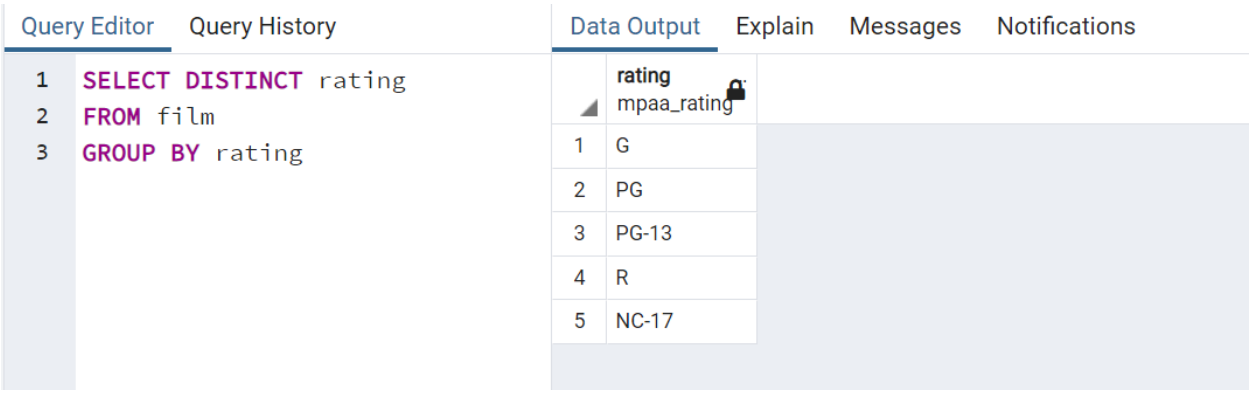
Answers to 3.6

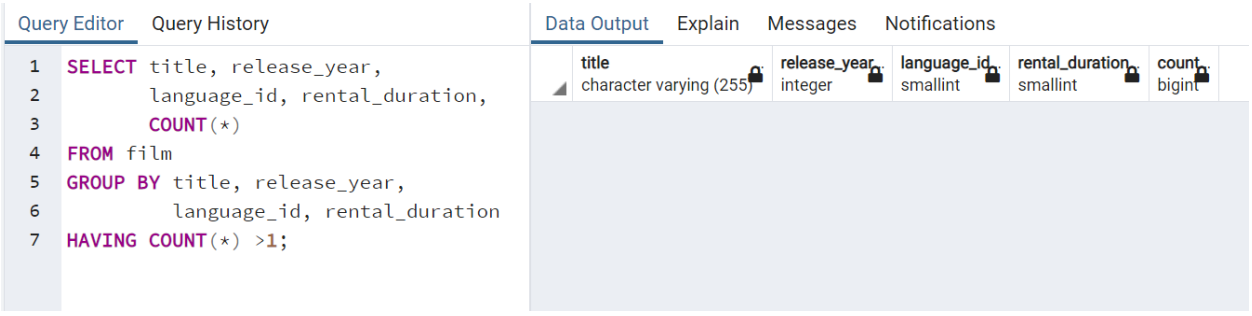
1. **Check for and clean dirty data:** Find out if the film table and the customer table contain any dirty data, specifically non-uniform or duplicate data, or missing values. Create a new "Answers 3.6" document and copy-paste your queries into it. Next to each query write 2 to 3 sentences explaining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

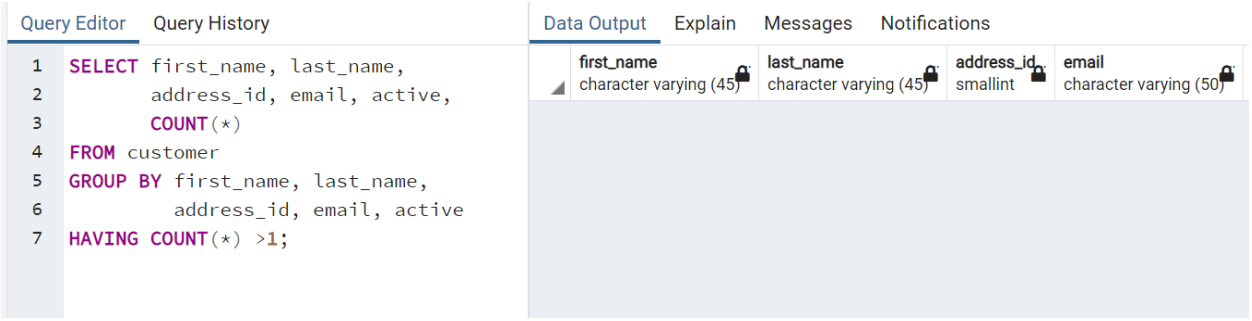
   ---Checking for rating values that don't fit in to uniformity---



If values are not showing uniformity, then a constant value for each category should be chosen and then the values should be all changed to that value for each category. For future entries, each column could have constraints put on it to make sure that there are no odd values entered in the future.

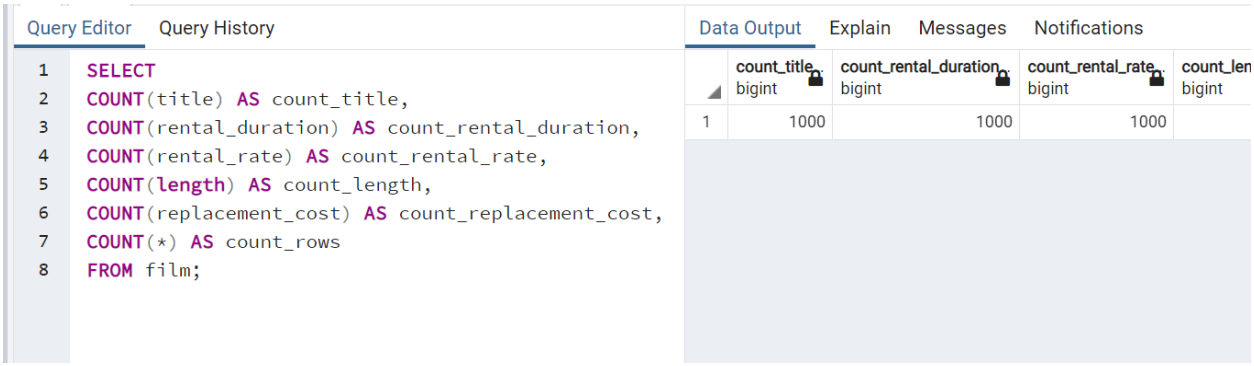
---Duplicate search---

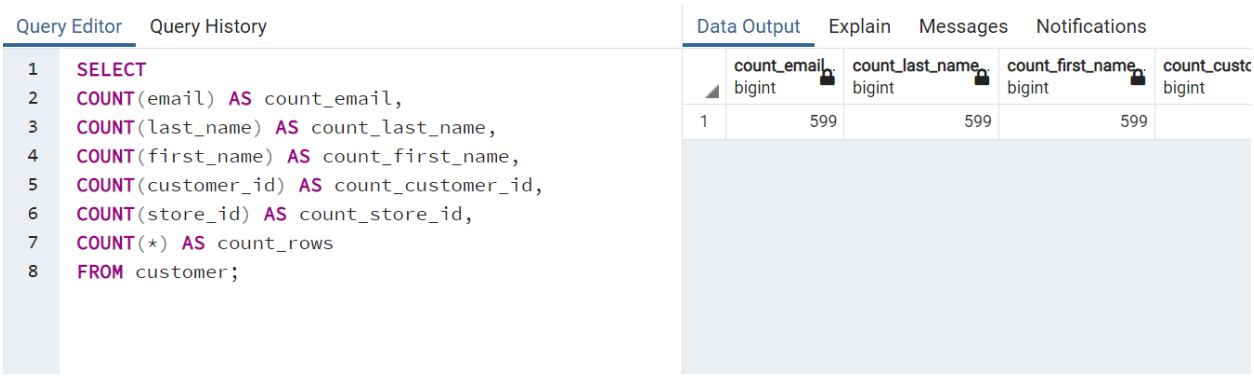




Values coming up as duplicate can either be hidden with specific query types or if it is necessary, they can be deleted. Another option you can limit the query to distinct values as well.

---Missing value search---





Any missing values can be filled in by finding column averages and inputting this value in to the blank spots. You could omit the entire column, however this is typically not a very desirable situation when it comes to data summarization.

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.

---descriptive statistics query film table---

Query Editor | Query History

```sql
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
```

Data Output | Explain | Messages | Notifications

	min_renatl_rate numeric	max_rental_rate numeric	avg_renatal_rate numeric	min_rental_duration smallint	max_rental_duration smallint	avg_rental_duration numeric
1	0.99	4.99	2.9800000000000000	3	7	4.9850000000000000

Data Output | Explain | Messages | Notifications

min_film integer	max_film integer	avg_film numeric	min_language smallint	max_language smallint	avg_language numeric	min_length smallint
1	1000	500.5000000000000000	1	1	1.0000000000000000000	46

lain Messages Notifications

max_length smallint	avg_length numeric	min_replacement_cost numeric	max_replacement_cost numeric	avg_replacement_cost numeric
185	115.2720000000000000	9.99	29.99	19.9840000000000000

---descriptive statistics query customer table---

Query Editor Query History

```
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;
```

Data Output Explain Messages Notifications

	min_active integer	max_active integer	avg_active numeric	min_address smallint	max_address smallint
1	0	1	0.97495826377295492487	5	605

Data Output Explain Messages Notifications

max_address smallint	avg_address numeric	min_customer integer	max_customer integer	avg_customer numeric	min_store smallint	max_store smallint
605	304.7245409015025042	1	599	300.0000000000000000	1	2

avg_store numeric
1.4557595993322204

---Finding the mode (most repeated) for non numeric columns film table---

Query Editor Query History

```
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 release_year)
AS year_value,
mode() WITHIN GROUP (ORDER BY title)
AS title_value
FROM film;
```

Data Output Explain Messages Notifications

	rating_value mpaa_rating	feature_value text[]	year_value integer	title_value character varying
1	PG-13	{Trailers,Commentaries,"Behind the Scenes"}	2006	Academy Dinosaur

---Finding the mode (most repeated) for non numeric columns customer table---

Query Editor | Query History

```
1  SELECT mode() WITHIN GROUP (ORDER BY first_name)
2  AS first_name_value,
3  mode() WITHIN GROUP (ORDER BY last_name)
4  AS last_name_value,
5  mode() WITHIN GROUP (ORDER BY email)
6  AS email_value
7  FROM customer
```

Data Output | Explain | Messages | Notifications

	first_name_value character varying	last_name_value character varying	email_value character varying
1	Jamie	Abney	aaron.selby@sakilacustomer.org

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.

   Without a doubt SQL would be much quicker and more efficient for the above work done. Excel would not take a really long time, however building the pivot tables is just not as efficient with a lot more clicks on the screen to get the same result. For SQL you would just have to know how to ask the question properly, then your entire answer is there in milliseconds.