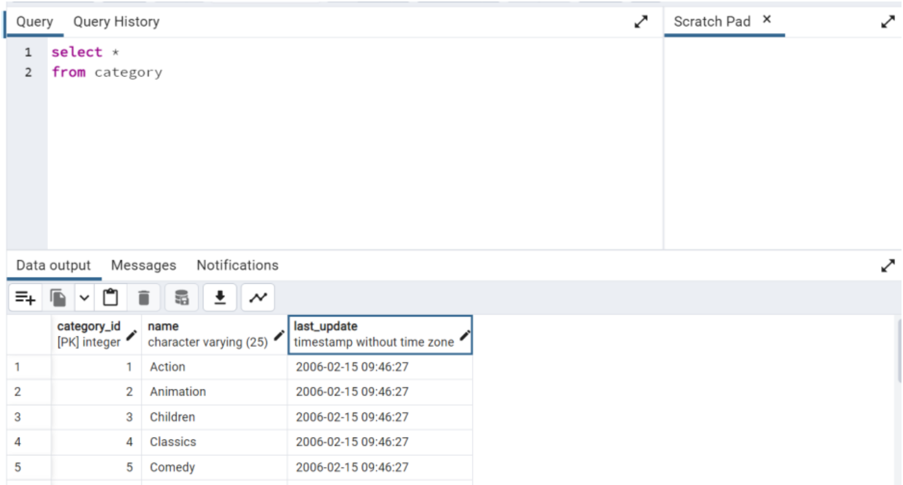
3.3: SQL for Data Analysts

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.



```
select *
from category
```

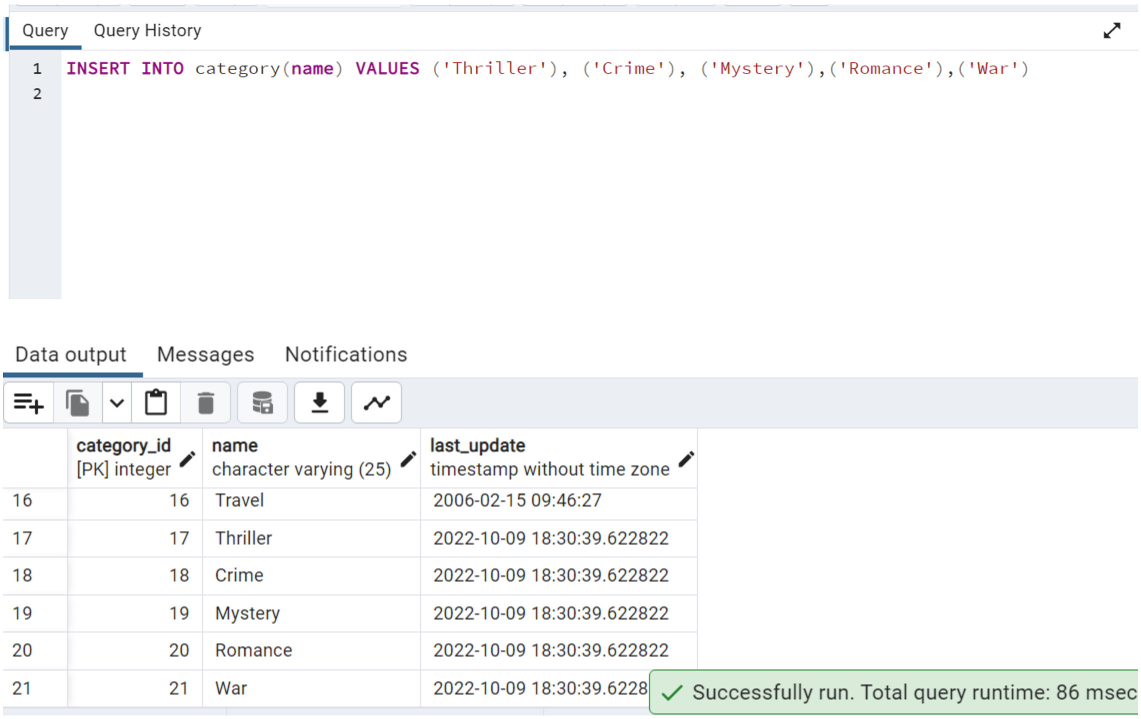
	category_id [PK] integer	name character varying (25)	last_update timestamp without time zone
1	1	Action	2006-02-15 09:46:27
2	2	Animation	2006-02-15 09:46:27
3	3	Children	2006-02-15 09:46:27
4	4	Classics	2006-02-15 09:46:27
5	5	Comedy	2006-02-15 09:46:27

```
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"
15      "Sports"        "2006-02-15 09:46:27"
16      "Travel"        "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.



```
INSERT INTO category(name) VALUES ('Thriller'), ('Crime'), ('Mystery'),('Romance'),('War')
```

	category_id [PK] integer	name character varying (25)	last_update timestamp without time zone
16	16	Travel	2006-02-15 09:46:27
17	17	Thriller	2022-10-09 18:30:39.622822
18	18	Crime	2022-10-09 18:30:39.622822
19	19	Mystery	2022-10-09 18:30:39.622822
20	20	Romance	2022-10-09 18:30:39.622822
21	21	War	2022-10-09 18:30:39.6228

✓ Successfully run. Total query runtime: 86 msec

- 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)
);
```

Two main constraints have been used on this statement: NOT NULL and PRIMARY KEY.

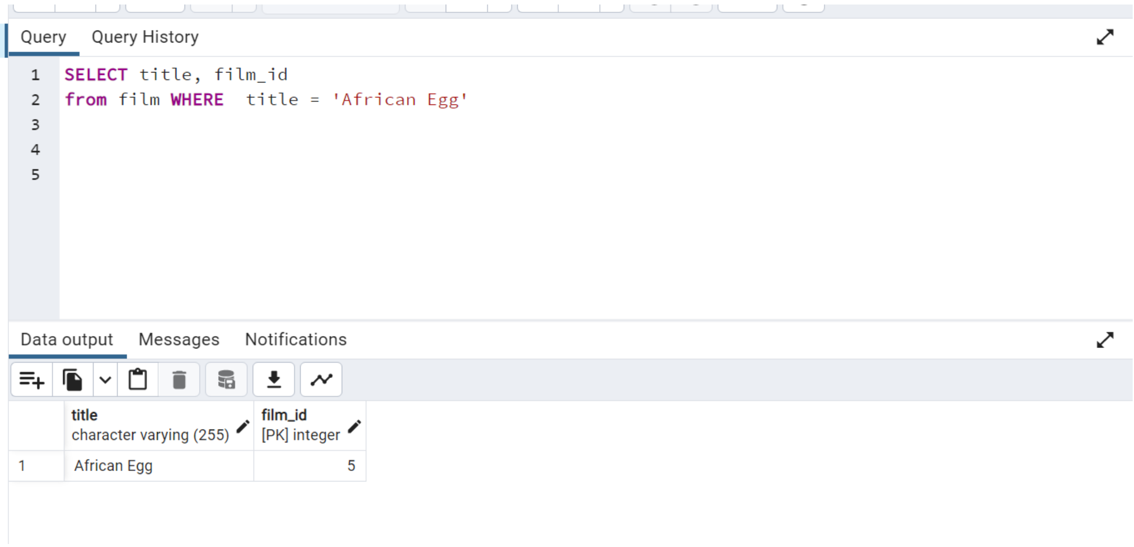
- NOT NULL: it ensures that a column can't have any empty or missing values.
- PRIMARY KEY: it gives each record in a table a unique ID, and it´s used to combine information from 2 or more tables

Both of them are important in order to keep an order on the formats of the values, and avoiding duplicates which make the combination impossible, or could also avoid values not permitted

Step 3:

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

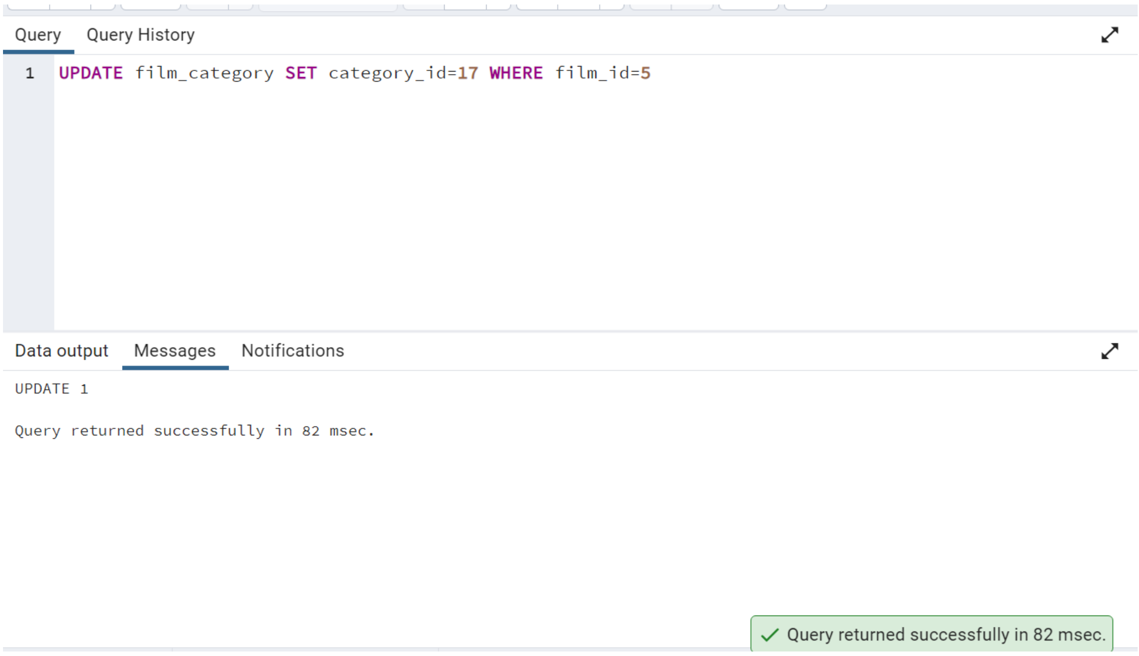
- Write the SELECT statement to find the film_id for the movie *African Egg*.



```
SELECT title, film_id
from film WHERE  title = 'African Egg'
```

	title character varying (255)	film_id [PK] integer
1	African Egg	5

- 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 document.

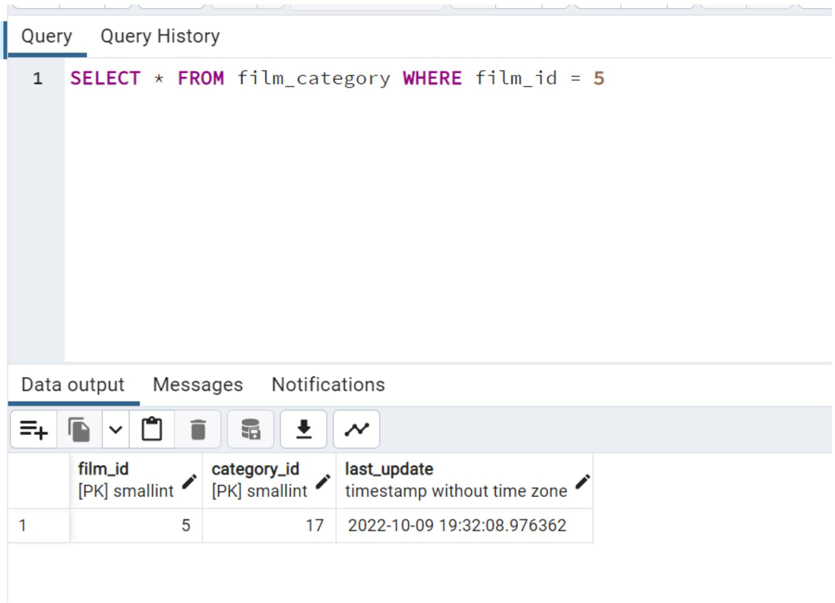


```
UPDATE film_category SET category_id=17 WHERE film_id=5
```

```
UPDATE 1

Query returned successfully in 82 msec.
```

✓ Query returned successfully in 82 msec.

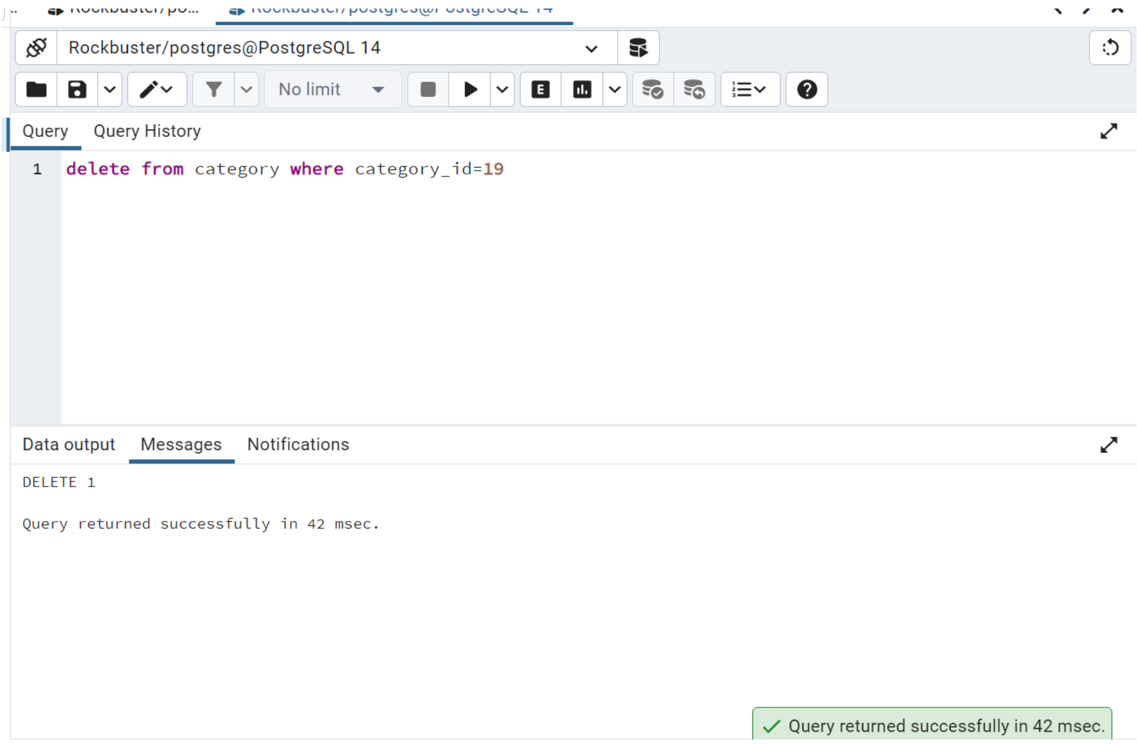


```sql
SELECT * FROM film_category WHERE film_id = 5
```

	film_id [PK] smallint	category_id [PK] smallint	last_update timestamp without time zone
1	5	17	2022-10-09 19:32:08.976362

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.



```sql
delete from category where category_id=19
```

```
DELETE 1

Query returned successfully in 42 msec.
```

Query returned successfully in 42 msec.

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.

As I was kind of skilled on Excel, and I´m starting with SQL, I´m still more confident on Excel of course. I do yet queries with fear and with errors. Naturally, the answers seem to be faster, but it´s only when you have understood the whole components of the data set and know where to look for.

I guess and I want to think that once we get more confident with SQL we are not going to go back to excel ever 😊

Step 6:

Save your "Answers 3.3" document as a PDF and upload it here for your tutor to review.

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 INT,
last_update TIMESTAMP NOT NULL DEFAULT now()
CONSTRAINT employee_pkey PRIMARY KEY (employee_id)
}
```