**Assignment 2 Report**

Matthew Antonis

200373088

COMP2003 Relational Database

Professor Lyle Wood

16/06/2023

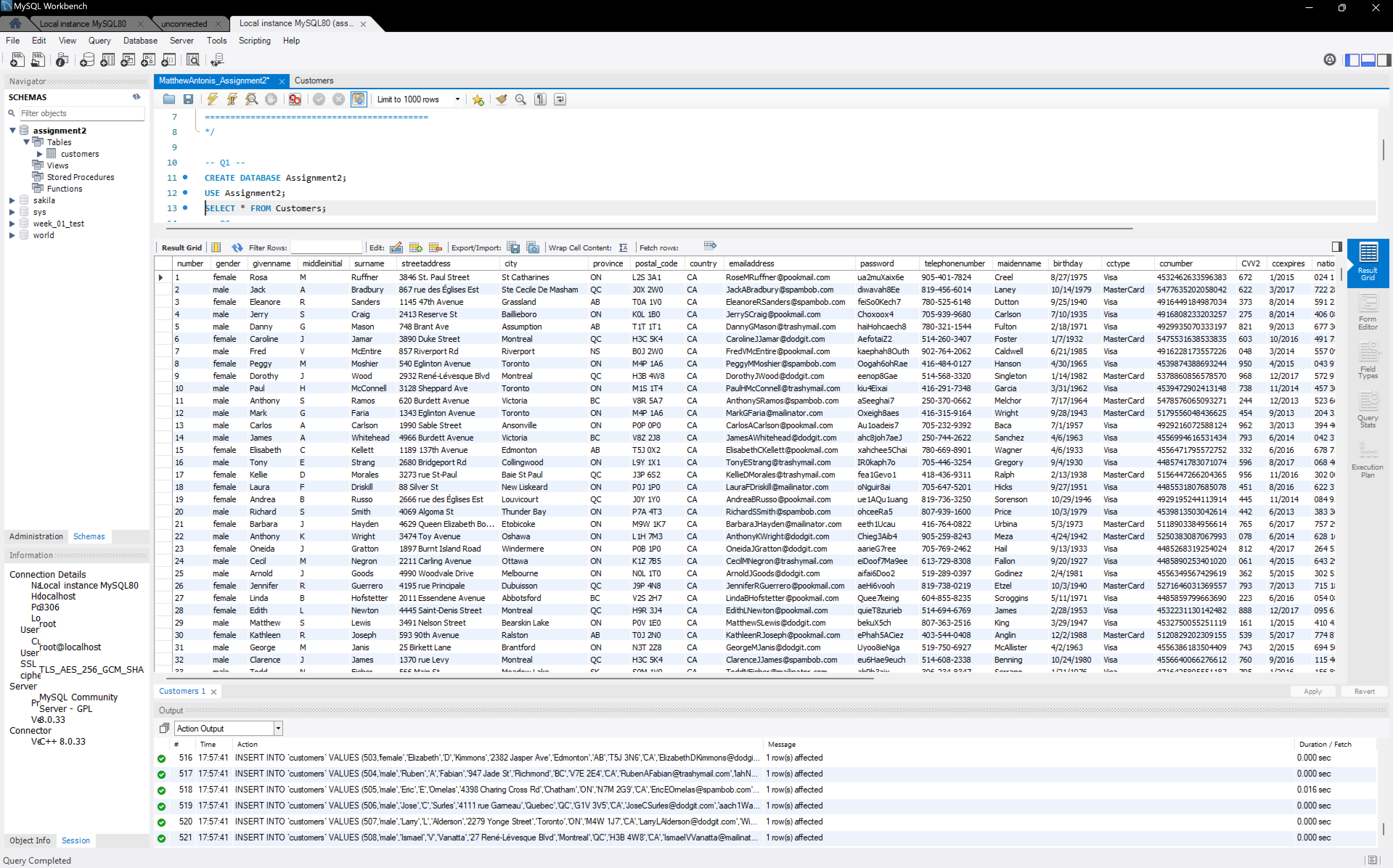
1. Make a database and name it ‘Assignment2’ and add the data from Customers.sql to your

Database.

**Script:**

* CREATE DATABASE Assignment2;
* USE Assignment2;
* SELECT \* FROM customers;

**Results:**



**Comments:**

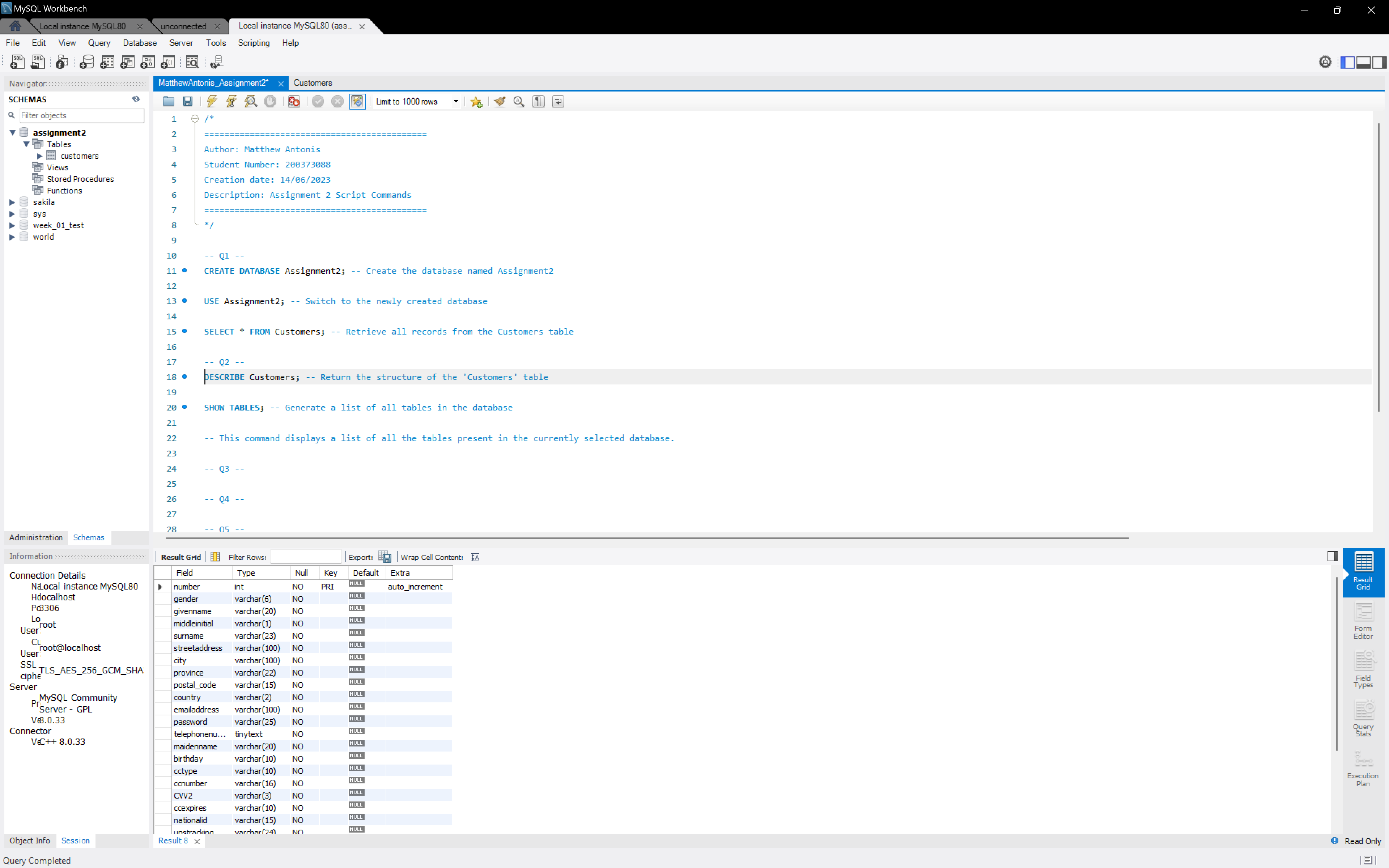
* Used commands “CREATE DATABASE” and “USE” first
* Copied over Customer.sql to mysql and ran the script
* Once the table was added I used the "SELECT \* FROM” Command

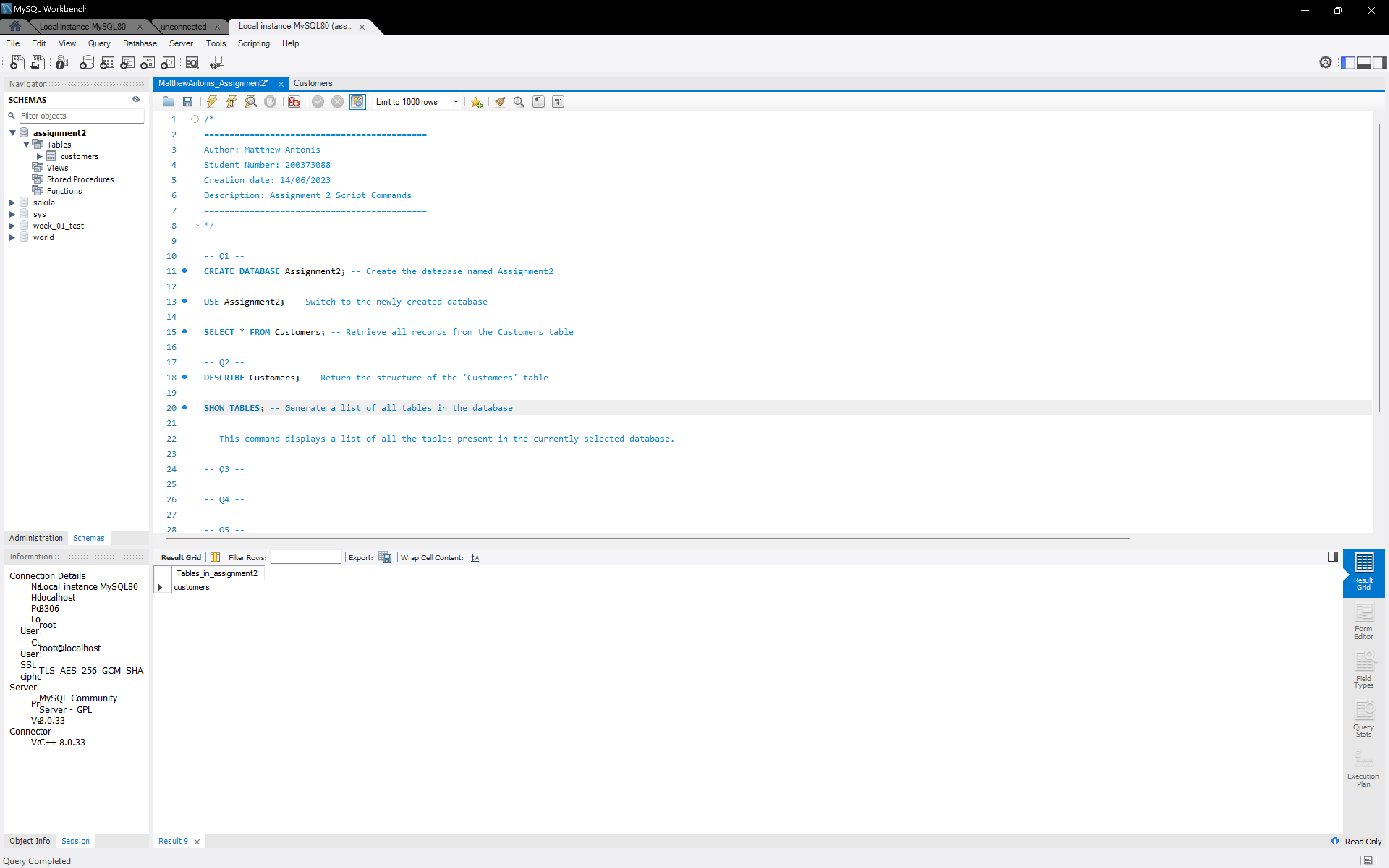
2. Use a single line SQL command to return the structure of the table from Q1. In your report, tell me what command you would use to generate a list of all the tables in your database.

**Script:**

* DESCRIBE Customers;
* SHOW TABLES;

**Results:**





**Comments:**

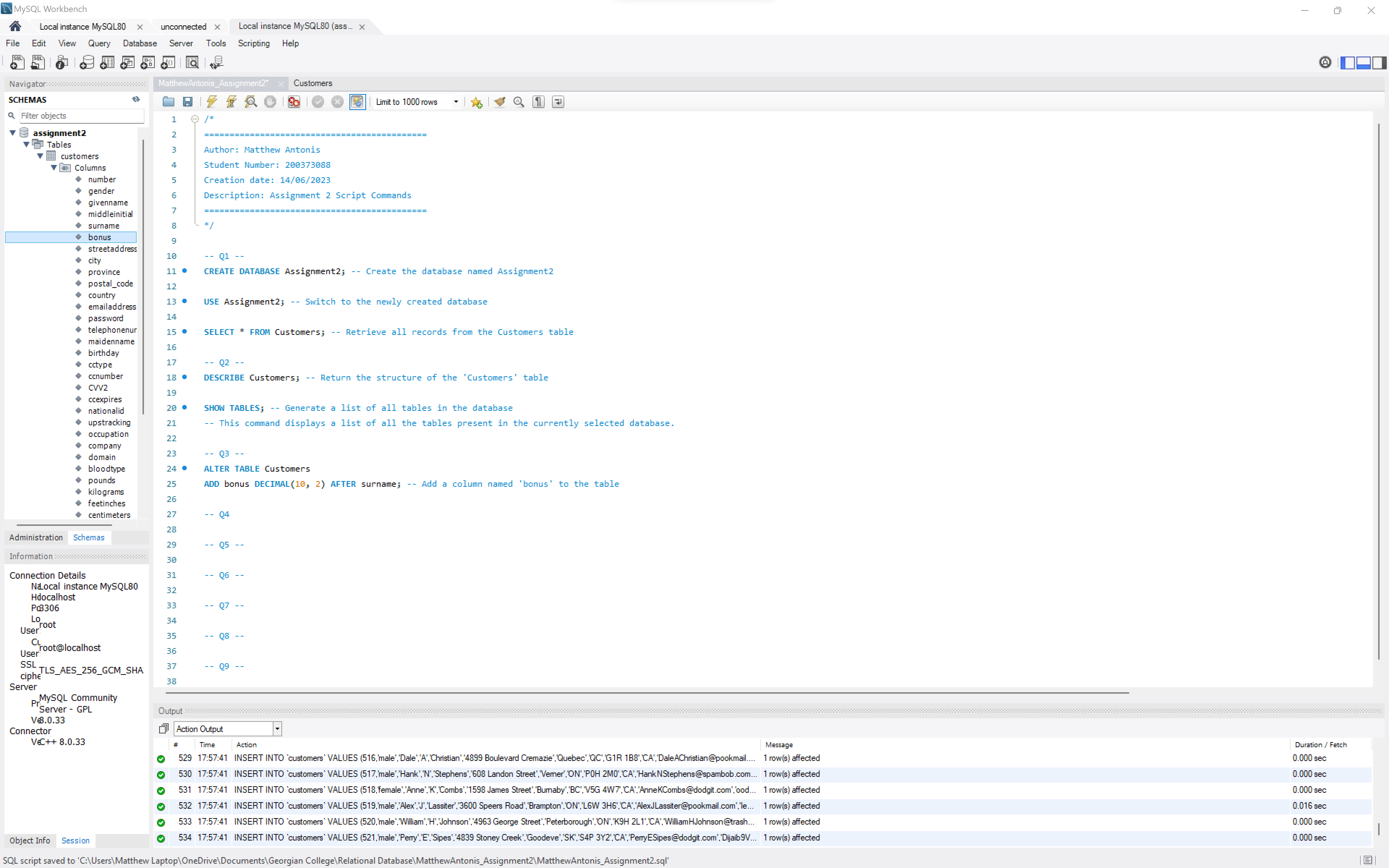
* I used the “SHOW TABLES” command because it displays a list of all the tables present in the currently selected database.

3. Add a column named bonus to your table. Place it after surname. Enable the users to store bonuses with values such as: 10,999.99 or 2.00 by choosing the proper data type for bonus. In your report, mention the data type you chose and another that you considered but chose not to use.

**Script:**

* ALTER TABLE Customers
* ADD bonus DECIMAL(10, 2) AFTER surname;

**Results:**



**Comments:**

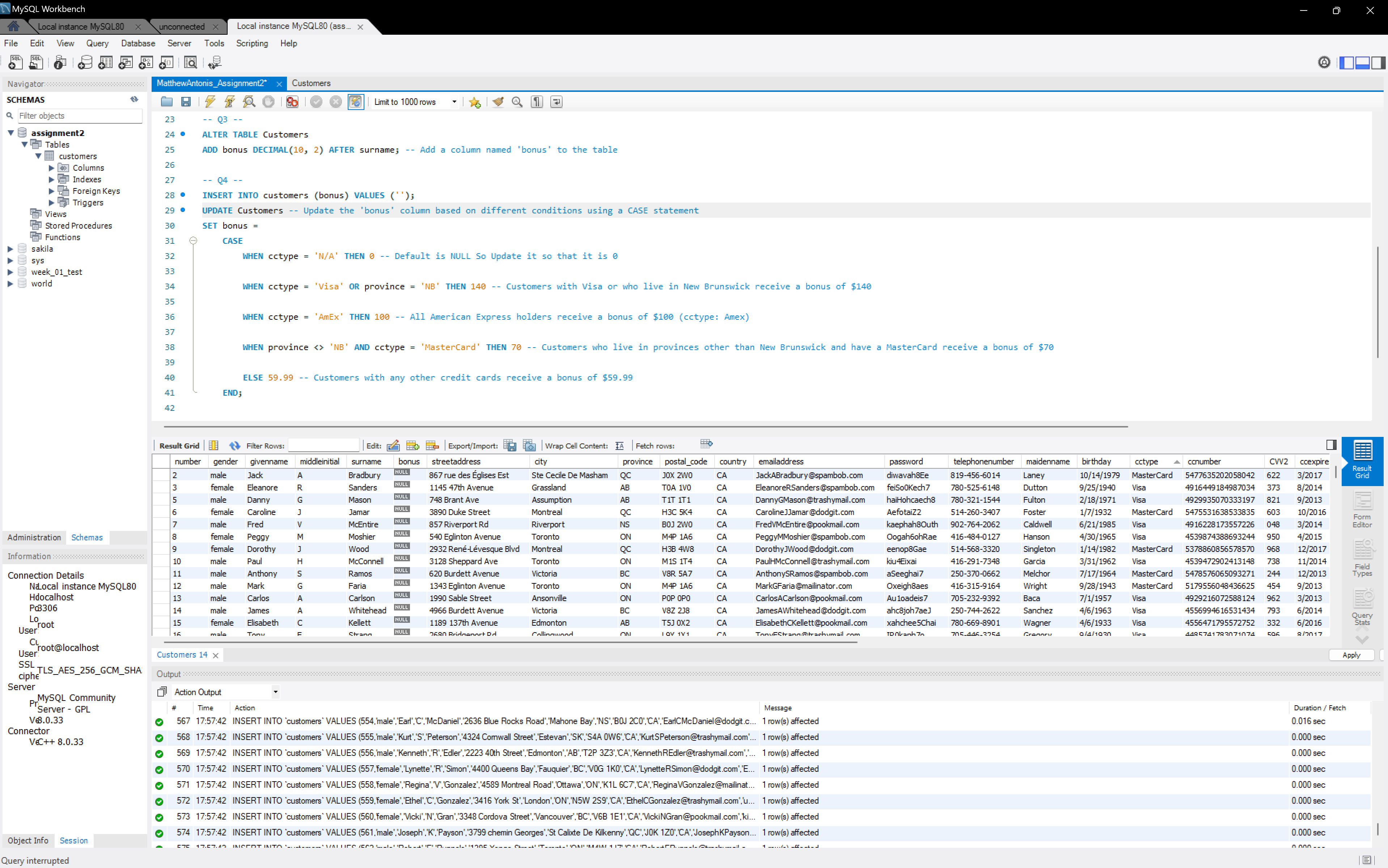
* DECIMAL data type is used with a precision of 10 and a scale of 2. This allows the 'bonus' column to store values up to 10 digits with 2 decimal places, accommodating the desired values like '10,999.99' or '2.00'.

4. Using a CASE command fill in the data for the bonus column. • Non-credit card holders will receive a bonus of zero (cctype: N/A). • Customers who hold a Visa or who live in New Brunswick the bonus will be $140 • All American Express holders will get $100 (cctype: Amex) • Customers who live in any province other than New Brunswick and have a MasterCard will get $70 as bonus. • Then, customers with any other credit cards will get $59.99 as bonus. In your report, explain the considerations that were made in constructing this statement.

**Script:**

* UPDATE Customers SET bonus =
  + CASE
    - WHEN cctype = 'N/A' THEN 0
    - WHEN cctype = 'Visa' OR province = 'NB' THEN 140
    - WHEN cctype = 'AmEx' THEN 100
    - WHEN province <> 'NB' AND cctype = 'MasterCard' THEN 70 ELSE 59.99
  + END;

**Results:**



**Comments:**

Considerations in constructing this statement:

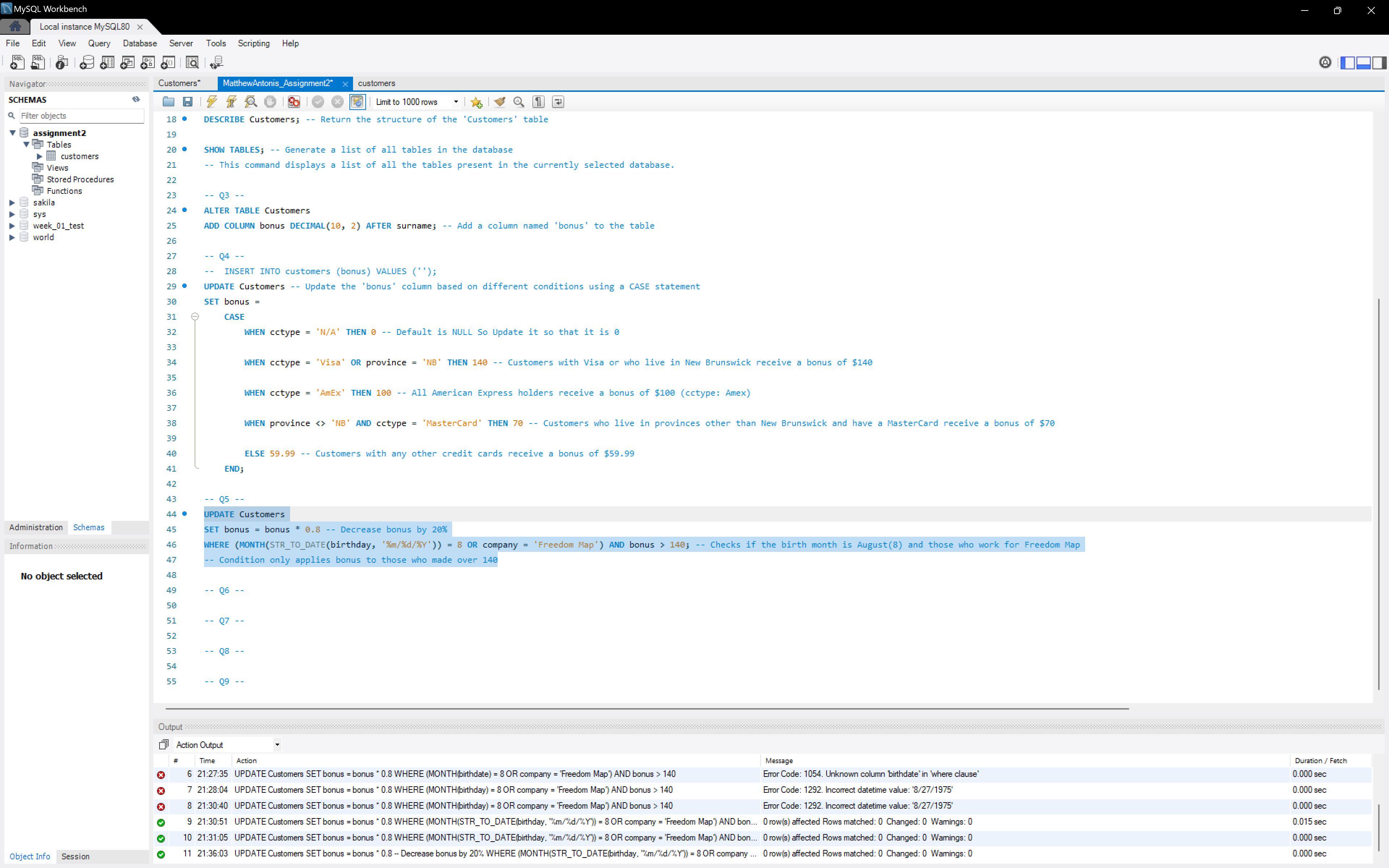
* Condition Order: The conditions are evaluated in the order they appear. It is important to consider the order of conditions to ensure that the most specific conditions are checked first. In this case, we check for 'N/A' cctype before checking for specific card types to cover non-credit card holders.
* Multiple Conditions: The CASE statement allows us to specify multiple conditions and assign different values based on each condition. Here, we consider various combinations of cctype and province to determine the bonus.
* Default Value: The final ELSE statement acts as a default value for customers who do not meet any of the specified conditions. In this case, they receive a bonus of $59.99.
* Data Integrity: It is important to ensure that the data in the 'cctype' and 'province' columns are accurate and consistent. The conditions in the CASE statement rely on the correctness of these values to determine the appropriate bonus.

5. Let’s decrease some of the bonuses. Please decrease the bonus by 20% for customers who are born in August or work for the company Freedom Map. Further, please only apply this decrease to the people who were getting a bonus of more than $140.

**Script:**

* UPDATE Customers
* SET bonus = bonus \* 0.8
  + WHERE (MONTH(STR\_TO\_DATE(birthday, '%m/%d/%Y')) = 8 OR company = 'Freedom Map') AND bonus > 140;

**Results:**



**Comments:**

* Converted from the 'birthday' string using STR\_TO\_DATE

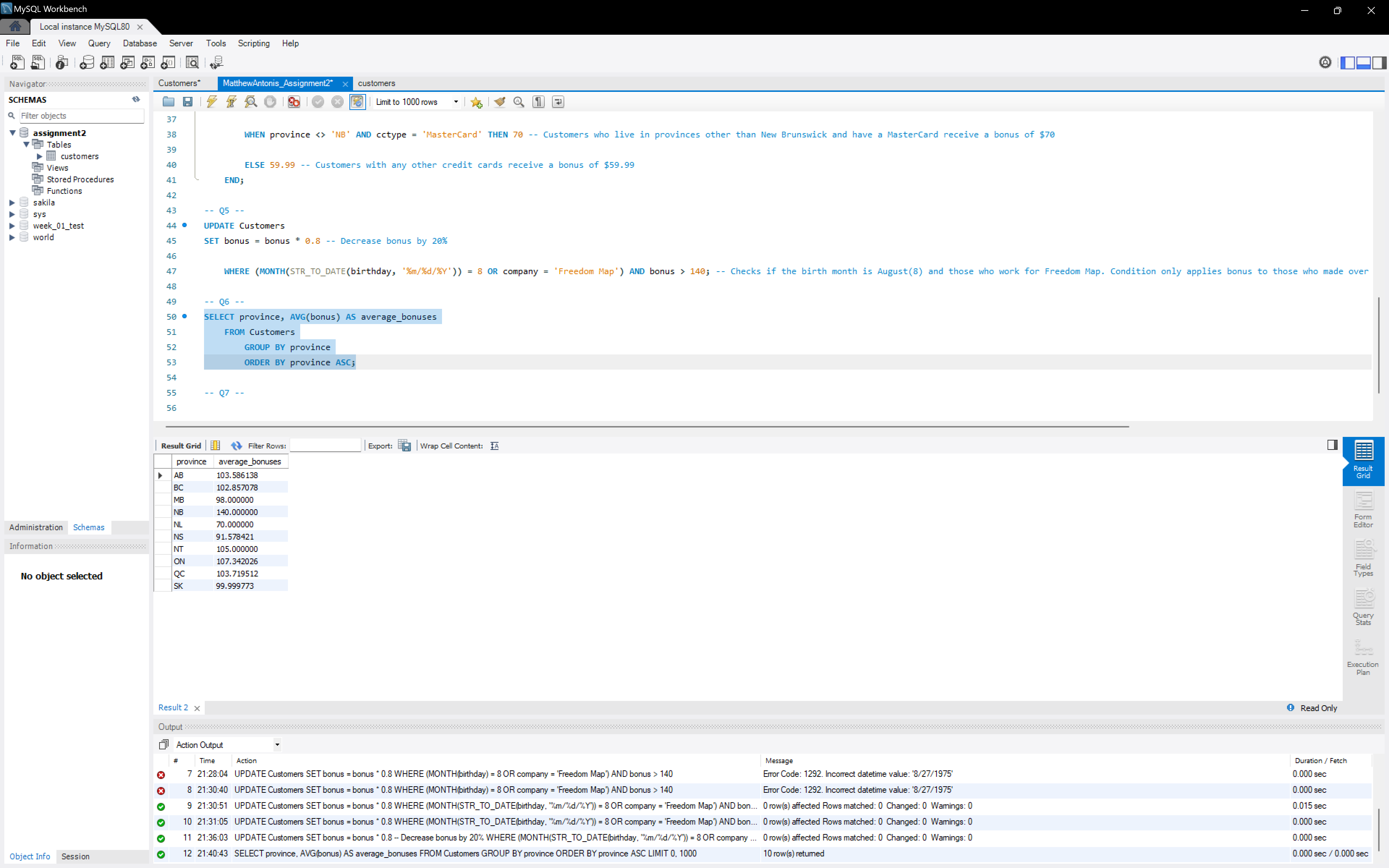
6. We need to calculate the average dollar value of bonuses being given to customers in each province. • List each province and tabulate the average value of bonus dollars provided to customers in that province and name the new column as average\_bonuses. • Sort the list alphabetically by province.

In your report, explain the considerations that were made in constructing this statement.

**Script:**

* SELECT province, AVG(bonus) AS average\_bonuses
* FROM Customers
  + GROUP BY province
  + ORDER BY province ASC;

**Results:**



**Comments:**

Considerations in constructing this statement:

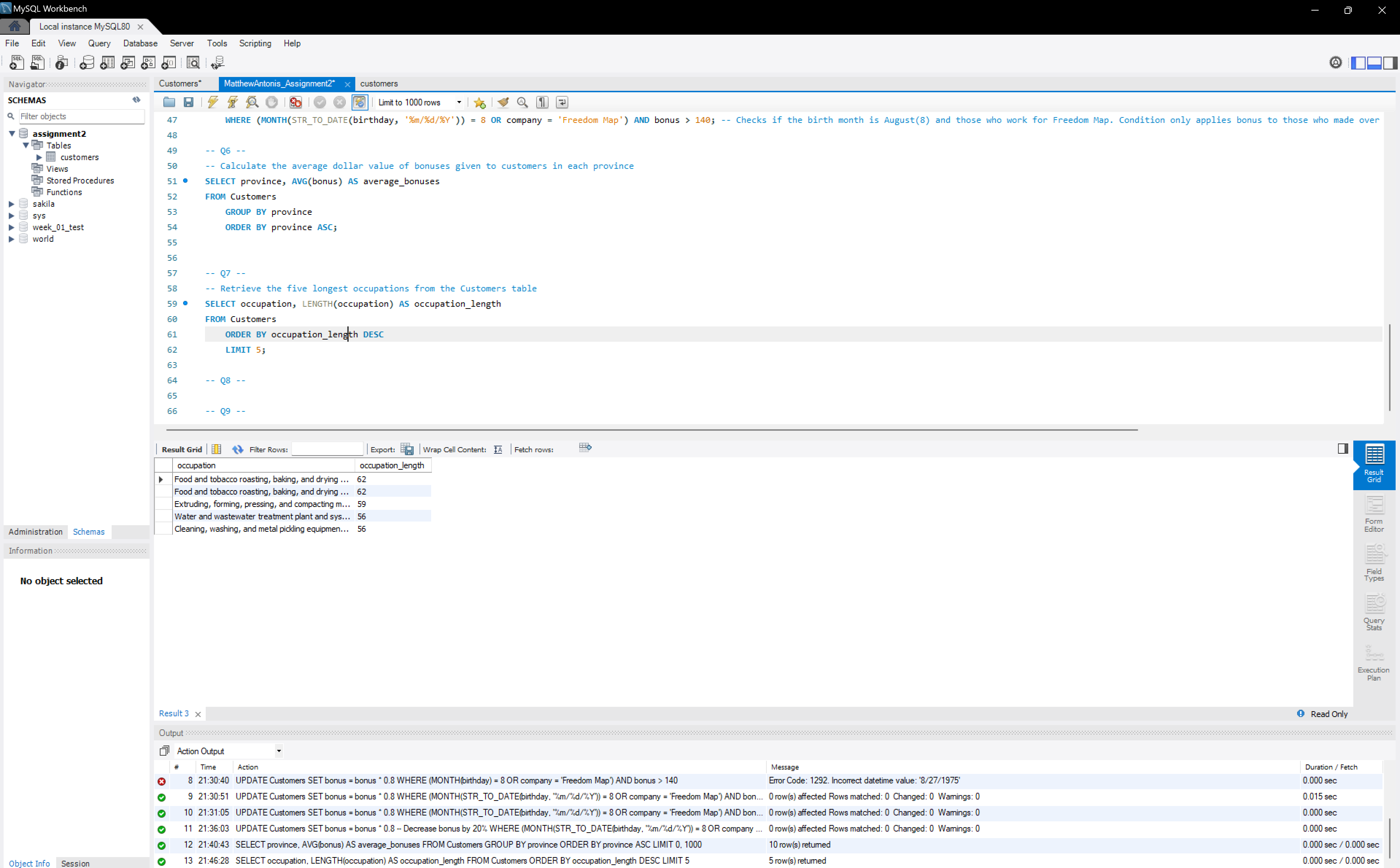
* Column selection: The SELECT clause specifies the 'province' column for identification and the 'bonus' column for calculating the average bonus value.
* Aggregation: The AVG() function is applied to the 'bonus' column to aggregate and calculate the average bonus within each province.
* Grouping: The GROUP BY clause groups the data by the 'province' column, enabling the calculation of average bonus per province.
* Column renaming: The AS keyword renames the calculated average bonus column as 'average\_bonuses' for better readability.
* Sorting: The ORDER BY clause sorts the result set in ascending order based on the 'province' column.

7. Generate a list displaying only the five longest occupations in the customers table. In two columns display the occupation, and the character length of the occupation in a field called occupation\_length. Please have the list sorted from longest to shortest. In your report, explain how you would return occupations 6 through 10 of this same list.

**Script:**

* SELECT occupation, LENGTH(occupation) AS occupation\_length
* FROM Customers
  + ORDER BY occupation\_length DESC
  + LIMIT 5;

**Results:**



**Comments:**

* Change/Add “LIMIT 5 OFFSET 5;”
* The OFFSET 5 clause skips the first five rows, allowing you to retrieve the occupations from the 6th row onward, giving you occupations 6 through 10 of the list.

8. Write a query to show customers’ full names, street address and city.

• Please return their last name in all capital letters, followed by their first name.

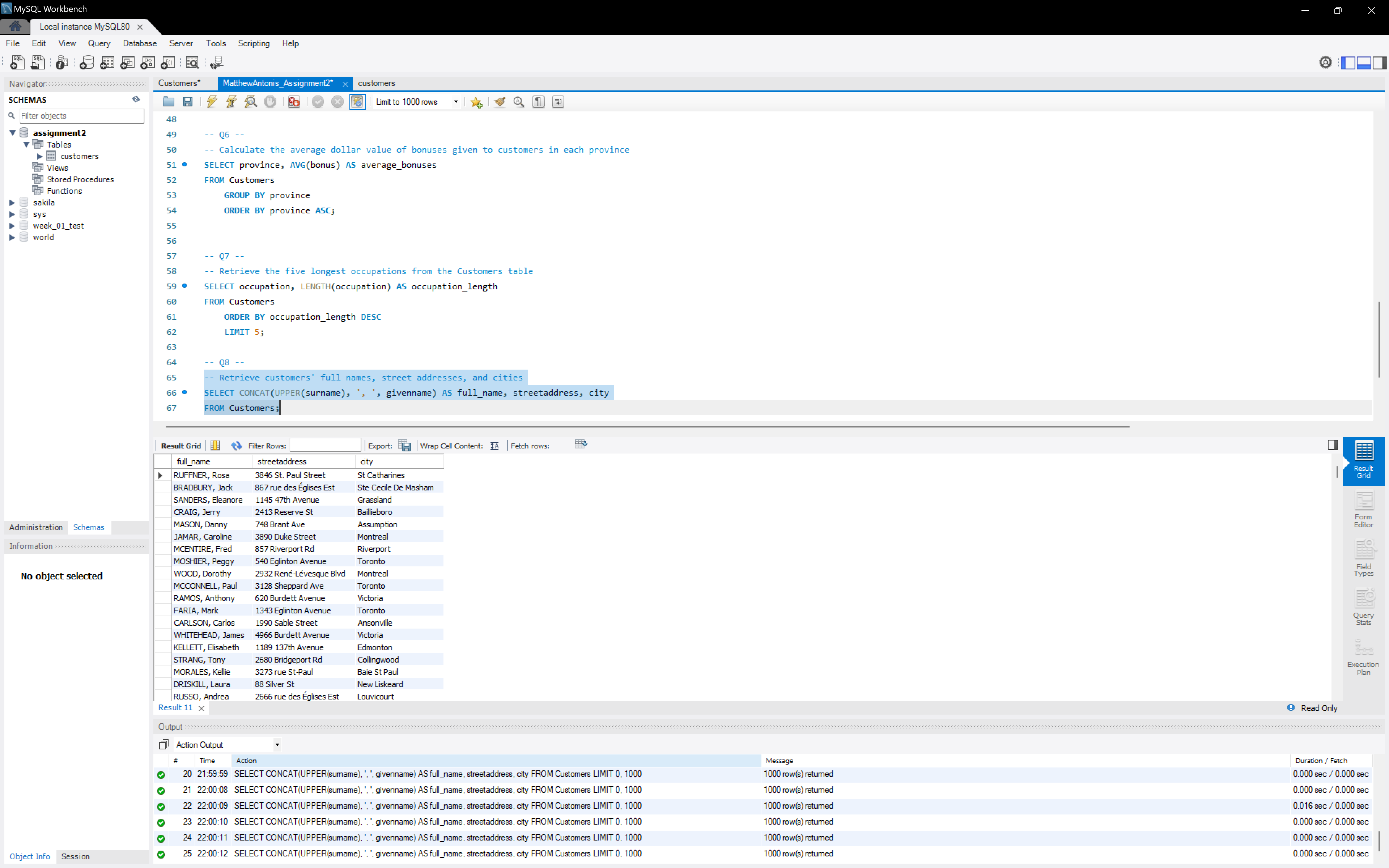
• Please call the name column full\_name and include both their first and last name as shown. full\_name, streetaddress, city

RUFFNER, Rosa 3846 St. Paul Street, St Catharines

**Script:**

* SELECT CONCAT(UPPER(surname), ', ', givenname) AS full\_name, streetaddress, city
* FROM Customers;

**Results:**



**Comments:**

* The CONCAT function is used to combine the last name, first name, and a comma separator. The UPPER function is applied to the last name to convert it to all capital letters.

9. Write a query to show how many customers have each type of credit card. In your result set,

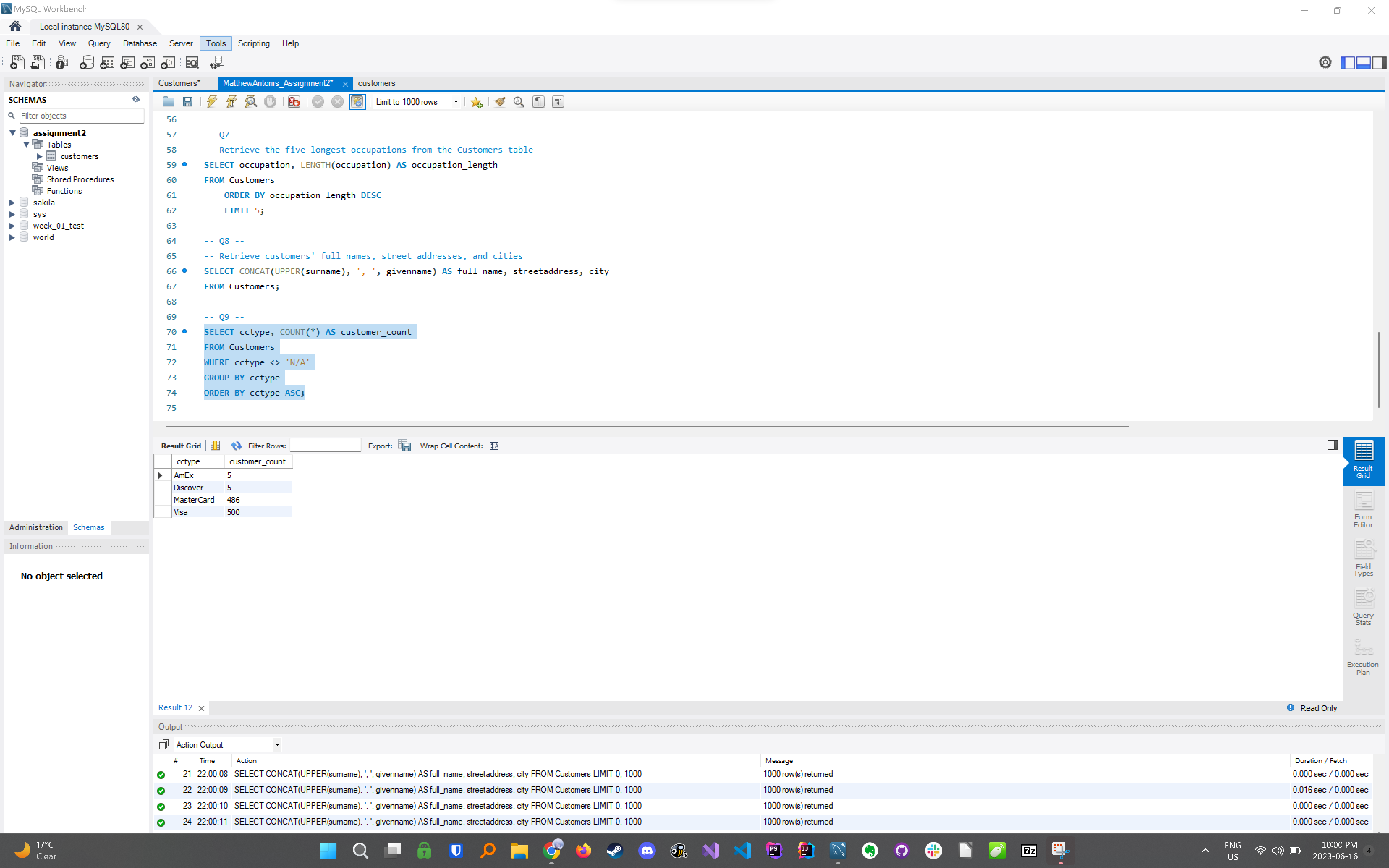
sort this table alphabetically by credit card type and exclude users who do not have a credit

card. I have provided an example table; these numbers are fictitious

**Script:**

* SELECT cctype, COUNT(\*) AS customer\_count
* FROM Customers
  + WHERE cctype <> 'N/A'
    - GROUP BY cctype
    - ORDER BY cctype ASC;

**Results:**



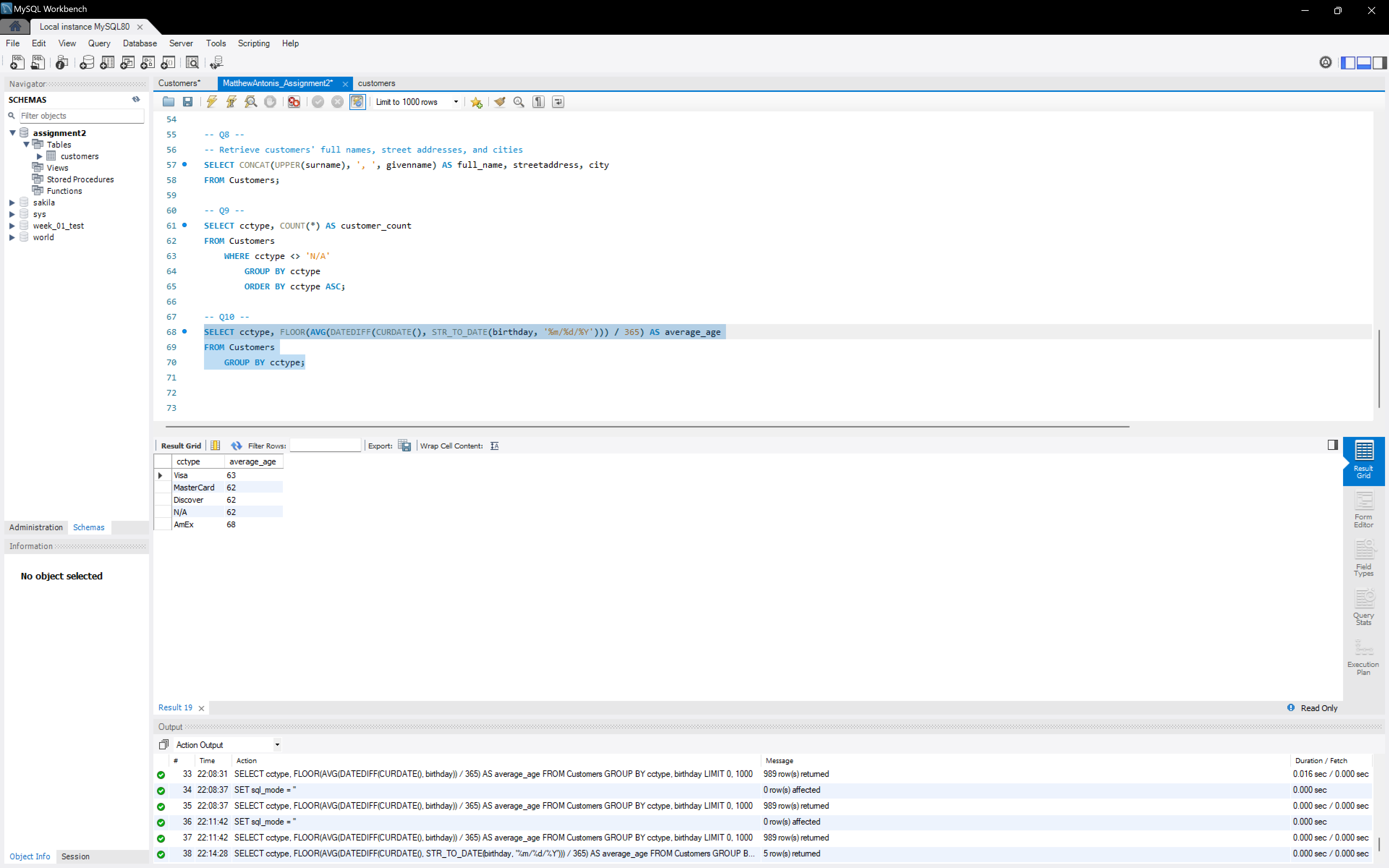
10. Write a query to calculate the average age (in years, rounded down to the nearest integer)

of all customers based on their birthdays, and group the results by cctype.

**Script:**

* SELECT cctype, FLOOR(AVG(DATEDIFF(CURDATE(), STR\_TO\_DATE(birthday, '%m/%d/%Y'))) / 365) AS average\_age
* FROM Customers
  + GROUP BY cctype;

**Results:**



**Comments:**

* The STR\_TO\_DATE function is used to convert the 'birthday' string into a date format that MySQL can understand. The format '%m/%d/%Y' specifies the month/day/year format.
* The DATEDIFF function calculates the difference in days between the current date (CURDATE()) and the converted 'birthday' date.
* The AVG function calculates the average age (in days) within each group of 'cctype'.
* The FLOOR function is used to round down the average age in days to the nearest integer when divided by 365, providing the average age in years.
* The GROUP BY clause groups the data by 'cctype' to calculate the average age per credit card type.

11. Write a query that will return the data required to build this pie graph. In your report

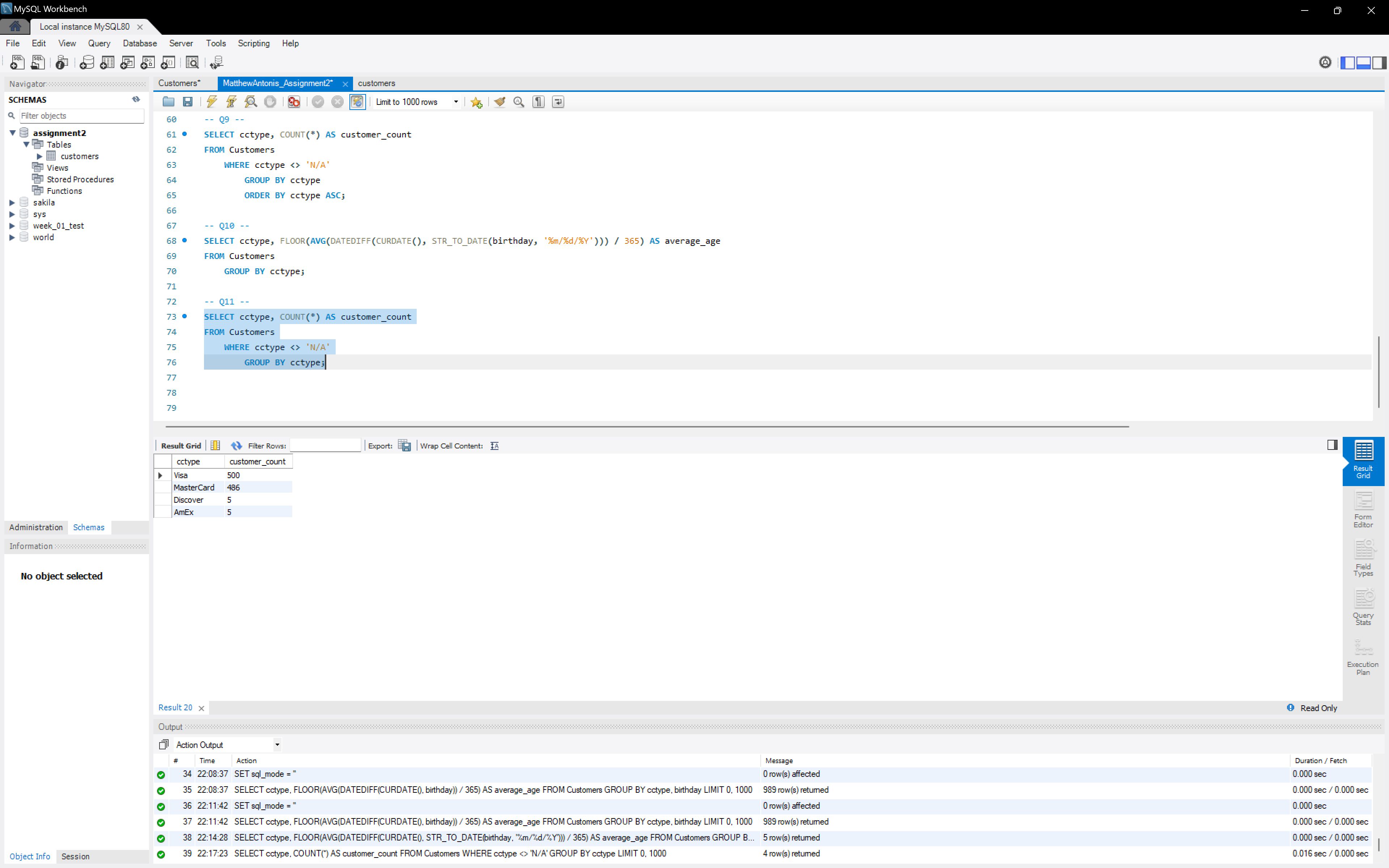
include a pie graph that you construct using the result set (you can use the software of your

choice to make the graph e.g., Excel).

**Script:**

* SELECT cctype, COUNT(\*) AS customer\_count
* FROM Customers
  + WHERE cctype <> 'N/A'
    - GROUP BY cctype;

**Results:**



**Comments:**

* This query calculates the count of customers for each credit card type (excluding those with 'N/A' as the credit card type) and groups the results by 'cctype'.
* The result set includes two columns: 'cctype' representing the credit card type and 'customer\_count' representing the count of customers for each credit card type.

**Pie Graph from Excel:**

