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)
#q.head()
print(q.head())
```

```
(85, 1)
             Movie_Name
0            Mastizaade
1  Dragonball Evolution
2             Loveyatri
3                Race 3
4                Gunday
Wall time: 378 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 p.name Director,m.title "Movie name",m.year
        from movie m left join m_director md on m.mid=md.mid left join person p on trim
(p.pid)=trim(md.pid)
        where trim(md.mid) in (select trim(mid)
                               from m_genre mg join genre g on trim(mg.gid) = Trim(g.g
id) where name like "%Comedy%")
                               and m.year%4=0
        order by m.year

        """

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

```
(246, 3)
                 Director  Movie name     year
0              Amit Mitra  Jagte Raho     1956
1            Chetan Anand    Funtoosh     1956
2             Satyen Bose     Jagriti     1956
3             Mohan Segal   New Delhi     1956
4              S.U. Sunny    Kohinoor     1960
..                    ...         ...      ...
241      Suzad Iqbal Khan     Monsoon   I 2015
242          Sunhil Sippy        Noor   I 2017
243           Abhinay Deo   Blackmail   I 2018
244   Parvati Balagopalan    Straight  II 2009
245         Kaizad Gustad     Jackpot  II 2013

[246 rows x 3 columns]
Wall time: 5.93 s
```

Out[4]:

	Director	Movie name	year
0	Amit Mitra	Jagte Raho	1956
1	Chetan Anand	Funtoosh	1956
2	Satyen Bose	Jagriti	1956
3	Mohan Segal	New Delhi	1956
4	S.U. Sunny	Kohinoor	1960

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 name
       from person
       where pid in (
                       select Trim(pid)
                       from m_cast mc
                       where mid in
                       (
                           select mid
                           from movie
                           where title like 'Alam ara'
                       )
                   )

        """

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

```
(9, 1)
                 Name
0   Prithviraj Kapoor
1             Zubeida
2       Jagdish Sethi
3       Master Vithal
4               Jillo
5             Sushila
6              Elizer
7           W.M. Khan
8         L.V. Prasad
Wall time: 115 ms
```

Out[5]:

	Name
0	Prithviraj Kapoor
1	Zubeida
2	Jagdish Sethi
3	Master Vithal
4	Jillo

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 name from person where pid in
        (
        select Trim(pid) from m_cast mc where
        mid in
        (
        select mid from movie where year<1970 or year>1990
        )
        )
        order by name


         """

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

```
(29661, 1)
                             Name
0                  'Ganja' Karuppu
1            'Lee' George Quinones
2       'Musafir' Radio Performing
3               'Nandha' Saravanan
4           'Om' Rakesh Chaturvedi
...                            ...
29656                  Zuri Echea
29657               Zuzanna Zajac
29658              Àaron Brewster
29659                 Éric Berger
29660             Ócsai Krisztián

[29661 rows x 1 columns]
Wall time: 711 ms
```

Out[6]:

	Name
0	'Ganja' Karuppu
1	'Lee' George Quinones
2	'Musafir' Radio Performing
3	'Nandha' Saravanan
4	'Om' Rakesh Chaturvedi

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 p.name,(select count(pid) from m_director md where md.pid=p.pid group by
pid having count(pid)>10) count
        from person p where p.pid in (
        select pid from m_director group by pid having count(pid)>10
        )
        order by count desc

        """
"""

select p.name,count(name) from person p join m_director md on trim(p.pid)=trim(md.pid)
        group by name having count(name)>10
        order by count(name) desc
"""
q4 = pd.read_sql_query(query, conn)
print(q4.shape)
print(q4)
q4.head()
```

(45, 2)

	Name	count
0	David Dhawan	39
1	Mahesh Bhatt	35
2	Ram Gopal Varma	30
3	Priyadarshan	30
4	Vikram Bhatt	29
5	Hrishikesh Mukherjee	27
6	Yash Chopra	21
7	Shakti Samanta	19
8	Basu Chatterjee	19
9	Subhash Ghai	18
10	Abbas Alibhai Burmawalla	17
11	Shyam Benegal	17
12	Rama Rao Tatineni	17
13	Manmohan Desai	16
14	Gulzar	16
15	Raj N. Sippy	16
16	Mahesh Manjrekar	15
17	Raj Kanwar	15
18	Rajkumar Santoshi	14
19	Rahul Rawail	14
20	Raj Khosla	14
21	Indra Kumar	14
22	Anurag Kashyap	13
23	Rakesh Roshan	13
24	Ananth Narayan Mahadevan	13
25	Dev Anand	13
26	Vijay Anand	13
27	K. Raghavendra Rao	13
28	Harry Baweja	13
29	Satish Kaushik	12
30	Madhur Bhandarkar	12
31	Prakash Jha	12
32	Rohit Shetty	12
33	Anees Bazmee	12
34	Anil Sharma	12
35	Nagesh Kukunoor	12
36	Prakash Mehra	12
37	Guddu Dhanoa	12
38	Umesh Mehra	12
39	Ketan Mehta	11
40	Mohit Suri	11
41	Sanjay Gupta	11
42	Nasir Hussain	11
43	Pramod Chakravorty	11
44	Govind Nihalani	11

Wall time: 91.8 ms

Out[7]:

	Name	count
0	David Dhawan	39
1	Mahesh Bhatt	35
2	Ram Gopal Varma	30
3	Priyadarshan	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 year,count(*) from movie where mid in (
        select mid from m_cast group by mid having count(pid)=(
        select count(pid) from m_cast mc where trim(pid) in (
        select trim(pid) from person p where gender like 'Female'
        )
        group by mid
        )
        )
        group by year

        """

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

```
(8, 2)
   year  count(*)
0  1964         1
1  1997         1
2  1999         1
3  2000         1
4  2001         1
5  2004         1
6  2013         2
7  2016         1
Wall time: 338 ms
```

Out[8]:

	year	count(*)
0	1964	1
1	1997	1
2	1999	1
3	2000	1
4	2001	1

In [ ]:

```
```

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 m1.year,cast((count(*)*100) as float)/(select count(*) from movie m2 whe
re m1.year=m2.year) as percentage,
        (select count(*) from movie m3 where m1.year=m3.year) as count from movie m1 wh
ere m1.mid in (
        select mid from m_cast group by mid having count(pid)=(
        select count(pid) from m_cast mc where trim(pid) in (
        select trim(pid) from person p where gender like 'Female'
        )
        group by mid
        )
        )
        group by year

        """

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

```
(8, 3)
   year  percentage  count
0  1964    7.142857     14
1  1997    1.851852     54
2  1999    1.515152     66
3  2000    1.562500     64
4  2001    1.408451     71
5  2004    0.970874    103
6  2013    1.574803    127
7  2016    0.847458    118
Wall time: 353 ms
```

Out[9]:

	year	percentage	count
0	1964	7.142857	14
1	1997	1.851852	54
2	1999	1.515152	66
3	2000	1.562500	64
4	2001	1.408451	71

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 title, (
        select count(distinct pid) from m_cast mc where mc.mid=m.mid) count
        from movie m where mid in (
        select mid from m_cast group by mid having count(distinct pid)=(
        select max(c) from (select mid,count(distinct pid) c from m_cast group by mi
d)))



        """

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

```
(1, 2)
           title  count
0  Ocean's Eight    238
Wall time: 440 ms
```

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 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)
print(q7)
q7.head()
```

```
(1, 3)
  decade_start  decade_end  num_movies
0         2008        2017        1128
Wall time: 308 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 b.number,b.actor,b.director from (select MAX(a.count) as number,a.director,a.actor from
        (select count(p.PID) as count ,p.PID as actor,md.PID as director from person as p left join m_cast as mc on p.PID=mc.PID
        inner join m_director as md on md.MID=mc.MID group by md.PID ,p.PID)
        as a group by a.actor) as b where b.director=(select PID from person where Name ='Yash Chopra')

        """

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

```
(0, 3)
Empty DataFrame
Columns: [number, actor, director]
Index: []
Wall time: 525 ms
```

Out[12]:

number	actor	director

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

query = """
        select distinct PID, Name
        from Person natural join M_Cast
        where Name <> 'Shah Rukh Khan' and MID in (select MID
        from M_Cast where PID in (select PID
        from Person natural join M_Cast
        where Name <> 'Shah Rukh Khan' and MID in (select MID
        from Person natural join M_Cast
        where Name = 'Shah Rukh Khan')))
        and PID not in (select PID
        from Person natural join M_Cast
        where Name <> 'Shah Rukh Khan' and MID in (select MID
        from Person natural join M_Cast
        where Name = 'Shah Rukh Khan'));

        """

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

```
(0, 2)
Empty DataFrame
Columns: [PID, Name]
Index: []
Wall time: 290 ms
```

Out[13]:

	PID	Name

In [ ]:

In [ ]: