**Project Part 4**

At this point your report should have a title page, a proposal which summarizes your project, descriptions of your entities and attributes, and data files in either XML or CSV format. Your GitHub should have your ERD file and 5 data files containing 3 entries each.   
  
Next your project will exercise your use of SQL commands by requiring you to create databases, load your data files into the databases, and then perform a few exercises. Remember to take screen captures of the results to prove that your command did what it needed to do. A simple status message of ‘ok’ is NOT acceptable.

**Data Definition Language Scripts**

For this segment of the project you must **A)** write or generate SQL commands to create your database tables and then **B)** load your data files into your tables. When you are finished, your database tables should contain all of the information that you described in your XML or CSV data sources.

**Document** your work in your **report** by showing the script and screen captures of the results, and explain what the script does for each statement.

The screen captures must show the contents of the database matching your design and the contents of your data files.

Also, save your scripts (table creation and loading command) to a text file that ends with the .sql extension, and save your script file to GitHub. Please consult the rubric for full details about what is expected in the report.

**CHEAT/COMPLIANCE WARNING:**

When you take a screen capture, please include a unique part of your system. If your document contains screen captures that are copies of other students’ captures in terms of image and data size, I will issue a 0. Also, use SQL that is compatible with mariaDB. If I cannot test your work, I will give you a 0.

**Procedure**

**Part A)**

You can write your scripts using **either** of two possible methods:  
  
1) Use Vertabelo to generate a script of SQL commands that build the database and its table structures. The instructions are below. This can be easy, but be careful that Vertabelo doesn’t add commands that don’t work. You can easily delete them if you get an error.

OR

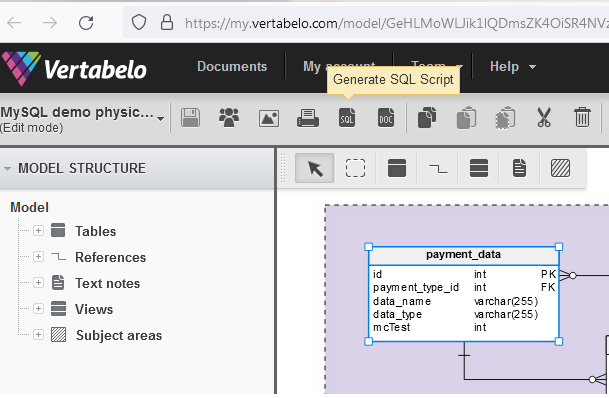
2) Write scripts manually to create the tables.

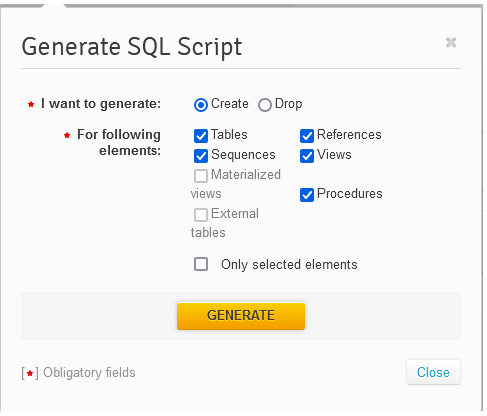
Whichever way you choose to build the tables, assign columns as a **Primary key** or **Foreign key** (where needed) so that if you modify a row in one table, the changes update in the other tables. Obviously, do not assign columns as keys if they are not shared across tables.

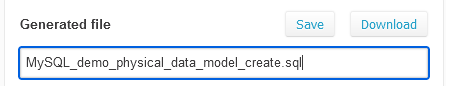
Whichever method you use to generate the databases, you still need to populate them with data. Copy your 5 data files to a directory convenient to the data table, then issue the appropriate ‘load’ command that loads either your XML or your CSV files. Be sure to include the path. If you use an absolute path, then you should have no trouble.

Here is the procedure for using Vertabelo:

a) Load your ERD file from the previous step of the project. If you don’t have one, then create the tables/database:



b) Using the SQL icon above the drawing area, choose to Generate SQL Script: 

c) Click on ‘Generate’ to export the SQL file:  


d) download, and copy the .sql file to your favorite directory where you administrate MariaDB. This is where you use the ‘SOURCE’ command to load and execute the sql file as a script of commands.

If you prefer using SQL to create the tables, then review the presentations or use mariadbtutorials.com for help.

**Part B) Load your data files**

Now that you have the tables created, you must complete this section by loading your data files with a ‘load’ command.

TO IMPORT DATA INTO MariaDB:

1. Use “LOAD DATA LOCAL INFILE ‘filename’ “ to load a delimited Excel file (usually a comma separated value, CSV).

<https://mariadb.com/kb/en/load-data-infile/>

You’ll notice the ‘set’ at the end gives you the option to manipulate data during the load. In fact, nearly all of those commands are optional since they are shown within square brackets, [ ]. Thus, this step is very simple! You may also redefine the delimiter character if you are not using a CSV.

2. If you are using XML files to store data, then see this example:

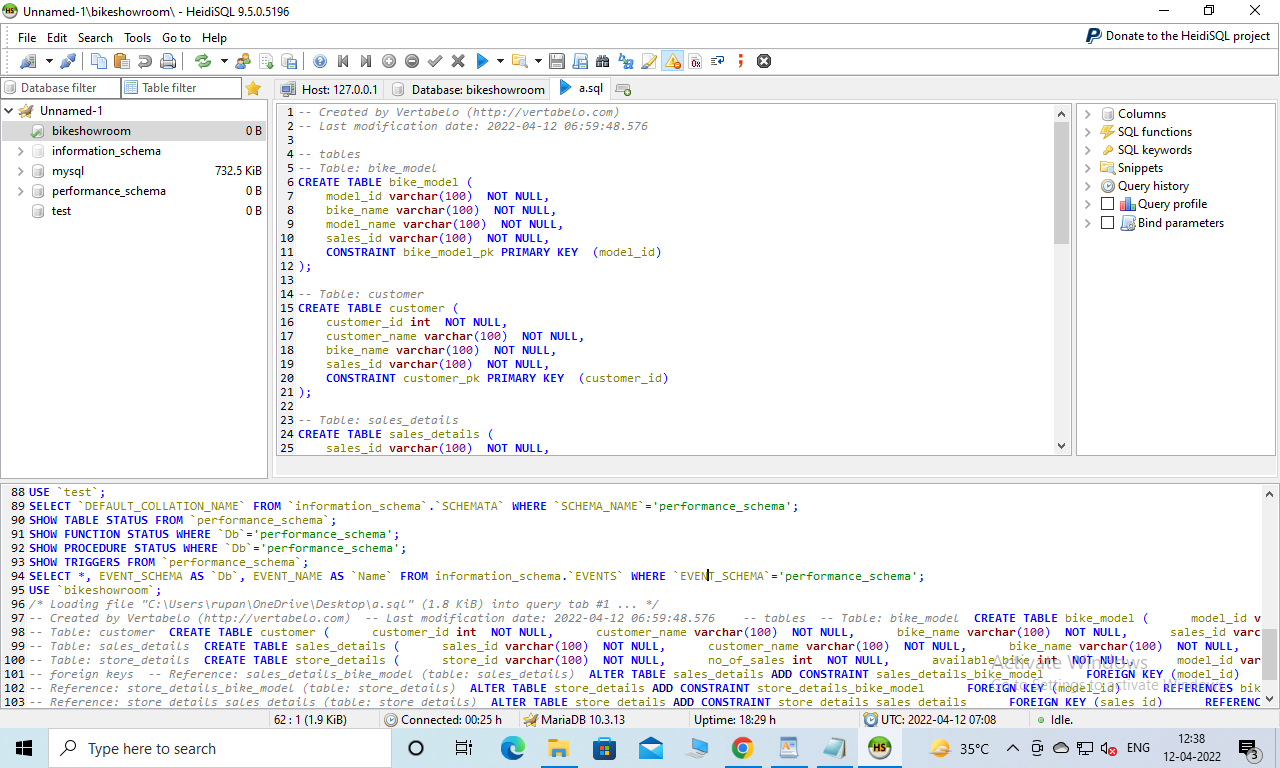
<https://mariadb.com/kb/en/load-xml/>

Again, a single command will load the contents of your file.   
  
If you get an error like the file is missing, then you need to either find the active directory and copy the data files there, or else include the absolute path to your files as part of the file name when you do the ‘load’.

Example:  
#> LOAD DATA LOCAL INFILE ‘/home/matt/cs509/products.CSV’

Rubric: Your work will be graded as follows:

* Database and table creation statements (manual or Vertebelo) saved as an sql script file (this is a simple text file with a .sql extention). Upload to GitHub. 8 points



* The SQL commands for populating the tables **and** what they do (explain each of the commands and options that you used in the script) documented in your report: 8 points

**Answer: In this project i choose a bike showroom database where i create 5 tables, for creating the table i use ‘create’ sql script using create statement i create 5 table : bike\_model,customer,sales\_details,store\_details**

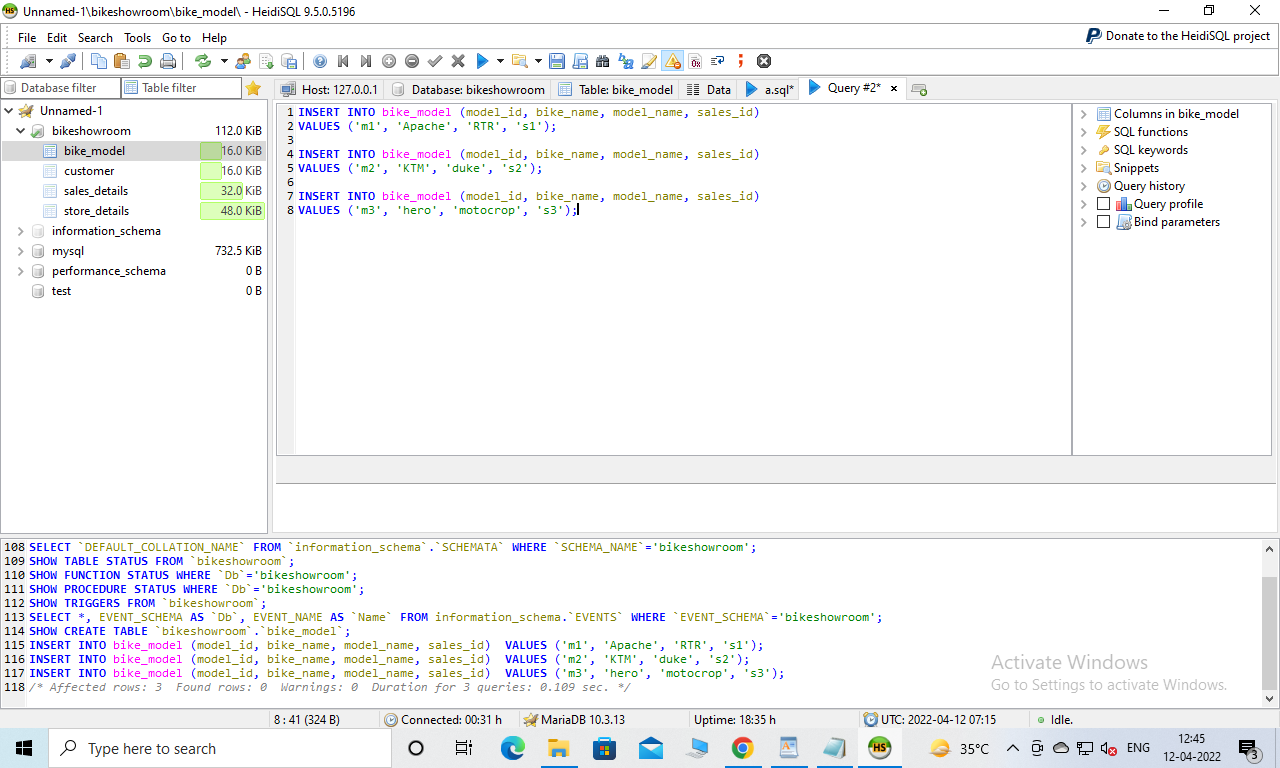
**in bike\_model table we store information like model\_id,bike\_name,model\_name,sales\_id**

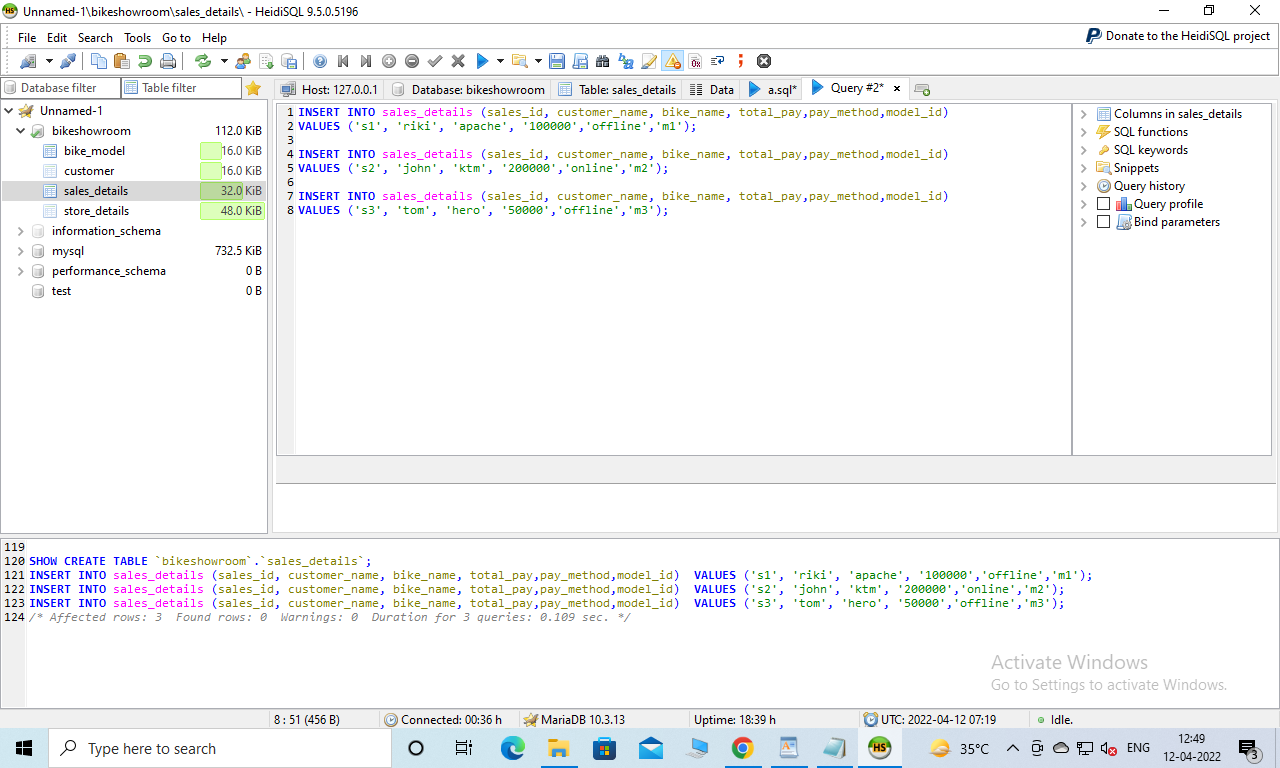
**in customer table we store information like customer\_id,customer\_name,bike\_name,sales\_id**

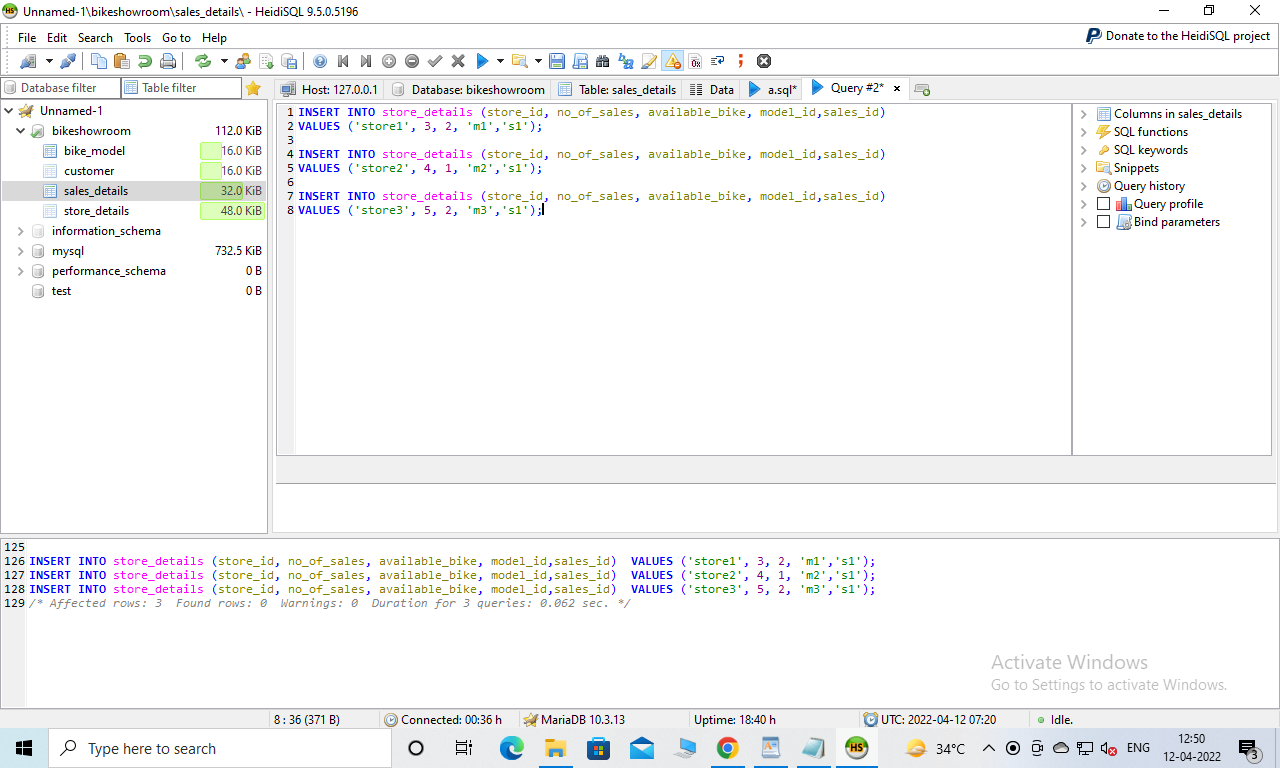
**in sales\_details table we store information like sales\_id,customer\_name,bike\_name,total\_pay,pay\_method,model\_id**

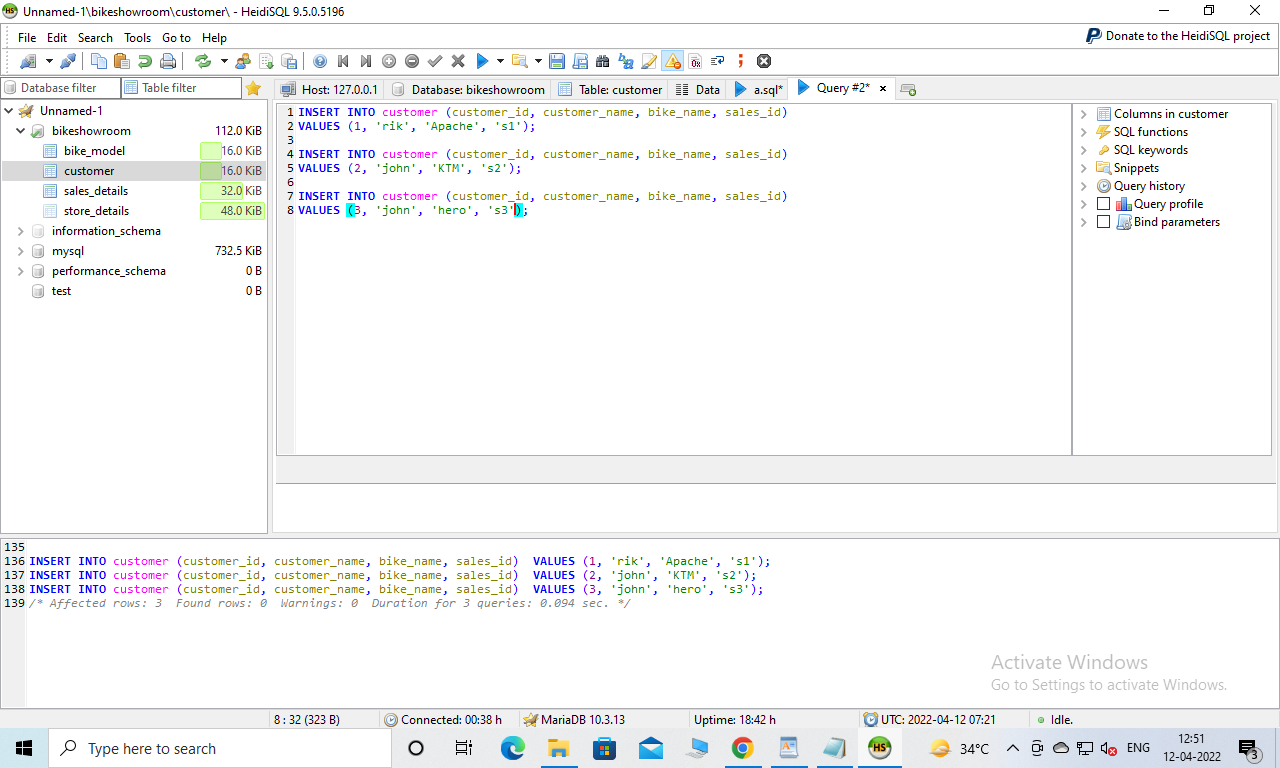
**in store\_details table we store information like store\_id,no of sales,available\_bike,model\_id,sales\_id.**

* Screenshots of your successful attempts to populate each table with at least three records from your data files: 4 points









Total points possible: 20

Don’t forget to do the exercises on the next page.

**Data Manipulation Language Scripts**

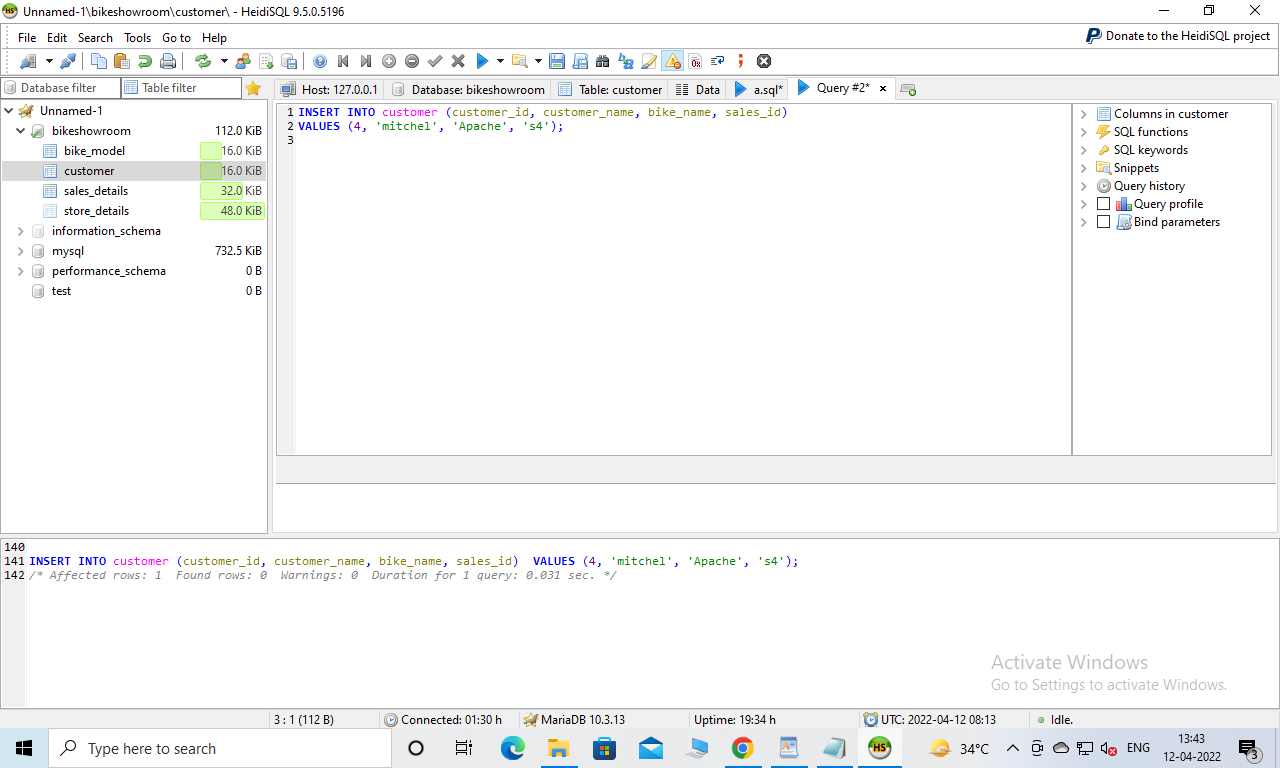
Write the SQL commands for twelve queries. Two queries should be insert statements, two should update statements, one should be a delete statement, one should be a simple select statement that selects a subset of the rows and columns from one table, two queries should be a select statements that select data from a joining of two tables, three queries should use summary functions to generate statistics about the data, and one query should be a multi-table query. Show the queries and screenshots of the results in your Word document **report**, and save your queries in a commented sql script to GitHub.

Rubric: Your work will be graded as follows:

* 1 point each for the two insert statements (2 pts total)

**Query1: INSERT INTO customer (customer\_id, customer\_name, bike\_name, sales\_id)**

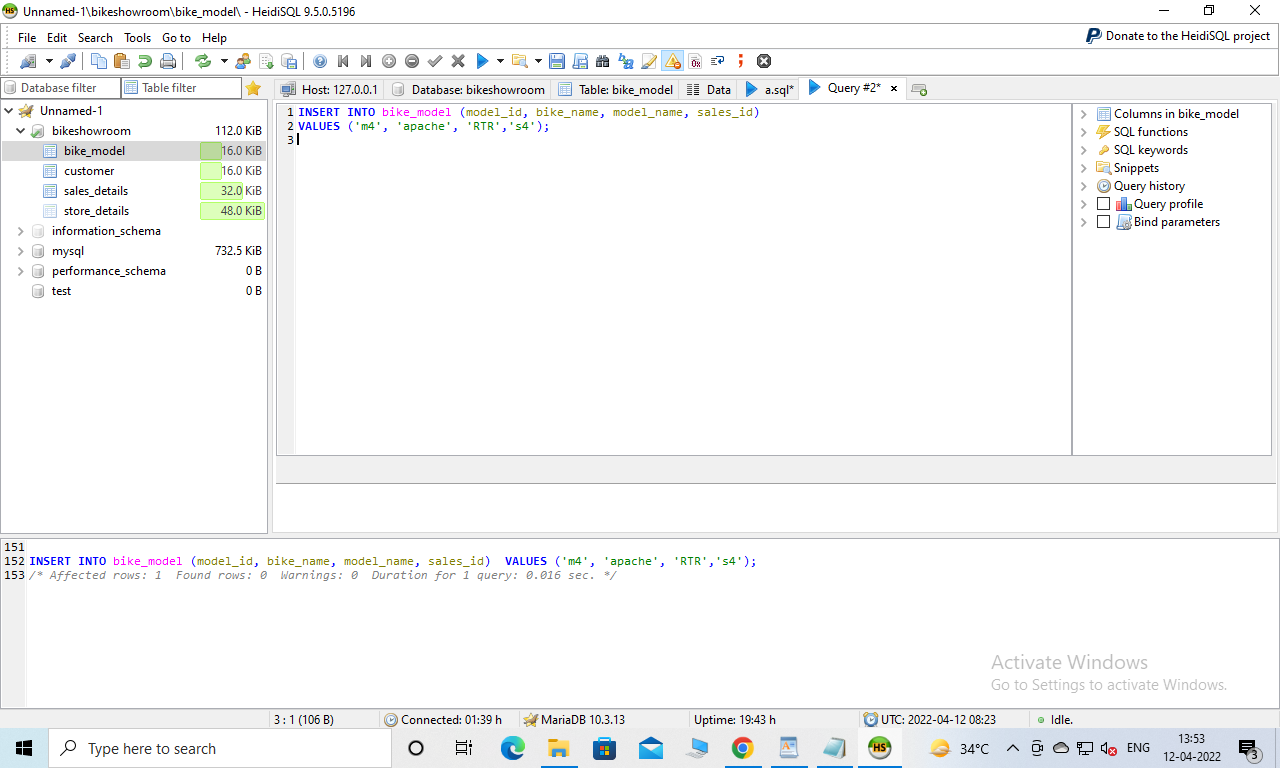
**VALUES (4, 'mitchel', 'Apache', 's4');**

****

**Query 2:**

**INSERT INTO bike\_model (model\_id, bike\_name, model\_name, sales\_id)**

**VALUES ('m4', 'apache', 'RTR','s4');**

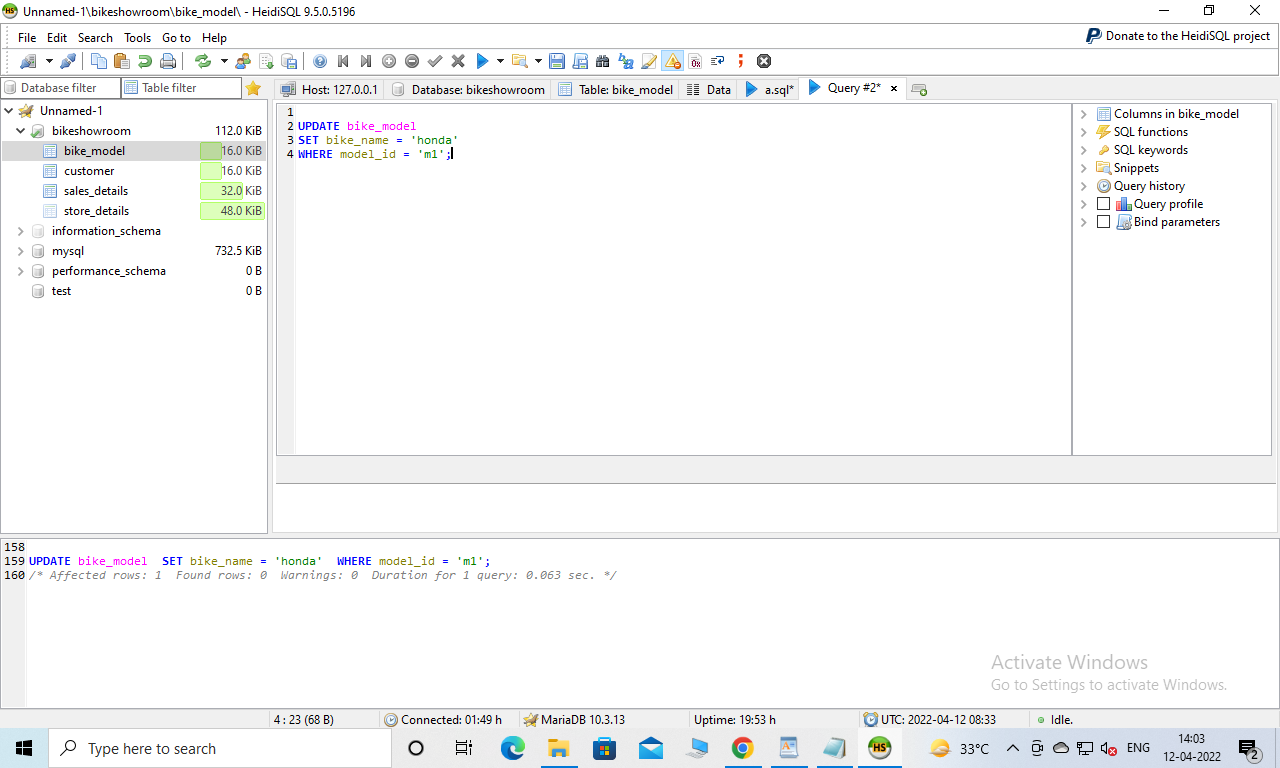
****

* 1 point each for the two update statements (2 pts total)

**Query 3: UPDATE bike\_model**

**SET bike\_name = 'honda'**

**WHERE model\_id = 'm1';**

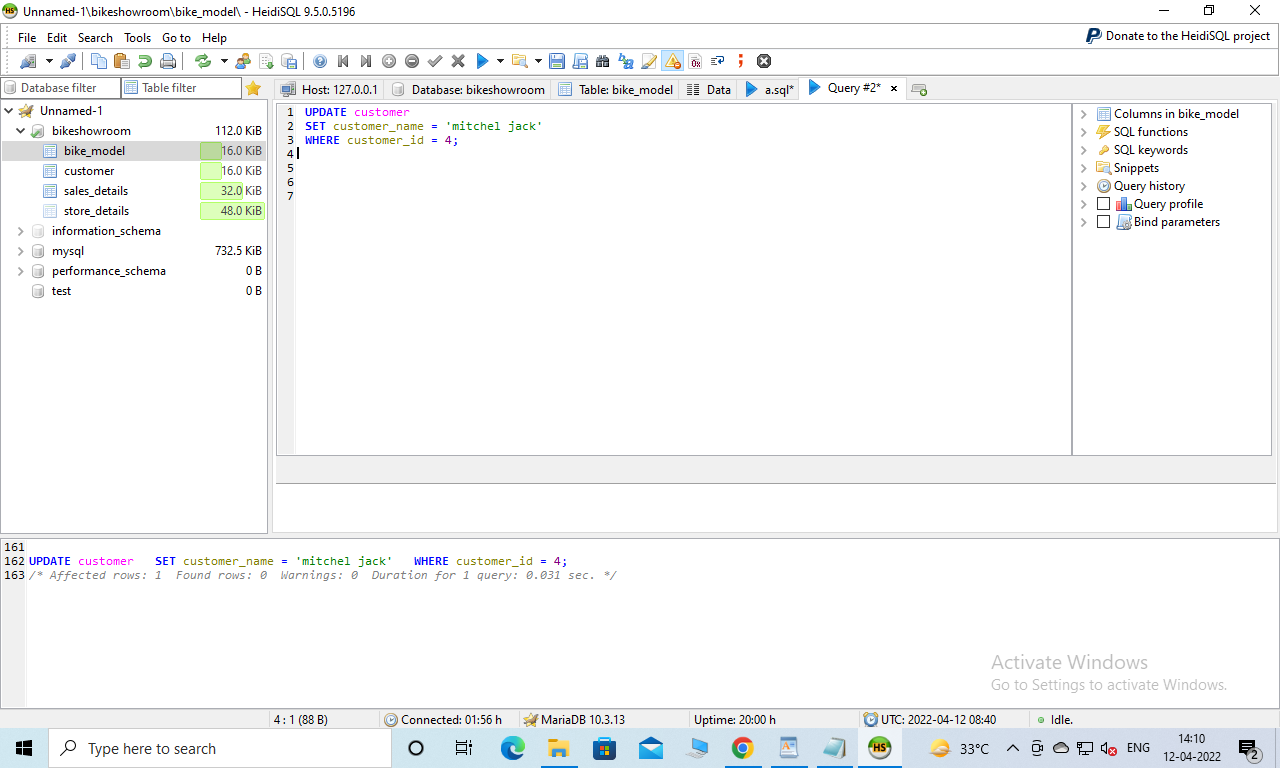
****

**Query4 :**

**UPDATE customer**

**SET customer\_name = 'mitchel jack'**

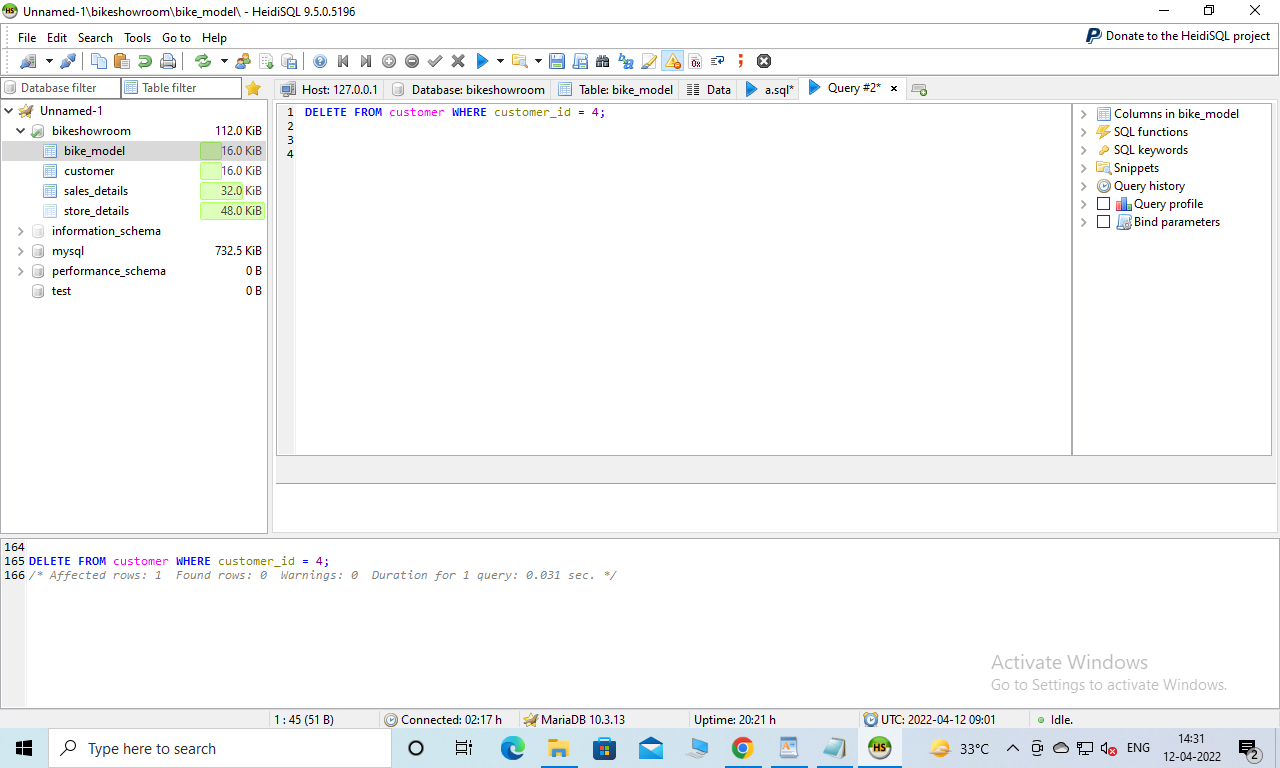
**WHERE customer\_id = 4;**

****

* 1 point for the delete statement (1 pt)

**Query 5**

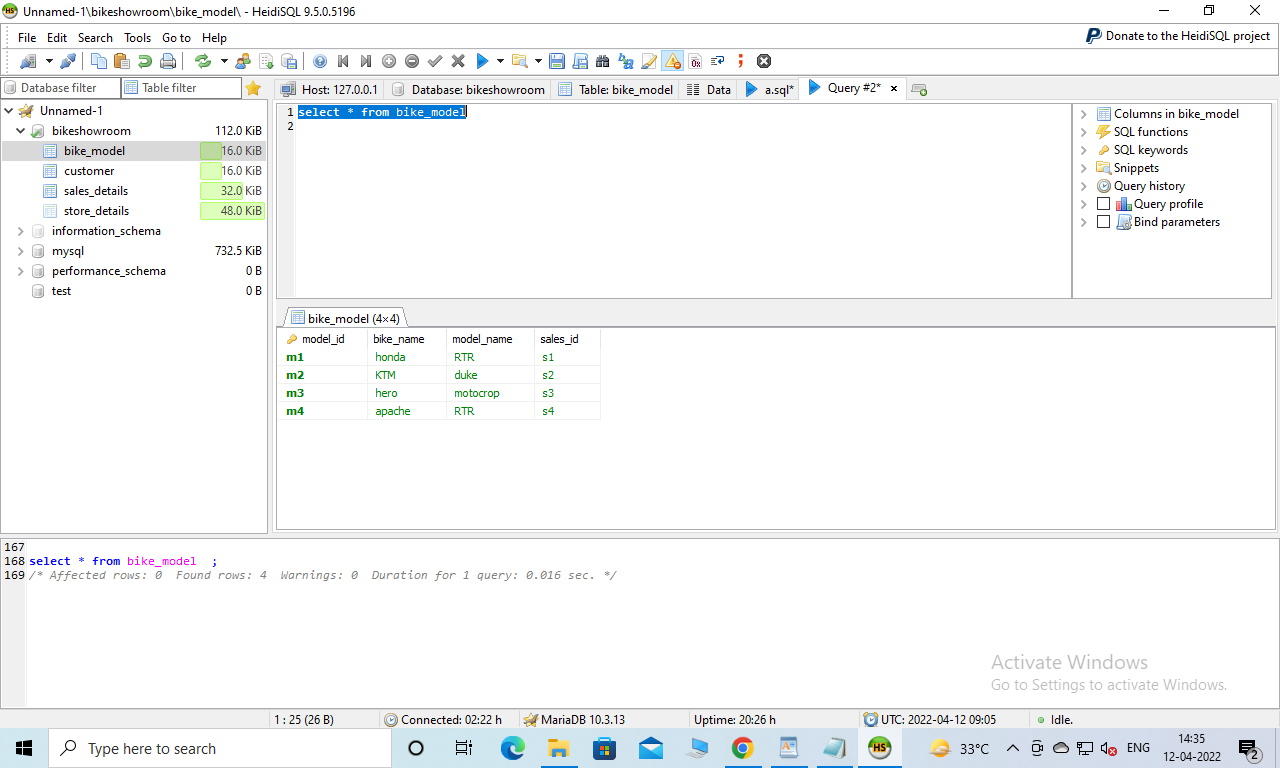
**DELETE FROM customer WHERE customer\_id = 4;**

****

* 1 point for the simple select statement (1 pt)

**Query 6:**

**select \* from bike\_model**

****

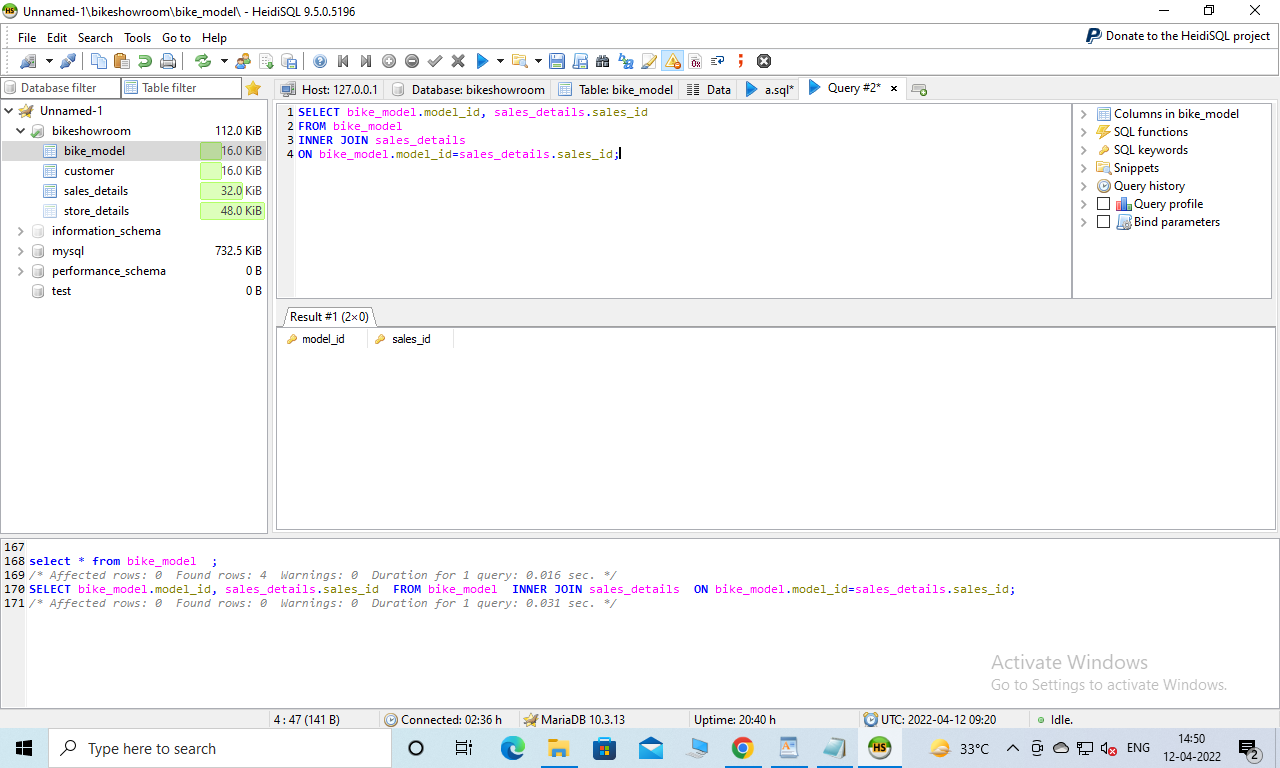
* 2 points each for the 2 join statements (4 pts total)

**Query 7: SELECT bike\_model.model\_id, sales\_details.sales\_id**

**FROM bike\_model**

**INNER JOIN sales\_details**

**ON bike\_model.model\_id=sales\_details.sales\_id;**

****

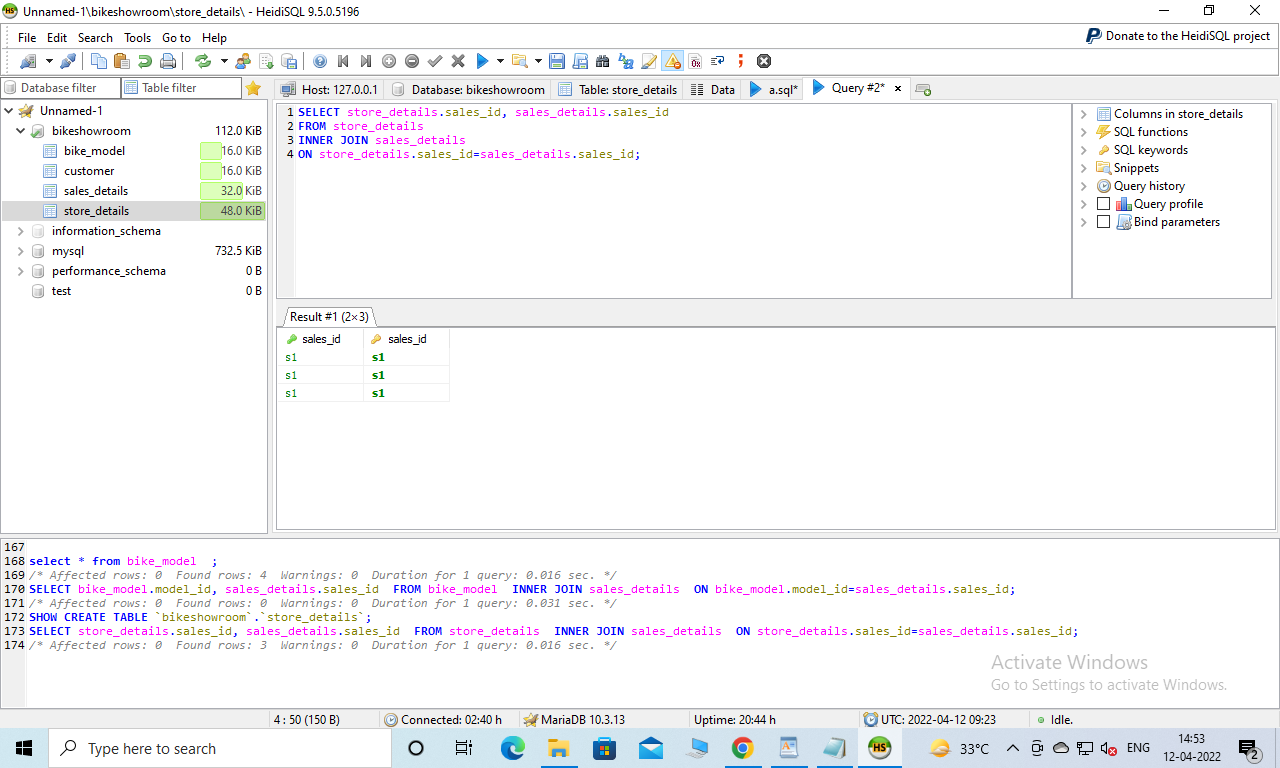
**Query 8:**

**SELECT store\_details.sales\_id, sales\_details.sales\_id**

**FROM store\_details**

**INNER JOIN sales\_details**

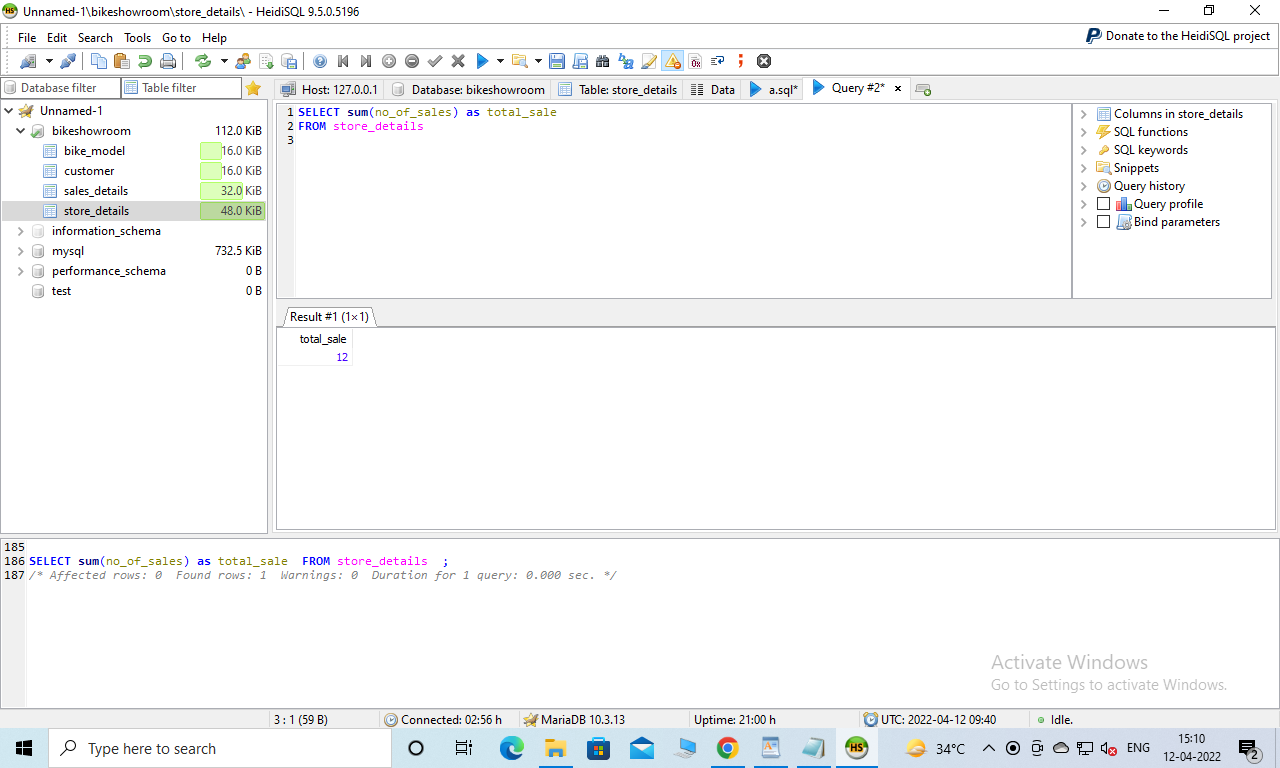
**ON store\_details.sales\_id=sales\_details.sales\_id;**

****

* 2 points each for the three queries that use summary statements (6 pts total)

**Query 9: SELECT sum(no\_of\_sales) as total\_sale**

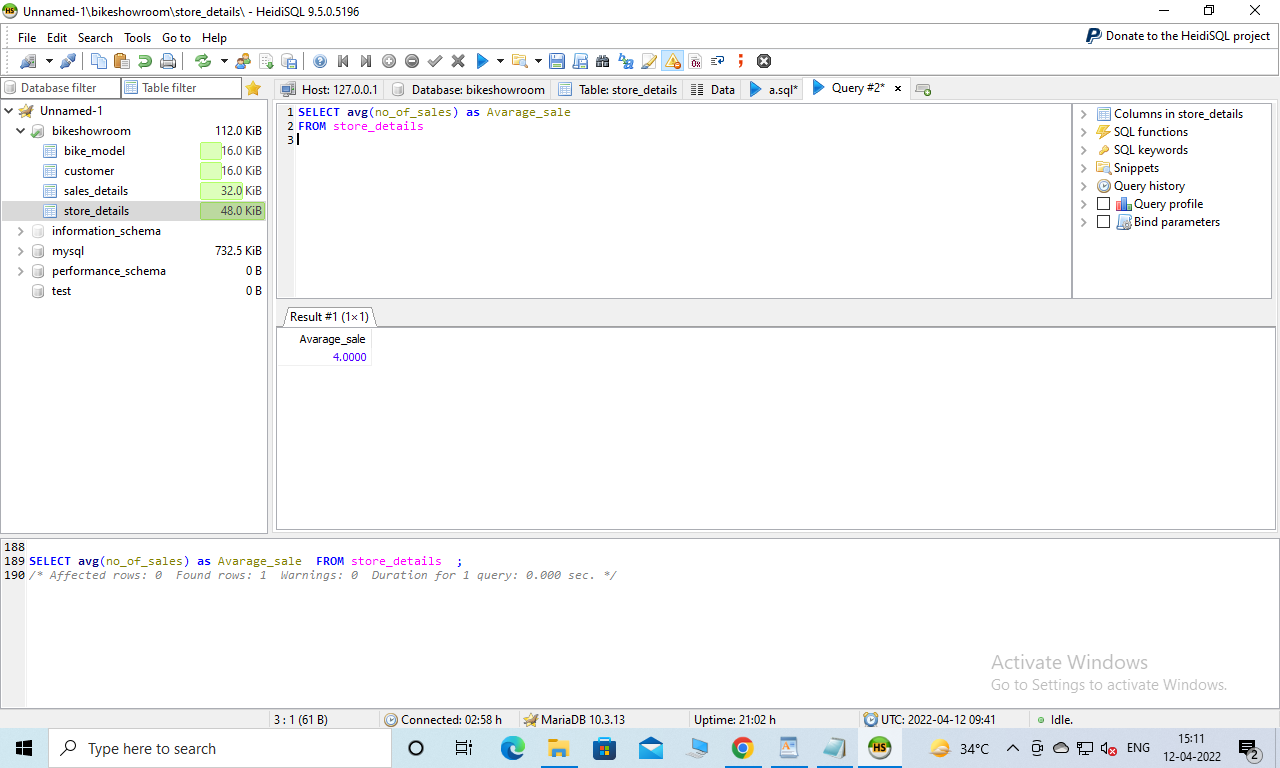
**FROM store\_details**

****

**Query 10:**

**SELECT avg(no\_of\_sales) as Avarage\_sale**

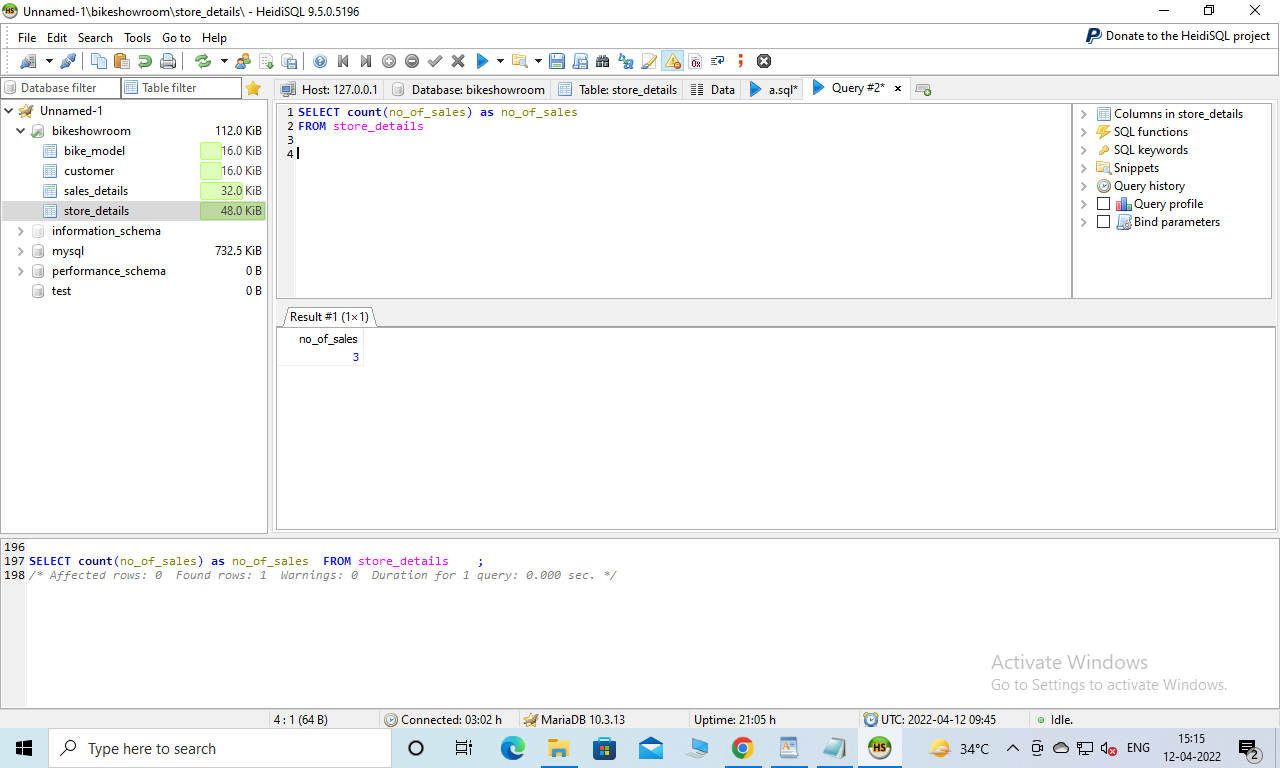
**FROM store\_details**

****

**Query 11:**

**SELECT count(no\_of\_sales) as no\_of\_sales**

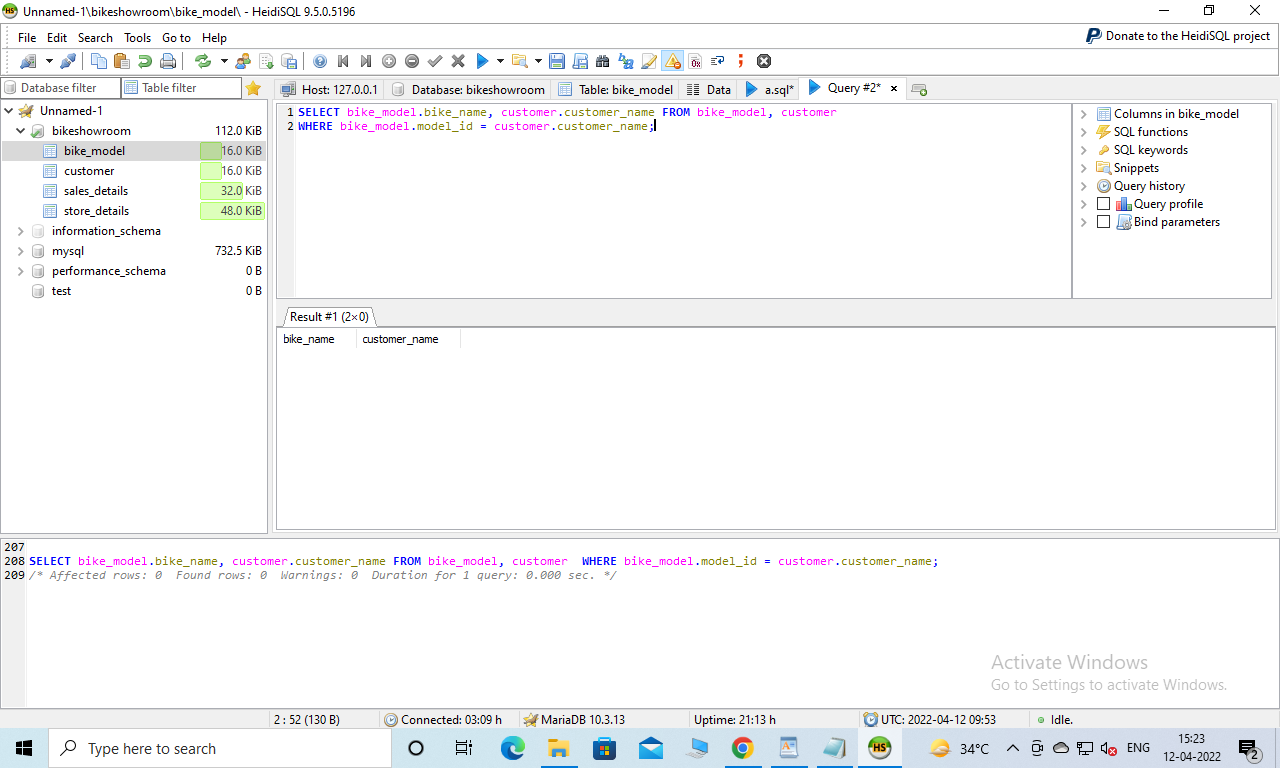
**FROM store\_details**

****

* 2 points for the multi-table query (2 pts)

**Query 12: SELECT bike\_model.bike\_name, customer.customer\_name FROM bike\_model, customer**

**WHERE bike\_model.model\_id = customer.customer\_name;**

****

* 12 points for showing the query and a screenshot of the corresponding result set back-to-back for each of these queries in your Word document. In short, you should have one screenshot for each query above for a total of 12 screenshots, and each screenshot that shows the effects of a query is worth a point.
* The query .sql file on GitHub containing all of the queries (10 pts)

Total points possible: 40

**WARNING:** Screen captures that just show an ‘ok’ message are NOT valid results. You must display the change to the database or the output of the query.

Also, I should be able to download your sql file and run it to confirm that the commands work. Of course I’ll have to create your database structures first using your script from the first half of this assignment.

**SPECIAL WARNING:** This is not the same assignment as the one I taught last year. If I see a submission that looks like last year’s project, I will issue 0’s for the entire project!