Databases & SQL for Analysts

3.3: SQL for Data Analysts

Step 1

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

```sql
SELECT DISTINCT name, category_id FROM public.category
ORDER BY category_id
```

name	category_id
Action	1
Animation	2
Children	3
Classics	4
Comedy	5
Documentary	6
Drama	7
Family	8
Foreign	9
Games	10
Horror	11
Music	12
New	13
Sci-fi	14
Sports	15
Travel	16

Step 2

- **Write an INSERT statement to add the following genres to the category table: Thriller, Crime, Mystery, Romance, and War:**

```sql
INSERT INTO public.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?**

```sql
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 category table has three constraints applied to its columns:

- The first is a NOT NULL constraint, which ensures that each column must contain a value.
- The second is a DEFAULT constraint, which sets a default value for the column if no value is provided.
- The third is a PRIMARY KEY constraint, which ensures that each row in the table has a unique identifier.

These constraints are important because they ensure the integrity of the data in the table. They also help to ensure that the data is consistent and can be easily retrieved.

Step 3

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

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

film_id	title
5	African Egg

- **Write an UPDATE command to change the category in the film_category:**

```
SELECT film_id, category_id FROM public.film_category
WHERE film_id = 5
```

film_id	category_id
5	8

```
UPDATE public.film_category
SET category_id = 17
WHERE film_id = 5
```

film_id	category_id
5	17

Step 4

- **Write a DELETE command to remove the mystery category from the category table:**

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

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
17	Thriller
18	Crime
20	Romance
21	War

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, the SQL commands used in step 1 to 4 can be executed by using the Find & Replace function and manually modify the data. Generally, SQL is more efficient than Excel, as it can process large amounts of data quickly and accurately. Furthermore, SQL is more flexible than Excel, as it allows for more complex queries and data manipulation.