Critical Thinking Assignment 5: Writing Queries

Alexander Ricciardi

Colorado State University Global

ITS410: Database Management

Dr. Murthy Rallapalli

May 18, 2025

Critical Thinking Assignment 5: Writing Queries

This documentation is part of the Critical Thinking 5 Assignment from ITS410: Database Management at Colorado State University Global. The documentation provides screenshots showcasing writing queries using MySQL and the My Guitar Shop database.

The Assignment Direction:

Writing Queries
Using the My Guitar Shop database you installed in Module 1, develop the following queries.

1. Write a SELECT statement that returns these columns:
   - The count of the number of orders in the Orders table
   - The sum of the tax_amount columns in the Orders table

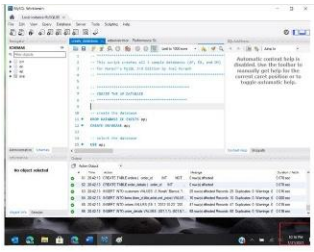
   Execute the query and take a screenshot of the query and the results.
2. Write a SELECT statement that returns one row for each category that has products with these columns:
   - The category_name column from the Categories table
   - The count of the products in the Products table
   - The list price of the most expensive product in the Products table.
   - Sort the result set so the category with the most products appears first.

   Execute the query, and take a screenshot of the query and the results.
3. Write a SELECT statement that returns one row for each customer that has orders with these columns:
   - The email_address column from the Customers table
   - The sum of the item price in the Order_Items table multiplied by the quantity in the Order_Items table
   - The sum of the discount amount column in the Order_Items table multiplied by the quantity in the Order_Items table
   - Sort the result set in descending sequence by the item price total for each customer.

   Execute the query and take a screenshot of the query and the results.
4. Write a SELECT statement that returns one row for each customer that has orders with these columns:
   - The email_address column from the Customers table
   - A count of the number of orders
   - The total amount for each order (*Hint: First, subtract the discount amount from the price. Then, multiply by the quantity.*)
   - Return only those rows where the customer has more than one order.
   - Sort the result set in descending sequence by the sum of the line item amounts.

   Execute the query and take a screenshot of the query and the results

All the screenshots should show current date. Example of screenshot.



Submit your labeled results screenshots in a Word file.

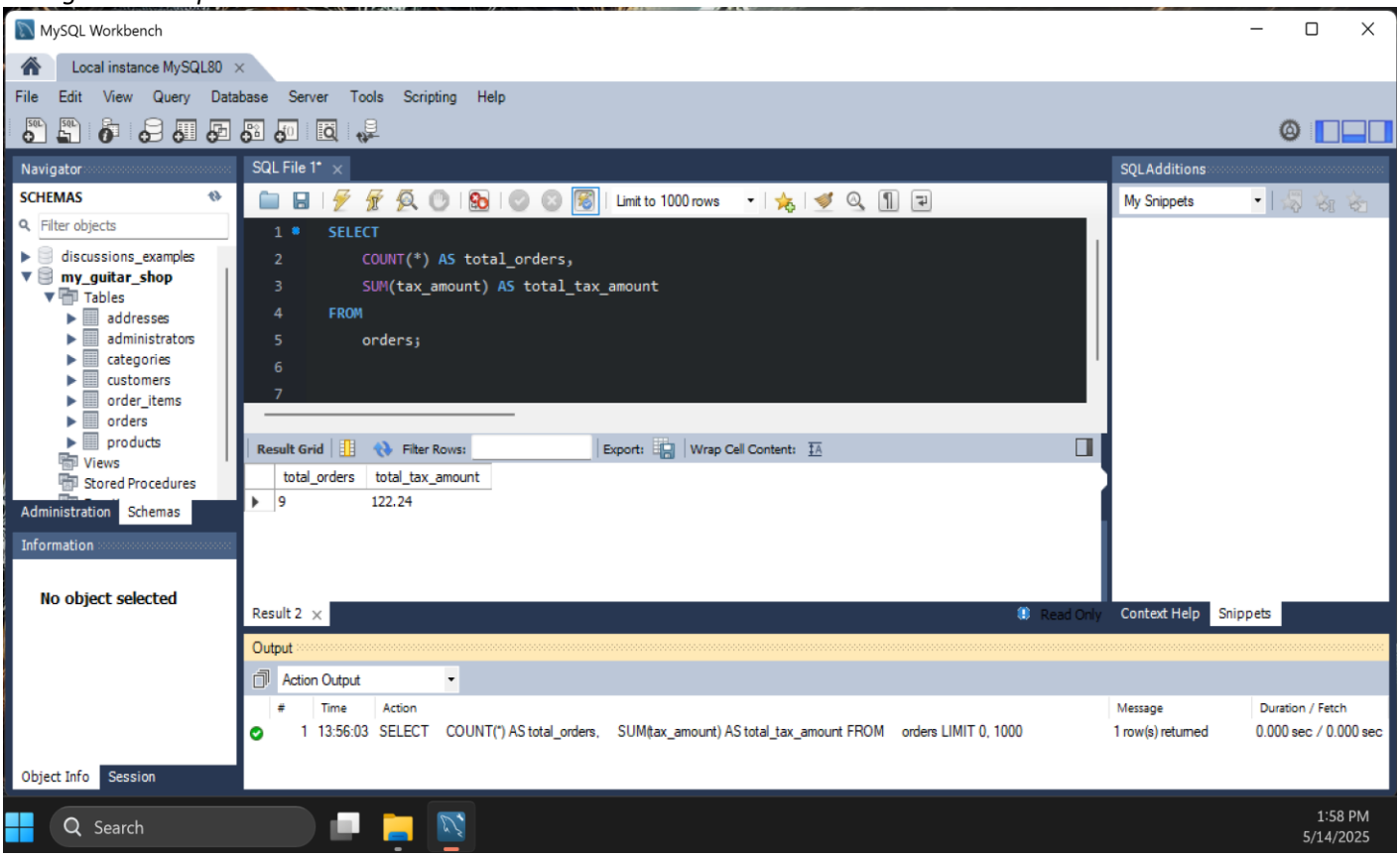
Screenshots

Step 1: Write a `SELECT` statement that returns these columns:

- The count of the number of orders in the `Orders` table
- The sum of the `tax_amount` columns in the `Orders` table

Figure 1

Assignment Steps 1



Note: The figure illustrates the MySQL Workbench result after performing steps 1.

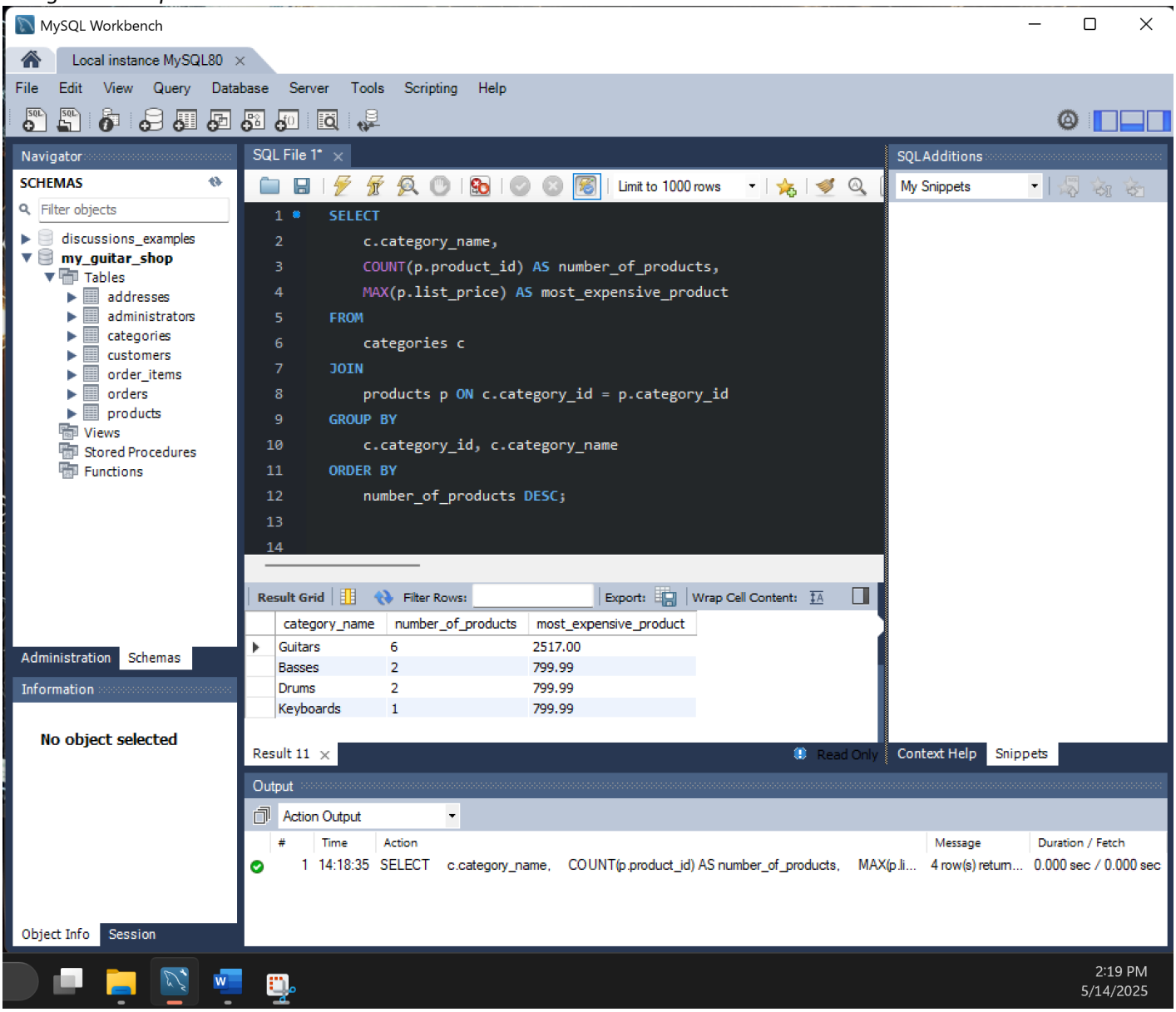
Please see the next page.

Step 2: Write a `SELECT` statement that returns one row for each category that has products with these columns:

- The `category_name` column from the `Categories` table
- The count of the products in the `Products` table
- The list price of the most expensive product in the `Products` table.
- Sort the result set so the category with the most products appears first.

Figure 2

Assignment Steps 2



Note: The figure illustrates the MySQL Workbench result after performing steps 2.

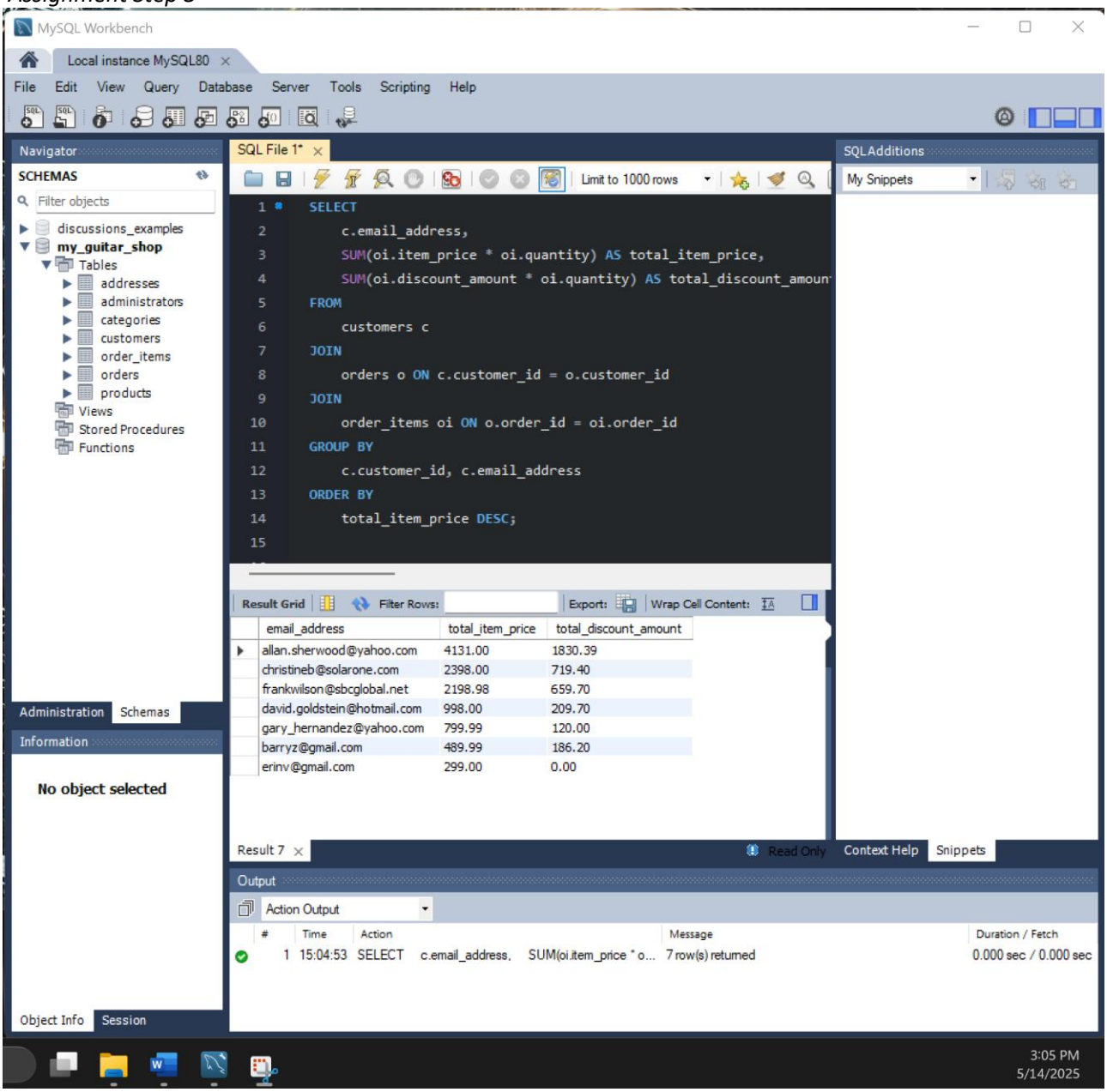
Please see the next page.

Step 3: Write a SELECT statement that returns one row for each customer that has orders with these columns:

- The `email_address` column from the Customers table
- The sum of the item price in the `Order_Items` table multiplied by the quantity in the `Order_Items` table
- The sum of the discount amount column in the `Order_Items` table multiplied by the quantity in the `Order_Items` table
- Sort the result set in descending sequence by the item price total for each customer.

Figure 3

Assignment Step 3



Note: The figure illustrates the MySQL Workbench result after performing steps 3.

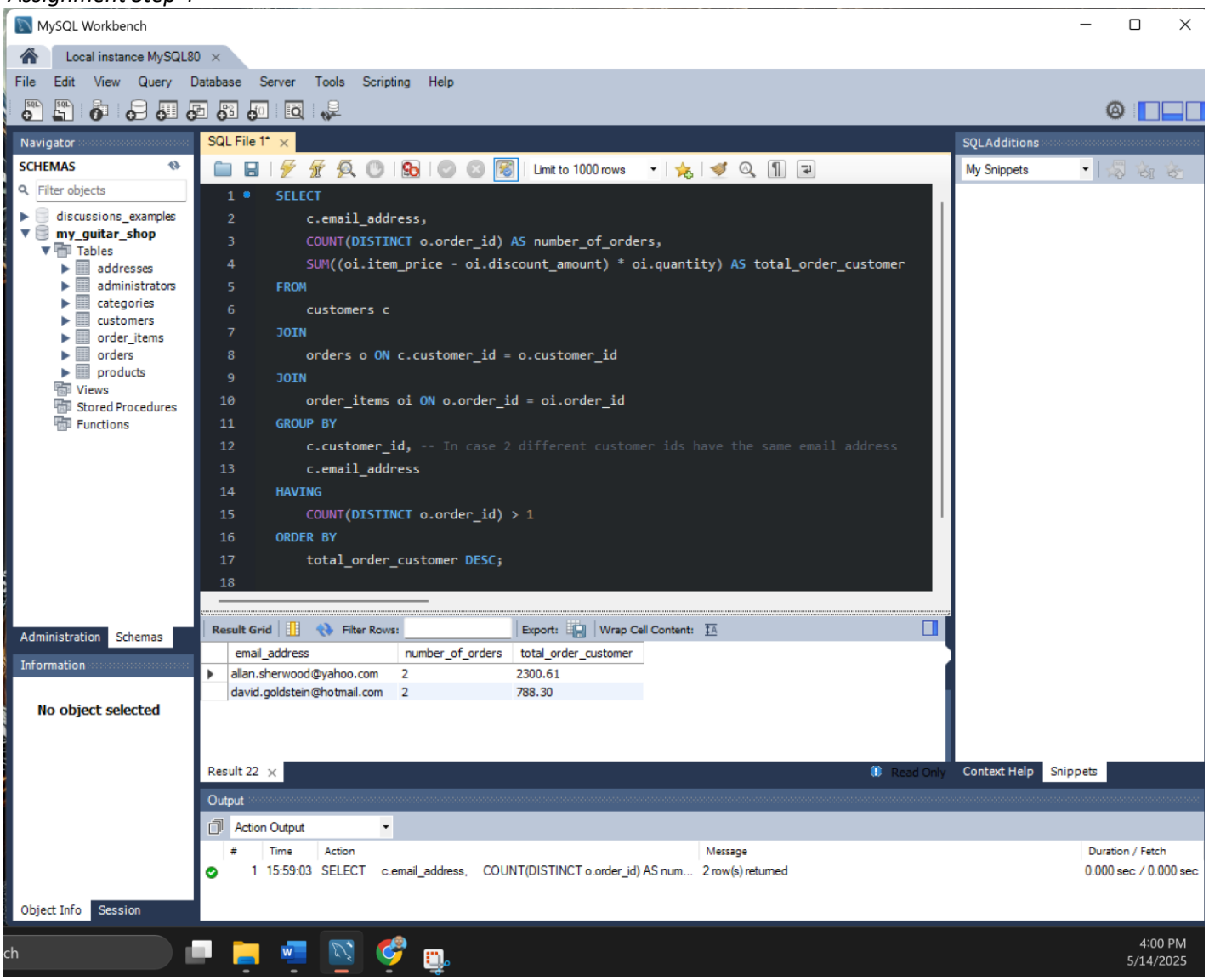
Step 4: Write a `SELECT` statement that returns one row for each customer that has orders with these columns:

- The `email_address` column from the `Customers` table
- A count of the number of orders
- The total amount for each order (*Hint: First, subtract the discount amount from the price. Then, multiply by the quantity.*)
- Return only those rows where the customer has more than one order.
- Sort the result set in descending sequence by the sum of the line item amounts.

Note that the directions do not ask to incorporate the taxes into the total.

Figure 4

Assignment Step 4



Note: The figure illustrates the MySQL Workbench result after performing steps 4.