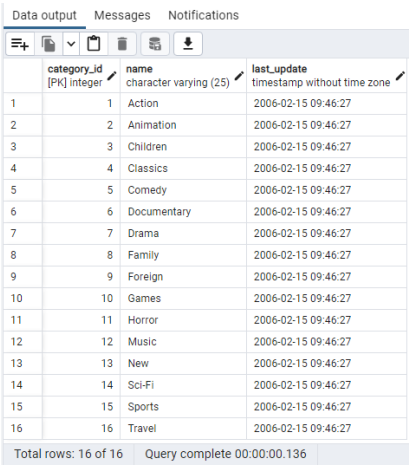
3.3 Data SQL for Data Analysts

1. **Step 1 Your first task is to find out what film genres already exist in the category table:**

   SELECT * FROM category



Data output | Messages | Notifications

	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
6	6	Documentary	2006-02-15 09:46:27
7	7	Drama	2006-02-15 09:46:27
8	8	Family	2006-02-15 09:46:27
9	9	Foreign	2006-02-15 09:46:27
10	10	Games	2006-02-15 09:46:27
11	11	Horror	2006-02-15 09:46:27
12	12	Music	2006-02-15 09:46:27
13	13	New	2006-02-15 09:46:27
14	14	Sci-Fi	2006-02-15 09:46:27
15	15	Sports	2006-02-15 09:46:27
16	16	Travel	2006-02-15 09:46:27

Total rows: 16 of 16 Query complete 00:00:00.136

2. **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')
   - NOT NULL: Missing value is no allowed.
     - Category_id: integer, numeric integer. DEFAULT, when a record with missing values is inserted, it will be assigned default value.
     - Name: text, character string with unlimited length.
     - last_update: DEFAULT, when a record with missing values is inserted, it will be assigned default value, which is now (time when the record is made).

   PRIMARY KEY: category_id is assigned as primary key and given a unique ID which can't contain any null or duplicate values.

3. **The genre for the movie African Egg needs to be updated to thriller.**

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

     SELECT film_id FROM film
     WHERE title = '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.

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

	film_id [PK] integer
1	5

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

	film_id [PK] smallint	category_id [PK] smallint	last_update timestamp without time zone
1	5	8	2006-02-15 10:07:09

```
UPDATE film_category
SET category_id = 3
WHERE film_id = 5
```

	film_id [PK] smallint	category_id [PK] smallint	last_update timestamp without time zone
1	5	3	2022-08-06 23:12:34.488753

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.**

   ```
   DELETE FROM category
   WHERE name = 'Mystery'
   ```

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.**

   Using Excel will be cumbersome especially with large scale data with many interconnected tables. We have to sort out the intended row and update it manually, which might prone to mistake. Additionally, imagine that we have many tables and working with Excel, it means that we have to work with multiple sheets or make it into one big table. However, Excel could be helpful when dealing with small data, simple and most people are already familiar with. SQL needs also access to the database (not all people are given access to it), while Excel does not need it.

Bonus Task

```
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)
);
```