Task 3 SQL commands

Answers 3.3

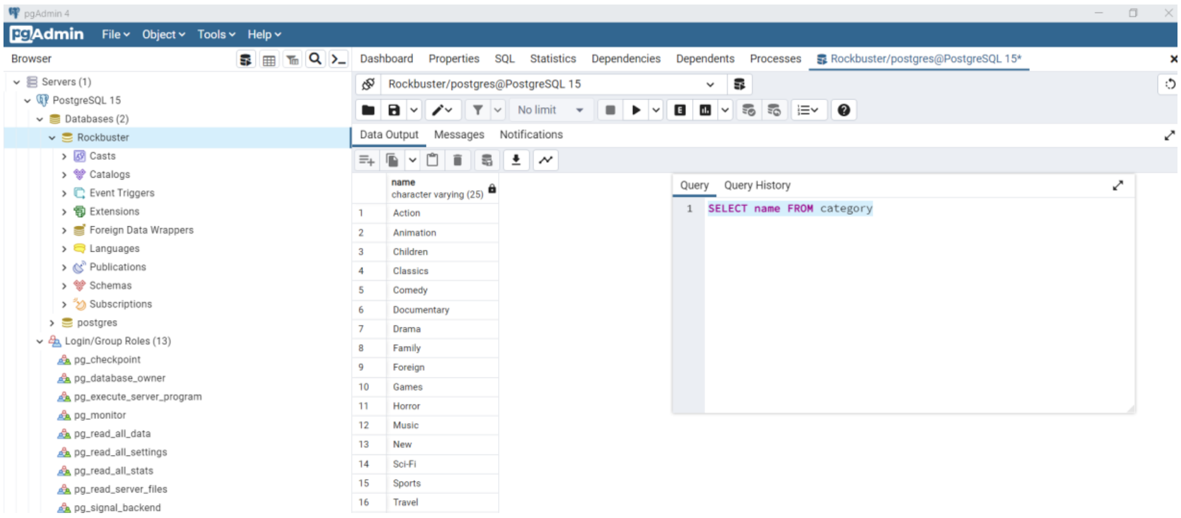
Context: Your line manager has told you they'd like to improve the movie search filters on the Rockbuster app and website. After thinking about the best way to do this, you decide to add more film categories so users can browse more genres they're interested in.

1. Find out what film genres already exist in the category table:

- Command used:

  SELECT name FROM category

Output:



```
SELECT name FROM category
```

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

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.

Commands used:

INSERT INTO category (name) VALUES (' Thriller ')

INSERT INTO category (name) VALUES ('Crime')

INSERT INTO category (name) VALUES (' Mystery')

INSERT INTO category (name) VALUES ('Romance')

INSERT INTO category (name) VALUES ('War')

Output:

"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"
15	"Sports"	"2006-02-15 09:46:27"
16	"Travel"	"2006-02-15 09:46:27"
17	"Thriller"	"2022-11-07 16:40:16.987543"
20	"Mystery"	"2022-11-07 16:44:54.756608"
21	"Romance"	"2022-11-07 16:45:03.305382"
22	"War"	"2022-11-07 16:45:11.378194"
23	"Crime"	"2022-11-07 16:49:48.483372"

- 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)
);
```

The statement below has the following constraints:

NOT NULL is applied to column headings category_id, name and last_update implying that for each of the referred columns null values cannot be accepted.

PRIMARY KEY, which means that the category_id column is a primary key, therefore a unique identifier that cannot be duplicated in the column.

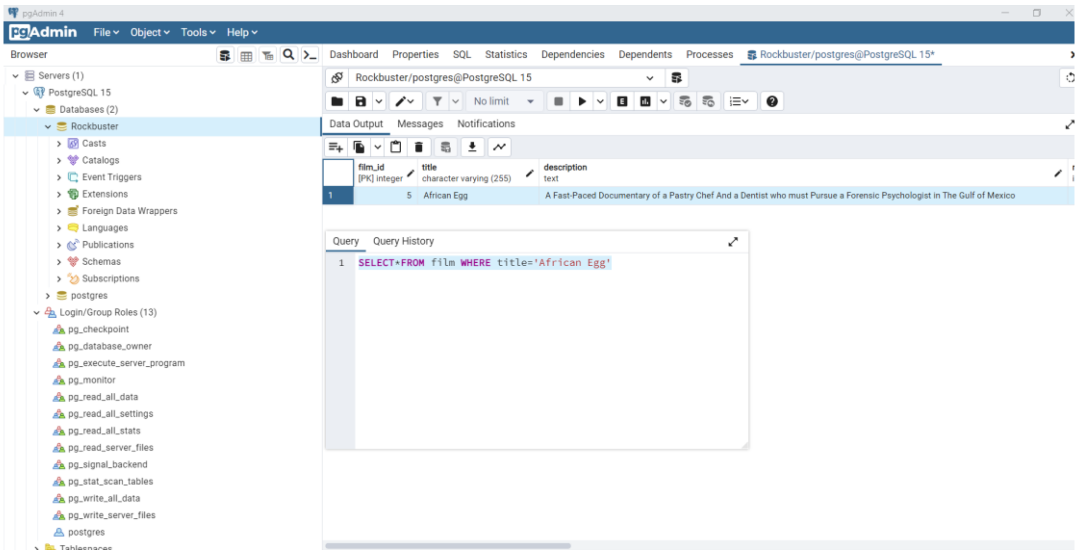
3. The genre for the movie *African Egg* needs to be updated to thriller. Work through the steps below to make this change:

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

Command used:

SELECT*FROM film WHERE title='African Egg'

Output:



```
   film_id        title                      description
   [PK] integer   character varying (255)    text
1  5              African Egg                A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico
```

```
SELECT*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.

film_ID for African Egg is 5

category_id for the African Egg is 8

category_id for thriller is 17

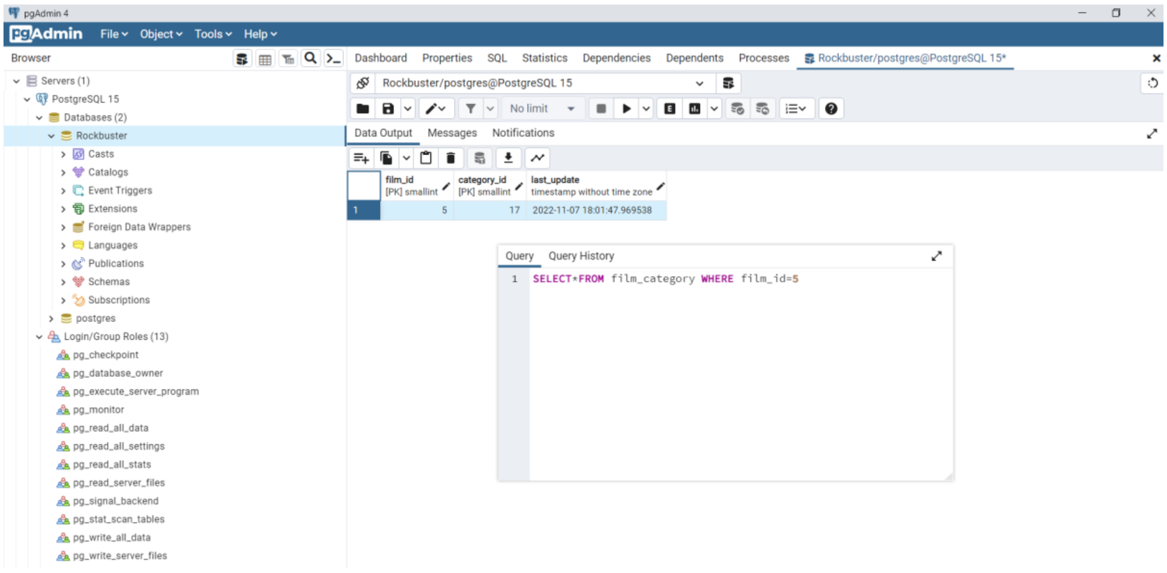
Command used:

UPDATE film_category SET category_id=17 WHERE film_id=5

Output:

UPDATE 1 Query returned successfully in 2 secs 231 msec.

Also, the print screen below, shows I looked into the category table, to see if the category had been updated to 17:



	film_id [PK] smallint	category_id [PK] smallint	last_update timestamp without time zone
1	5	17	2022-11-07 18:01:47.969538

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

Step 4:

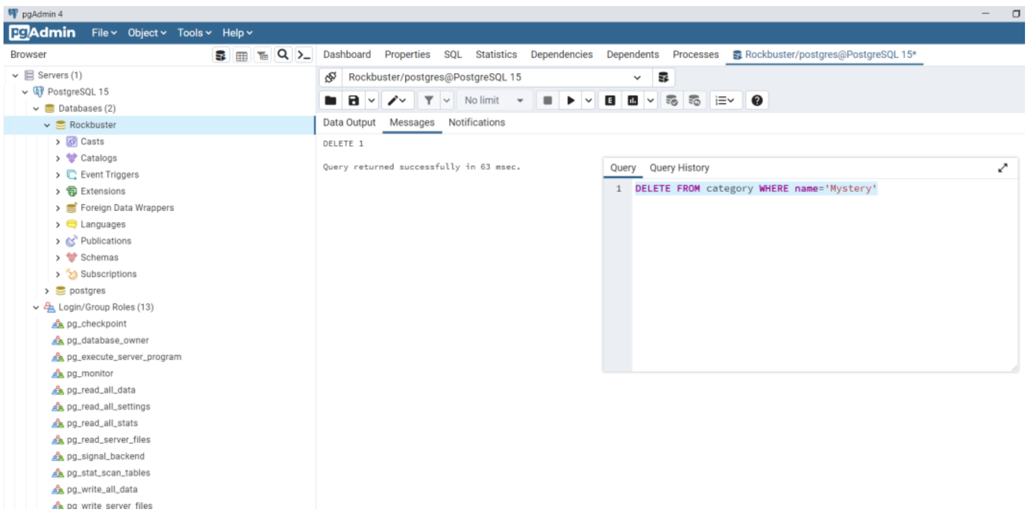
Since there aren’t many movies in the mystery category, you and your manager decide to remove it from the category table. Write a DELETE command to do so and copy-paste it into your answers document.

Command used:

DELETE FROM category WHERE name='Mystery'

Remove mystery movies category

‘Mystery’ was deleted:



```
DELETE 1

Query returned successfully in 63 msec.
```

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

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.

To complete step 1 in excel would as easy and in SQL, I would just use the filter tool and know how many genres existed.

Now, to insert new genres, I would have to make sure it would not compromise the data already in the excel table, because unlike SQL where I have a set of interrelated tables, in excel I would be likely working with one table. I would probably have to the VLOOKUP tool to then connect the data to new categories, so would be a much more complicated process.

For step 3, I could filter out the name of the movie and just change its genre in the spreadsheet. It would be pretty simple, while in SQL I had to find information in different tables (i.e., the film_id and the category table) to then make a change in another table (i.e., film_category).

Finally for step 4, I would have to delete those rows in excel after filtering them out, yet in SQL a simple command does the task.

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 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)
}
```

The typos I see in the code above are the following:

Should be CREATE TABLE – not CREATE TBL

The table name should be lower case, no in capital letters (e.g. 3EMPLOYEES) and it should not start with a number. The first constraint should be NOT NULL instead of NOT EMPTY. Finally, the brackets are also wrong should be these ones: ()

There are other things that did not work in the code, this code below worked:

CREATE TABLE employees

(employee_name VARCHAR(50),

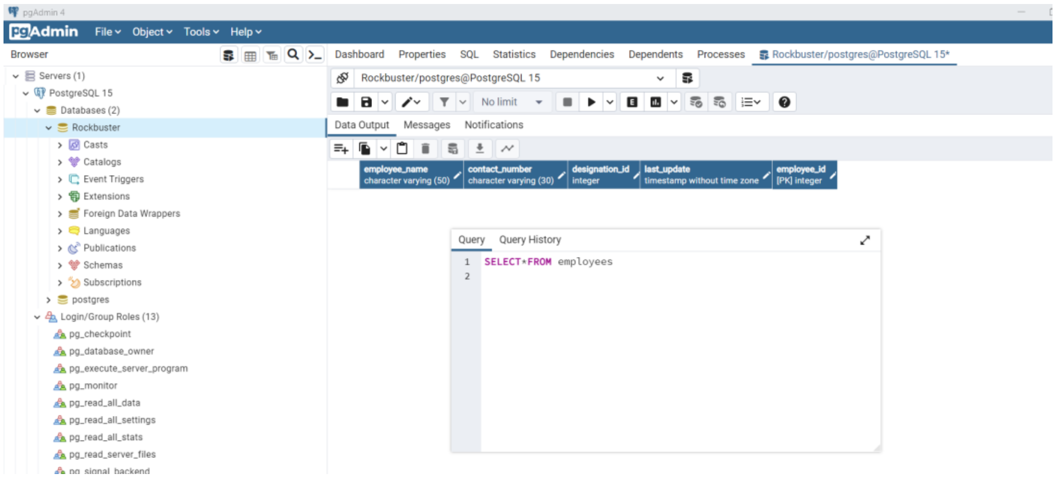
contact_number VARCHAR(30) ,

designation_id INT,

last_update TIMESTAMP NOT NULL,

employee_id INT PRIMARY KEY)

And this table was the result:



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

```
SELECT*FROM employees
```