SQL Assignment

In [1]:

```
import pandas as pd
import sqlite3
```

In [2]:

```
conn = sqlite3.connect("Db-IMDB-Assignment.db")
```

Sample Code

In [3]:

```
%%time
# Write your sql query below

query = """
        SELECT TRIM(Movie.title) AS 'Movie_Name'
        FROM Movie
        WHERE Movie.rating < 3

        """

q = pd.read_sql_query(query, conn)
print(q.shape)
print(q.head())
```

```
(85, 1)
             Movie_Name
0            Mastizaade
1  Dragonball Evolution
2             Loveyatri
3                Race 3
4                Gunday
Wall time: 355 ms
```

Q1 --- List all the directors who directed a 'Comedy' movie in a leap year. (You need to check that the genre is 'Comedy’ and year is a leap year) Your query should return director name, the movie name, and the year.

In [4]:

```
%%time
# Write your sql query below

query = """
        SELECT Person.name,Movie.title,Movie.year
        FROM Movie
        JOIN M_Director ON trim(Movie.mid)=trim(M_Director.mid)
        JOIN Person ON trim(M_Director.pid)=trim(Person.pid)
        WHERE trim(Movie.mid) in
        (SELECT trim(M_Genre.mid)
        FROM M_Genre
        JOIN Genre on trim(M_Genre.gid)=trim(Genre.gid)
        WHERE trim(Genre.name) like '%Comedy%')
        AND Movie.year%4==0

        """


q1 = pd.read_sql_query(query, conn)
print(q1.shape)
q1.head()
```

```
(246, 3)
Wall time: 7.11 s
```

Out[4]:

	Name	title	year
0	Milap Zaveri	Mastizaade	2016
1	Danny Leiner	Harold & Kumar Go to White Castle	2004
2	Anurag Kashyap	Gangs of Wasseypur	2012
3	Frank Coraci	Around the World in 80 Days	2004
4	Griffin Dunne	The Accidental Husband	2008

Q2 --- List the names of all the actors who played in the movie 'Anand' (1971)

In [5]:

```
%%time
# Write your sql query below

query = """
        SELECT Person.name
        FROM Movie
        JOIN M_Cast ON trim(Movie.mid)=trim(M_Cast.mid)
        JOIN Person ON trim(M_Cast.pid)=trim(Person.pid)
        WHERE Movie.title like 'Anand'

        """


q2 = pd.read_sql_query(query, conn)
print(q2.shape)
q2.head()
```

```
(17, 1)
Wall time: 491 ms
```

Out[5]:

	Name
0	Rajesh Khanna
1	Amitabh Bachchan
2	Sumita Sanyal
3	Ramesh Deo
4	Seema Deo

Q3 --- List all the actors who acted in a film before 1970 and in a film after 1990. (That is: < 1970 and > 1990.)

In [6]:

```
%%time
# Write your sql query below

query = """
        SELECT Person.name
        FROM Person
        WHERE trim(Person.pid) IN
        (SELECT trim(M_Cast.pid)
        FROM M_Cast
        JOIN Movie ON trim(M_Cast.mid)=trim(Movie.mid)
        WHERE Movie.year<1970
        INTERSECT
        SELECT trim(M_Cast.pid)
        FROM M_Cast
        JOIN Movie ON trim(M_Cast.mid)=trim(Movie.mid)
        WHERE Movie.year>1990)

        """


q3 = pd.read_sql_query(query, conn)
print(q3.shape)
q3.head()
```

```
(333, 1)
Wall time: 2min 34s
```

Out[6]:

	Name
0	Rishi Kapoor
1	Amitabh Bachchan
2	Asrani
3	Zohra Sehgal
4	Parikshat Sahni

Q4 --- List all directors who directed 10 movies or more, in descending order of the number of movies they directed. Return the directors' names and the number of movies each of them directed.

In [7]:

```
%%time
# Write your sql query below

query = """
        SELECT Person.name,count(*) AS Count
        FROM Person
        JOIN M_Director ON trim(Person.pid)=trim(M_Director.pid)
        JOIN Movie ON trim(M_Director.mid)=trim(Movie.mid)
        GROUP BY Person.Name
        HAVING count(*)>=10
        ORDER BY count(*) DESC
        """

q4 = pd.read_sql_query(query, conn)
print(q4.shape)
q4.head()
```

```
(58, 2)
Wall time: 1min 27s
```

Out[7]:

	Name	Count
0	David Dhawan	39
1	Mahesh Bhatt	36
2	Priyadarshan	30
3	Ram Gopal Varma	30
4	Vikram Bhatt	29

Q5.a --- For each year, count the number of movies in that year that had only female actors.

In [8]:

```
%%time
# Write your sql query below

query = """
        SELECT Movie.year,count(*)
        FROM Movie
        JOIN
        (SELECT trim(M_Cast.mid) as mid,count(*)
        FROM M_Cast
        WHERE trim(M_Cast.pid)IN
        (SELECT trim(Person.pid)
        FROM Person
        WHERE Person.gender like 'Female')
        GROUP BY trim(M_Cast.mid)
        HAVING count(*)=
        (SELECT count(*)
        FROM M_Cast MC
        WHERE MC.mid=M_Cast.mid)) C
        ON trim(Movie.mid)=trim(c.mid)
        GROUP BY Movie.year
        """

q5a = pd.read_sql_query(query, conn)
print(q5a.shape)
q5a.head()
```

```
(4, 2)
Wall time: 1min 26s
```

Out[8]:

	year	count(*)
0	1939	1
1	1999	1
2	2000	1
3	I 2018	1

Q5.b --- Now include a small change: report for each year the percentage of movies in that year with only female actors, and the total number of movies made that year. For example, one answer will be: 1990 31.81 13522 meaning that in 1990 there were 13,522 movies, and 31.81% had only female actors. You do not need to round your answer.

In [9]:

```
%%time
# Write your sql query below

query = """
        SELECT Movie.year,cAST(count(*) AS FLOAT)/(SELECT count(*) FROM Movie M WHERE M
ovie.year=M.year) AS Percentage,
        (SELECT count(*) FROM Movie M WHERE Movie.year=M.year) AS Total
        FROM Movie
        JOIN
        (SELECT trim(M_Cast.mid) as mid,count(*)
        FROM M_Cast
        WHERE trim(M_Cast.pid)IN
        (SELECT trim(Person.pid)
        FROM Person
        WHERE Person.gender like 'Female')
        GROUP BY trim(M_Cast.mid)
        HAVING count(*)=
        (SELECT count(*)
        FROM M_Cast MC
        WHERE MC.mid=M_Cast.mid)) C
        ON trim(Movie.mid)=trim(c.mid)
        GROUP BY Movie.year

        """

q5b = pd.read_sql_query(query, conn)
print(q5b.shape)
q5b.head()
```

```
(4, 3)
Wall time: 1min 39s
```

Out[9]:

	year	Percentage	Total
0	1939	0.500000	2
1	1999	0.015152	66
2	2000	0.015625	64
3	2018	0.100000	10

Q6 --- Find the film(s) with the largest cast. Return the movie title and the size of the cast. By "cast size" we mean the number of distinct actors that played in that movie: if an actor played multiple roles, or if it simply occurs multiple times in casts, we still count her/him only once.

In [10]:

```
%%time
# Write your sql query below

query = """
        SELECT Movie.title,
        (SELECT count(M_Cast.pid) FROM M_Cast GROUP BY M_Cast.mid HAVING count(DISTINCT
M_Cast.pid)=
        (SELECT max(c) FROM (SELECT M_Cast.mid,count(distinct M_Cast.pid) c FROM M_Cast
group by M_Cast.mid))
        ) count
        FROM Movie
        WHERE Movie.mid =
        (SELECT M_Cast.mid FROM M_Cast GROUP BY M_Cast.mid HAVING count(distinct M_Cas
t.pid)=
        (SELECT max(c) FROM (select M_Cast.mid,count(distinct M_Cast.pid) c from M_Cast
GROUP BY M_Cast.mid)))

        """


q6 = pd.read_sql_query(query, conn)
print(q6.shape)
q6.head()
```

```
(1, 2)
Wall time: 1.03 s
```

Out[10]:

	title	count
0	Ocean's Eight	238

Q7 --- A decade is a sequence of 10 consecutive years. For example, say in your database you have movie information starting from 1965. Then the first decade is 1965, 1966, ..., 1974; the second one is 1967, 1968, ..., 1976 and so on. Find the decade D with the largest number of films and the total number of films in D.

In [11]:

```
%%time
# Write your sql query below

query = """
        SELECT y.year as decade_start, y.year + 9 as decade_end,
        count(*) as num_movies
        FROM (SELECT distinct Movie.year FROM Movie) y
        JOIN Movie m ON m.year >= y.year
        AND m.year < y.year + 10
        GROUP BY y.year
        ORDER BY count(*) DESC
        LIMIT 1

        """

q7 = pd.read_sql_query(query, conn)
print(q7.shape)
q7.head()
```

```
(1, 3)
Wall time: 391 ms
```

Out[11]:

	decade_start	decade_end	num_movies
0	2008	2017	1128

Q8 --- Find all the actors that made more movies with Yash Chopra than any other director.

In [12]:

```
%%time
# Write your sql query below

query = """
        SELECT Person.name
        FROM Person
        WHERE Person.pid IN
        (SELECT trim(M_Cast.pid)
        FROM M_Cast
        JOIN M_Director ON trim(M_Cast.mid)=trim(M_Director.mid)
        WHERE trim(M_Director.pid)=
        (SELECT trim(Person.pid)
        FROM Person
        WHERE Person.name like '%Yash Chopra%'))
        GROUP BY Person.name
        HAVING count(*)=
        (SELECT count(*)
        FROM M_Cast
        JOIN M_Director ON trim(M_Cast.mid)=trim(M_Director.mid)
        WHERE trim(Person.pid)=trim(M_Cast.pid)
        GROUP BY trim(M_Director.pid)
        ORDER BY count(*) DESC
        LIMIT 1
        )
        """

q8 = pd.read_sql_query(query, conn)
print(q8.shape)
q8.head()
```

```
(211, 1)
Wall time: 1min 18s
```

Out[12]:

	Name
0	Abbie Murphy
1	Akhtar Mirza
2	Akhtar-Ul-Iman
3	Aloka Mukherjee
4	Amir Zadey

Q9 --- The Shahrukh number of an actor is the length of the shortest path between the actor and Shahrukh Khan in the "co-acting" graph. That is, Shahrukh Khan has Shahrukh number 0; all actors who acted in the same film as Shahrukh have Shahrukh number 1; all actors who acted in the same film as some actor with Shahrukh number 1 have Shahrukh number 2, etc. Return all actors whose Shahrukh number is 2.

In [13]:

```
%%time
# Write your sql query below
# As in given dataset, There is no data related to "Shahrukh Khan". So the output of qu
ery is 0
query ="""
       SELECT Person.Name
       FROM Person
       WHERE trim(Person.pid) IN
       (SELECT distinct trim(M_Cast.pid)
       FROM M_Cast
       WHERE trim(M_Cast.mid) IN
       (SELECT distinct trim(M_Cast.mid)
       FROM M_Cast
       WHERE trim(M_Cast.mid) IN
       (SELECT trim(M_Cast.mid)
       FROM M_Cast
       WHERE trim(M_Cast.pid) IN
       (SELECT trim(M_Cast.pid)
       FROM M_Cast
       WHERE trim(M_Cast.mid) IN
       (SELECT trim(M_Cast.mid)
       FROM M_Cast
       WHERE trim(M_Cast.pid)=
       (SELECT trim(Person.pid)
       FROM Person
       WHERE Person.name like '%Shah Rukh Khan%'))
       AND trim(M_Cast.pid)<>(SELECT trim(Person.pid)
       FROM Person
       WHERE Person.name like '%Shah Rukh Khan%'))))
       AND trim(M_Cast.pid) NOT IN
       (SELECT trim(M_Cast.pid)
       FROM M_Cast
       WHERE trim(M_Cast.mid) IN
       (SELECT trim(M_Cast.mid)
       FROM M_Cast
       WHERE trim(M_Cast.pid)=
       (SELECT trim(Person.pid)
       FROM Person
       WHERE Person.name like '%Shah Rukh Khan%'))))
       """

q9 = pd.read_sql_query(query, conn)
print(q9.shape)
q9.head()
```

```
(25698, 1)
Wall time: 1.33 s
```

Out[13]:

	Name
0	Freida Pinto
1	Rohan Chand
2	Damian Young
3	Waris Ahluwalia
4	Caroline Christl Long

In [ ]: