Portfolio Project

Alexander Ricciardi

Colorado State University Global

ITS410: Database Management

Dr. Murthy Rallapalli

June 8, 2025

Portfolio Project

This documentation is part of the Portfolio Project from ITS410: Database Management at Colorado State University Global. The project consists of two parts, part 1, a lesson learned reflection report, and part 2, a database design and analysis.

Part 1: Lessons Learned and Reflection

The ITS410: Data Management course at Colorado State University Global was a challenging and very satisfying course. I really enjoyed learning about Rational Data Base Management System (RDBMS), more specifically data management using MySQL and MySQL workbench. I learn valuable lessons from the fundamental concepts of data management to more advanced concepts such as tuning an securing rational database using MySQL. This essay provides an overview of what I learn from each module of the ITS410 course and a reflection on how these lessons provided me with skills to further my endeavor .

Lesson Learn

In module 1, "Getting Started with MySQL Workbench," I learn about relational database concepts such as tables, unique keys, and data relationships. I also learn about Structured Query Language (SQL) use for data manipulation and management within RDBSM. Additionally, I spend a significant amount of time learning about and installing MySQL and MySQL Workbench, and then utilizing Workbench to create initial databases and load data using provided SQL scripts.

In module 2, "MySQL Data Types and Retrieve Data," I learned about the different data types use by MySQL to store data. The module also introduced me to the data retrieval statement: the `SELECT` query, and its clauses like `FROM`, `WHERE`, `ORDER BY`, and `LIMIT` used to fetch and organize data within tables. In the next module, module 3, "Retrieve Data from Multiple Tables," I learned concepts for querying and combining data from multiple tables using table and column aliases for improving code readability. As well as the SQL `JOIN` commands and their various types such as `INNER JOIN` for matching rows, `LEFT JOIN` and `RIGHT JOIN` for including non-matching rows from one table, `CROSS JOIN` for Cartesian products, and `SELF JOIN` for relating rows within the same table.

The module 4, "Create Tables, Insert, Update, and Import Data," introduced me to the Data Definition Language (DDL) and Data Manipulation Language (DML) operations which are part of SQL. Operations such as the `CREATE TABLE` statement combined with clauses like `NOT NULL`, `DEFAULT` values, `AUTO_INCREMENT`, and primary keys. The module also introduced me to INSERT statement for adding single or multiple rows, to `UPDATE` command for modifying existing records and the `WHERE` clause to specify the conditions that determines which rows should be updated. Following module 4, module 5, "Grouping and Summarizing Data," introduced me to data analysis and summarization. I learned how to use the `GROUP BY` clause in conjunction with aggregate functions like `SUM`, `COUNT`, and `AVG` to generate data summaries. It also explained how to use the `HAVING` clause to filter grouped summaries. The module also introduced me to methods for combining distinct query results vertically using the `UNION` and `INTERSECT` operators, as well as the concept of subqueries to performed complex data retrieval.

In module 6, "Stored Procedures and Functions," I learned how to create and use stored procedures and user-defined functions, which allow the implementation of reusable SQL code directly within the database. I learned about their syntax used to create, call, alter, and drop these stored procedures. Additionally, I learned about their benefits such as improving application security, modularity, reduced

network traffic, reusability, and functionality for executing complex operations. The next module, module 7, “Database Tuning and Security,” introduced me to database tunning, backup, and recovery. The module explained the importance of database indexes, particularly when using MySQL's B-tree indexing structure for searching and fetching data. It also demonstrated how to use statements such as `EXPLAIN`, and considering object characteristics like table size to analysis queries for optimization purposes. Additionally, it described the best practices for securing a database.

The final module, module 8, “Backup and Recovery,” introduced me to database backup and recovery best practices. It described how these practices are crucial for preventing catastrophic data losses. I learned how to perform logical database backups by using the `mysqldump` program utility, which creates SQL script files that can be used to recreate the database schemas and data. I also learned about physical backups, which are direct copy of the database files and directories. Finally, I learned about database replication, which is a live slave copies of a master database that can live in one or more slave servers.

Reflection

I enjoyed learning about data management in MySQL; the course provided me with a robust foundation for data management and data system analysis. Additionally, what I learned through this course give me the skills needed to choose data types for a specific data application, to create and modify databases, how to modify and manage those databases, how to administered data management system-based MySQL, how to keep those systems secure, how to protect them from catastrophic data losses, and how to maintain and optimize these systems. All these skills that I acquired throughout this course are crucial for effectively managing and administrate data systems, which is an essential component of information technology and businesses functionality. These acquired skills provide me with a practical toolkit that is essential for my future endeavors as a computer science student and as a computer science professional. They will allow me to confidently design, implement, and maintain RDBMS, optimize my queries, implement data security measures and maintain data integrity within those systems.

Conclusion

The ITS410: Data Management course at Colorado State University Global provided me with fundamentals knowledge and understanding of data management, more specifically Rational Data Base Management System (RDBMS). In this course I learned how to use MySQL and MySQL Workbench to create, modify, manage, and administer complex rational databases. This provided me with the knowledge, critical thinking skills, and the technical skills that I need as a computer science student and professional to tackle real-world database challenges and contribute to the development of database projects. Furthermore, I enjoy the course and believe that the skills and knowledge that I acquired will be immensely beneficial in my future academic and professional endeavors in computer science.

Part 2: Queries

The Assignment Direction:

Using the My Guitar Shop database you installed in Module 1, develop the following queries.
SUBMIT A SCREENSHOT OF EACH STEP.

1. Write a SELECT statement that returns these column names and data from the Products table:

product_name	The product_name column
list_price	The list_price column
discount_percent	The discount_percent column
discount_amount	A column that's calculated from the previous two columns
discount_price	A column that's calculated from the previous three columns

   - Round the discount_amount and discount_price columns to two decimal places.
   - Sort the result set by the discount_price column in descending sequence.
   - Use the LIMIT clause so the result set contains only the first five rows.
   - Submit a screenshot.

2. Write a SELECT statement that returns these column names and data from the Order_Items table:

item_id	The item_id column
item_price	The item_price column
discount_amount	The discount_amount column
quantity	The quantity column
price_total	A column that's calculated by multiplying the item price by the quantity
discount_total	A column that's calculated by multiplying the discount amount by the quantity
item_total	A column that's calculated by subtracting the discount amount from the item price and then multiplying by the quantity

   - Only return rows where the item_total is greater than 500.
   - Sort the result set by the item_total column in descending sequence.
   - Submit a screenshot.

3. Write a SELECT statement that returns the product_name and list_price columns from the Products table.
   - Return one row for each product that has the same list price as another product.
   - Hint: Use a self-join to check that the product_id columns aren't equal but the list_price columns are equal.
   - Sort the result set by the product_name column. Submit a screenshot.

4. Write a SELECT statement that returns these two columns:

category_name	The category_name column from the Categories table
product_id	The product_id column from the Products table

   - Return one row for each category that has never been used.
   - Hint: Use an outer join and only return rows where the product_id column contains a null value.
   - Submit a screenshot.

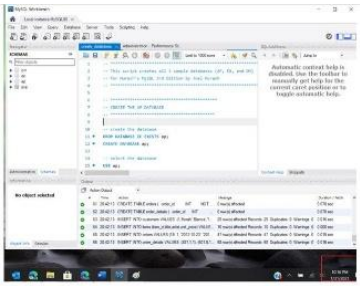
5. Write an INSERT statement that adds this row to the Customers table:
   email_address: rick@raven.com
   password: (empty string)
   first_name: Rick
   last_name: Raven
   - Use a column list for this statement.
   - Submit a screenshot.

6. Write a SELECT statement that answers this question: Which customers have ordered more than one product?
   - Return these columns:
     The email_address column from the Customers table
     The count of distinct products from the customer's orders
   - Sort the result set in ascending sequence by the email_address column.
   - Submit a screenshot.

7. Write a SELECT statement that answers this question: What is the total quantity purchased for each product within each category?
   - Return these columns
     - The category_name column from the category table
     - The product_name column from the products table
     - The total quantity purchased for each product with orders in the Order_Items table
     - Use the WITH ROLLUP operator to include rows that give a summary for each category name as well as a row that gives the grand total.
     - Use the IF and GROUPING functions to replace null values in the category_name and product_name columns with literal values if they're for summary rows.
   - Submit a screenshot.

8. Write and execute a script that creates a user with a username using your firstname initial and lastname and password of your choosing. This user should be able to connect to MySQL from any computer.
   - This user should have SELECT, INSERT, UPDATE, and DELETE privileges for the Customers, Addresses, Orders, and Order_Items tables of the My Guitar Shop database.
   - However, this user should only have SELECT privileges for the Products and Categories tables. -
   - Also, this user should not have the right to grant privileges to other users.
   - Check the privileges for the user by using the SHOW GRANTS statement.
   - Submit a screenshot.

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

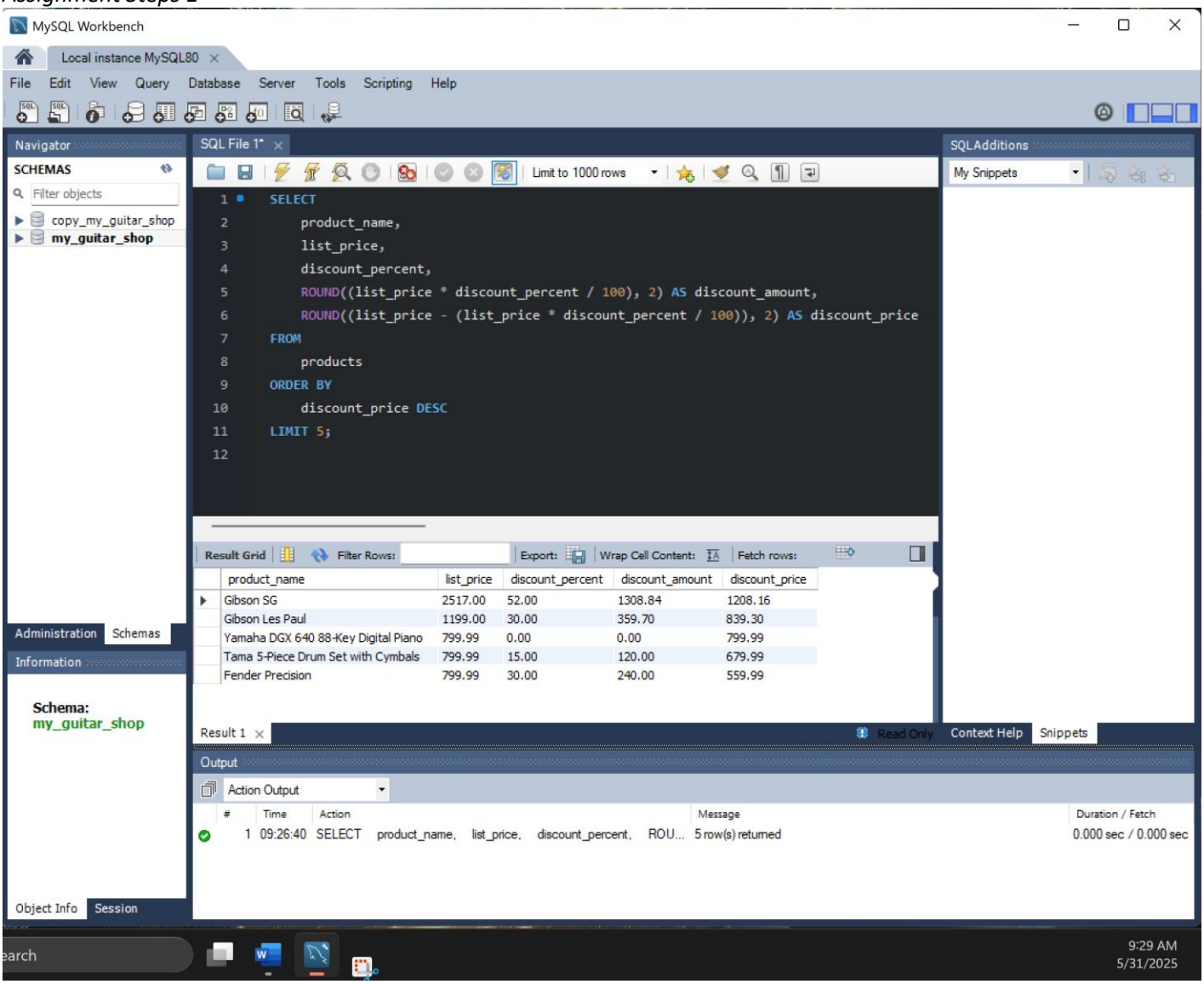


Submit your labeled results screenshots in a Word file.

Step 1: Write a SELECT statement that returns these column names and data from the Products table.

Figure 1

Assignment Steps 1



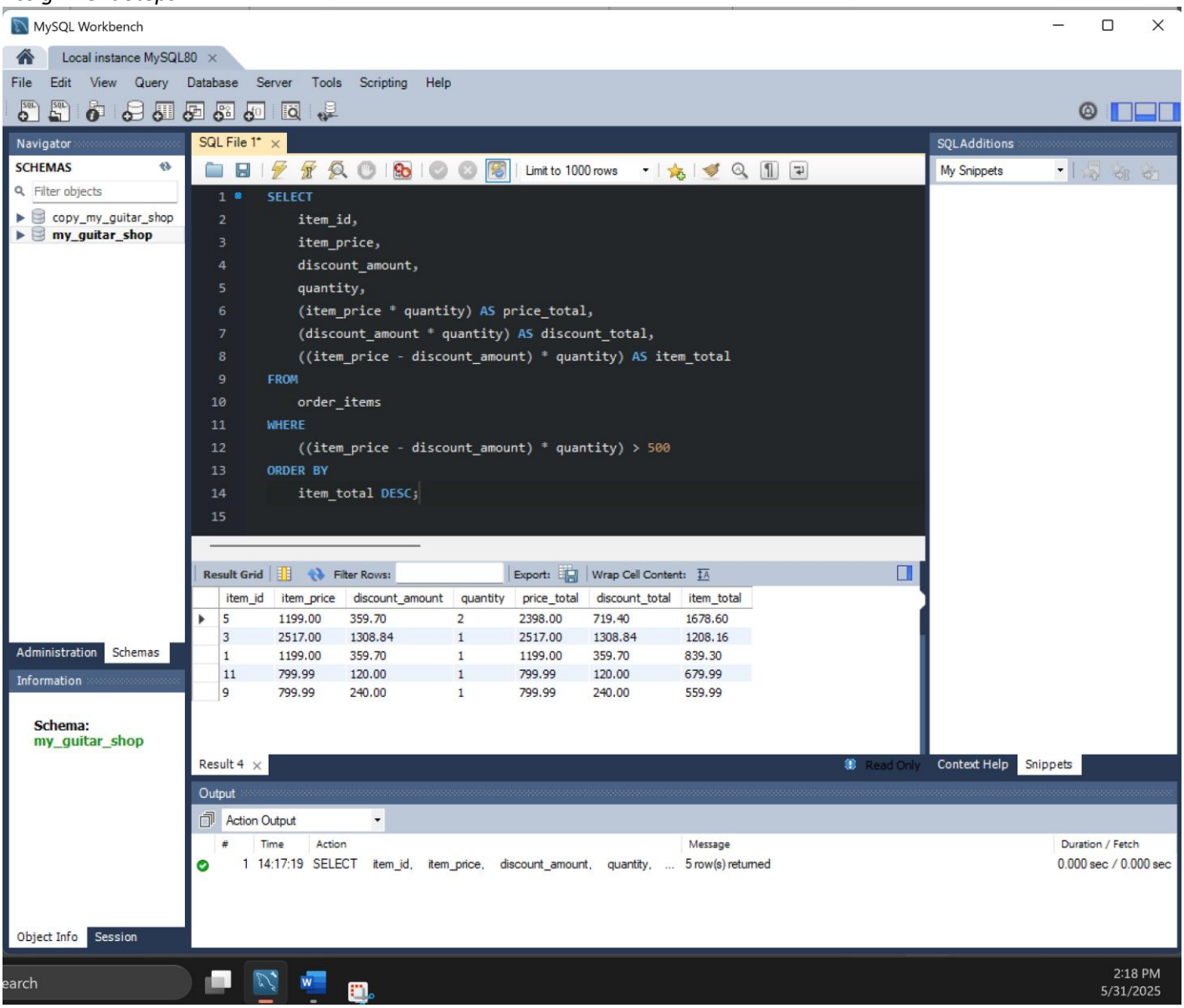
Note: The figure illustrates the MySQL Workbench result after performing steps 1.

Please see the next page.

Step 2: Write a SELECT statement that returns these column names and data from the Order_Items.

Figure 2

Assignment Steps 2



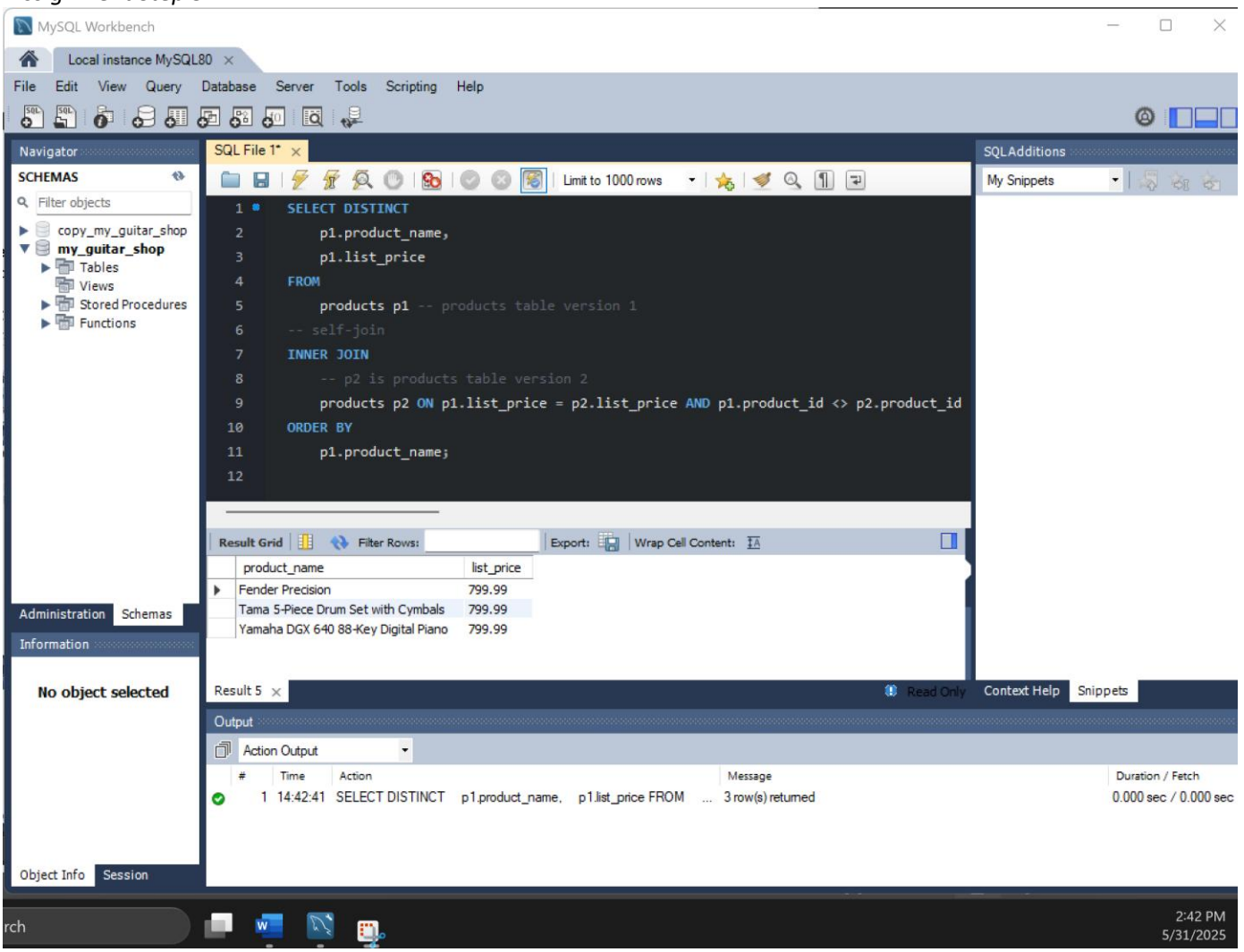
Note: The figure illustrates the MySQL Workbench result after performing steps 2.

Please see the next page.

Step 3: Write a SELECT statement that returns the product_name and list_price columns from the Products table.

Figure 3

Assignment Step 3



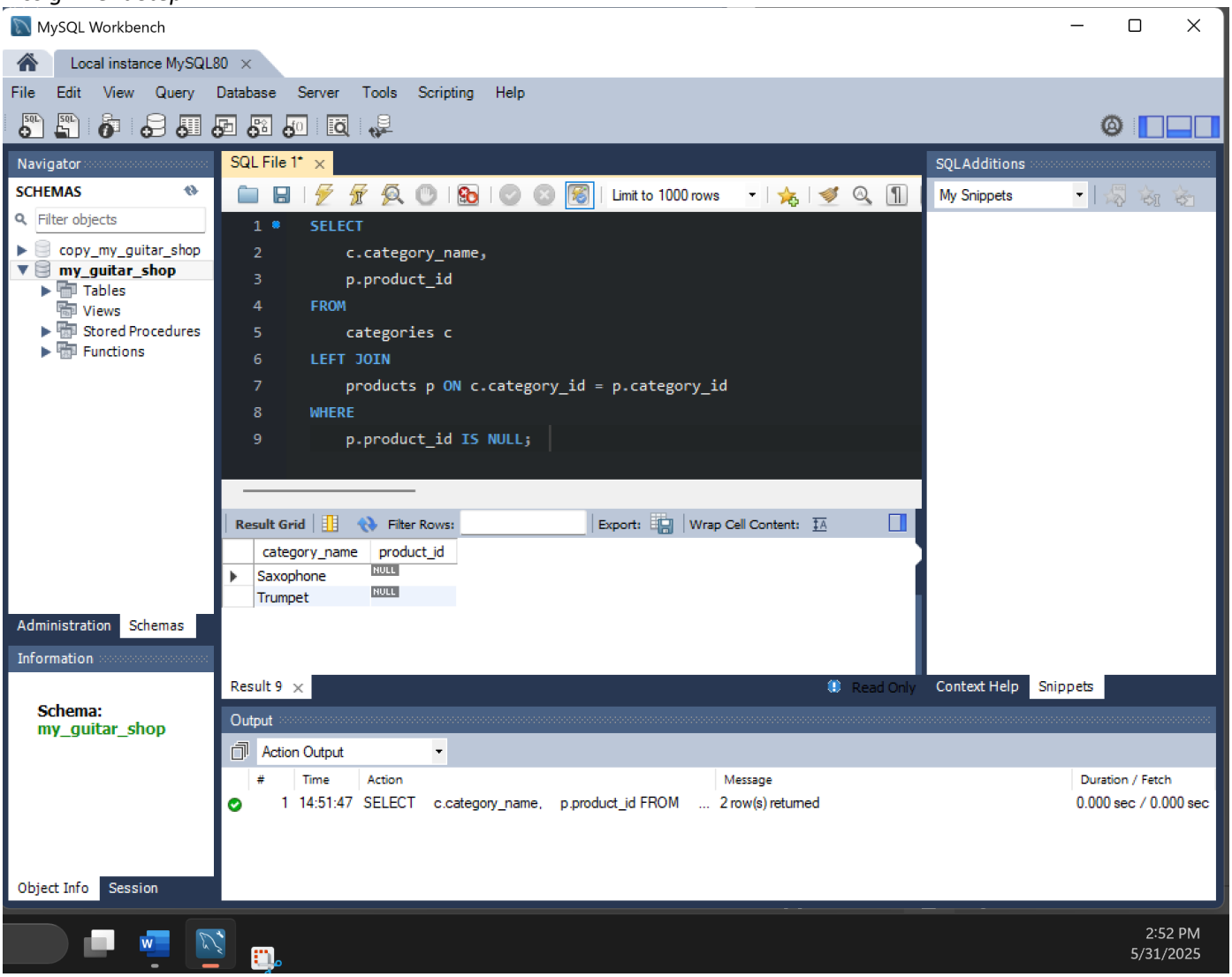
Note: The figure illustrates the MySQL Workbench result after performing steps 3.

Please see the next page.

Step 4: Write a SELECT statement that returns these two columns : category_name and product_id.

Figure 4

Assignment Step 4



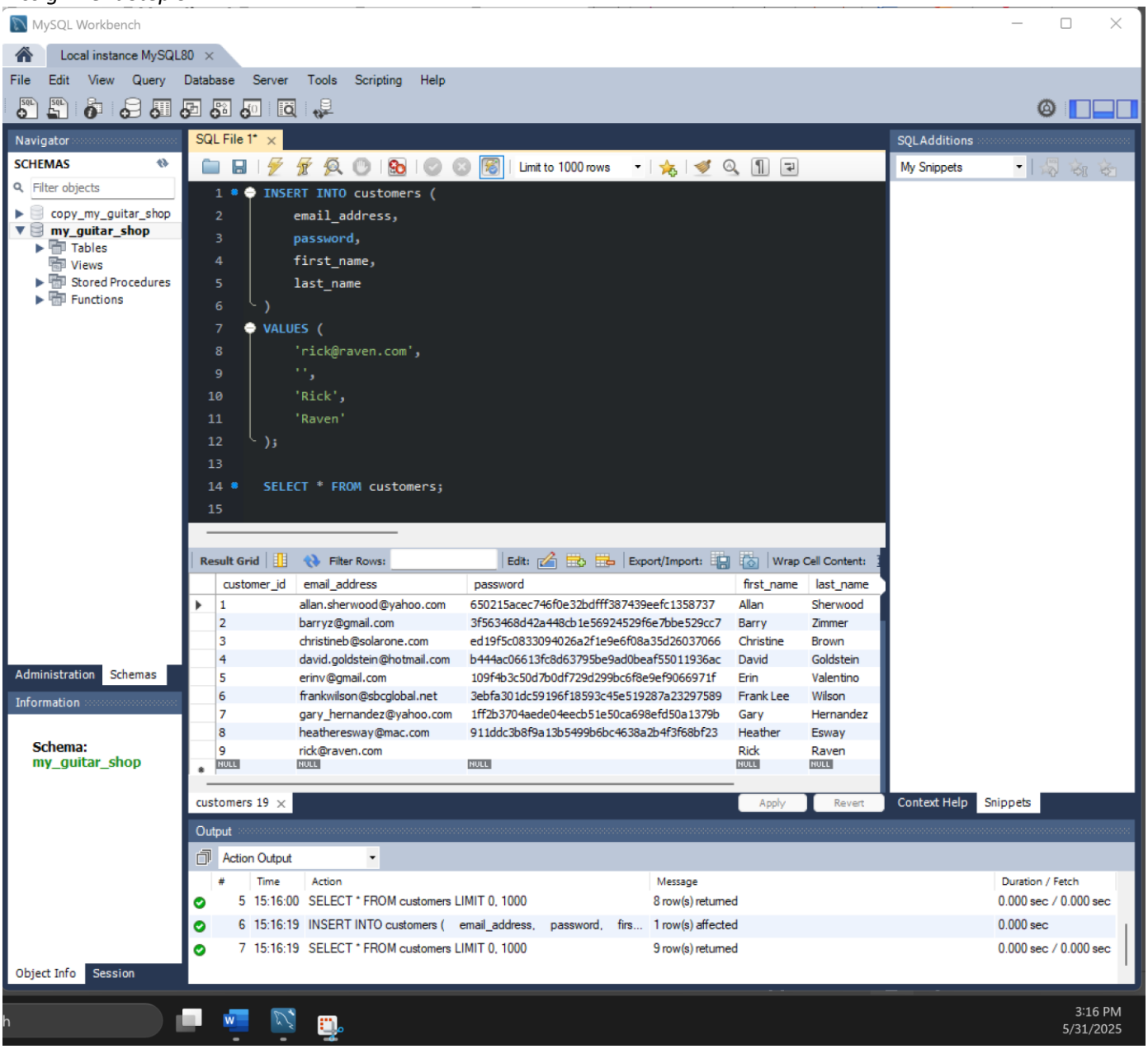
Note: The figure illustrates the MySQL Workbench result after performing steps 4.

Please see the next page.

Step 5: Write an INSERT statement that adds this row to the Customers table.

Figure 5

Assignment Step 5



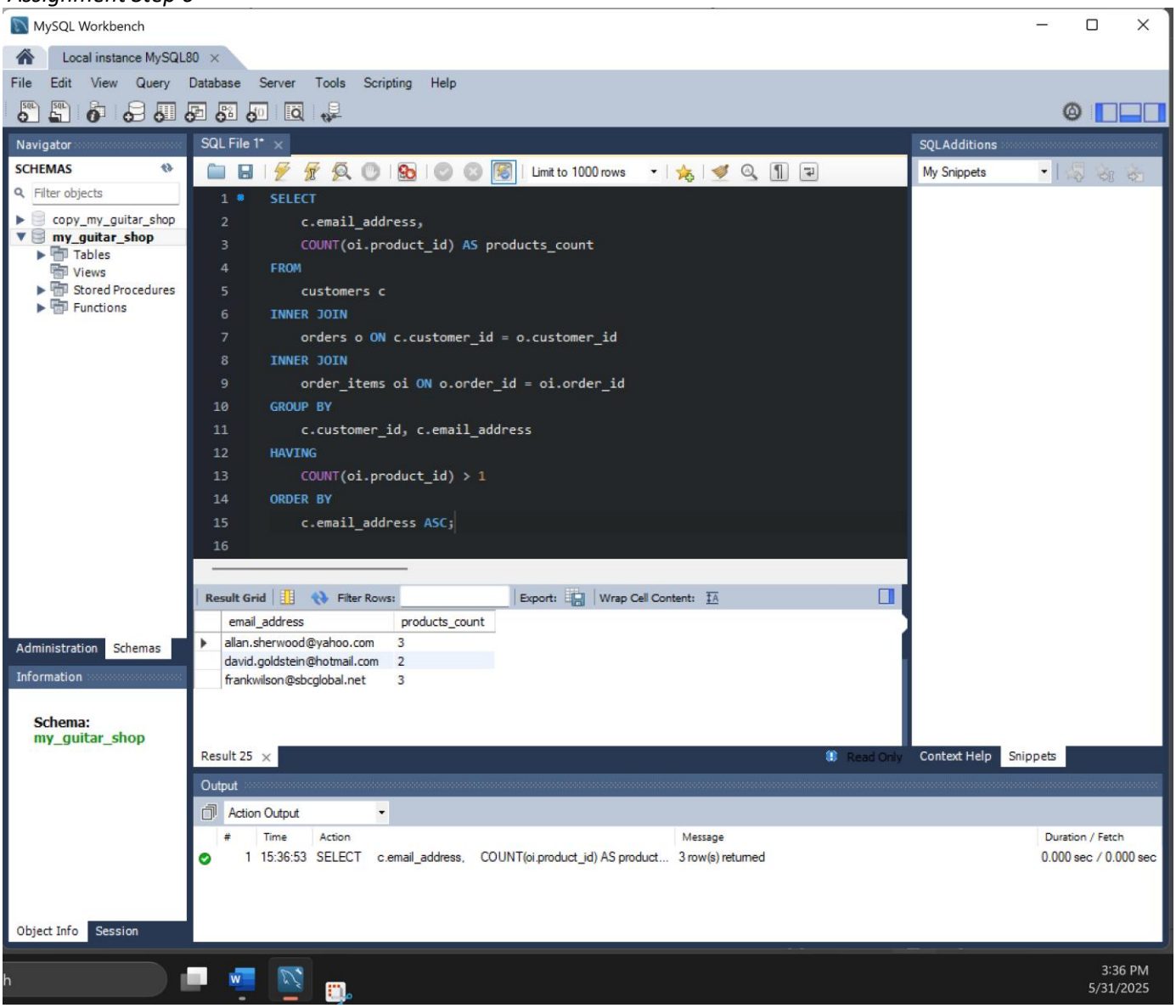
Note: The figure illustrates the MySQL Workbench result after performing steps 5.

Please see the next page.

Step 6: Write a SELECT statement that answers this question: Which customers have ordered more than one product?

Figure 6

Assignment Step 6



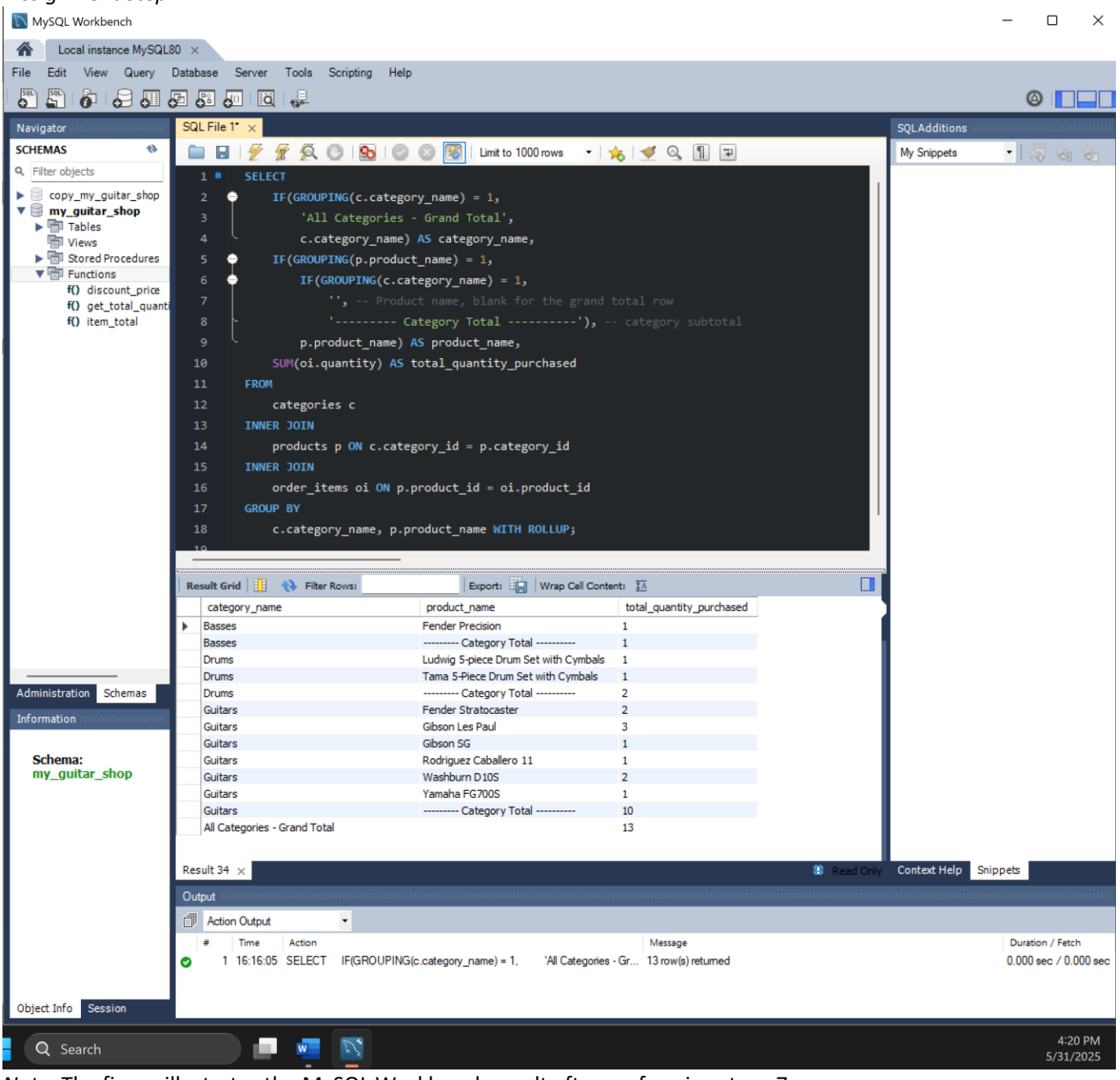
Note: The figure illustrates the MySQL Workbench result after performing steps 6. It shows the emails of customers that ordered more than one product and the number of products they ordered.

Please see the next page.

Step 7: Write a SELECT statement that answers this question: What is the total quantity purchased for each product within each category?

Figure 7

Assignment Step 7



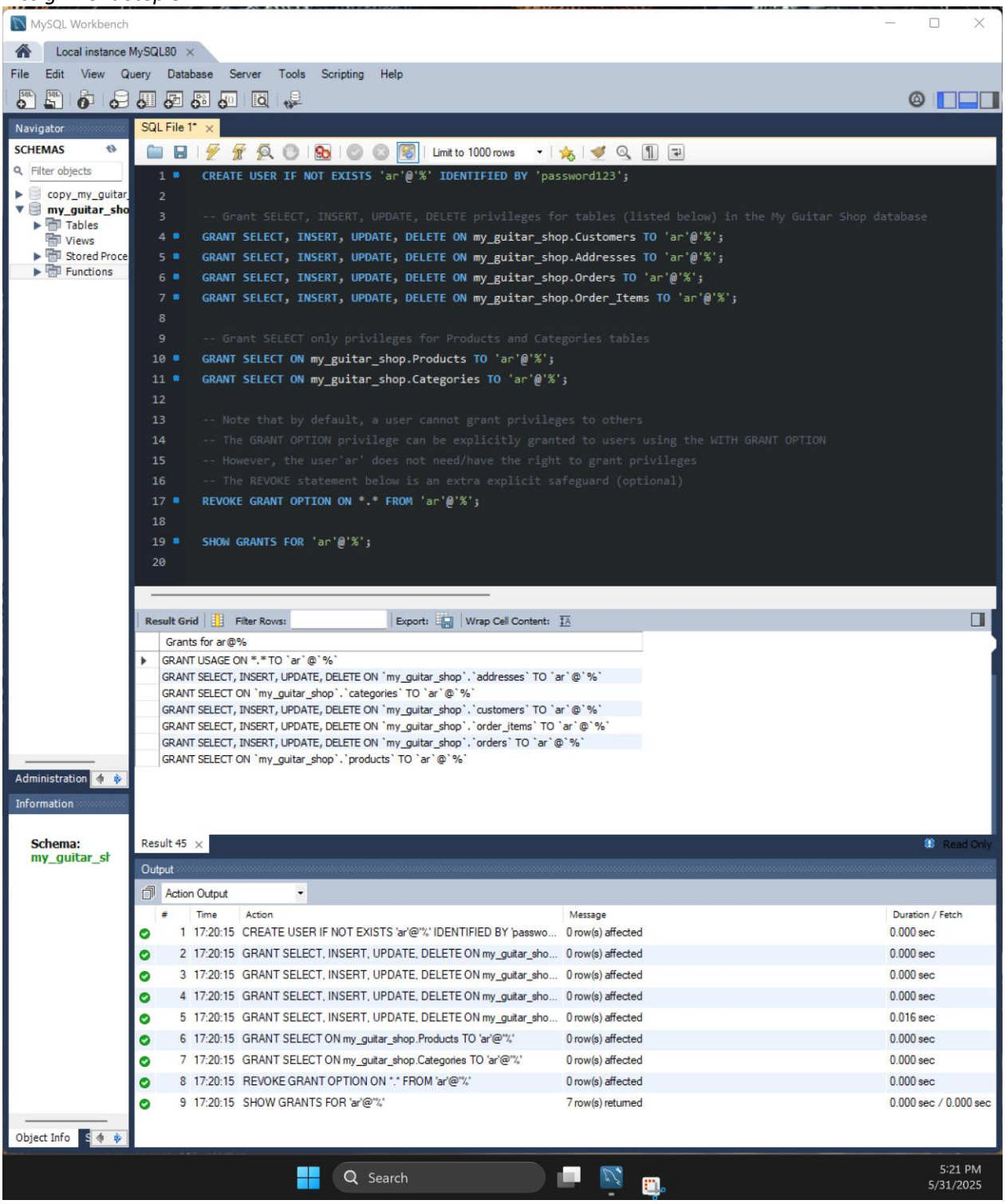
Note: The figure illustrates the MySQL Workbench result after performing steps 7.

Please see the next page.

Step 8: Write and execute a script that creates a user with a username using your firstname initial and lastname and password of your choosing.

Figure 8

Assignment Step 8



Note: The figure illustrates the MySQL Workbench result after performing steps 8.