Exercise 1

In this exercise you will:

- Create a small database with 3 tables
- Populate the database with some data
- Make queries to the database

The Data

In the file `kubrick_films.csv` you will find a set of data about movies directed by Stanley Kubrick, including date, budget, box office sales, and leading actors.

Task 1 - Create the database

1. Open the file `schema.sql` which contains some stubs for CREATE statements for the database. Complete the statements such that you end up with the following three tables.

    - `film` table
        - must have an `id` primary key generated by the DB
        - must have a `title` of type VARCHAR(128)
        - must have a `year` of type INTEGER
        - may have a `duration` of type INTEGER
        - may have a `budget` of type INTEGER
        - may have a `boxoffice` of type INTEGER

        - may have a `genre` of type VARCHAR(32)
        - may have a `genre_complex` of type VARCHAR(64)
        - may have a `rating` of type REAL
        - may have a `imdb_votes` of type REAL
    - `actor` table
        - must have an `id` primary key generated by the DB
        - must have a `name` field of type VARCHAR(64) that is unique
    - `starring` table
        - must have a `filmid` foreign key of type INTEGER
        - must have a `actorid` foreign key of type INTEGER
        - If the primary key that either foreign key points to is deleted, then delete the row in `starring` too.

2. Now create the database by entering the following at the command line:

```
sqlite3 kubrick.db
```

3. Inside the SQLite shell, use the `.read` command to execute your statements:

```
sqlite> .read schema.sql
```

Task 2 - Populate the database

1. Open the file `populate.sql` and add INSERT statements for each

film in the CSV file.

    - ***I did this for you***

2. Inside the SQLite shell, use the `.read` command to execute your statements:

```
sqlite> .read populate.sql
```

Task 3 - Query the database

Open the file `query.sql` and write queries to answer the questions below. Copy and paste each query into the SQLite shell to test your results againts the expected results.

1. Query 1: Get the title and year for films made before 1970

    - Expected result:

```
Fear and Desire|1953
Killer's Kiss|1955
The Killing|1956
Paths of Glory|1957
Spartacus|1960
Lolita |1962
Dr. Strangelove|1964
2001: A Space Odyssey|1968
```

2. Query 2: Get the title, year, and boxoffice for all films, sorted by boxoffice, descending

    - Expected result:

```
2001: A Space Odyssey|1968|190000
Eyes Wide Shut |1999|162000
Spartacus|1960|60000
Full Metal Jacket|1987|46300
Shining |1980|44000
A Clockwork Orange|1971|26589
Barry Lyndon  |1975|20000
Dr. Strangelove|1964|9440
Lolita |1962|9250
Fear and Desire|1953|0
Killer's Kiss|1955|0
The Killing|1956|0
Paths of Glory|1957|0
```

3. Query 3: Get the title of all movies starring “Sterling Hayden”

    - Expected result:

```
The Killing
Dr. Strangelove
```

4. Query 4: Get the title and rating of all movies with a rating greater

than or equal to 8.0, in ascending order

- Expected result:

```
Spartacus|8.0
The Killing|8.1
Barry Lyndon  |8.1
2001: A Space Odyssey|8.4
A Clockwork Orange|8.4
Full Metal Jacket|8.4
Paths of Glory|8.5
Shining |8.5
Dr. Strangelove|8.6
```

5. Query 5: Get the title, year, and imdb_votes of the ten films with the largest imdb_vote, sorted by imdb_votes descending

- Expected result:

```
Shining |1980|346.241
A Clockwork Orange|1971|328.338
Full Metal Jacket|1987|280.283
2001: A Space Odyssey|1968|253.251
Dr. Strangelove|1964|231.79
Eyes Wide Shut |1999|154.685
Spartacus|1960|72.569
Paths of Glory|1957|72.249
```

```
Barry Lyndon  |1975|64.023
The Killing|1956|40.759
```

6. Query 6: Get the title of every film that has the *word* “A” in it (not the letter, but the word).
    - Expected result:

```
2001: A Space Odyssey
A Clockwork Orange
```

7. Query 7a and 7b
    - Query 7a: Get the total number of roles stored in the starring table
        - Expected result:

```
65
```

    - Query 7b: Now get the number of unique actors recorded in the starring table
        - Expected result:

```
61
```

8. Get the minimum, maximum, average and sum total of the boxoffice numbers for all films
    - Expected result:

```
0|190000|43659.9230769231|567579.0
```

9. Three parts
    - 9a. Add new table and insert some data
        - Create a new table `director` with an `id` primary key and a `name`.
        - Insert “Stanley Kubrick” and “Terry Gilliam” to the new table
        - Change the film table to have a foreign key `directorid`.
        - Insert an entry for “Brazil (1985)” linked to Gilliam (ignore other attributes)
        - *I did all of this for you*
    - 9b. Use a natural join to get titles and director names.
        - Since you haven’t added foreign keys to Kubrick yet, this should only return results for Gilliam
        - Expected result:

```
Brazil|Terry Gilliam
Fisher King|Terry Gilliam
12 Monkeys|Terry Gilliam
```

- 9c. Now do a left outer join on film and director.
    - This should get you all the films with nulls for Kubrick.
    - Expected result:

```
Fear and Desire|
Killer's Kiss|
The Killing|
Paths of Glory|
Spartacus|
Lolita |
Dr. Strangelove|
2001: A Space Odyssey|
A Clockwork Orange|
Barry Lyndon  |
Shining |
Full Metal Jacket|
Eyes Wide Shut |
Get Out|
Brazil|Terry Gilliam
Fisher King|Terry Gilliam
```

```
12 Monkeys|Terry Gilliam
```

10. Three Parts

    - 10a. Insert directorid foregin keys for all of Kubrick's films
        - *I did this for you*
    - 10b. Get the list of genres of Kubrick's movies and the number of films in that genre in descending order
        - Expected result:

```
Drama|6
Adventure|3
Comedy|1
Horror|1
Romance|1
Thriller|1
```

    - 10c. Perform the same query but get just the genres in which he made only one film
        - Expected result:

```
Comedy|1
Horror|1
```

```
Romance|1
Thriller|1
```