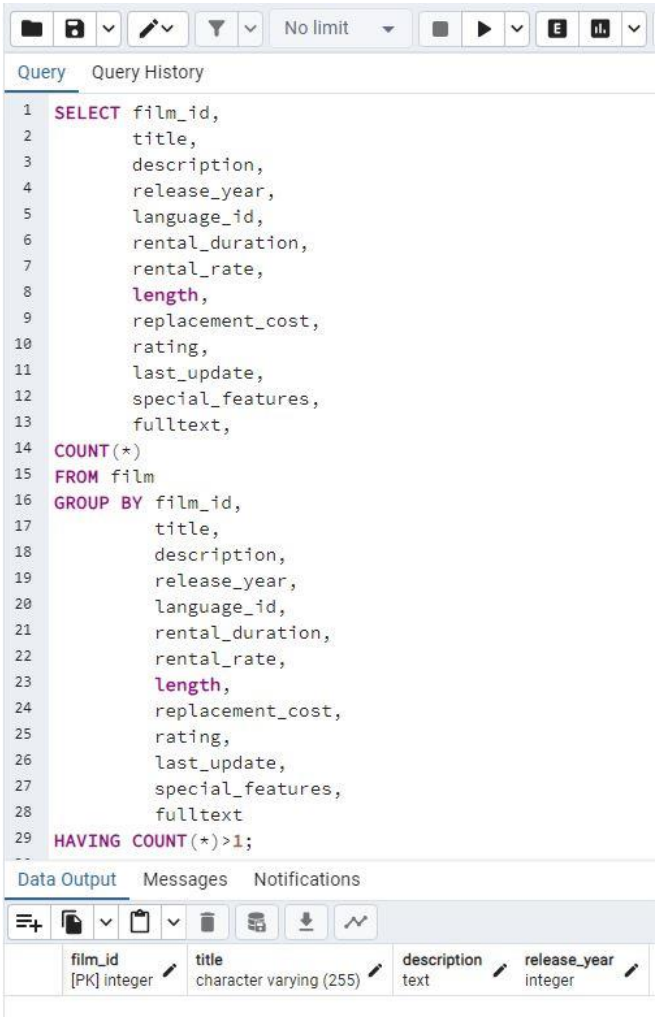
Data Immersion 3.6

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).

Duplicate Data

Film Table



I found no duplicates in the film_table and customer_table; if duplicates were present, I would have removed them using GROUP BY or DISTINCT, or created a view table to manage the duplicates.

Customer Table

Query Query History

```sql
SELECT customer_id
       store_id,
       first_name
       last_name,
       email,
       address_id,
       activebool,
       create_date,
       last_update,
       active,
COUNT(*)
FROM customer
GROUP BY customer_id,
         store_id,
         first_name,
         last_name,
         email,
         address_id,
         activebool,
         create_date,
         last_update,
         active
HAVING COUNT(*)<1;
```

Data Output Messages Notifications

store_id integer	last_name character varying (45)	email character varying (50)

Non-uniform Data

Film Table

Query Query History

```sql
SELECT DISTINCT film_id,
                title,
                description,
                release_year,
                language_id,
                rental_duration,
                rental_rate,
                length,
                replacement_cost,
                rating,
                last_update,
                special_features,
                fulltext
FROM film
ORDER BY film_id;
```

Data Output Messages Notifications

	film_id [PK] integer	title character varying (255)	description text
1	1	Academy Dinosaur	A Epic Drama of a Femir
2	2	Ace Goldfinger	A Astounding Epistle of
3	3	Adaptation Holes	A Astounding Reflection
4	4	Affair Prejudice	A Fanciful Documentary
5	5	African Egg	A Fast-Paced Document
6	6	Agent Truman	A Intrepid Panorama of a
7	7	Airplane Sierra	A Touching Saga of a Hu
8	8	Airport Pollock	A Epic Tale of a Moose A
9	9	Alabama Devil	A Thoughtful Panorama
10	10	Aladdin Calendar	A Action-Packed Tale of
11	11	Alamo Videotape	A Boring Epistle of a But
12	12	Alaska Phantom	A Fanciful Saga of a Hur
13	13	Ali Forever	A Action-Packed Drama
14	14	Alice Fantasia	A Emotional Drama of a
15	15	Alien Center	A Brilliant Drama of a Ca
16	16	Alley Evolution	A Fast-Paced Drama of a

There were no uniformity issues in the mentioned tables; however, if encountered, I would correct them using SQL commands like UPDATE, GROUP BY, or DISTINCT to ensure data consistency.

Customer Table

Query Query History

```sql
SELECT DISTINCT film_id,
                title,
                description,
                release_year,
                language_id,
                rental_duration,
                rental_rate,
                length,
                replacement_cost,
                rating,
                last_update,
                special_features,
                fulltext
FROM film
ORDER BY film_id;
```

Data Output Messages Notifications

	film_id [PK] integer	title character varying (255)	description text
1	1	Academy Dinosaur	A Epic Drama of a Femir
2	2	Ace Goldfinger	A Astounding Epistle of
3	3	Adaptation Holes	A Astounding Reflection
4	4	Affair Prejudice	A Fanciful Documentary
5	5	African Egg	A Fast-Paced Document
6	6	Agent Truman	A Intrepid Panorama of a
7	7	Airplane Sierra	A Touching Saga of a Hu
8	8	Airport Pollock	A Epic Tale of a Moose A
9	9	Alabama Devil	A Thoughtful Panorama
10	10	Aladdin Calendar	A Action-Packed Tale of
11	11	Alamo Videotape	A Boring Epistle of a But
12	12	Alaska Phantom	A Fanciful Saga of a Hur
13	13	Ali Forever	A Action-Packed Drama
14	14	Alice Fantasia	A Emotional Drama of a
15	15	Alien Center	A Brilliant Drama of a Ca
16	16	Alley Evolution	A Fast-Paced Drama of a

Missing Values

Film Table

Query Query History

```sql
WHERE (film_id,
       title,
       description,
       release_year,
       language_id,
       rental_duration,
       rental_rate,
       length,
       replacement_cost,
       rating,
       last_update,
       special_features,
       fulltext) IS NULL
ORDER BY film_id;
```

Data Output Messages Notifications

film_id [PK] integer	title character varying (255)	description text

I encountered no missing values in the tables, but if necessary, I would impute missing data using SQL syntax to fill values with column averages or other relevant statistics, applying the WHERE clause to specify conditions for imputing data.

Customer Table

Query Query History

```sql
SELECT *
FROM customer
WHERE (store_id,
       first_name,
       last_name,
       email,
       address_id,
       activebool,
       create_date,
       last_update,
       active) IS NULL;
```

Data Output Messages Notifications

customer_id [PK] integer	store_id smallint	first_name character varyi

2. Summarise 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.

Numeric

Query Query History

```sql
SELECT
  MIN(release_year) AS min_release_year,
  MIN(rental_duration) AS min_rentdur,
  MIN(rental_rate) AS min_rate,
  MIN(length) AS min_leng,
  MIN(replacement_cost) AS min_replac,
  MAX(release_year) AS max_release_year,
  MAX(rental_duration) AS max_rentdur,
  MAX(rental_rate) AS max_rate,
  MAX(length) AS max_leng,
  MAX(replacement_cost) AS max_replac,
  AVG(release_year) AS avg_release_year,
  AVG(rental_duration) AS avg_rentdur,
  AVG(rental_rate) AS avg_rate,
  AVG(length) AS avg_leng,
  AVG(replacement_cost) AS avg_replac
FROM film;
```

Data Output Messages Notifications

	min_release_year integer	min_rentdur smallint	min_rate numeric	min_leng smallint
1	2006	3	0.99	46

The data could reveal insights such as customer preferences and rental behaviours, using SQL functions like MIN, MAX, AVG for numeric data and MODE for non-numeric data to summarize customer interactions and movie popularity, although the customer_table lacks numerical data for detailed statistical analysis.

Non-numeric

Query Query History

```sql
SELECT
  MODE () WITHIN GROUP (ORDER BY customer_id) AS modal_customer_id,
  MODE () WITHIN GROUP (ORDER BY store_id) AS modal_store_id,
  MODE () WITHIN GROUP (ORDER BY first_name) AS modal_first_name,
  MODE () WITHIN GROUP (ORDER BY last_name) AS modal_last_name,
  MODE () WITHIN GROUP (ORDER BY email) AS modal_email,
  MODE () WITHIN GROUP (ORDER BY address_id) AS modal_address_id,
  MODE () WITHIN GROUP (ORDER BY activebool) AS modal_activebool,
  MODE () WITHIN GROUP (ORDER BY create_date) AS modal_create_date,
  MODE () WITHIN GROUP (ORDER BY last_update) AS modal_last_update,
  MODE () WITHIN GROUP (ORDER BY active) AS modal_active
FROM customer;
```

Data Output Messages Notifications

	modal_customer_id integer	modal_store_id smallint	modal_first_name character varying	modal_last_name character varying	modal_em character v
1	1	1	Jamie	Abney	aaron.selb

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.

From my perspective, if the task at hand involves data profiling for smaller datasets and a need for quick, visual insights, I would lean towards Excel. Its robust features for visualisation allow for immediate graphical representation of data, which can be very intuitive for spotting trends and outliers. However, when dealing with large datasets or requiring scalability, SQL is a more

powerful choice. Despite having a steeper learning curve and being less user-friendly, SQL can handle complex queries and large volumes of data efficiently. It's a tool designed with performance in mind, capable of profiling data directly within the database, reducing the need for data transfer. In essence, Excel shines for accessibility and visualisation, while SQL excels in data handling capacity and is well-suited for in-depth, large-scale data profiling tasks. I definitely find it logistically easier to use Excel though and it user face more appealing.