**Option #1 and #2**

Liam Bergerson

CSU - Global

ITS410-1

K. Gregar-Skillman

5/15/22

**Part 1: Lessons Learned and Reflection**

Databases are incredibly useful with storing the data of an organization. Databases can range from information on a sheet of paper, such as an employee attendance sheet, or a digital database that stores customer names, addresses, credit card info, etc. In order to interact with those large, digital databases, a Database Management System (DBMS) is needed. A DBMS is used to view and manipulate the data within databases. One such DBMS that I have had the pleasure to experience MySQL, a relational DBMS.

A relational database is a type of database that houses data in the form of tables with rows and columns. Relational databases can also relate information in different tables with each other. MySQL is also open-source which means that it is completely free of charge; no licenses or subscriptions are needed to use MySQL. MySQL utilizes the Structured Query Language (SQL) in order to develop queries that allow the user to view, manipulate, add, or delete data in the database.

SQL is incredibly easy to understand and is not case-sensitive unlike Python or JavaScript. SQL allows you to write a select statement as “SELECT”, “SeLEcT”, or “select” and the query will still be executed. However, since SQL is not case-sensitive, it can be hard to read queries when they need to be analyzed. Guidelines are created in order to remediate this issue. Whenever using clauses, they need to be capitalized. For example, the select clause must look like this, “SELECT”. Another guideline is that every nested statement must begin with a new level of indentation.

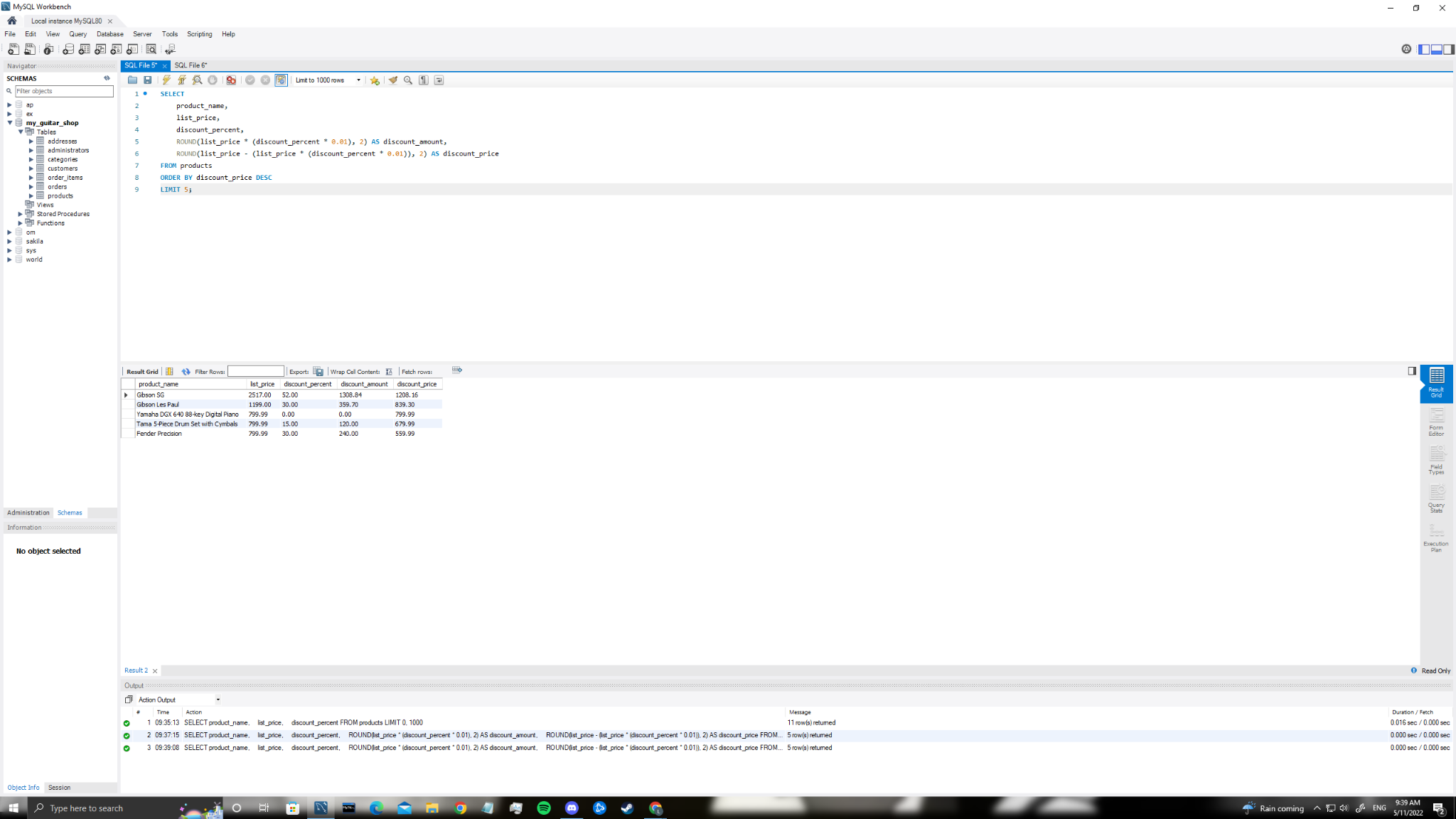
There are four clauses that are used the most within DBMSs. They are the SELECT, INSERT INTO, UPDATE, and DELETE clauses. The SELECT clause is used to manipulate data in one or more tables and display the results based on the parameters set within the SELECT clause. The INSERT INTO clause is used to add new data to the table. For example, if a new employee joins an organization, they will need to add that employee's information to the database. This can be achieved using the INSERT INTO clause. The UPDATE clause is used to modify data within a table. For example, if an employee changes their email address, the UPDATE clause can be used to update the email address to the new one. Lastly, the DELETE clause is used to delete data from a table. For example, if an employee leaves the organization, their employee information will need to be deleted from the database.

Not so much with the SELECT clause, but with the other three, caution must be advised. Accidental changes can be made using the INSERT INTO, UPDATE, or DELETE clauses. For example, when deleting an employee from the database, another employee may be accidentally deleted from the database. It is important to make a backup of the database before any of these three clauses are executed. If any data is accidentally updated or deleted, the backup can be restored, therefore, no data is permanently lost.

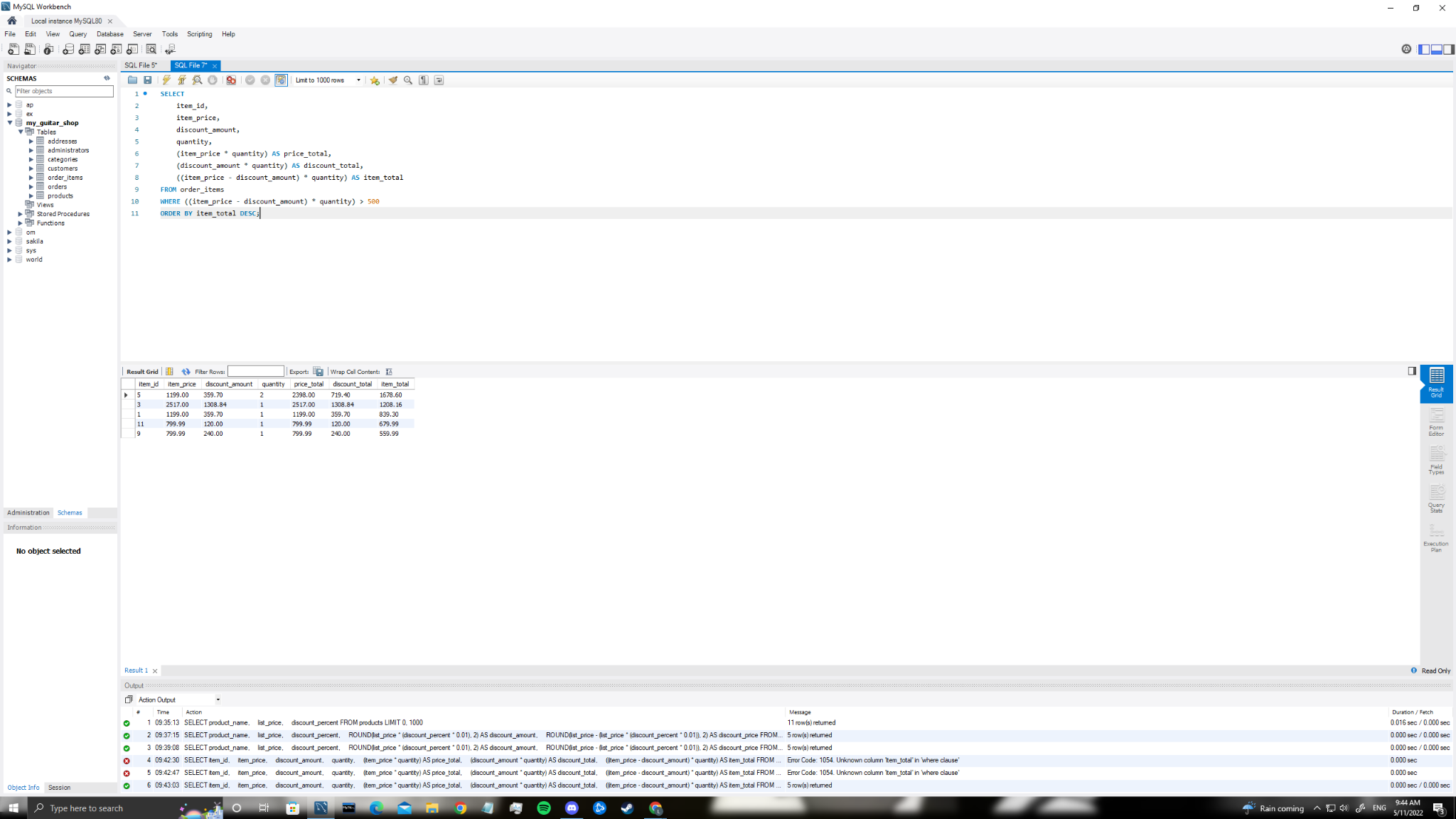
Making backups before executing these statements will ensure more effective database management because the chance of trying to manually recover data lost is much lower. The creation of roles and users for those roles will allow for more efficient database management. Instead of everyone having root access, users will be assigned roles with the least amount of privileges in order to do their job. Not only is this a more efficient approach because it prevents users from accidentally deleting or modifying data on the database, it increases the security of the database. If there are more users that have elevated privileges, it gives attackers more user accounts to utilize in order to steal data from a database. If the principle of least privilege is implemented, attackers will not have the elevated privileges if they breach a user account.

**Part 2: Queries**

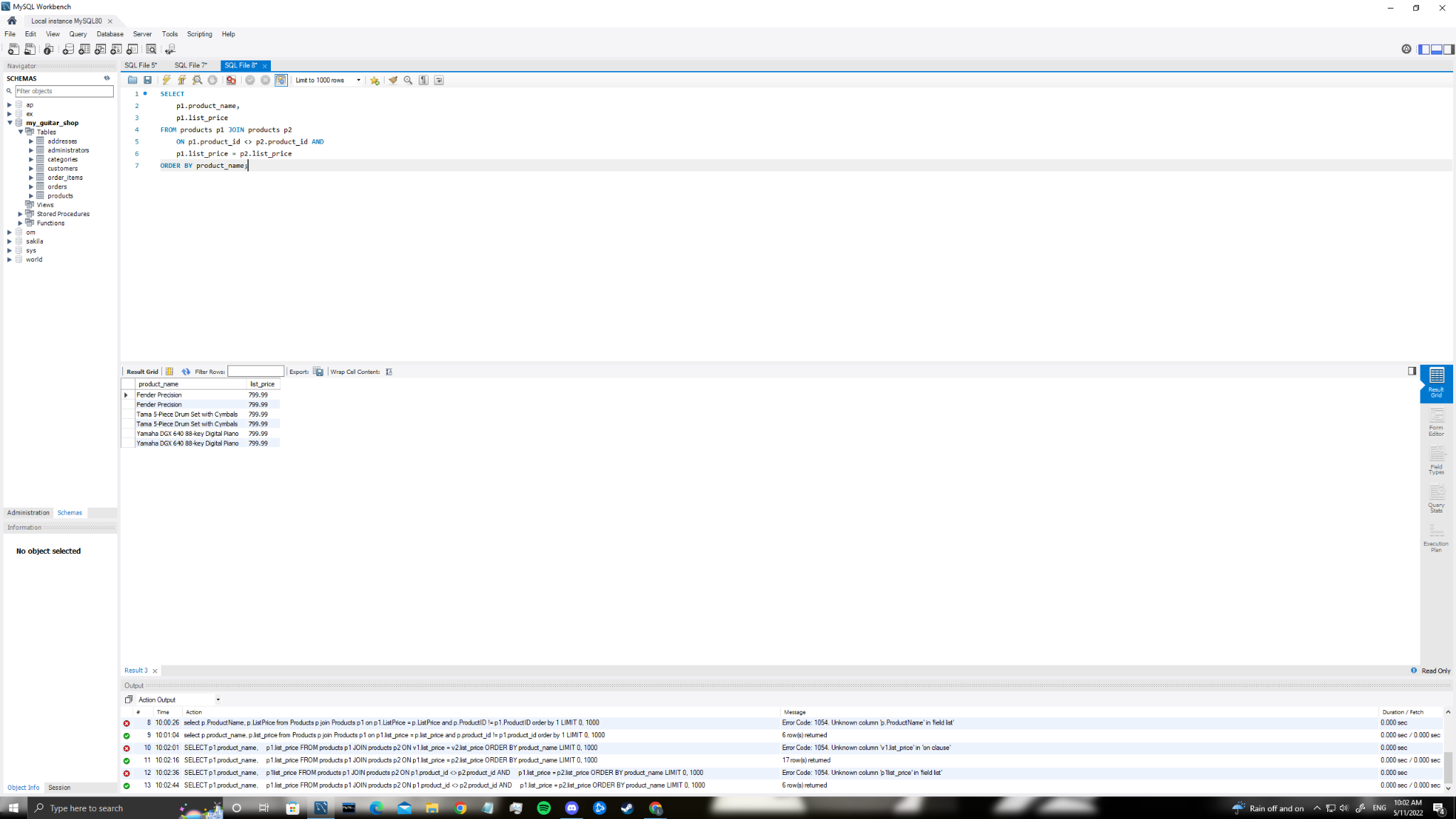
**Step 1**

****

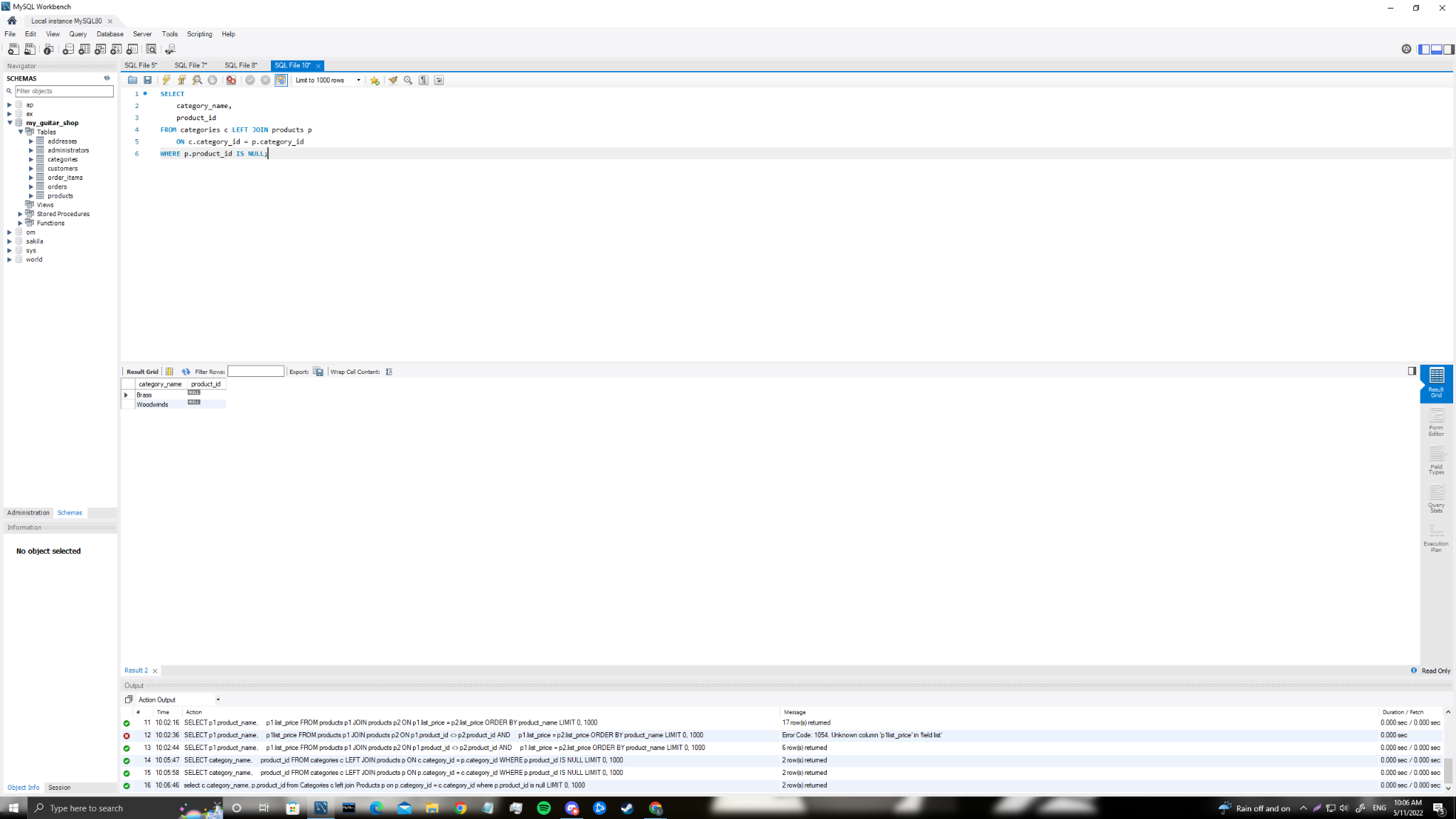
**Step 2**



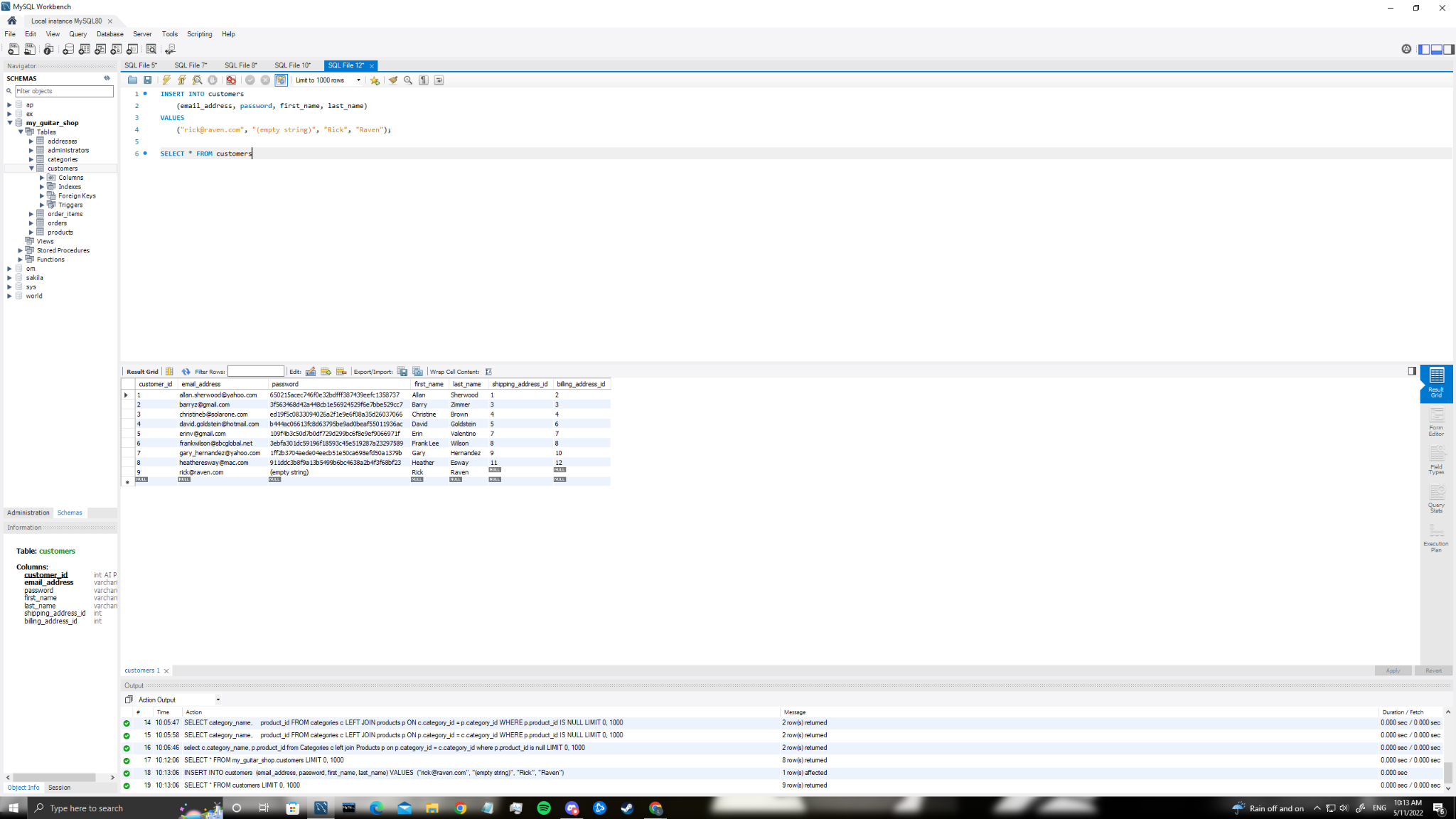
**Step 3**

****

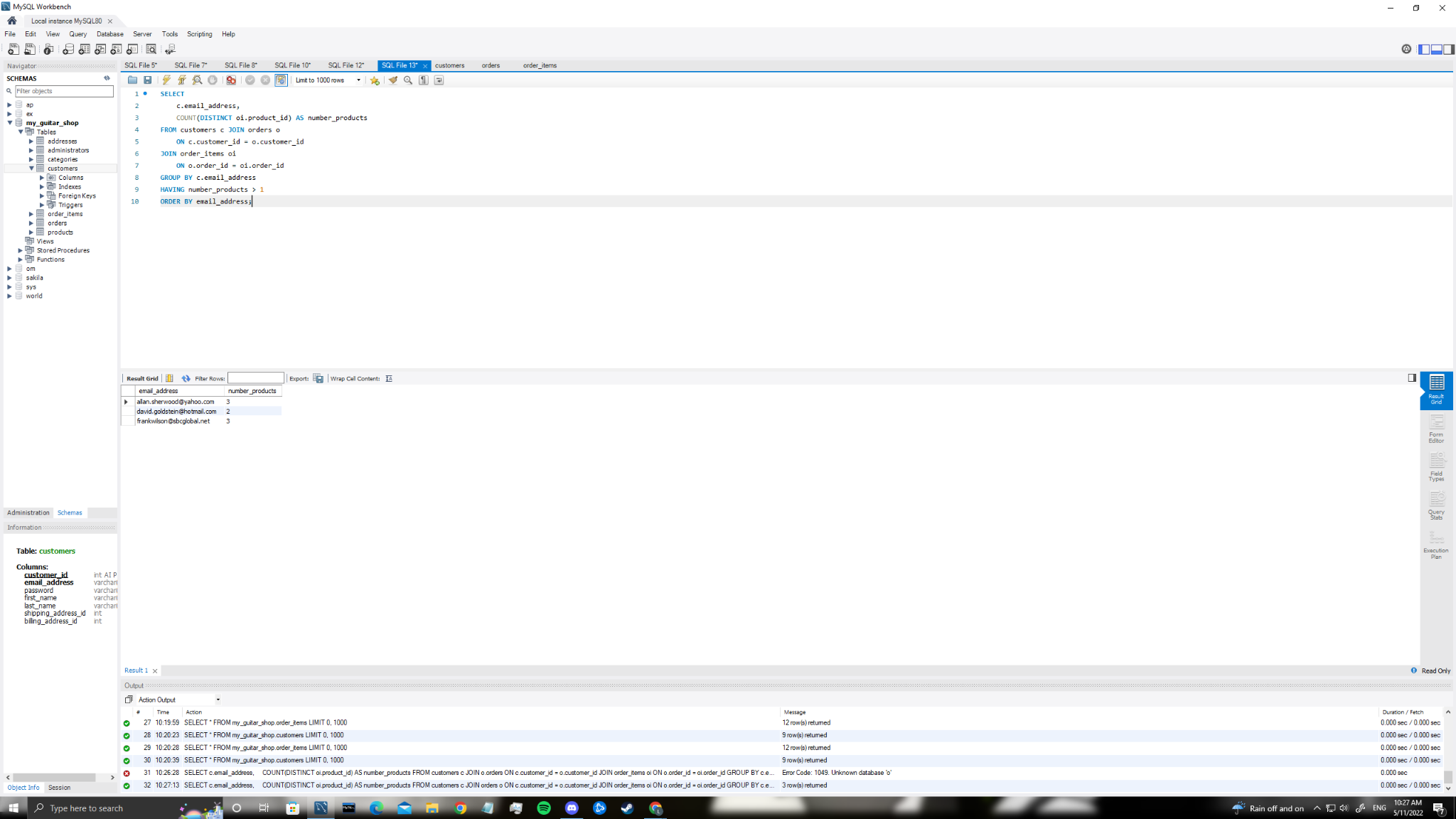
**Step 4**

****

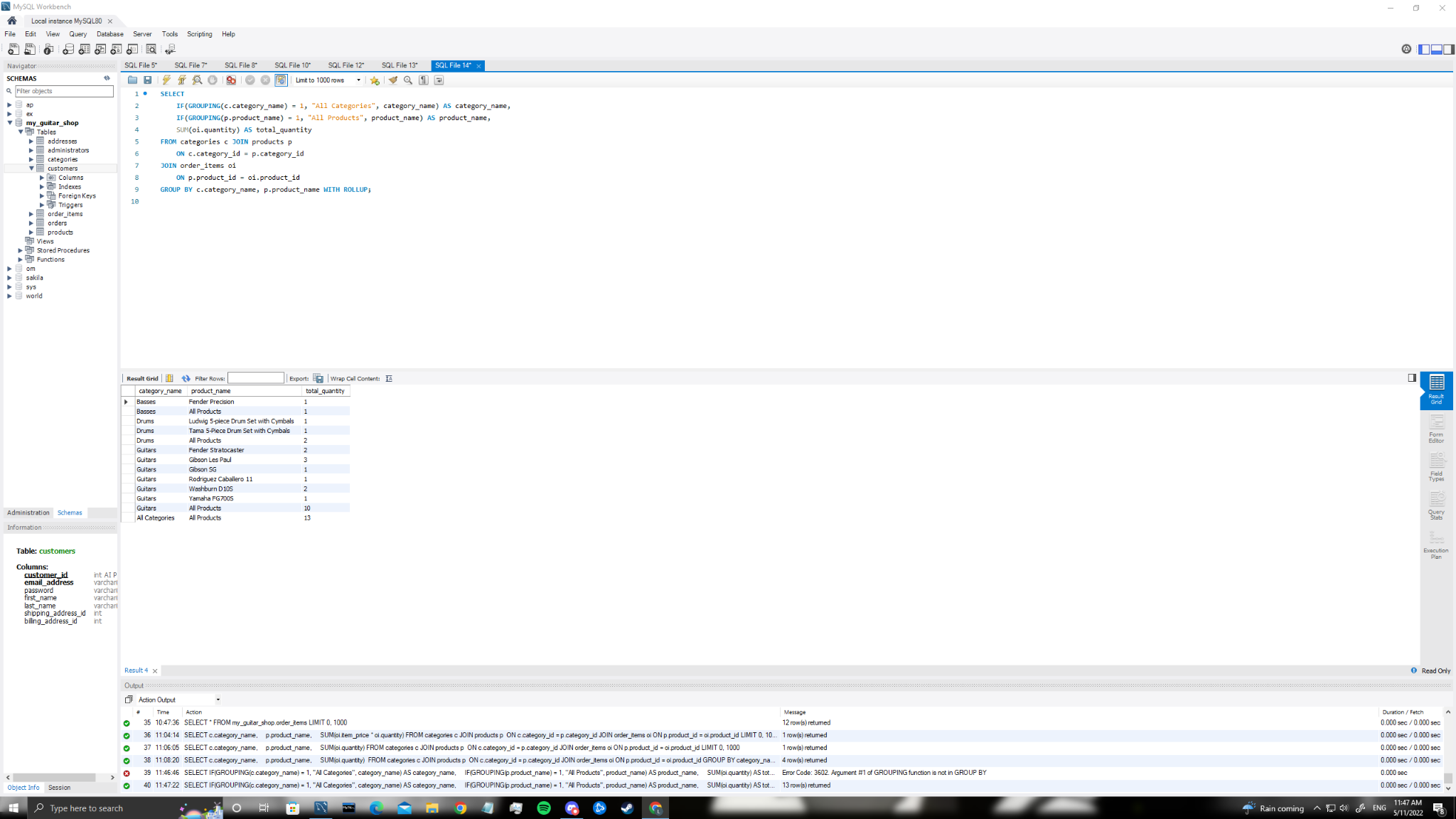
**Step 5**

****

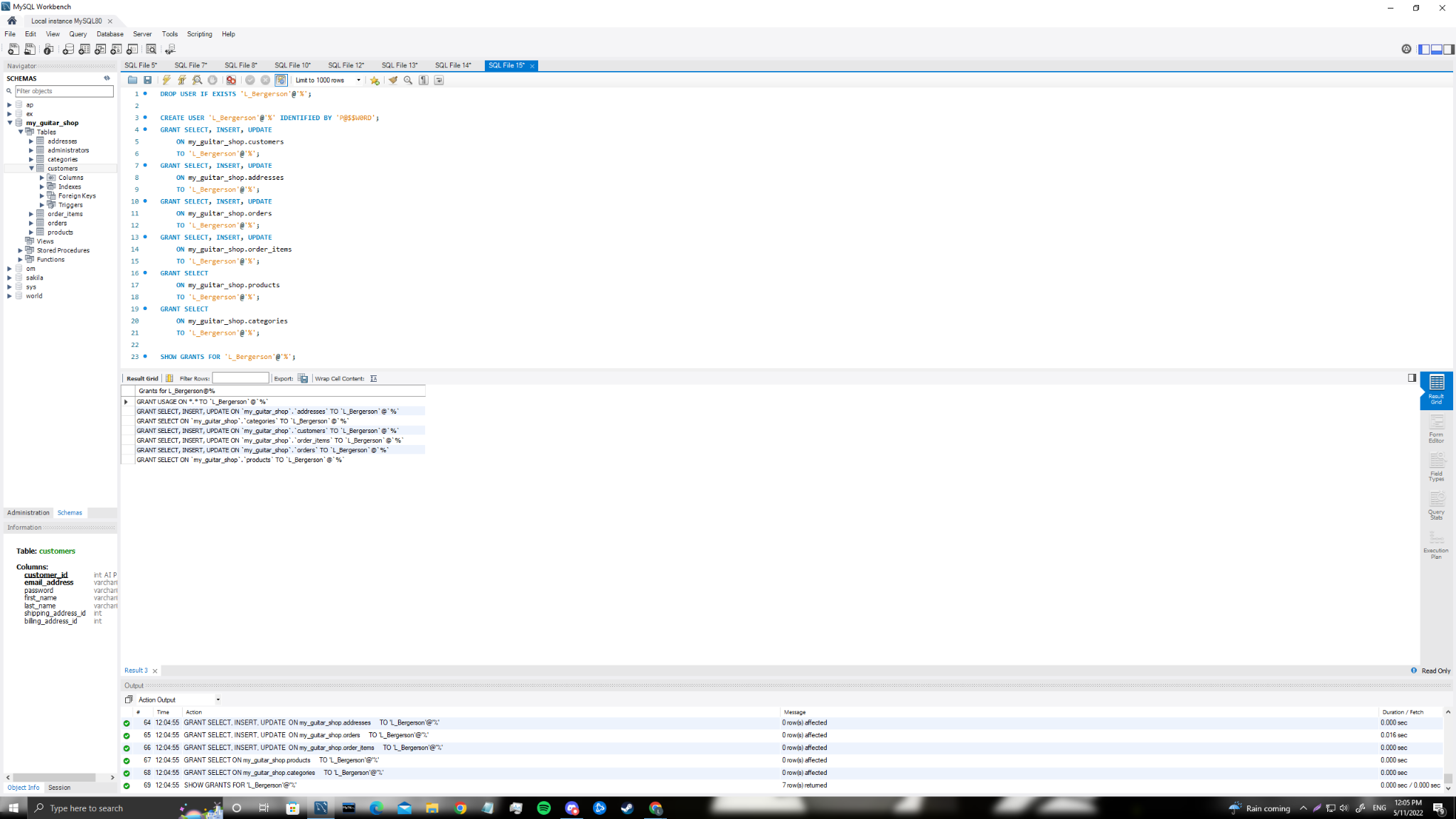
**Step 6**

****

**Step 7**

****

**Step 8**

****