

## Summarizing & Cleaning Data in SQL

### 1. Check for and clean dirty data:

#### Nonuniform Data

#### FILM

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane displays the database structure, with 'Tables (16)' expanded and 'title' selected. The main pane shows a SQL query in the 'Query' editor:

```
1 SELECT DISTINCT film_id,  
2 title,  
3 description,  
4 release_year,  
5 language_id,  
6 rental_duration,  
7 rental_rate,  
8 length,  
9 replacement_cost,  
10 rating,  
11 last_update  
12 FROM film  
13  
14  
15
```

Below the query editor, the 'Data Output' pane displays the results of the query. The table has 16 columns: film\_id, title, description, release\_year, language\_id, rental\_duration, rental\_rate, and length. The results show 16 rows of data, including titles like 'Party Knock', 'Million Ace', 'Forever Candidate', 'Amelie Heffingtons', 'Virgin Daisy', 'Jedi Beneath', 'Donnie Ailey', 'Pittsburgh Hunchback', 'Ark Ridgmont', 'Edge Kissing', 'Volcano Texas', 'Dragon Squad', 'Caddyshack Jedi', 'Dracula Crystal', 'Dogma Family', and 'Head Stranger'.

film_id	title	description	release_year	language_id	rental_duration	rental_rate	length
660	Party Knock	A Fateful Display of a Technical Writer And a Butler who must Battle a Sumo Wrestler in An Abandoned Mine Shaft	2006	1	7	2.99	
578	Million Ace	A Brilliant Documentary of a Womanizer And a Squirrel who must Find a Technical Writer in The Sahara Desert	2006	1	4	4.99	
328	Forever Candidate	A Unbelievable Panorama of a Technical Writer And a Man who must Pursue a Frisbee in A U-Boat	2006	1	7	2.99	
20	Amelie Heffingtons	A Boring Drama of a Woman And a Squirrel who must Conquer a Student in A Baloon	2006	1	4	4.99	
944	Virgin Daisy	A Awe-Inspiring Documentary of a Robot And a Mad Scientist who must Reach a Database Administrator in A Shark Tank	2006	1	6	4.99	
479	Jedi Beneath	A Astounding Reflection of a Explorer And a Dentist who must Pursue a Student in Nigeria	2006	1	7	0.99	
241	Donnie Ailey	A Awe-Inspiring Tale of a Butler And a Frisbee who must Vanquish a Teacher in Ancient Japan	2006	1	4	0.99	
682	Pittsburgh Hunchback	A Thrilling Epistle of a Boy And a Boat who must Find a Student in Soviet Georgia	2006	1	4	4.99	
38	Ark Ridgmont	A Beautiful Yarn of a Pioneer And a Monkey who must Pursue a Explorer in The Sahara Desert	2006	1	6	0.99	
272	Edge Kissing	A Beautiful Yarn of a Composer And a Mad Cow who must Redeem a Mad Scientist in A Jet Boat	2006	1	5	4.99	
949	Volcano Texas	A Awe-Inspiring Yarn of a Hunter And a Feminist who must Challenge a Dentist in The Outback	2006	1	6	0.99	
250	Dragon Squad	A Taut Reflection of a Boy And a Walress who must Outgun a Teacher in Ancient China	2006	1	4	0.99	
111	Caddyshack Jedi	A Awe-Inspiring Epistle of a Woman And a Madman who must Fight a Robot in Soviet Georgia	2006	1	3	0.99	
249	Dracula Crystal	A Thrilling Reflection of a Feminist And a Cat who must Find a Frisbee in An Abandoned Fun House	2006	1	7	0.99	
239	Dogma Family	A Brilliant Character Study of a Database Administrator And a Monkey who must Succumb a Astronaut in New Orleans	2006	1	5	4.99	
408	Head Stranger	A Thoughtful Saga of a Hunter And a Crocodile who must Confront a Dog in The Gulf of Mexico	2006	1	4	4.99	

Total rows: 1000 of 1000 Query complete 00:00:00.141 Ln 6, Col 17

## CUSTOMER

The screenshot shows the pgAdmin 4 interface with a query executed against the 'customer' table. The query is a SELECT DISTINCT statement listing various customer attributes. The results are displayed in a table with 17 columns and 17 rows of data.

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	address_id smallint	activebool boolean	create_date date	last_update timestamp without time zone	active integer
1	357	1	Keith	Rico	keith.rico@sakilacustomer.org	362	true	2006-02-14	2013-05-26 14:49:45.738	1
2	171	2	Dolores	Wagner	dolores.wagner@sakilacustomer.org	175	true	2006-02-14	2013-05-26 14:49:45.738	1
3	139	1	Amber	Dixon	amber.dixon@sakilacustomer.org	143	true	2006-02-14	2013-05-26 14:49:45.738	1
4	471	1	Dean	Sauer	dean.sauer@sakilacustomer.org	476	true	2006-02-14	2013-05-26 14:49:45.738	1
5	594	1	Eduardo	Hiatt	eduardo.hiatt@sakilacustomer.org	600	true	2006-02-14	2013-05-26 14:49:45.738	1
6	401	2	Tony	Caraniza	tony.caraniza@sakilacustomer.org	406	true	2006-02-14	2013-05-26 14:49:45.738	1
7	157	2	Darlene	Rose	darlene.rose@sakilacustomer.org	161	true	2006-02-14	2013-05-26 14:49:45.738	1
8	154	2	Michele	Grant	michele.grant@sakilacustomer.org	158	true	2006-02-14	2013-05-26 14:49:45.738	1
9	530	2	Darryl	Ashcraft	darryl.ashcraft@sakilacustomer.org	536	true	2006-02-14	2013-05-26 14:49:45.738	1
10	493	1	Brent	Harkins	brent.harkins@sakilacustomer.org	498	true	2006-02-14	2013-05-26 14:49:45.738	1
11	542	2	Lornie	Tridao	lornie.tridao@sakilacustomer.org	548	true	2006-02-14	2013-05-26 14:49:45.738	1
12	566	1	Casey	Mena	casey.mena@sakilacustomer.org	572	true	2006-02-14	2013-05-26 14:49:45.738	1
13	166	2	Holly	Fox	holly.fox@sakilacustomer.org	190	true	2006-02-14	2013-05-26 14:49:45.738	1
14	128	1	Marjorie	Tucker	marjorie.tucker@sakilacustomer.org	132	true	2006-02-14	2013-05-26 14:49:45.738	1
15	466	1	Leo	Ebert	leo.ebert@sakilacustomer.org	471	true	2006-02-14	2013-05-26 14:49:45.738	1
16	494	2	Ramon	Choate	ramon.choate@sakilacustomer.org	499	true	2006-02-14	2013-05-26 14:49:45.738	1
17	178	2	Marion	Snyder	marion.snyder@sakilacustomer.org	182	true	2006-02-14	2013-05-26 14:49:45.738	1

Total rows: 599 of 599 Query complete 00:00:00.191 Ln 12, Col 1

If there were non-uniform for the tables above, I would use UPDATE and SET commands.

## Duplicate Date

## FILM

The screenshot shows the pgAdmin 4 interface with a query executed against the 'film' table. The query is a SELECT statement with GROUP BY and HAVING clauses. The results are displayed in a table with 13 columns and 1 row of data.

	film_id [PK] integer	title character varying (255)	release_year integer	language_id smallint	rental_duration smallint	rental_rate numeric (4,2)	length smallint	replacement_cost numeric (5,2)	rating mpaa_rating	last_update timestamp without time zone	special_features text[]	count bigint
1												

Total rows: 0 of 0 Query complete 00:00:00.097 Ln 25, Col 19

## CUSTOMER

The screenshot shows the pgAdmin 4 interface with a SQL query executed against the 'customer' table. The query is as follows:

```
4 address_id,  
5 activebool,  
6 create_date,  
7 last_update,  
8 active,  
9 COUNT(*)  
10 FROM customer  
11 GROUP BY customer_id,  
12 store_id,  
13 first_name,  
14 last_name,  
15 email,  
16 address_id,  
17 activebool,  
18 create_date,  
19 last_update,  
20 active  
21 HAVING COUNT(*) > 1;
```

The Data Output tab shows the following columns:

customer_id	store_id	first_name	last_name	email	address_id	activebool	create_date	last_update	active	count
[PK] integer	smallint	character varying (45)	character varying (45)	character varying (50)	smallint	boolean	date	timestamp without time zone	integer	bigint

Total rows: 0 of 0 Query complete 00:00:00.093 Ln 17, Col 7

There are no duplicate values in both film and customer. If there were, I would fix it by deleting the duplicate values or creating a VIEW table displaying only the duplicate values.

## Missing Data

## FILM

The screenshot shows the pgAdmin 4 interface with a SQL query executed against the 'film' table. The query is as follows:

```
1 SELECT *  
2 from film  
3 WHERE (film_id,  
4 title,  
5 release_year,  
6 language_id,  
7 rental_duration,  
8 rental_rate,  
9 length,  
10 replacement_cost,  
11 rating,  
12 last_update)  
13 is NULL
```

The Data Output tab shows the following columns:

film_id	title	description	release_year	language_id	rental_duration	rental_rate	length	replacement_cost	rating	last_update	special_features	fulltext
[PK] integer	character varying (255)	text	integer	smallint	smallint	numeric (4,2)	smallint	numeric (5,2)	mpaa_rating	timestamp without time zone	text[]	tsvector

Total rows: 0 of 0 Query complete 00:00:00.139 Ln 13, Col 11

## CUSTOMER

The screenshot shows the pgAdmin 4 interface. On the left is the 'Object Explorer' tree, which is expanded to show the 'public' schema under the 'postgres' database. The 'Tables (16)' folder is selected, and the 'actor' table is highlighted. The main pane displays a SQL query in the 'Query' tab:

```

1 SELECT *
2 FROM customer
3 WHERE (customer_id,
4        store_id,
5        first_name,
6        last_name,
7        email,
8        address_id,
9        activebool,
10       create_date,
11       last_update,
12       active)
13       IS NULL
14
15
16

```

Below the query editor, the 'Data Output' tab shows the results of the query. The results are displayed in a table with 11 columns: customer\_id, store\_id, first\_name, last\_name, email, address\_id, activebool, create\_date, last\_update, and active. The table is currently empty, showing only the column headers and their data types.

customer_id	store_id	first_name	last_name	email	address_id	activebool	create_date	last_update	active
[PK] integer	integer	character varying (45)	character varying (45)	character varying (50)	smallint	boolean	date	timestamp without time zone	integer

At the bottom of the interface, the status bar indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.096'.

There are no missing values but if there were, I could impute values there aren't a lot of missing data.

## 2. Summarize your data:

## Numerical – Film

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows a tree view of the database structure, including 'public' schema, 'Aggregates', 'Collations', 'Domains', 'FTS Configurations', 'FTS Dictionaries', 'FTS Parsers', 'FTS Templates', 'Foreign Tables', 'Functions', 'Materialized Views', 'Operators', 'Procedures', 'Sequences', and 'Tables (16)'. The 'Tables (16)' folder is expanded, showing tables like 'actor', 'address', 'category', 'city', 'country', 'customer', 'film', 'film\_actor', 'film\_category', 'inventory', 'language', 'payment', 'rental', 'staff', 'store', 'title', 'Trigger Functions', 'Types', 'Views', 'Subscriptions', 'postgres', 'Login/Group Roles', and 'Tablespaces'.

The main pane shows the 'Query Editor' for the 'Rockbuster/postgres@PostgreSQL 16' connection. The query is as follows:

```

1 SELECT MIN (release_year) AS min_release_year,
2 MAX (release_year) AS max_release_year,
3 AVG (release_year) AS avg_release_year,
4 MIN (rental_duration) AS min_rental_duration,
5 MAX (rental_duration) AS max_rental_duration,
6 AVG (rental_duration) AS avg_rental_duration,
7 MIN (rental_rate) AS min_rental_rate,
8 MAX (rental_rate) AS max_rental_rate,
9 AVG (rental_rate) AS avg_rental_rate,
10 MIN (length) AS min_length,
11 MAX (length) AS max_length,
12 AVG (length) AS avg_length,
13 MIN (replacement_cost) AS min_replacement_cost,
14 MAX (replacement_cost) AS max_replacement_cost
15 FROM film
16

```

The 'Data Output' tab shows the results of the query. The table has 13 columns: min\_release\_year (integer), max\_release\_year (integer), avg\_release\_year (numeric), min\_rental\_duration (smallint), max\_rental\_duration (smallint), avg\_rental\_duration (numeric), min\_rental\_rate (numeric), max\_rental\_rate (numeric), avg\_rental\_rate (numeric), min\_length (smallint), max\_length (smallint), and avg\_length (numeric). The results are as follows:

	min_release_year integer	max_release_year integer	avg_release_year numeric	min_rental_duration smallint	max_rental_duration smallint	avg_rental_duration numeric	min_rental_rate numeric	max_rental_rate numeric	avg_rental_rate numeric	min_length smallint	max_length smallint	avg_length numeric
1	2006	2006	2006.0000000000000000	3	7	4.9850000000000000	0.99	4.99	2.9800000000000000	46	185	115.272000

At the bottom, the status bar indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.114'.

## Non-numerical – Film

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane displays the database schema, with the 'film' table selected under the 'public' schema. The main query editor displays the following SQL query:

```
1 SELECT MODE () WITHIN GROUP (ORDER BY title) AS title_modal_value,  
2 MODE () WITHIN GROUP (ORDER BY special_features) AS special_features_modal_value  
3 FROM film  
4  
5  
6  
7  
8
```

The 'Data Output' pane at the bottom shows the results of the query:

	title_modal_value	special_features_modal_value
1	Academy Dinosaur	{Trailers,Commentaries,'Behind the Scenes'}

Total rows: 1 of 1 Query complete 00:00:00.128 Ln 5, Col 1

## Non-numerical – CUSTOMER

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane displays the database schema, with the 'customer' table selected under the 'public' schema. The main query editor displays the following SQL query:

```
1 SELECT MODE () WITHIN GROUP (ORDER BY first_name) AS first_name_modal_value,  
2 MODE () WITHIN GROUP (ORDER BY last_name) AS last_name_modal_value,  
3 MODE () WITHIN GROUP (ORDER BY email) AS email_modal_value,  
4 MODE () WITHIN GROUP (ORDER BY address_id) AS address_id_modal_value,  
5 MODE () WITHIN GROUP (ORDER BY activebool) AS activebool_modal_value  
6 FROM customer  
7  
8  
9  
10  
11
```

The 'Data Output' pane at the bottom shows the results of the query:

	first_name_modal_value	last_name_modal_value	email_modal_value	address_id_modal_value	activebool_modal_value
1	Jamie	Abney	aaron.selby@sakilacustomer.org	5	true

Total rows: 1 of 1 Query complete 00:00:00.067 Ln 11, Col 1

### 3. Reflect on your work:

SQL is very efficient for advanced data tasks and handling large volumes of data. One thing I love about SQL is it prevents you from accidentally deleting or editing data and it tells you when your query is incorrect. But it does require a lot of practice and learning. I find that

although it's great, I do struggle with remembering the syntax or knowing which and when to use them.

Excel on the other hand is user friendly but it's geared towards basic data tasks and handling smaller volumes of data. I think both tool is great depending on what you're using it for and I do really enjoy learning SQL and pretty fascinated on what it can do.