Databases & SQL for Analysts

Task 3.3: SQL for Data Analysts

Ngawang Dhundup

LINK NAME:

Directions:

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.

	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 00:00:00.086

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')

- 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 NOTNULL constraint prevents any empty values
- The PRIMARY KEY gives each record a unique id

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 *
- FROM Film
- WHERE title = ('African Egg')
- Film_id =5
- category_id = 8

- 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;

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.

- 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.

- With Excel it would be harder to insert the different genres and make them match each table. I believe the ease of sql is how you can link between multiple tables and change accordingly. Changing the genre for the "African Egg" movie was a bit more difficult in sql in excel I would imagine since in excel it is as easy as filtering for the title and just changing it. Deleting a whole genre was much easier in sql however,. In sql pulling up multiple tables to double check is so much easier compared to excel.

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_3
 {
employee_id VARINT(30) NOT NULL DEFAULT
name VARCHAR(255),
contact_number VARCHAR(30) ,
designation_id INT,
last_update TIMESTAMP NOT NULL DEFAULT now()
CONSTRAINT employee_pkey PRIMARY KEY (employee_id)
};
```