Career Foundry

Data Analytics Immersion

A3.E3

Kendra Jackson

Step 1:

b) SELECT * FROM category

c)

"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"

Step 2:

a) INSERT INTO category(name) VALUES ('Thriller'),('Crime'),('Mystery'),('Romance'),('War')

b) Constraints:
   a. NOT NULL
      i. Used within the category_id, name, and last_update to ensure the columns cannot contain empty or no (null) data.
   b. PRIMARY KEY
      i. Used within the category_id column to set it as the primary key, this ensures each value within this column is unique and not missing
   c. DEFAULT
      i. Used within the category_id and last_update columns to assign a default value when no value provided (important as these columns cannot contain no values so when a no value occurs, the default value is used)

- Constraints are important as they ensure data integrity and consistency as well as allow for efficient data management

Step 3

a. SELECT * FROM film WHERE title = 'African Egg'
    i. Film ID = 5
b. SELECT film_id, catgegory_id FROM film_category WHERE film_id = 5
    i. Current ID = 8 (Family)
    ii. Category ID = 17 (Thriller)
    iii. Found link between tables by using data map document
c. Update film_category SET category_id = 17 WHERE film_id = 5

Step 4

a. DELETE FROM category WHERE name = 'Mystery'

Step 5:

Pros:

- SQL is more efficient and faster (must have the correct command and the knowledge on how to write queries)
- Process is done by the computer and therefore has less errors (not manual like Excel can be)
- Quick and accurate to perform changes (in excel replacing or deleting means you may have to filter and sift through results)

Cons:

- Requires ERD or data dictionary
- Updating certain things in excel seemed easier with find and replace
- Need to know query language and commands for the correct result
- Requires more time and practice (it's a new language)
- Need to know command execution order vs query order

Bonus Task

```
CREATE Table employee
(
employee_id SERIAL NOT NULL,
name VARCHAR(50),
contact_number VARCHAR(30),
designation_id INT,
last_update TIMESTAMP(6) NOT NULL DEFAULT now()
CONSTRAINT employee_pkey PRIMARY KEY (employee_id)
);
```