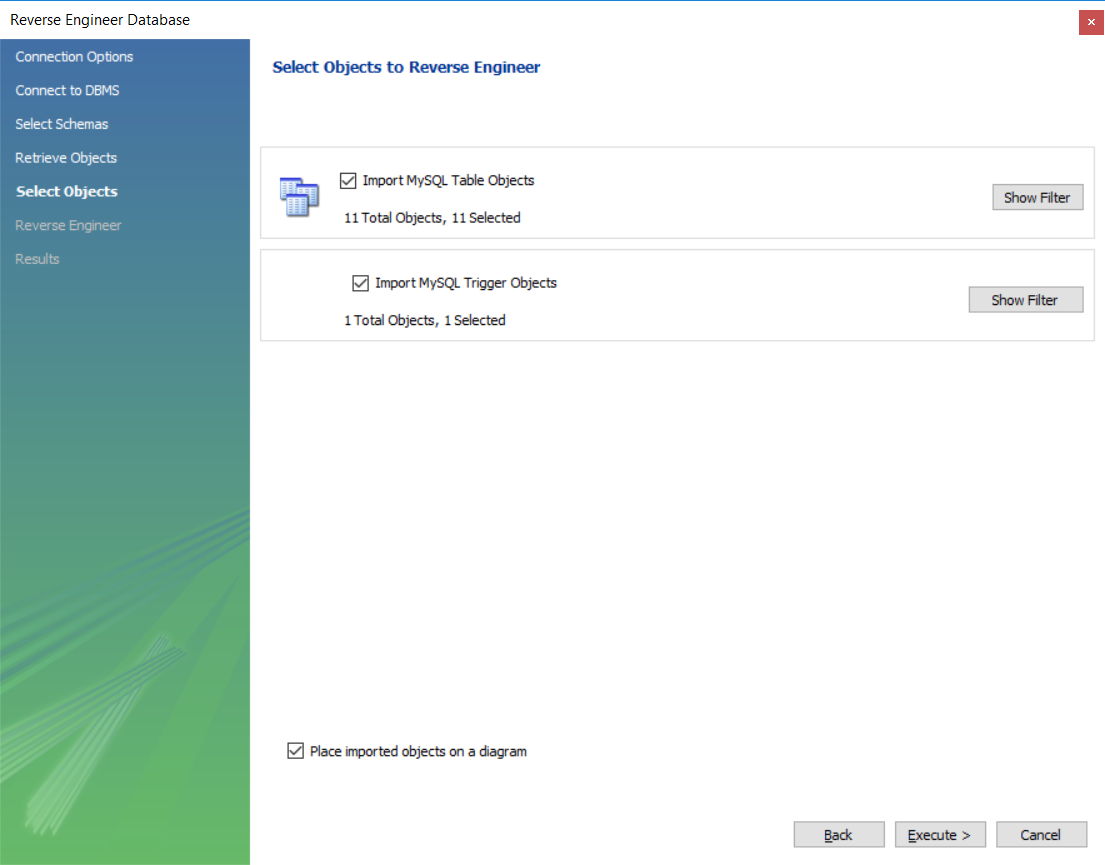


* + 1. Place your screenshot below:

Graphical user interface, text, application, table

Description automatically generated

1. You will now reverse-engineer the data model diagram from the lab6 physical schema. On the MySQL Workbench menu, select Database > Reverse Engineer.
   1. When prompted, make sure that only the “lab6’ schema is selected.
   2. You should see a screen like:

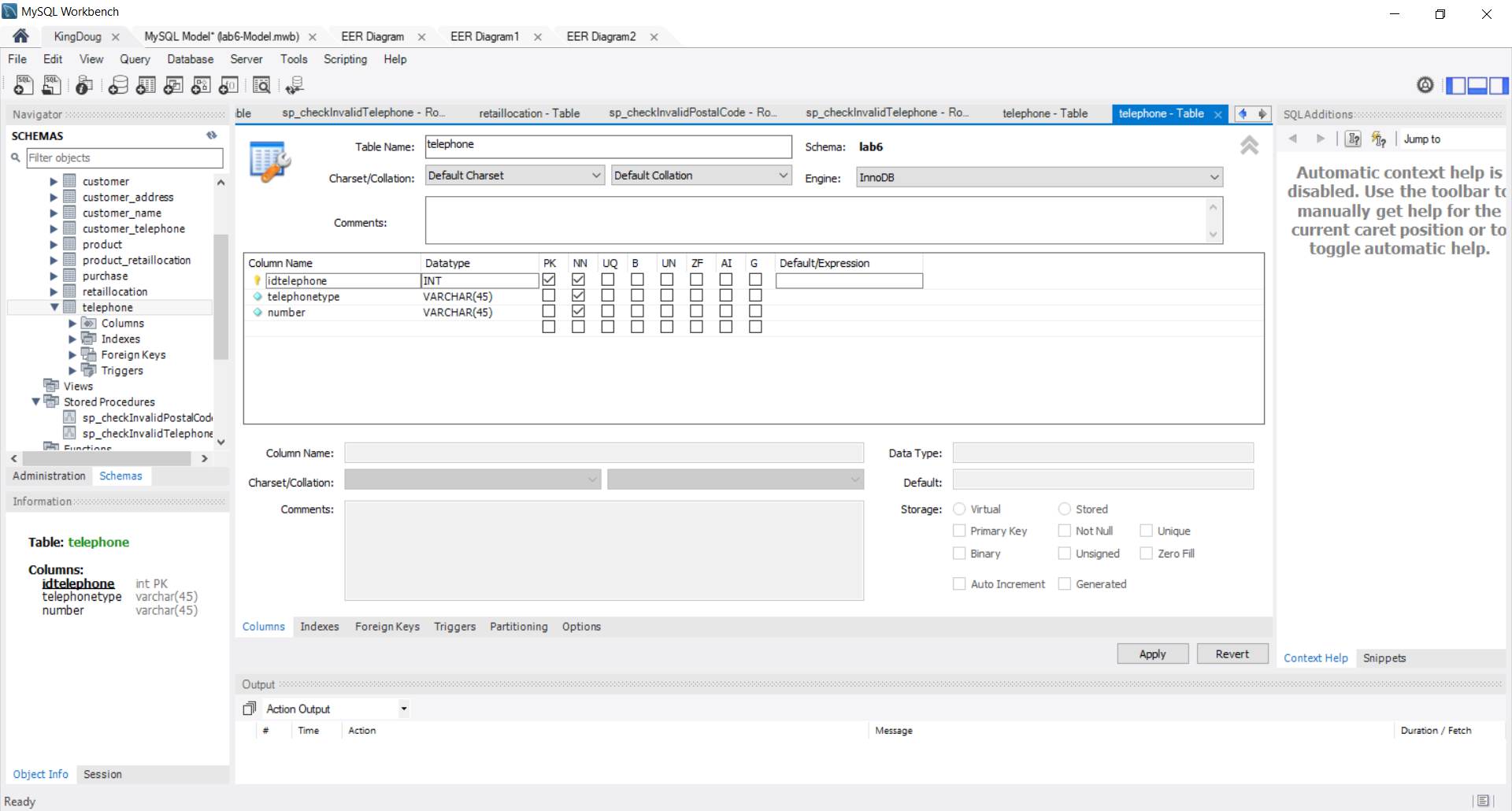


* + 1. Then click “Execute” and the model will be created!
    2. Spend some time to fix the layout by moving the tables in the ER diagram/model so that there are no crossing lines, and re-size the tables so that you can see all of the field names in the diagram. Once you have fixed the layout, save the model/diagram in a file (by clicking on the Save icon) as “lab6-Model-Initial.mwb”. Paste a copy of the diagram below:

Graphical user interface

Description automatically generated with medium confidence

1. **Return to you default connection to lab6.** Use the “lab6-StoredProcs.sql” file to create the two stored procedures to be used for data validation. Take a look at the file, but it likely does not need any editing.
2. Select the telephone table, and from the right-mouse click menu select “Alter table”. You should see something like this:



* 1. Notice the set of tabs at the bottom of the screen: “Columns”, “Indexes”, etc. Select the Triggers tab:
     1. Create the following “Before Insert” trigger.

CREATE DEFINER = CURRENT\_USER TRIGGER `lab6`.`telephone\_BEFORE\_INSERT` BEFORE INSERT ON `telephone` FOR EACH ROW

BEGIN

CALL sp\_checkInvalidTelephone (NEW.`number`);

END

* + 1. And use a CALL to the same stored procedure to create the “Before Update” trigger.
    2. Provide screenshot(s) showing your two new triggers. (Before Update & Before Insert):

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* 1. **Use the diagram** to navigate to the two tables containing postal codes and change the size of the postal code fields to be 6 (rather than 45). Also create “Before Insert” and “Before Update” triggers to call the stored procedure that checks that the format matches the regular expression pattern [a-Z][0-9][a-Z][0-9][a-Z][0-9] (This is very similar to what you did for the telephone number validation.)
     1. Provide screen shots showing your modified tables with the triggers:
        1. address table:

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

* + - 1. retaillocation table:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

* 1. **Save your work.**

1. Now we will forward engineer the changes from our model to our lab6 database.
   1. Select Database > Forward Engineer from the MySQL menu.
   2. Keep the defaults, so that the tables, data, and triggers all get pushed to the lab6 schema.
   3. After the forward engineering is successful.
      1. Open your default lab6 connection and select the telephone table.
      2. When you have selected the table, a small table icon appears in the navigator to allow you to view/edit the table contents. If you click on the table icon, a pane appears with the results of a “select \* from *tablename;”* query.
         1. Try an edit to one of the rows - fix one of the pre-existing telephone numbers to match the new format. When you click on “Apply” the planned SQL “UPDATE” statement pops-up, and you can confirm.
            1. Try both correct and incorrect formats. Provide a screenshot of the result of an incorrect format.

Graphical user interface, application

Description automatically generated

* + - * 1. Fix the format of the pre-existing data, and try to insert a new row.

Provide a screenshot of your updated table contents

Graphical user interface, text, application, email

Description automatically generated

* + - 1. Now navigate to the “address” table and verify that your triggers are working.
         1. Provide a screenshot of trying to update a row with an invalid postal code format.

Graphical user interface, application

Description automatically generated

1. Make sure you save your model and exit.
2. Once you have embedded all of your screenshots, submit the file in Brightspace and you’re done!