Step 1:

```
SELECT category_id, name FROM category;
```

```
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"
```

Step 2:

```
INSERT INTO category(category_id,name)
VALUES('17','Thriller');

INSERT INTO category(category_id,name)
VALUES('18','Crime');
```

```
INSERT INTO category(category_id,name)
VALUES('19','Mystery');
```

```
INSERT INTO category(category_id,name)
VALUES('20','Romance');
```

```
INSERT INTO category(category_id,name)
VALUES('21','War');
```

The NOT NULL constraint makes it so that if someone attempts to add data with missing values into the category table that data would be rejected outright. This is to prevent any data in this table from being incomplete.

The PRIMARY KEY constraint makes the category_id column into that tables data grain. This can help prevent any duplicated values from getting into the table.

Step 3:

SELECT film_id, title FROM film;	African Egg’s film id is 5
SELECT film_id, category_id FROM film_category;	African Egg’s category id is 8
SELECT category_id, name FROM category;	African Egg’s category is Family

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

Step 4:

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

Step 5:

Within an Excel sheet all of the different columns would be grouped into one “table”. So if you wanted to change something, you would have to change **MANY** rows withing the data set. SQL streamlines this entire prosses by separating the columns into different tables. Although the one down side to having multiple tables is that you have to navigate through each one like a maze, while in Excel everything is just right there.