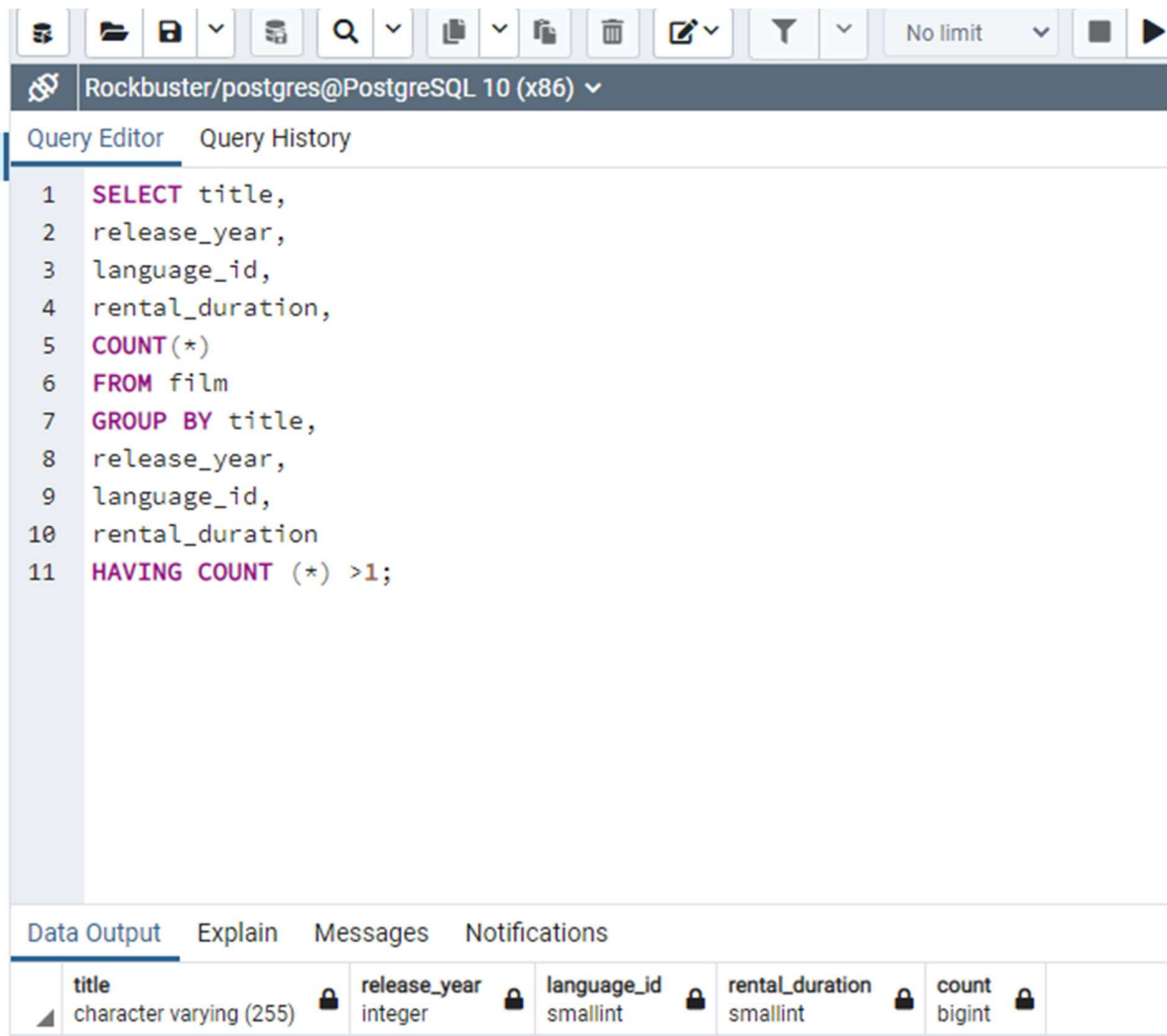


Summarizing & Cleaning Data in SQL

Question 1

Film Table



The screenshot shows a PostgreSQL query editor interface. At the top, there is a toolbar with various icons for file operations, search, and execution. Below the toolbar, the connection name "Rockbuster/postgres@PostgreSQL 10 (x86)" is displayed. The "Query Editor" tab is active, showing a SQL query with line numbers 1 through 11. The query is a SELECT statement that groups data from the "film" table by title, release_year, language_id, and rental_duration, and filters out groups with a count of 1 or less. At the bottom, the "Data Output" tab is active, showing the column definitions for the query results.

```
1 SELECT title,  
2 release_year,  
3 language_id,  
4 rental_duration,  
5 COUNT(*)  
6 FROM film  
7 GROUP BY title,  
8 release_year,  
9 language_id,  
10 rental_duration  
11 HAVING COUNT (*) >1;
```

title	release_year	language_id	rental_duration	count
character varying (255)	integer	smallint	smallint	bigint

Customer Table

The screenshot shows a PostgreSQL query editor interface. The title bar indicates the user is 'Rockbuster/postgres@PostgreSQL 10 (x86)'. The 'Query Editor' tab is active, displaying a SQL query. The query is designed to find customers who have duplicate information based on their first name, last name, email, and address ID. The query uses a GROUP BY clause on these fields and a HAVING clause to filter for groups with a count greater than 1. Below the query editor, the 'Data Output' tab is active, showing the schema of the result set. The schema includes columns for customer_id (integer, primary key), store_id (smallint), first_name (character varying (45)), last_name (character varying (45)), email (character varying (50)), and count (bigint). The result set is currently empty.

```
1 SELECT customer_id,  
2 store_id,  
3 first_name,  
4 last_name,  
5 email,  
6 COUNT(*)  
7 FROM customer  
8 GROUP BY  
9 customer_id,  
10 store_id,  
11 first_name,  
12 last_name,  
13 email,  
14 address_id  
15 HAVING COUNT(*)>1;
```

customer_id	store_id	first_name	last_name	email	count
[PK] integer	smallint	character varying (45)	character varying (45)	character varying (50)	bigint

There are no duplicates in the above.

Question 2

Film table with non-numerical columns

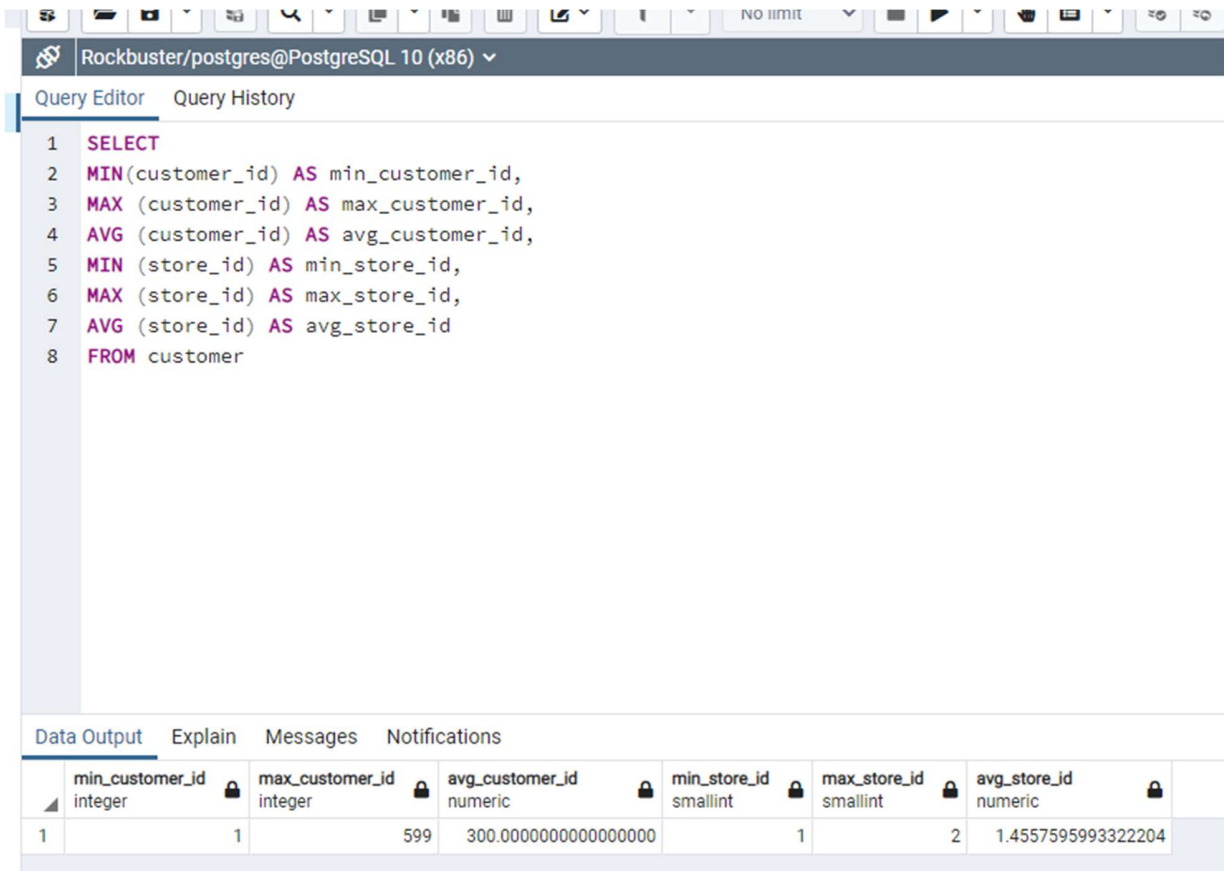
The screenshot shows a PostgreSQL query editor interface. The query editor contains the following SQL query:

```
1 SELECT mode() WITHIN GROUP (ORDER BY film_id) AS modal_film_id,  
2 mode () WITHIN GROUP (ORDER BY title) AS modal_title,  
3 mode () WITHIN GROUP (ORDER BY description) AS modal_description,  
4 mode () WITHIN GROUP (ORDER BY rating) AS modal_rating  
5 FROM film
```

The query has been executed, and the results are displayed in the Data Output tab. The results show a single row of data:

modal_film_id	modal_title	modal_description	modal_rating
1	Academy Dinosaur	A Action-Packed Character Study of a Astronaut And a Explorer who must Reach a Monkey in A MySQL Convention	PG-13

Customer table with numerical columns



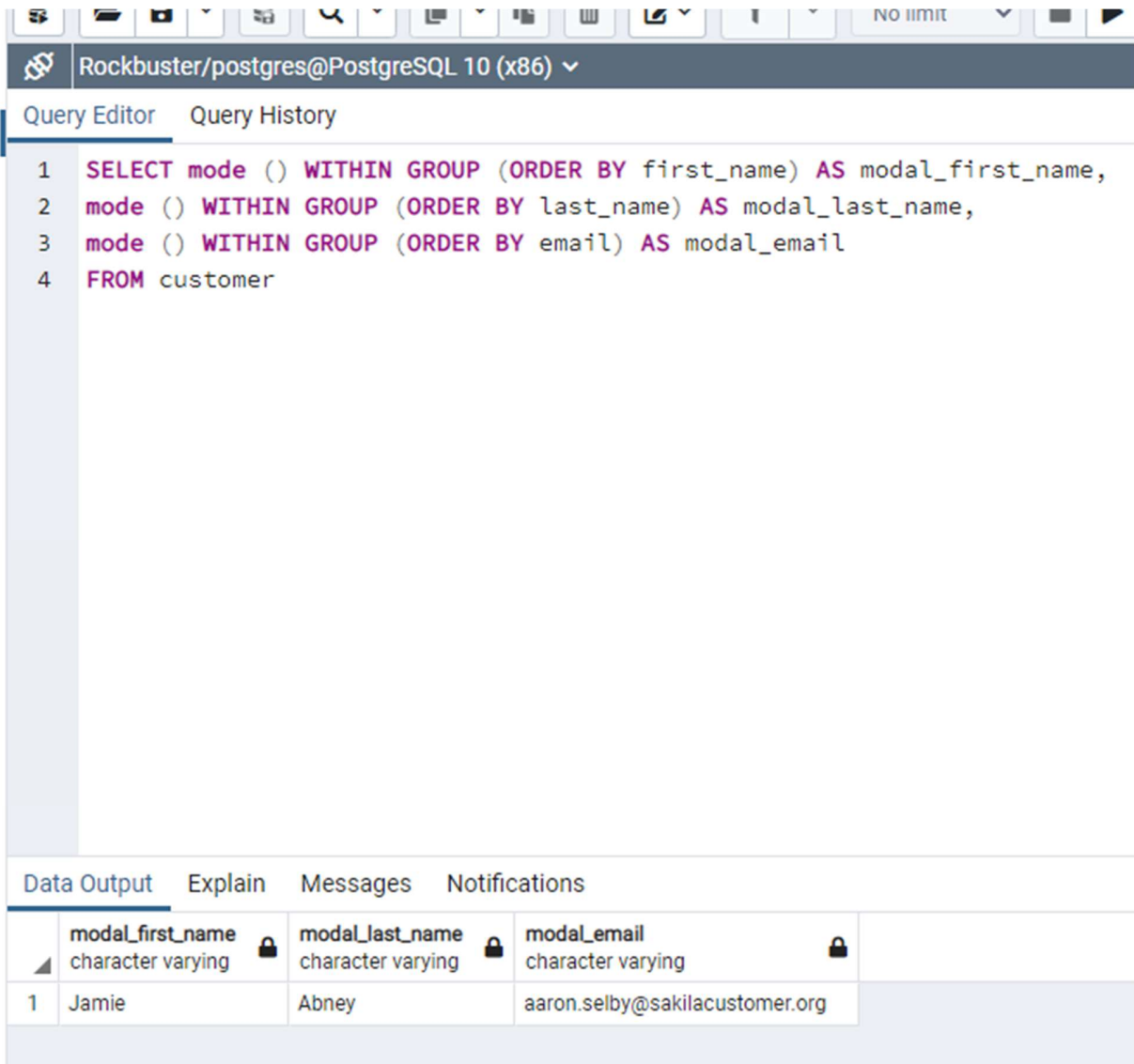
The screenshot shows a PostgreSQL query editor interface. At the top, the connection is set to 'Rockbuster/postgres@PostgreSQL 10 (x86)'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying a SQL query. The query is as follows:

```
1 SELECT
2 MIN(customer_id) AS min_customer_id,
3 MAX (customer_id) AS max_customer_id,
4 AVG (customer_id) AS avg_customer_id,
5 MIN (store_id) AS min_store_id,
6 MAX (store_id) AS max_store_id,
7 AVG (store_id) AS avg_store_id
8 FROM customer
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query in a table format. The table has 7 columns: min_customer_id, max_customer_id, avg_customer_id, min_store_id, max_store_id, and avg_store_id. The data is as follows:

	min_customer_id integer	max_customer_id integer	avg_customer_id numeric	min_store_id smallint	max_store_id smallint	avg_store_id numeric
1	1	599	300.0000000000000000	1	2	1.4557595993322204

Customer table with non-numerical columns



Rockbuster/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```
1 SELECT mode () WITHIN GROUP (ORDER BY first_name) AS modal_first_name,
2 mode () WITHIN GROUP (ORDER BY last_name) AS modal_last_name,
3 mode () WITHIN GROUP (ORDER BY email) AS modal_email
4 FROM customer
```

Data Output Explain Messages Notifications

	modal_first_name character varying	modal_last_name character varying	modal_email character varying	
1	Jamie	Abney	aaron.selby@sakilacustomer.org	

Question 3

When it comes to Data profiling I think it is much each to use SQL dependent on the size of the Database.