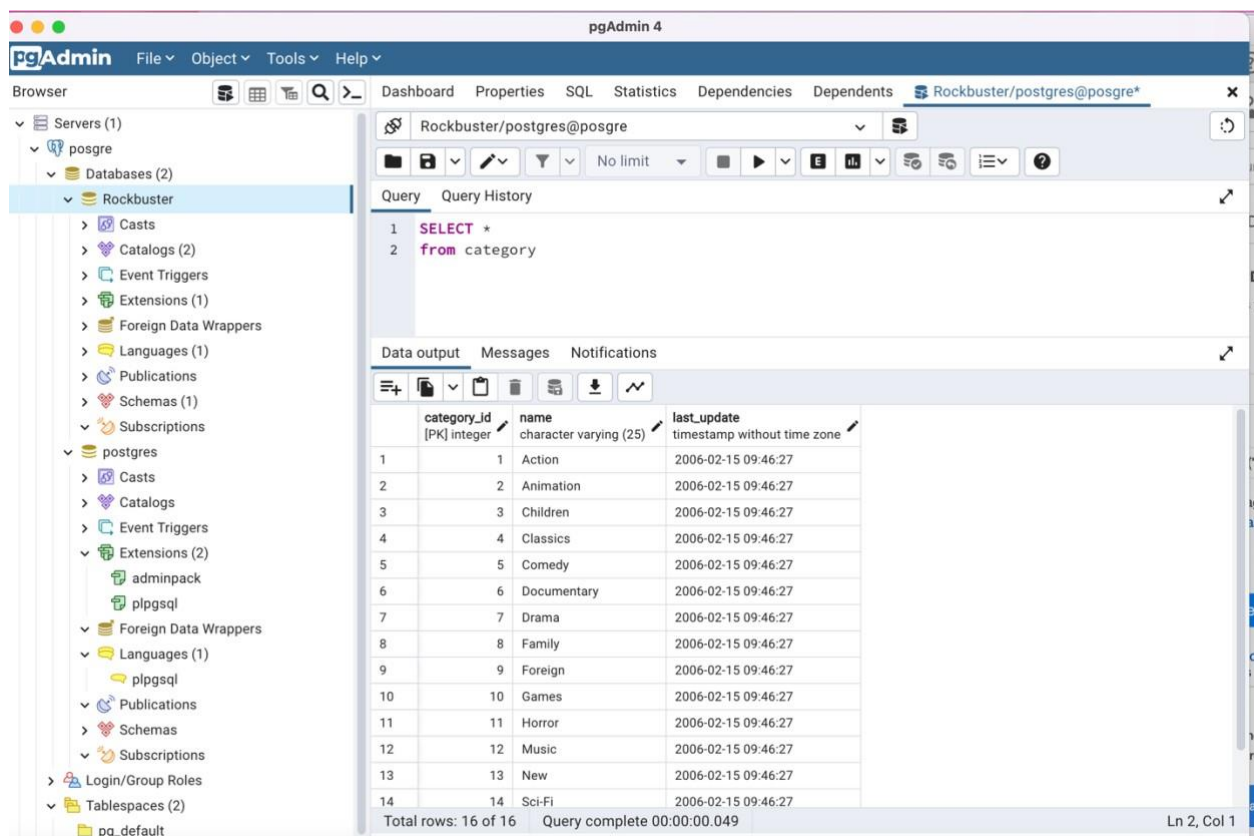


3.3: SQL for Data Analysts

Step 1:

Your first task is to find out what film genres already exist in the category table:

- Open pgAdmin 4, click the Rockbuster database, and open the Query Tool.
- Write a **SELECT** command to find out what film genres exist in the category table.
- Copy-paste the output into your answers document or write the answers out—it's up to you. Make sure to include the category ID for each genre.



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'Rockbuster' database selected under the 'posgre' server. The main pane shows a SQL query: `SELECT * from category`. Below the query, the 'Data output' tab is active, displaying a table with 16 rows. The table has three columns: 'category_id' (integer), 'name' (character varying (25)), and 'last_update' (timestamp without time zone). The data shows 14 rows of film genres, with the last two rows being 'New' and 'Sci-Fi'. The status bar at the bottom indicates 'Total rows: 16 of 16' and 'Query complete 00:00:00.049'.

category_id	name	last_update
1	Action	2006-02-15 09:46:27
2	Animation	2006-02-15 09:46:27
3	Children	2006-02-15 09:46:27
4	Classics	2006-02-15 09:46:27
5	Comedy	2006-02-15 09:46:27
6	Documentary	2006-02-15 09:46:27
7	Drama	2006-02-15 09:46:27
8	Family	2006-02-15 09:46:27
9	Foreign	2006-02-15 09:46:27
10	Games	2006-02-15 09:46:27
11	Horror	2006-02-15 09:46:27
12	Music	2006-02-15 09:46:27
13	New	2006-02-15 09:46:27
14	Sci-Fi	2006-02-15 09:46:27

Step 2:

You're ready to add some new genres! Write an **INSERT** statement to add the following genres to the category table: Thriller, Crime, Mystery, Romance, and War:

- Copy-paste your **INSERT** commands into your answers document.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane shows a tree view with 'Servers (1)' containing 'posgre', which has 'Databases (2)' including 'Rockbuster'. The 'Query' tab is active, showing a SQL query:

```

1 INSERT INTO category (name)
2 VALUES ('Thriller'), ('Crime'), ('Mystery'), ('Romance'), ('War');
3 SELECT * FROM category
4

```

Below the query, the 'Data output' tab shows the results of the query. The table has three columns: 'category_id [PK] integer', 'name character varying (25)', and 'last_update timestamp without time zone'. The results are as follows:

category_id [PK] integer	name character varying (25)	last_update timestamp without time zone
8	Family	2006-02-15 09:46:27
9	Foreign	2006-02-15 09:46:27
10	Games	2006-02-15 09:46:27
11	Horror	2006-02-15 09:46:27
12	Music	2006-02-15 09:46:27
13	New	2006-02-15 09:46:27
14	Sci-Fi	2006-02-15 09:46:27
15	Sports	2006-02-15 09:46:27
16	Travel	2006-02-15 09:46:27
17	War	2022-09-02 20:55:06.477843
18	Crime	2022-09-02 20:55:06.477843
19	Mystery	2022-09-02 20:55:06.477843
20	Romance	2022-09-02 20:55:06.477843
21	War	2022-09-02 20:55:06.477843

At the bottom, it says 'Total rows: 99 of 99' and 'Query complete 00:00:00.085'.

- The **CREATE** statement below shows the constraints on the category table. Write a short paragraph explaining the various constraints that have been applied to the columns. What do these constraints do exactly? Why are they important?

```

CREATE TABLE category
(
    category_id integer NOT NULL DEFAULT nextval('category_category_id_seq'::regclass),
    name text COLLATE pg_catalog."default" NOT NULL,
    last_update timestamp with time zone NOT NULL DEFAULT now(), CONSTRAINT
    category_pkey PRIMARY KEY (category_id)
);

```

NOT NULL this ensures that the column cannot have empty or missing values

category_id : (data type is integer) Value cannot be null

name: (data type is text) Values cannot be null last_update: (data type is timestamp with time zone) value cannot be null

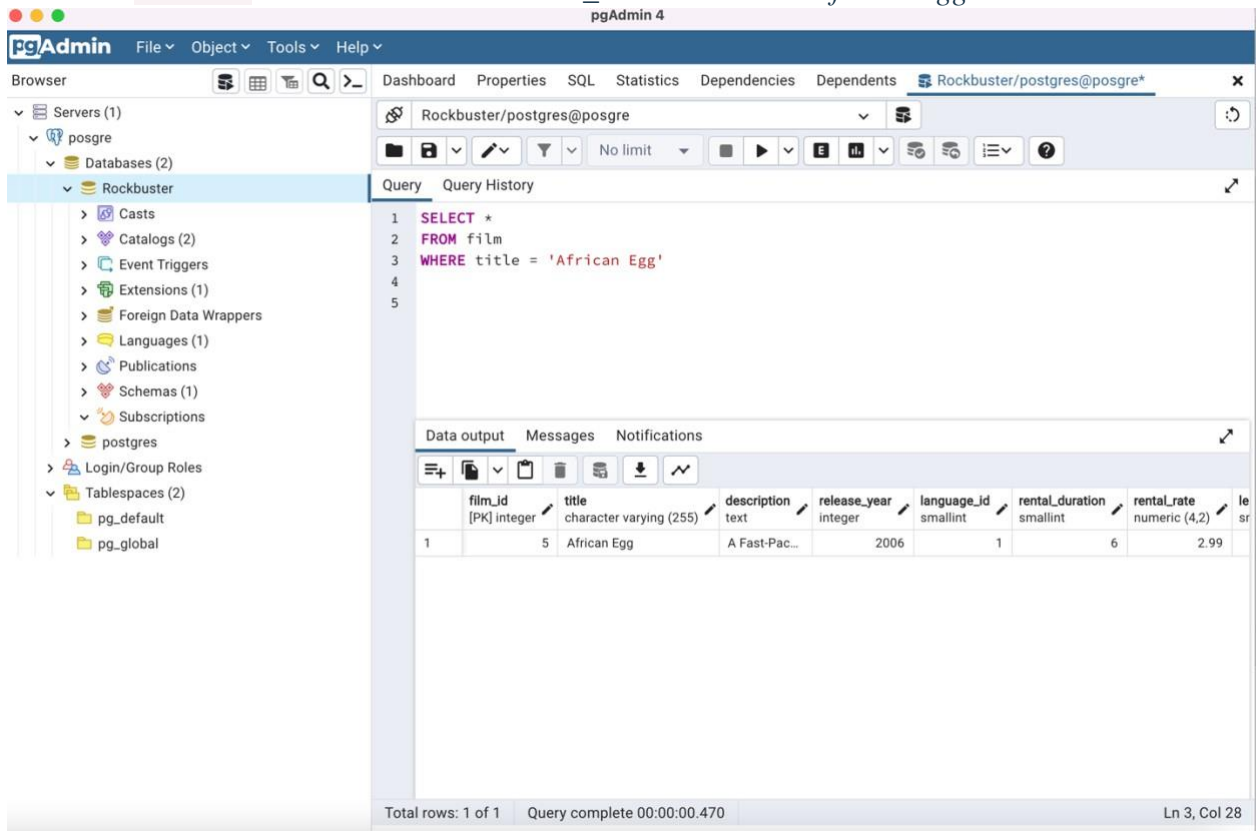
PRIMARY KEY is a unique identifier for each record in a table. `Category_pkey` which is the `category_id` is set as the primary key.

Constraints are important because they can help you make sure that the values in each column are consistently formatted. They can also help you make sure values in a column are unique, not null, or even check for values that don't belong.

Step 3:

The genre for the movie *African Egg* needs to be updated to thriller. Work through the steps below to make this change:

- Write the **SELECT** statement to find the `film_id` for the movie *African Egg*.



The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree is expanded to 'Rockbuster'. The 'Query' tab is active, showing the following SQL query:

```
1 SELECT *
2 FROM film
3 WHERE title = 'African Egg'
4
5
```

Below the query, the 'Data output' tab shows the results of the query. The results are displayed in a table with the following columns and values:

film_id [PK] integer	title character varying (255)	description text	release_year integer	language_id smallint	rental_duration smallint	rental_rate numeric (4,2)	length integer
1	5	African Egg	A Fast-Pac...	2006	1	6	2.99

The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.470'.

- Once you have the `film_ID` and `category_ID`, write an **UPDATE** command to change the category in the `film_category` table (not the `category` table). Copy-paste this command into your answers docum

Help ▾

Dashboard Properties SQL Statistics Dependencies Dependents Rockbuster/postgres@posgre*

Rockbuster/postgres@posgre

No limit

Query Query History

```
1 SELECT * FROM film WHERE title = 'African Egg';
2 SELECT category_id FROM film_category WHERE film_id = 5;
3 UPDATE film_category SET category_id = 32 WHERE film_id = 5;
4
```

Data output Messages Notifications

UPDATE 1

Query returned successfully in 53 msec.

No limit

Query Query History

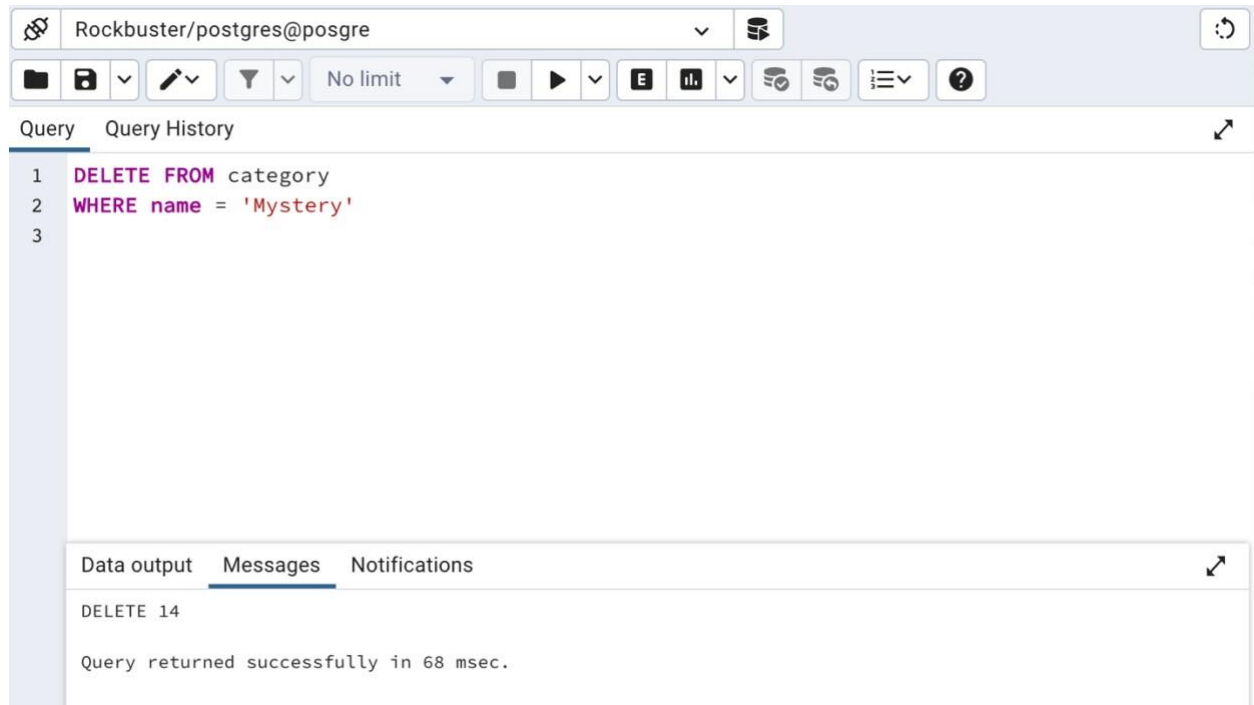
```
1 SELECT * FROM film_category WHERE film_id = 5
2
3
```

Data output Messages Notifications

	film_id [PK] smallint	category_id [PK] smallint	last_update timestamp without time zone
1	5	32	2022-09-03 08:27:23.664716

Step 4:

Since there aren't many movies in the mystery category, you and your manager decide to remove it from the category table. Write a **DELETE** command to do so and copy-paste it into your answers document.



Step 5:

Based on what you've learned so far, think about what it would be like to complete steps 1 to 4 with Excel instead of SQL. Are there any pros and cons to using SQL? Write a paragraph explaining your answer.

when using Excel, I needed to take more steps to replace data, find the desired table, use the filter, select the desired category and replace it. SQL searches for the required table automatically. Some easier steps for example the updating of the category could be easily achieved with excel's search and replace function.

Bonus Task

The SQL query below contains some typos. See if you can fix it based on what you've learned so far about SQL and data types; then try running it in pgAdmin 4. If the query works, copy it into your Answers 3.3 document.

If you get this you're a SQL champ!

```
CREATE TBL 3EMPLOYEES
```

```
{  
employee_id VARINT(30) NOT  
EMPTY name VARCHAR(50),  
contact_number VARCHAR(30) ,  
designation_id INT,  
last_update TIMESTAMP NOT NULL DEF now()  
CONSTRAIN employee_pkey PRIMARY KEY (employee_id)  
}
```

```
CREATE TABLE employees
```

```
(  
employee_id VARCHAR(30) NOT NULL,  
name VARCHAR(50),  
contact_number VARCHAR(30),  
designation_id INTEGER,  
last_update TIMESTAMP NOT NULL DEFAULT now(),  
CONSTRAINT employee_pkey PRIMARY KEY (employee_id)
```

The screenshot shows a PostgreSQL client interface with the following components:

- Top Bar:** Displays the connection string "Rockbuster/postgres@posgre" and a refresh button.
- Toolbar:** Contains icons for file operations, filters, execution (play button), and other database management functions.
- Query Editor:** Shows a SQL query:

```
1  
2 SELECT * FROM employees;  
3
```
- Data Output Panel:** Displays the results of the query in a table format with the following columns:

employee_id	name	contact_number	designation_id	last_update
[PK] character varying (30)	character varying (50)	character varying (30)	integer	timestamp without time zone