3.3: SQL for Data Analysts

Step 1:

Query Editor | Query History

```
1  SELECT category_id, name FROM category
```

Data Output | Explain | Messages | Notifications

	category_id [PK] integer	name character varying (25)
1	1	Action
2	2	Animation
3	3	Children
4	4	Classics
5	5	Comedy
6	6	Documentary
7	7	Drama
8	8	Family
9	9	Foreign
10	10	Games
11	11	Horror
12	12	Music
13	13	New
14	14	Sci-Fi
15	15	Sports
16	16	Travel

Step 2:

Query Editor | Query History

```
1  INSERT INTO category(name) VALUES ('Thriller');
2  INSERT INTO category(name) VALUES ('Crime');
3  INSERT INTO category(name) VALUES ('Mystery');
4  INSERT INTO category(name) VALUES ('Romance');
5  INSERT INTO category(name) VALUES ('War')
```

17	17	Thriller	2022-04-28 11:19:44.145034
18	18	Crime	2022-04-28 11:19:44.145034
19	19	Mystery	2022-04-28 11:19:44.145034
20	20	Romance	2022-04-28 11:19:44.145034
21	21	War	2022-04-28 11:19:44.145034

```
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)
);
```

Constraint NOT NULL ensures none of the cells are empty in that column. The category id column must be an integer and cannot be null. The name column must be text and cannot be null. The last update column must be a timestamp with time zone and cannot be null. Constraint primary key makes all values in the column into a primary key for category id.

Step 3:

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

Data Output | Explain | Messages | Notifications

	film_id [PK] integer	title character varying (255)
1	5	African Egg

```
1  SELECT film_id, category_id FROM film_category WHERE film_id = 5
```

Data Output | Explain | Messages | Notifications

	film_id [PK] smallint	category_id [PK] smallint
1	5	8

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

Data Output | Explain | Messages | Notifications

```
UPDATE 1

Query returned successfully in 47 msec.
```

```
1  SELECT * FROM film_category WHERE film_id = 5
```

Data Output | Explain | Messages | Notifications

	film_id [PK] smallint	category_id [PK] smallint	last_update timestamp without time zone
1	5	17	2022-04-28 11:58:02.736147

Step 4:

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

Data Output | Explain | Messages | Notifications

```
DELETE 1

Query returned successfully in 54 msec.
```

Query Editor | Query History

```
1  SELECT * From category WHERE name = 'Mystery'
```

Data Output | Explain | Messages | Notifications

	category_id [PK] integer	name character varying (25)	last_update timestamp without time zone

Step 5:

When working with large data sets and tables that are linked SQL makes searching for and changing information much easier and quicker than you can in Excel. Excel is better when you are working with smaller amounts of data and some data visualizations. In SQL constraints also helps make sure data is entered and in the correct format.

Bonus:

Query Editor | Query History

```
1  CREATE TABLE Employees_3
2  (
3  employee_id VARCHAR(30) NOT NULL,
4  name VARCHAR(50),
5  contact_number VARCHAR(30),
6  designation_id INT,
7  last_update TIMESTAMP NOT NULL DEFAULT now(),
8  CONSTRAINT employee_pkey PRIMARY KEY (employee_id)
9  )
```

Data Output | Explain | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 84 msec.
```