**Exercise 3.3**

**Step 1**

**SELECT \***

**FROM category**

|  |  |  |
| --- | --- | --- |
| **category\_id** | **name** | **last\_update** |
| 1 | Action | 2006-02-15 9:46 AM |
| 2 | Animation | 2006-02-15 9:46 AM |
| 3 | Children | 2006-02-15 9:46 AM |
| 4 | Classics | 2006-02-15 9:46 AM |
| 5 | Comedy | 2006-02-15 9:46 AM |
| 6 | Documentary | 2006-02-15 9:46 AM |
| 7 | Drama | 2006-02-15 9:46 AM |
| 8 | Family | 2006-02-15 9:46 AM |
| 9 | Foreign | 2006-02-15 9:46 AM |
| 10 | Games | 2006-02-15 9:46 AM |
| 11 | Horror | 2006-02-15 9:46 AM |
| 12 | Music | 2006-02-15 9:46 AM |
| 13 | New | 2006-02-15 9:46 AM |
| 14 | Sci-Fi | 2006-02-15 9:46 AM |
| 15 | Sports | 2006-02-15 9:46 AM |
| 16 | Travel | 2006-02-15 9:46 AM |

**Step 2**

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

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');

* 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.

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)

);

* What do these constraints do exactly?
  + **NOT NULL:**
    - This constraint on category\_id, name, and last\_update means these columns cannot be empty or left blank.
  + **DEFAULT:**
    - The category\_id column has a default value set using a sequence (nextval('category\_category\_id\_seq'::regclass)). This means every new record added to the table will automatically get a unique identifier from this sequence.
    - The last\_update column has a default value set to the current timestamp (now()), so if not specified, each new record will have the current date and time recorded automatically.
  + **PRIMARY KEY:**
    - The PRIMARY KEY constraint on the category\_id column ensures that each record in the table has a unique identifier.
* Why are they important?
  + **NOT NULL:**
    - This constraint is significant as it ensures that no important columns are left blank, maintaining complete and accurate records.
  + **DEFAULT:** 
    - The default sequence for category\_id automatically assigns a unique identifier to each record, saving time and effort while avoiding conflicts.
    - The default value for last\_update tracks when each record was added, providing helpful information for data tracking and auditing.
  + PRIMARY KEY:
    - This constraint guarantees that each record can be uniquely identified, allowing efficient data retrieval and avoiding duplication in the table.

**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.

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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.

**Pros:**

When looking at SQL vs Excel, some pros would be efficiency, data integrity, precision and scalability, just to name a few. As SQL is designed to manage and manipulate data efficiently, it makes it quick and easy to do so without risking the data integrity and while being able to be extremely precise by targeting specific data. This minimizes the risk of changing records that you did not intend to. SQL is also more suited to working with large datasets and complex queries, making it more scalable for managing extensive databases.

**Cons:**

Some cons on the other hand, would definitely be the learning curve and error sensitivity. Since SQL requires a specific syntax, there is a steeper learning curve associated to becoming familiar with the interface compared to Excel. Also, errors in SQL commands are not so easily undone as just hitting the undo button in Excel, and errors in SQL could lead to unintended changes in the database, or even lost data if not done correctly.