3.3: SQL for Data Analysts

- Write a SELECT command to find out what film genres exist in the category table.

category_id	Name
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

- Copy-paste your INSERT commands into your answers document.

Answer:

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')

Or we can also write this query as:

```
INSERT INTO category(category_id,name) Values(17,'Thriller'),(18,'Crime'),
        (19,'Mystery'),(20,'Romance'),(21,'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)
);
```

Answer: The NOT NULL constraints have been applied on category_id which means this column must not contain empty and missing values. This is important because category_id is a primary key which can not contain any duplicate values. The category_id has data type called integer which means we can only assign integer values. The combination of these constraints ensures that the 'category_id' column is an essential location for finding specific information because it will always have a value and be unique for each row in the table. The 'DEFAULT' constraints used to specify a default value to be automatically entered if no other value is provided.

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

Answer:

```
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.

Answer:

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

- Write a DELETE command to do so and copy-paste it into your answers document.

Answer:

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

- 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.

Answer:

If I would have used excel to complete steps 1 to 4 then this will take more time and more steps to complete the task. In SQL we can get results in a single line of command within less than milliseconds. For example, updating a film's category_id. While it is relatively simple to find the film and update the category manually in Excel, I feel like it's a lot more efficient to write a quick command and have the program do that. But as I learn more SQL, my preferences would be SQL for data analysis.

Pros of SQL:

- **Faster Query Processing:** we can retrieve Large amount of data quickly and efficiently
- **No Coding Skills:** it is a very User-friendly language. large number of lines of code is not required.
- **Standardized Language:** Due to it's documentation and long establishment over years, it provides a uniform platform worldwide to all its users.
- **Scalability:** SQL databases can handle large volumes of data and can be scaled up or down as per the requirements of the application.
- **Security:** SQL databases have built-in security features that help protect data from unauthorized access, such as user authentication, encryption, and access control.
- **Data Integrity:** This has constraints such as unique keys, primary keys, and foreign keys, which help prevent data duplication and maintain data accuracy.

- **Backup and Recovery:** SQL databases have built-in backup and recovery tools that help recover data in case of system failures, crashes, or other disasters.
- **Data Consistency:** This ensure consistency of data across multiple tables through the use of transactions, which ensure that changes made to one table are reflected in all related tables.

Cons of SQL:

Complex Interface: SQL has a difficult interface that makes few users uncomfortable while dealing with the database.

Cost: Some versions are costly and hence, programmers cannot access it.

Limited Flexibility: SQL databases are less flexible when it comes to handling unstructured or semi-structured data, as they require data to be structured into tables and columns.

Limited Query Performance: SQL databases may have limited query performance when dealing with large datasets, as queries may take longer to process than in-memory databases.

Complexity: SQL databases can be complex to set up and manage, requiring skilled database administrators to ensure optimal performance and maintain data integrity.

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.

Answer: when I first execute this query then I got this error message.

```
ERROR:  syntax error at or near "TBL"
LINE 1: CREATE TBL 3EMPLOYEES
                ^

SQL state: 42601
Character: 8
```

Then i made some changes:

```
CREATE table EMPLOYEES
 (
employee_id INTEGER 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)
)

Query returned successfully.
```