

Step 1:

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

| 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 |

Step 2:

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

The NOT NULL constraint is used in the category_id, name and last_update columns in order to ensure there are no missing values in these fields. The PRIMARY KEY constraint is applied to the category_id column. This will create a unique ID for each record in the table, which cannot contain duplicate or null values. These are important because they allow for linking data between multiple tables.

Step 3:

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

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

Step 4:

```
DELETE FROM category  
WHERE category_id = 19
```

Step 5:

SQL is great for pulling the exact information necessary, without having to sort through large worksheets or waiting for Excel to load them. At this point, the only downside I've encountered with SQL is double-checking my work to ensure my queries actually did what I intended.

Bonus Task:

```
CREATE TABLE employees_3  
(  
  employee_id INT NOT NULL,  
  name VARCHAR(50),  
  contact_number VARCHAR(30),  
  designation_id INTEGER,  
  last_update TIMESTAMP NOT NULL DEFAULT now(),  
  CONSTRAINT employee_pkey PRIMARY KEY (employee_id)  
);
```