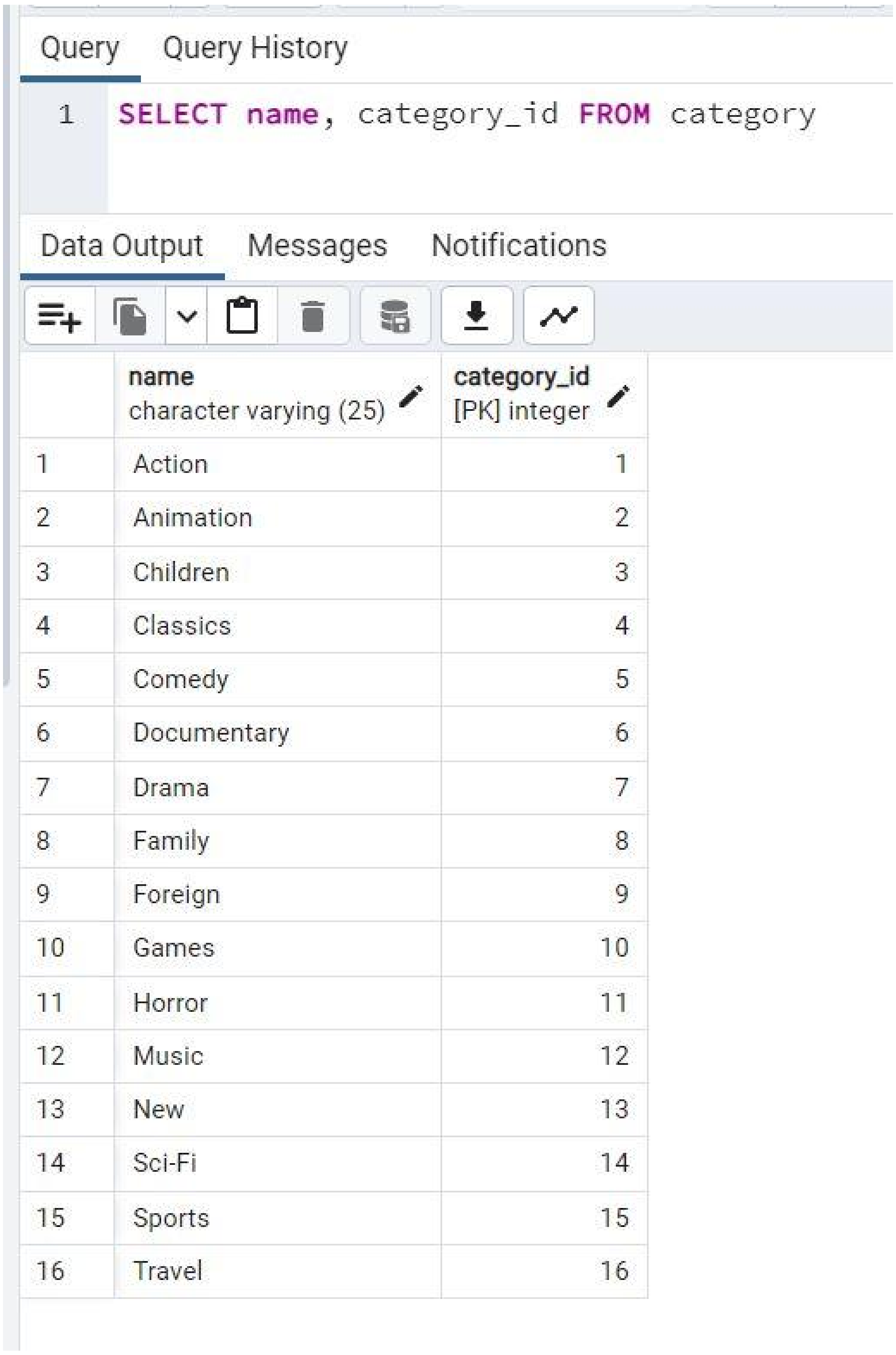
Step 1:

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

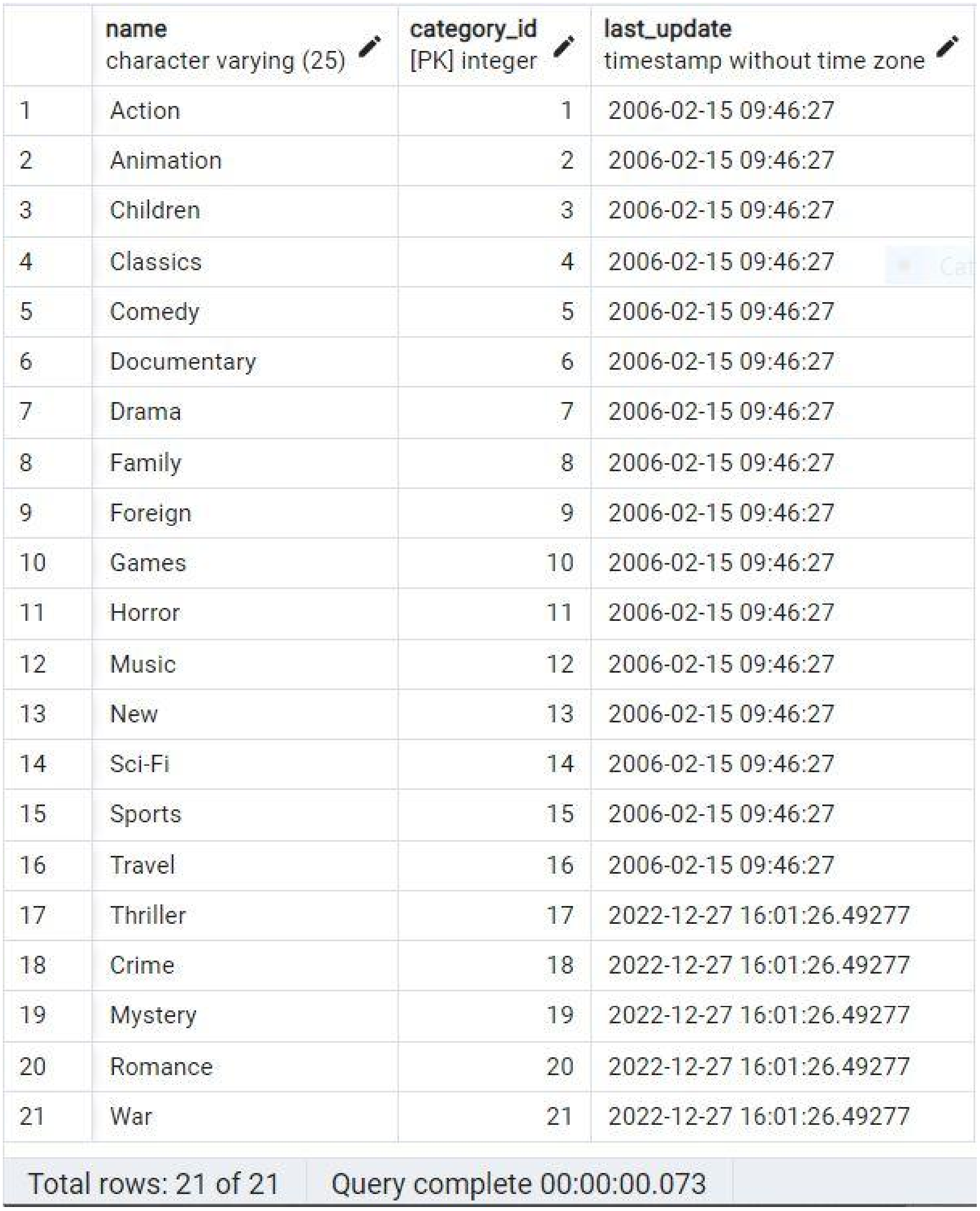


```sql
SELECT name, category_id FROM category
```

	name character varying (25)	category_id [PK] integer
1	Action	1
2	Animation	2
3	Children	3
4	Classics	4
5	Comedy	5
6	Documentary	6
7	Drama	7
8	Family	8
9	Foreign	9
10	Games	10
11	Horror	11
12	Music	12
13	New	13
14	Sci-Fi	14
15	Sports	15
16	Travel	16

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.

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



	name character varying (25)	category_id [PK] integer	last_update timestamp without time zone
1	Action	1	2006-02-15 09:46:27
2	Animation	2	2006-02-15 09:46:27
3	Children	3	2006-02-15 09:46:27
4	Classics	4	2006-02-15 09:46:27
5	Comedy	5	2006-02-15 09:46:27
6	Documentary	6	2006-02-15 09:46:27
7	Drama	7	2006-02-15 09:46:27
8	Family	8	2006-02-15 09:46:27
9	Foreign	9	2006-02-15 09:46:27
10	Games	10	2006-02-15 09:46:27
11	Horror	11	2006-02-15 09:46:27
12	Music	12	2006-02-15 09:46:27
13	New	13	2006-02-15 09:46:27
14	Sci-Fi	14	2006-02-15 09:46:27
15	Sports	15	2006-02-15 09:46:27
16	Travel	16	2006-02-15 09:46:27
17	Thriller	17	2022-12-27 16:01:26.49277
18	Crime	18	2022-12-27 16:01:26.49277
19	Mystery	19	2022-12-27 16:01:26.49277
20	Romance	20	2022-12-27 16:01:26.49277
21	War	21	2022-12-27 16:01:26.49277

Total rows: 21 of 21 Query complete 00:00:00.073

- 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 Constraint:** This constraint makes sure that no columns have any missing numbers.
    - category_id : (integer) there can be no null value.
    - name: (text) there can be no null value.
    - last_update: (timestamp with time zone) there can be no null value.

- There is also a **DEFAULT** command that gives a default value to an affected field in order to avoid null values

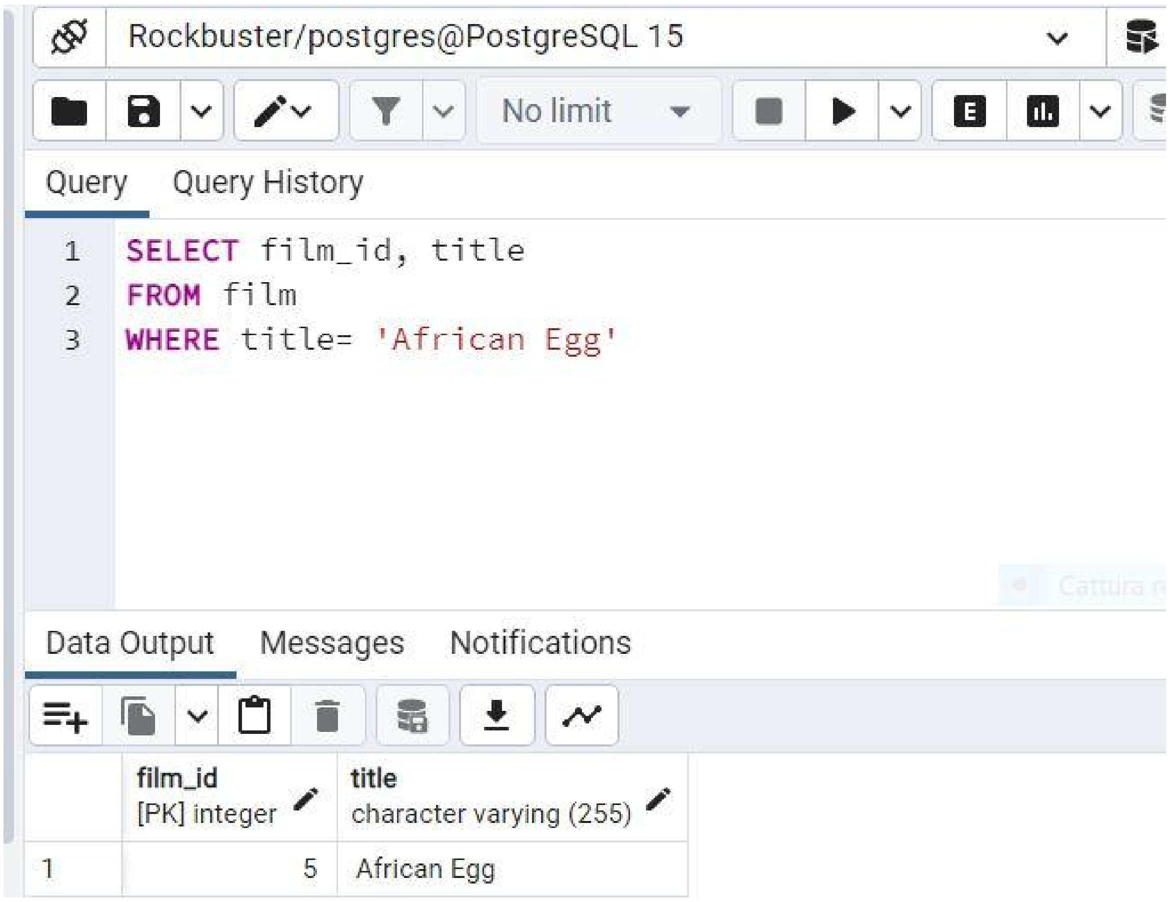
- **PRIMARY KEY Constraint**: it transforms the entire column's values into a primary key, which serves as a single point of identification for an entire table row.

- **Constraints** describe what type of data a table or column can accept and are often applied when a table is created. Constraints, when used appropriately, make searching the database faster and easier. In some circumstances, they may even serve as a data quality check.

Step 3:

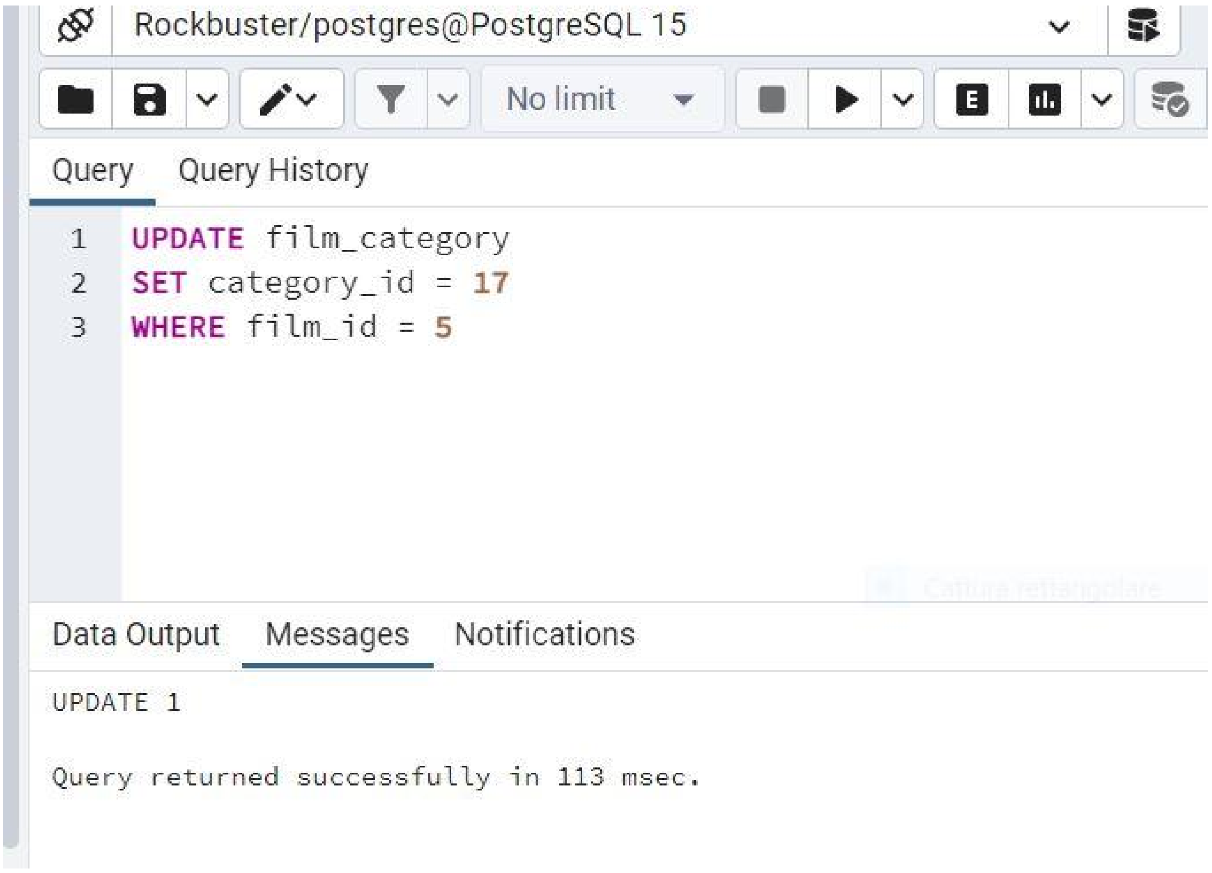
The genre for the movie *African Egg* needs to be updated to thriller. Work through the steps below to make this change:

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



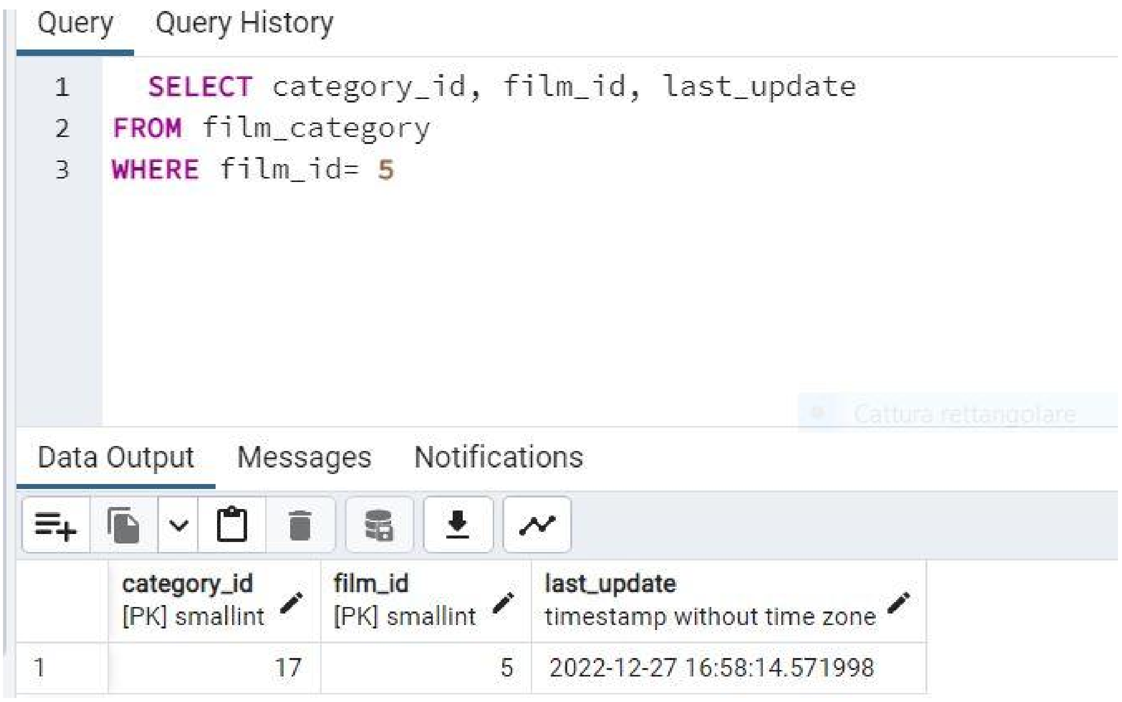
```sql
SELECT film_id, title
FROM film
WHERE title= 'African Egg'
```

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



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

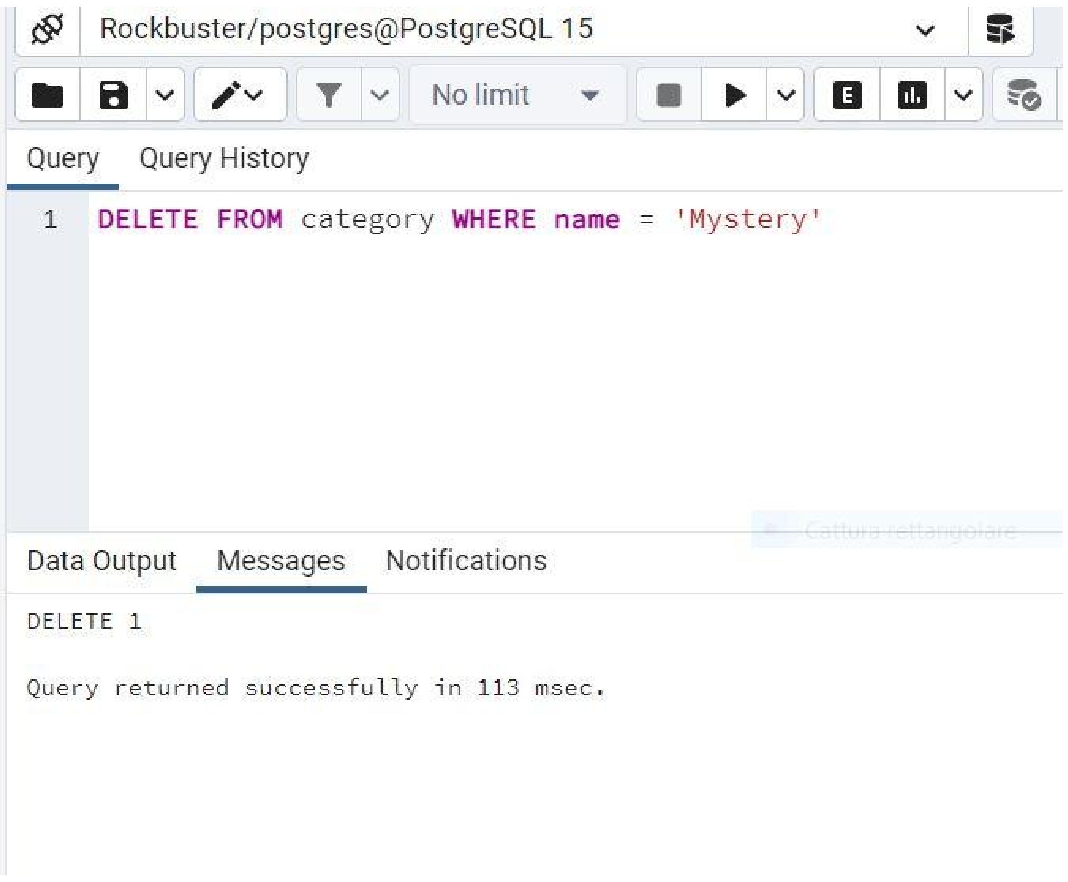
```
UPDATE 1

Query returned successfully in 113 msec.
```



```sql
  SELECT category_id, film_id, last_update
FROM film_category
WHERE film_id= 5
```

	category_id [PK] smallint	film_id [PK] smallint	last_update timestamp without time zone
1	17	5	2022-12-27 16:58:14.571998

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 name = 'Mystery'
```

```
DELETE 1

Query returned successfully in 113 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.

In Excel, I would have manually inserted all the genres, updated the genre with the Find & Replace command, and manually removed the genre Mystery. I would have also utilized pivot tables to obtain all the necessary data.

Everything was much simpler to do using SQL, but the disadvantage is that it requires some expertise and familiarity with commands. Excel is simpler, but it takes more time.

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.

```
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)
) ;
```