CST8215 – Lab 8 Submission

Student Number: \_\_\_\_\_\_41019241\_\_\_\_\_\_

Student Name: \_\_\_\_\_\_Adam Di Cioccio\_\_\_\_\_\_\_\_\_\_\_\_

Section Number: \_\_321\_\_ (e.g. 322, 323, 363)

Date: \_\_November 16, 2020\_\_\_

Procedure

Where you see *Provide screenshot here* you are being asked to take the screenshot from PG Admin (both the top pane where you have typed in the SQL statement and the bottom pane showing the results of the SQL statement).

A table joined with itself is a SELF JOIN. It happens in a unary relationship, when a table with a foreign key references the primary key in the same table.

Think of a JOIN ON two tables which are same; each row of one table is combined with each row of the other table. There is no explicit statement for a SELF JOIN.

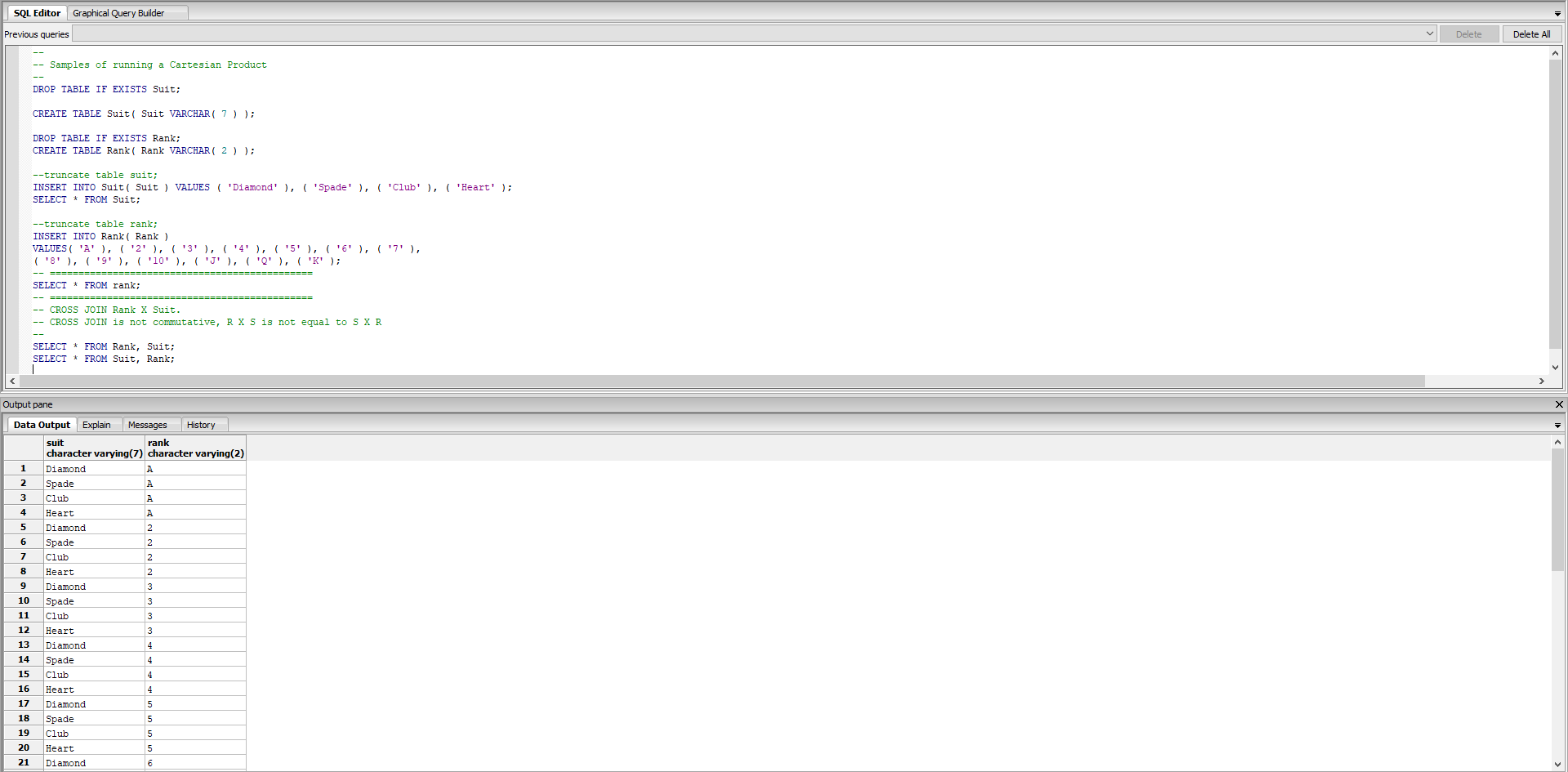
Examples of SELF JOIN: Consider an employee table, with EmployeeID, Name and ManagerID. Most employees will have a manager whose ID will be associated with the employee record.

Query 1

Run the script Deck.sql.

Observe the output of a CARTESIAN PRODUCT, also known as CROSS JOIN.

*Provide screenshot here of the query statements*



Query 2

Run the scripts SELF JOIN DDL, SELF JOIN DML and SELF JOIN Query.

Observe how the Manager’s data, (i.e. ManagerID needs to be inserted first and then the employee records). Alternatively, ManagerID can be updated later.

Observe the two ways in which a constraint can be implemented.

**First** at the time of creating the table **or** by later using ALTER TABLE clause.

Execute the ‘SELF JOIN’ query and confirm that it returns the correct ‘Employee to Manager’ results.

*Provide screenshot here of the query statement*



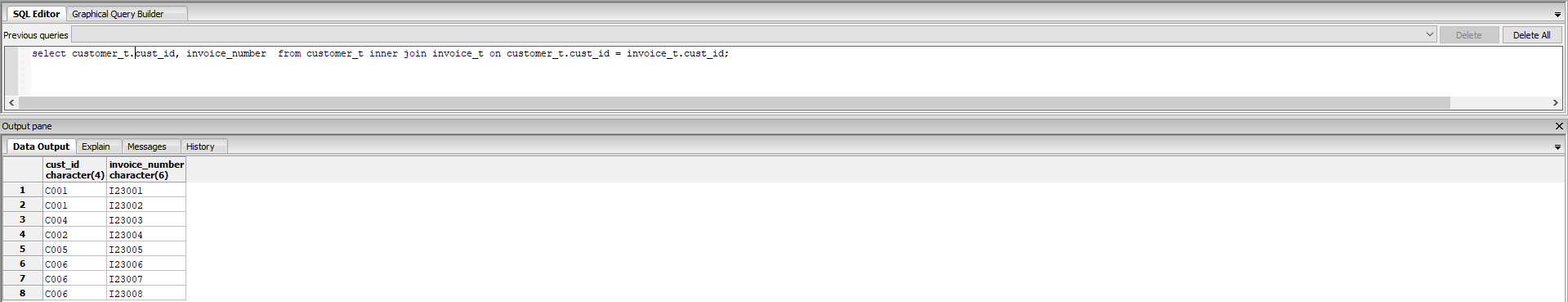
**Additional ‘JOIN’ practice questions**

The following queries will use your **Inventory** Database created in **Lab 4**.

It is recommended to have the ERD of Lab 4 at hand while answer this lab so that learners may understand the table relationships. Before beginning, it is suggested to study the ERD diagram of Inventory database and identify key relationship and column names.

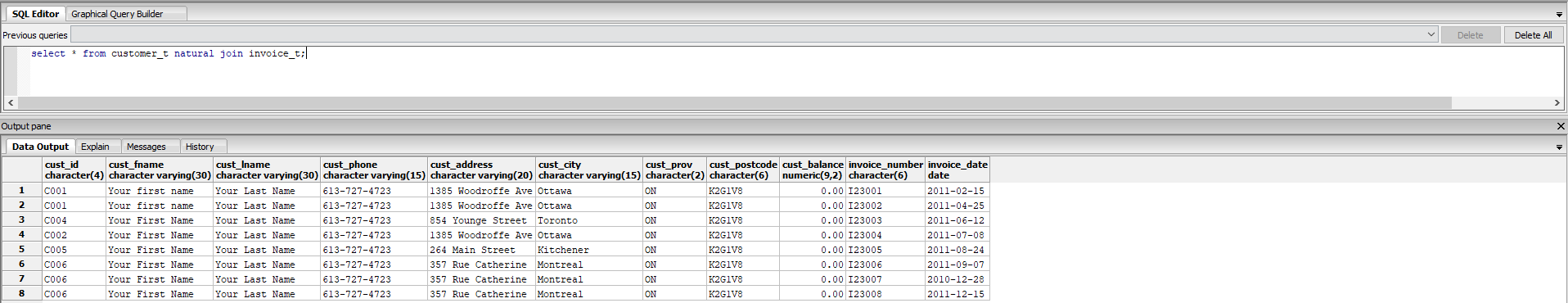
**Query 1: List all customer along with their invoice details using an inner join**

*Provide screenshot here of the query statement*



**Query 2: Edit Query 1 to use natural join instead of inner join. Discuss the benefits of using natural join over inner join.**

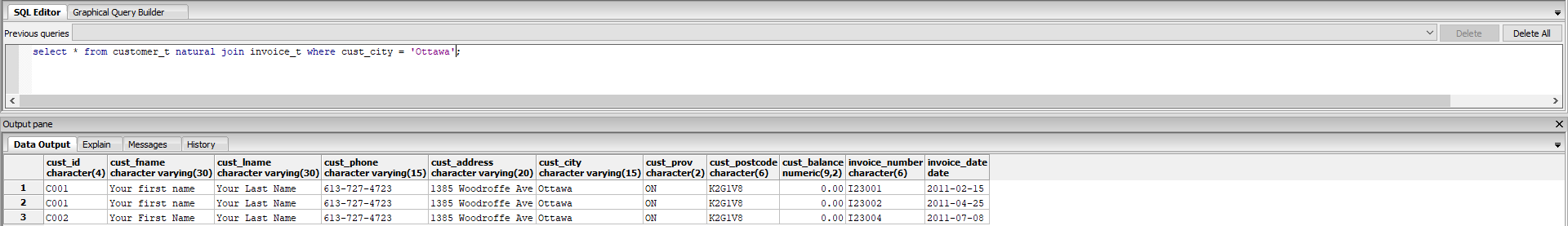
*Provide screenshot here of the query statement*



**Query 3: Edit Query2 to display only customers from Ottawa.**

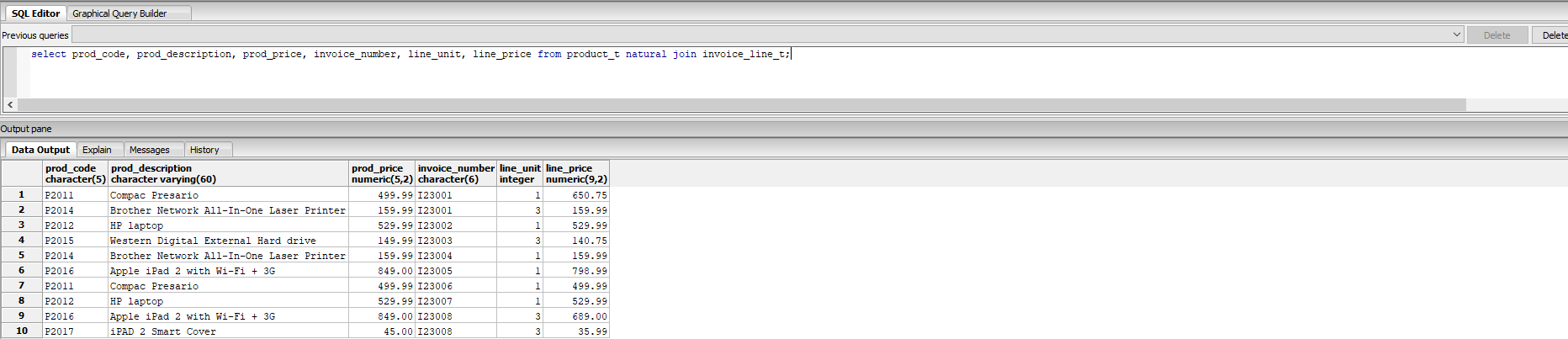
**Note: If you do not have customers in Ottawa, use any other city.**

*Provide screenshot here of the query statement*



**Query 4: List prod\_code, prod\_description, prod\_price along with invoice\_number, line\_unit and line\_price information. Use either inner join or natural join.**

*Provide screenshot here of the query statement*



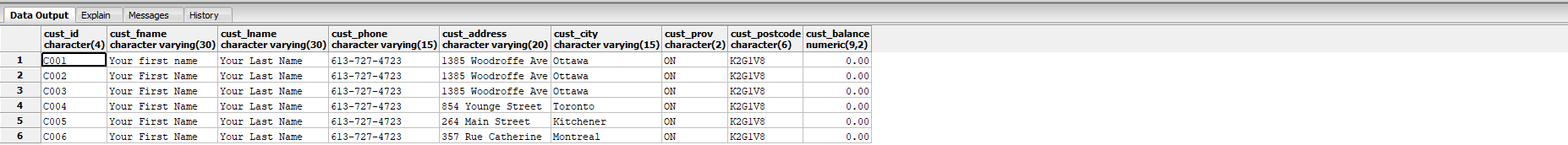
**Query 5: List cust\_id, cust\_fname, cust\_lname and invoice\_number for only those customers who have not placed any order i.e. Customers who do not have a record in invoice\_t table. Hint: Use Outer Joins with WHERE is null clause.**

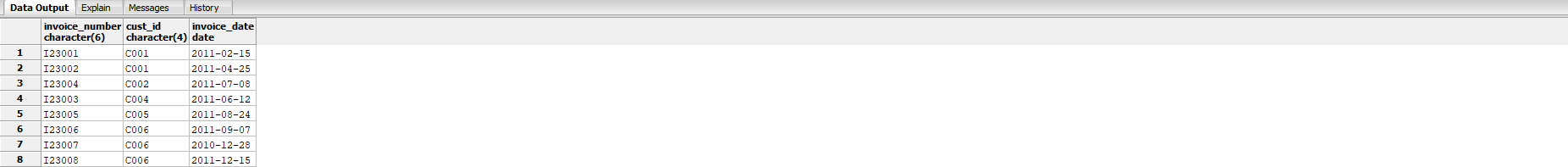
SELECT \* FROM customer\_t;

SELECT \* FROM invoice\_t order by cust\_id;

Note: Using the above two queries, identify which customer does not have a record in invoice.

*Provide screenshot here of the query statement*

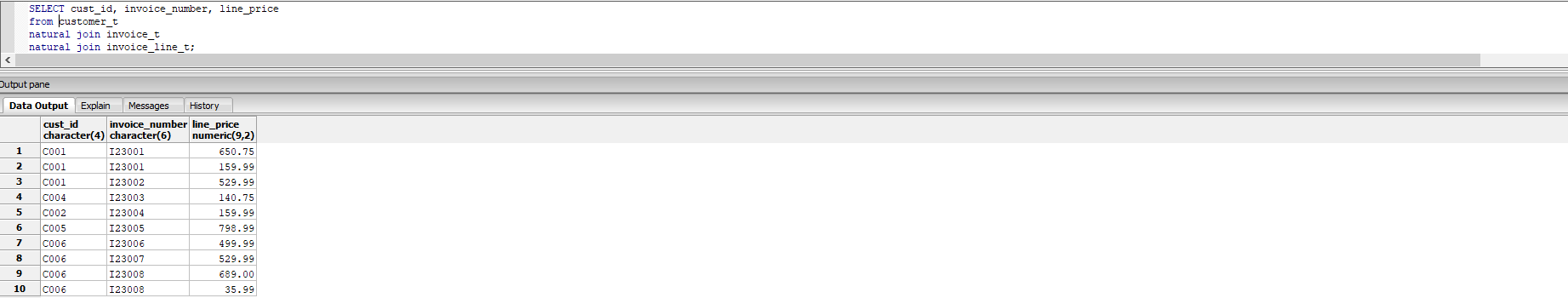




**Query 5: List Customer details along with invoice\_number and line\_price.**

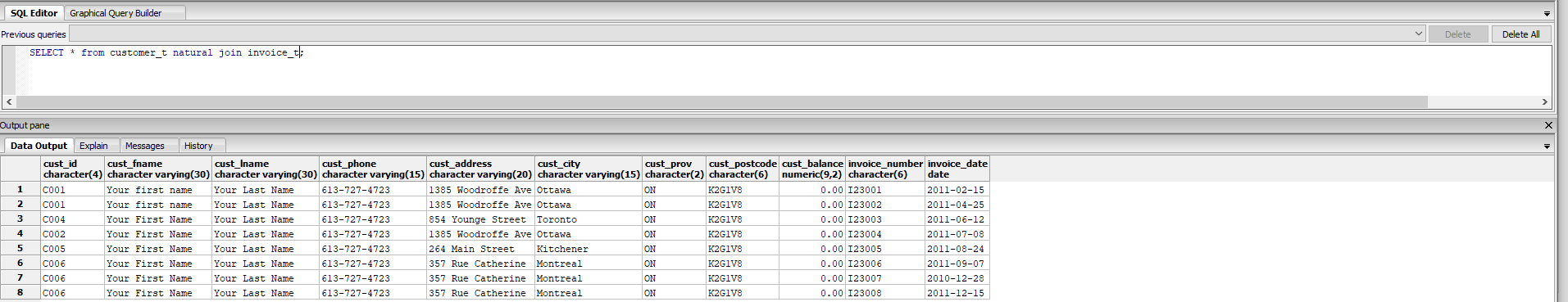
**Hint: This will require joining customer\_t, invoice\_t and invoice\_line\_t tables.**

*Provide screenshot here of the query statement*



**Query 6: Find and list which customer has ordered which product.** Hint: This will require you to identify relationship from Customer\_T to Invoice\_T, Invoice\_T to Invoice\_Line\_T and Invoice\_Line\_T to Product\_T.

*Provide screenshot here of the query statement*



**Query 7: Edit Query 6 to include cust\_city column and then display customer and product details for only customer in Ottawa. Hint: Use Query 3 as reference.**

*Provide screenshot here of the query statement*

