**Directions**

Rockbuster's database engineers have loaded some new data into the database, and your manager has asked you to clean and profile it. Follow the instructions below to complete their request:

1. **Check for and clean dirty data:** Find out if the film table and the customer table contain any dirty data, specifically non-uniform or duplicate data, or missing values. Create a new “Answers 3.6” document and copy-paste your queries into it. Next to each query write 2 to 3 sentences explaining how you would clean the data (even if the data is not dirty).

**FILM TABLE**

Graphical user interface, text, application

Description automatically generated

**CUSTOMER TABLE**

Graphical user interface, application

Description automatically generated

There is no returned duplicate value.

There are two ways of dealing with duplicate value

* Create a virtual table “View” where unique records can be selected
* Delete duplicate record from the table or View However, if altering table is not permitted, we can use GROUP BY or DISTINCT to select unique record.

**NON-UNIFORM**

Graphical user interface, text, application

Description automatically generated

Values are homogeneous, but in order to fix it on the table if there was an issue, I would use the UPDATE command combined with SET and WHERE, to replace the values that should be differently represented.

**Incorrect Data**

The only way to identify incorrect data is to go to the source to find out if the values are correct or not. That said, there are times when common sense can help you figure out if a value is right or wrong.

**Missing Data**

This can also be fixed with logic, as you can use commands to better visualize the data and find missing or null information. In some cases where a column has too much information missing, it might even be valid to remove it from queries. A solution for it after finding the missing data, is to replace it with the average of the remaining informed data (if appropriate). The following command can assist with that:

UPDATE tablename

SET = AVG(col1)

WHERE col1 IS NULL

1. **Summarize your data:** Use SQL to calculate descriptive statistics for both the film table and the customer table. For numerical columns, this means finding the minimum, maximum, and average values. For non-numerical columns, calculate the mode value. Copy-paste your SQL queries and their outputs into your answers document.

**FILM(NUMERICAL COLUMNS)**

--descriptive statistics for numerical columns for film table

SELECT MIN(rental\_rate) AS min\_renatl\_rate,

MAX(rental\_rate) AS max\_rental\_rate,

AVG(rental\_rate) AS avg\_renatal\_rate,

MIN(rental\_duration) AS min\_rental\_duration,

MAX(rental\_duration) AS max\_rental\_duration,

AVG(rental\_duration) AS avg\_rental\_duration,

MIN(film\_id) AS min\_film,

MAX(film\_id) AS max\_film,

AVG(film\_id) AS avg\_film,

MIN(language\_id) AS min\_language,

MAX(language\_id) AS max\_language,

AVG(language\_id) AS avg\_language,

MIN(length) AS min\_length,

MAX(length) AS max\_length,

AVG(length) AS avg\_length,

MIN(replacement\_cost) AS min\_replacement\_cost,

MAX(replacement\_cost) AS max\_replacement\_cost,

AVG(replacement\_cost) AS avg\_replacement\_cost

FROM film;

Graphical user interface, text, application

Description automatically generated

**FILM(NON-NUMERICAL COLUMNS)**

--mode value for non-numerical columns for film table

SELECT mode() WITHIN GROUP (ORDER BY rating)

AS rating\_value,

mode() WITHIN GROUP (ORDER BY special\_features)

AS Feature\_value,

mode() WITHIN GROUP (ORDER BY title)

AS title\_value

FROM film;

Graphical user interface, text, application

Description automatically generated

**CUSTOMER(NUMERICAL COLUMNS)**

--descriptive statistics for numerical columns for customer table

SELECT MIN(active) AS min\_active,

MAX(active) AS max\_active,

AVG(active) AS avg\_active,

MIN(address\_id) AS min\_address,

MAX(address\_id) AS max\_address,

AVG(address\_id) AS avg\_address,

MIN(customer\_id) AS min\_customer,

MAX(customer\_id) AS max\_customer,

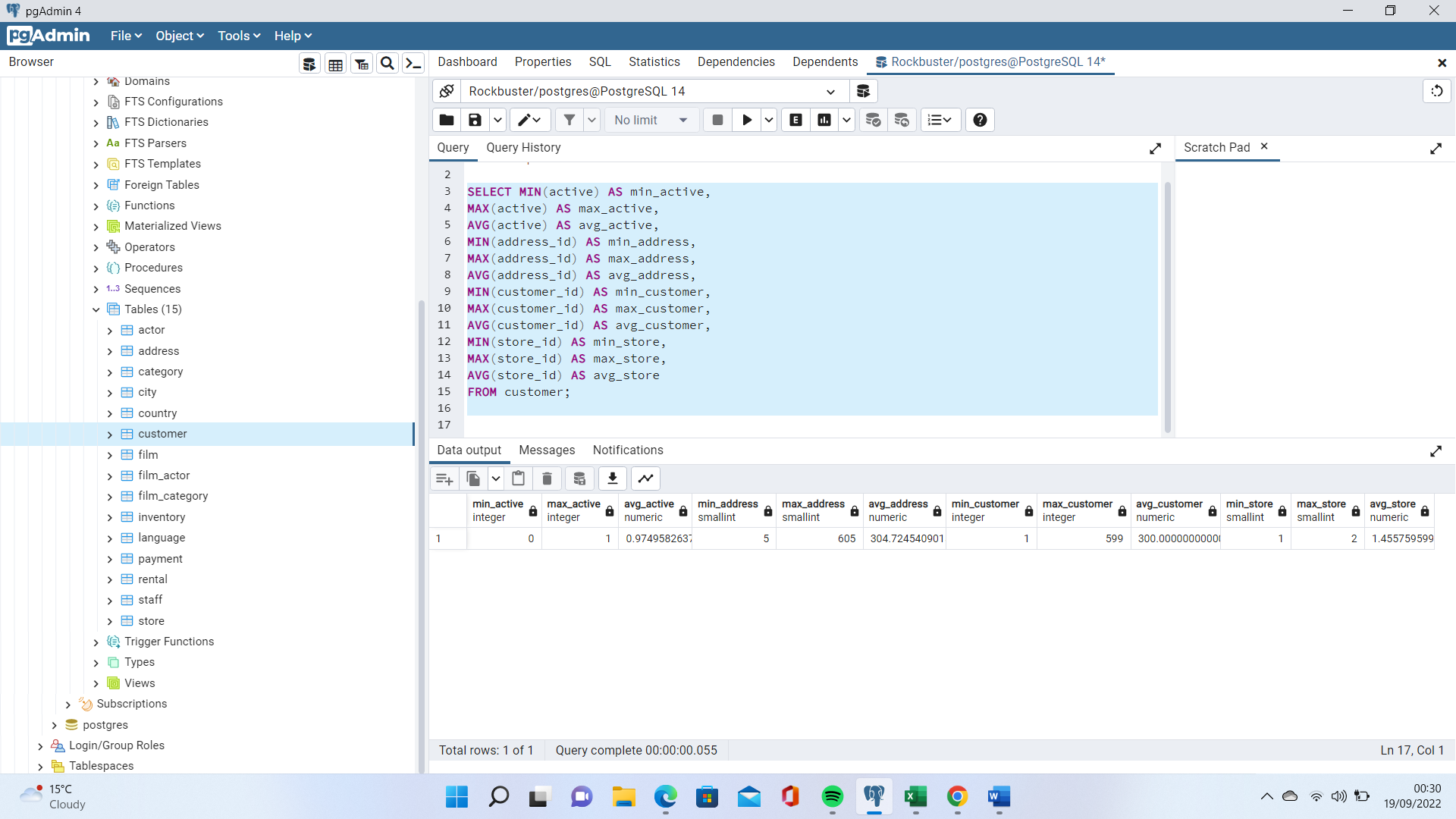
AVG(customer\_id) AS avg\_customer,

MIN(store\_id) AS min\_store,

MAX(store\_id) AS max\_store,

AVG(store\_id) AS avg\_store

FROM customer;



**CUSTOMER(NON-NUMERICAL COLUMNS)**

--mode value for non-numerical columns for customer table

SELECT mode() WITHIN GROUP (ORDER BY first\_name)

AS first\_name\_value,

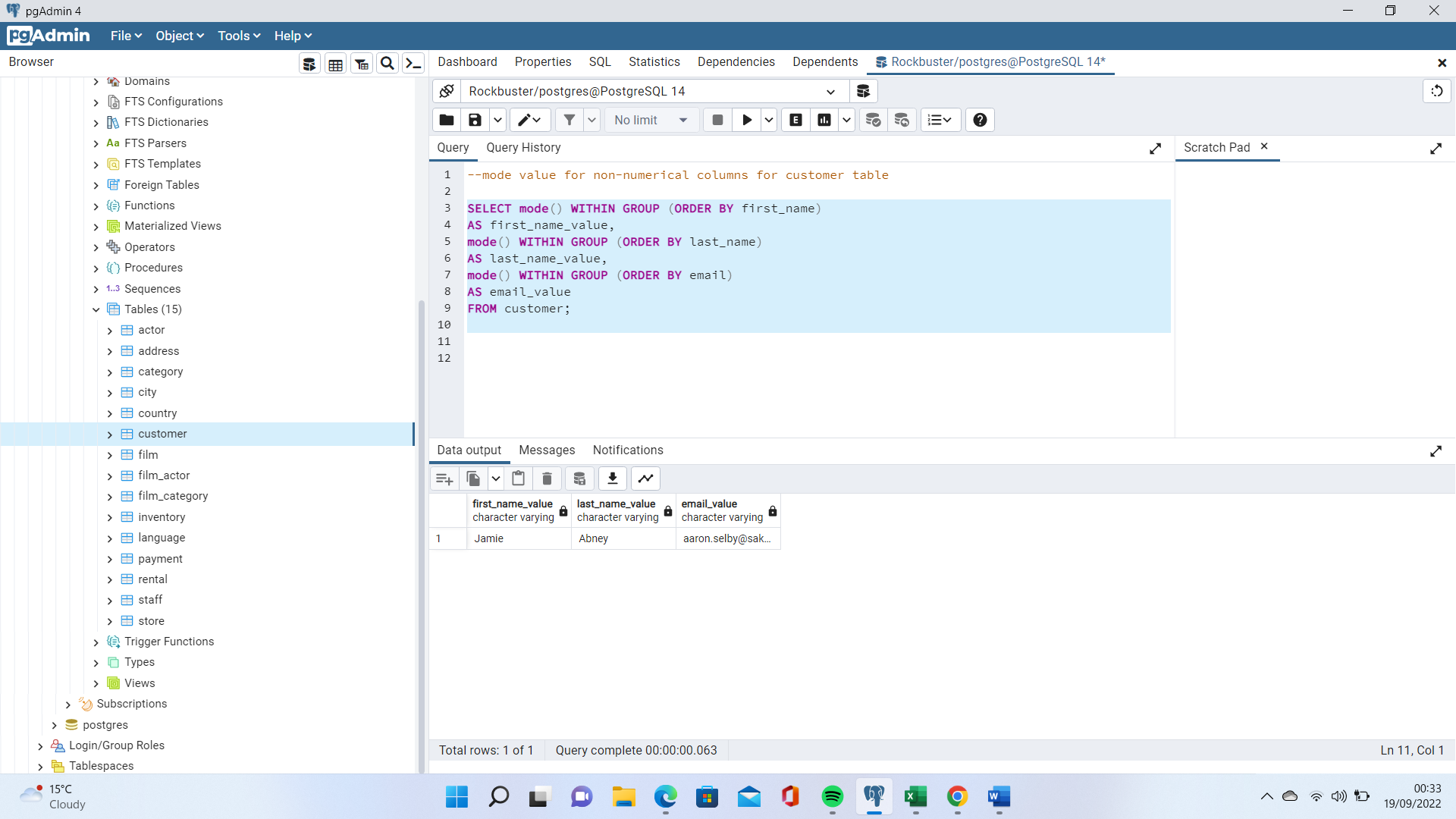
mode() WITHIN GROUP (ORDER BY last\_name)

AS last\_name\_value,

mode() WITHIN GROUP (ORDER BY email)

AS email\_value

FROM customer;



1. **Reflect on your work: Back in Achievement 1 you learned about data profiling in Excel. Based on your previous experience, which tool (Excel or SQL) do you think is more effective for data profiling, and why? Consider their respective functions, ease of use, and speed. Write a short paragraph in the running document that you have started.**

SQL is much better suited for data profiling, even when working with a smaller data set in Excel it would take longer to compile these as there are multiple steps involved in the process. With SQL once the query has been written it can be applied time and again without much effort.