

Task 3.6: Summarizing & Cleaning Data in SQL

Directions

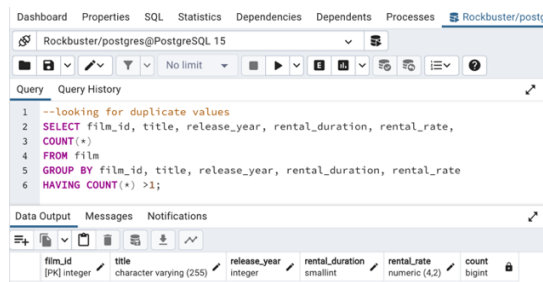
Rockbuster's database engineers have loaded some new data into the database, and your manager has asked you to clean and profile it. Follow the instructions below to complete their request:

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).\

a. Duplicate data from film

--looking for duplicate values

```
SELECT film_id, title, release_year, rental_duration, rental_rate,  
COUNT(*)  
FROM film  
GROUP BY film_id, title, release_year, rental_duration, rental_rate  
HAVING COUNT(*) >1;
```



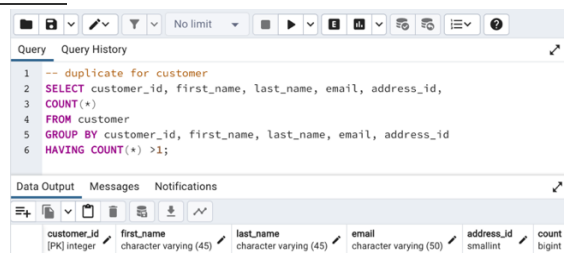
The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 --looking for duplicate values  
2 SELECT film_id, title, release_year, rental_duration, rental_rate,  
3 COUNT(*)  
4 FROM film  
5 GROUP BY film_id, title, release_year, rental_duration, rental_rate  
6 HAVING COUNT(*) >1;
```

The Data Output section shows the following columns:

film_id	title	release_year	rental_duration	rental_rate	count
[PK] integer	character varying (255)	integer	smallint	numeric (4,2)	bigint

b. Data from customer table



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 -- duplicate for customer  
2 SELECT customer_id, first_name, last_name, email, address_id,  
3 COUNT(*)  
4 FROM customer  
5 GROUP BY customer_id, first_name, last_name, email, address_id  
6 HAVING COUNT(*) >1;
```

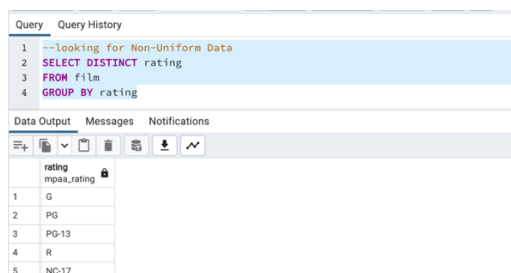
The Data Output section shows the following columns:

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

c. Non-uniform data

--looking for Non-Uniform Data

```
SELECT DISTINCT rating  
FROM film  
GROUP BY rating
```



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 --looking for Non-Uniform Data  
2 SELECT DISTINCT rating  
3 FROM film  
4 GROUP BY rating
```

The Data Output section shows the following columns:

rating
mpaa_rating
1 G
2 PG
3 PG-13
4 R
5 NC-17

2. **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.

```
SELECT
MIN(customer_id) AS min_customer_id,
MAX(customer_id) AS max_customer_id,
AVG(customer_id) AS avg_customer_id,
MIN(store_id) AS min_store_id,
MAX(store_id) AS max_store_id,
AVG(store_id) AS avg_store_id,
MIN(address_id) AS min_address_id,
MAX(address_id) AS max_address_id,
AVG(address_id) AS avg_address_id,
MIN(create_date) AS min_create_date,
MAX(create_date) AS max_create_date,
MODE() WITHIN GROUP (ORDER BY create_date) AS create_date,
MIN(last_update) AS min_last_update,
MAX(last_update) AS max_last_update,
MODE() WITHIN GROUP (ORDER BY last_update) AS last_update,
MODE() WITHIN GROUP (ORDER BY first_name) AS first_name,
MODE() WITHIN GROUP (ORDER BY last_name) AS last_name,
MODE() WITHIN GROUP (ORDER BY email) AS email,
MODE() WITHIN GROUP (ORDER BY create_date) AS create_date,
MODE() WITHIN GROUP (ORDER BY active) AS mode_active
FROM customer;
```

Query Query History											
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	MIN(address_id) AS min_address_id,										
9	MAX(address_id) AS max_address_id,										
10	AVG(address_id) AS avg_address_id,										
11	MIN(create_date) AS min_create_date,										
12	MAX(create_date) AS max_create_date,										
13	MODE() WITHIN GROUP (ORDER BY create_date) AS create_date,										
14	MIN(last_update) AS min_last_update,										
15	MAX(last_update) AS max_last_update,										
16	MODE() WITHIN GROUP (ORDER BY last_update) AS last_update,										
17	MODE() WITHIN GROUP (ORDER BY first_name) AS first_name,										
18	MODE() WITHIN GROUP (ORDER BY last_name) AS last_name,										
19	MODE() WITHIN GROUP (ORDER BY email) AS email,										
20	MODE() WITHIN GROUP (ORDER BY create_date) AS create_date,										
21	MODE() WITHIN GROUP (ORDER BY active) AS mode_active										
22	FROM customer;										

Data Output Messages Notifications											
	min_customer_id	max_customer_id	avg_customer_id	min_store_id	max_store_id	avg_store_id	min_address_id	max_address_id	avg_address_id	min_create_date	max_create_date
	integer	integer	numeric	smallint	smallint	numeric	smallint	smallint	numeric	date	date
1	1	599	300.0000000	1	2	1.455759599	5	605	304.7245409	2006-02-14	2006-02-14

Data Output Messages Notifications										
	max_create_date	create_date	min_last_update	max_last_update	last_update	first_name	last_name	email	create_date	mode_active
	date	date	timestamp with time zone	timestamp with time zone	timestamp with time zone	character varying	character varying	character varying	date	integer
1	2006-02-14	2006-02-14	2013-05-...	2013-05-...	2013-05-...	Jamie	Abney	aaron.sel...	2006-02-14	1

3. **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.

Excel vs SQL, which one is more effective?

Excel is an excellent tool to utilize when dealing with a small amount of data but depending on what data you need it can be a hassle. Once you learn all the wording for maximizing SQL it can be a faster tool to utilize.

4. Save your “Answers 3.6” document as a PDF and upload it here for your tutor to review.