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 *
        from person
        """

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

```
(37566, 4)
Wall time: 198 ms
```

Out[3]:

	index	PID	Name	Gender
0	0	nm0000288	Christian Bale	Male
1	1	nm0000949	Cate Blanchett	Female
2	2	nm1212722	Benedict Cumberbatch	Male
3	3	nm0365140	Naomie Harris	Female
4	4	nm0785227	Andy Serkis	Male

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 [43]:

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

query = """
        SELECT p.Name,mov.year,mov.title
        FROM Person p
        JOIN M_Director d ON p.PID=d.PID
        JOIN Movie mov ON d.MID=mov.MID
        JOIN M_Genre g ON mov.MID=g.MID
        JOIN Genre gen ON g.GID=gen.GID
        WHERE gen.Name like "%Comedy%"
        OR gen.Name like"Comedy%"
        OR gen.Name like"%Comedy"

        INTERSECT
        SELECT p.Name,mov.year,mov.title
        FROM Person p
        JOIN M_Director d ON p.PID=d.PID
        JOIN Movie mov ON d.MID=mov.MID
        WHERE ((CAST(substr(mov.year,-4) as integer)%4 == 0 AND CAST(substr(mov.year,-4) as
integer)<> 0) OR (CAST(substr(mov.year,-4) as integer)%400 == 0))

        """

q = pd.read_sql_query(query, conn)
print(q.shape)
q.head(50)
```

```
(232, 3)
Wall time: 224 ms
```

Out[43]:

	Name	year	title
0	A. Bhimsingh	1968	Sadhu Aur Shaitaan
1	A. Bhimsingh	1972	Joroo Ka Ghulam
2	Abbas Tyrewala	2008	Jaane Tu... Ya Jaane Na
3	Abhishek Jain	2012	Kevi Rite Jaish
4	Abhishek Sharma	2016	Tere Bin Laden: Dead Or Alive
5	Aditya Chopra	2008	Rab Ne Bana Di Jodi
6	Aditya Chopra	2016	Befikre
7	Akashdeep	2016	Santa Banta Pvt Ltd
8	Anand Balraj	2012	Daal Mein Kuch Kaala Hai
9	Anees Bazmee	2008	Singh Is Kinng
10	Anurag Basu	2012	Barfi!
11	Anurag Kashyap	2012	Gangs of Wasseypur
12	Arbaaz Khan	2012	Dabangg 2
13	Ashish R. Mohan	2012	Khiladi 786
14	Ashwini Chaudhary	2012	Jodi Breakers
15	Aziz Mirza	1992	Raju Ban Gaya Gentleman
16	Aziz Mirza	2000	Phir Bhi Dil Hai Hindustani
17	Aziz Mirza	2008	Kismat Konnection
18	Basu Chatterjee	1976	Chhoti Si Baat
19	Basu Chatterjee	1980	Apne Paraye
20	Basu Chatterjee	1980	Man Pasand
21	Basu Chatterjee	1984	Lakhon Ki Baat
22	Bhagyaraj	1996	Mr. Bechara
23	Bhappi Sonie	I 1968	Brahmachari
24	Bimal Roy	1960	Parakh
25	Brij	1972	Victoria No. 203
26	Brij	1980	Bombay 405 Miles
27	Chandrakant Singh	2008	Rama Rama Kya Hai Dramaaa
28	Chetan Anand	1956	Funtoosh
29	Chi Gurudutt	2008	Kaamannana Makkalu
30	Danny Leiner	2004	Harold & Kumar Go to White Castle
31	David Dhawan	1992	Bol Radha Bol
32	David Dhawan	1996	Loafer
33	David Dhawan	1996	Saajan Chale Sasural
34	David Dhawan	2000	Chal Mere Bhai
35	David Dhawan	2000	Dulhan Hum Le Jayenge

	Name	year	title
36	David Dhawan	2000	Kunwara
37	David Dhawan	2004	Mujhse Shaadi Karogi
38	Deepak Anand	1992	Yaad Rakhegi Duniya
39	Deepak S. Shivdasani	2008	Mr. White Mr. Black
40	Dibakar Banerjee	2008	Oye Lucky! Lucky Oye!
41	Eeshwar Nivas	2008	De Taali
42	Eeshwar Nivas	2008	My Name Is Anthony Gonsalves
43	Farah Khan	2004	Main Hoon Na
44	Frank Coraci	2004	Around the World in 80 Days
45	Ganapathy Bharat	2004	Hari Om
46	Ganesh Acharya	2008	Money Hai Toh Honey Hai
47	Gauri Shinde	2012	English Vinglish
48	Govind Menon	2004	Kis Kis Ki Kismat
49	Griffin Dunne	2008	The Accidental Husband

In [ ]:

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 p.name
            FROM Person p
            WHERE trim(p.PID,' ') IN
            (
                SELECT trim(mcast.PID,' ')
                FROM M_Cast mcast
                WHERE mcast.MID IN
                (
                    SELECT mov.MID
                    FROM Movie mov
                    WHERE lower(mov.title)="anand"
                )
            )


        """

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

```
(17, 1)
Wall time: 76.8 ms
```

Out[5]:

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

	Name
4	Seema Deo
5	Asit Kumar Sen
6	Dev Kishan
7	Atam Prakash
8	Lalita Kumari
9	Savita
10	Brahm Bhardwaj
11	Gurnam Singh
12	Lalita Pawar
13	Durga Khote
14	Dara Singh
15	Johnny Walker
16	Moolchand

In [ ]:

```

```

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 pers.PID
        FROM Person pers
        JOIN M_Cast mcast ON pers.PID=trim(mcast.PID)
        JOIN Movie mov ON mcast.MID=mov.MID
        WHERE CAST(substr(mov.year,-4) as integer)<1970

        INTERSECT

        SELECT pers.PID
        FROM Person pers
        JOIN M_Cast mcast ON pers.PID=trim(mcast.PID)
        JOIN Movie mov ON mcast.MID=mov.MID
        WHERE CAST(substr(mov.year,-4) as integer)>1990

        """

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

```
(300, 1)
Wall time: 679 ms
```

Out[6]:

	PID
0	nm0000821
1	nm0003987
2	nm0004334
3	nm0004429
4	nm0004432

Q4 --- List all directors who directed 10 movies or more, in descending order of

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,COUNT(md.MID) cnt
        FROM Person p JOIN M_Director md ON p.PID=md.PID
        GROUP BY md.PID
        HAVING cnt>9
        ORDER BY cnt DESC

        """

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

```
(58, 2)
Wall time: 53.9 ms
```

Out[7]:

	Name	cnt
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 CAST(substr(mov.year,-4) as integer) year ,COUNT(DISTINCT mov.MID) count
        FROM Movie mov
        WHERE trim(mov.MID)NOT IN
            (
                SELECT trim(mcast.MID)
                FROM M_Cast mcast
                WHERE trim(mcast.PID) IN
                    (
                        SELECT p.PID
                        FROM Person p
                        WHERE p.Gender!='Female'
                    )
            )
        GROUP BY CAST(substr(mov.year,-4) as integer)
        """

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

```
(4, 2)
Wall time: 161 ms
```

Out[8]:

	year	count
0	1939	1
1	1999	1
2	2000	1
3	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 [14]:

```
Query="""
            SELECT table1.movie_count,table2.female_movie_count,
(table2.female_movie_count*100.00)/table1.movie_count as percent
    FROM
        (
            SELECT CAST(substr(mov1.year,-4) as integer) as year,COUNT(DISTINCT mov1.mid) as
movie_count
            FROM movie mov1
            GROUP BY CAST(substr(mov1.year,-4) as integer)
        ) as table1
     JOIN
        (
          SELECT CAST(substr(mov.year,-4) as integer)as year ,COUNT(DISTINCT mov.MID)
female_movie_count
          FROM Movie mov
          WHERE trim(mov.MID)NOT IN
            (
                SELECT trim(mcast.MID)
                FROM M_Cast mcast
                WHERE trim(mcast.PID) IN
                    (
                        SELECT p.PID
                        FROM Person p
                        WHERE p.Gender!='Female'
                    )
            )
          GROUP BY CAST(substr(mov.year,-4) as integer)
        ) as table2
    ON table1.year=table2.year
"""
q5b = pd.read_sql_query(Query, conn)
print(q5b.shape)
print(q5b.head(10))
```

```
(4, 3)
   movie_count  female_movie_count    percent
0            2                   1  50.000000
1           66                   1   1.515152
2           64                   1   1.562500
3          104                   1   0.961538
```

In [ ]:

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 [16]:

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

query = """
        SELECT mov.title,COUNT(DISTINCT mcast.PID) AS cnt
        FROM Movie mov JOIN M_Cast mcast
        ON mov.MID=mcast.MID
        GROUP BY mcast.MID
        ORDER BY cnt DESC
        """

q6 = pd.read_sql_query(query, conn)
print(q6.shape)
q6.head(10)
```

```
(3473, 2)
Wall time: 332 ms
```

Out[16]:

	title	cnt
0	Ocean's Eight	238
1	Apaharan	233
2	Gold	215
3	My Name Is Khan	213
4	Captain America: Civil War	191
5	Geostorm	170
6	Striker	165
7	2012	154
8	Pixels	144
9	Yamla Pagla Deewana 2	140

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 [18]:

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

query = """
        SELECT y.year as decade_start ,y.year + 9 as decade_end ,count(*) as movie_count
        FROM (
                SELECT  DISTINCT mov.year
                FROM Movie mov
            )
        AS y JOIN Movie M ON m.year>=y.year and M.year<=y.year +9
        GROUP BY y.year
        ORDER BY movie_count desc

        """

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

```
(78, 3)
Wall time: 153 ms
```

Out[18]:

	decade_start	decade_end	movie_count
0	2008	2017	1126
1	2009	2018	1116
2	2005	2014	1113
3	2007	2016	1112
4	2004	2013	1098

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

In [33]:

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

query =  """
            with count_movies
            as
             (
              select actors,director,distinct_movies
              from
              (
               select trim(movie_cas.pid) actors, trim(movie_direc.pid) director,count(distinct mov
ie_direc.mid)distinct_movies
               from m_director movie_direc join m_cast movie_cas on trim(movie_direc.mid) = trim(mo
vie_cas.mid)
               group by actors,director
              )
             ),
            max_mov as
             (
               select actors,director,distinct_movies
               from count_movies
               where (actors,distinct_movies) in
                                     (
                                        select actors, max(distinct_movies) distinct_movies
                                        from count_movies group by actors
                                     )
             )
            select person.name as actor_name ,distinct_movies
            from person
            JOIN(
                select actors,distinct_movies
                from (
                        select actors, director, distinct_movies
                        from max_mov
                        where actors in
                            (
                               select actors
                               from max_mov
                               group by actors
                               )
                        )
                where director in
                        (
                            select pid
                            from person where name like "%Yash Chopra%"
                        )
                    ) as t1
            ON person.pid=t1.actors
            Order by distinct_movies desc
        """

q8 = pd.read_sql_query(query, conn)
print(q8.shape)
q8.head(10)
```

```
(245, 2)
```

```
Wall time: 6min 6s
```

Out[33]:

	actor_name	distinct_movies
0	Jagdish Raj	11
1	Manmohan Krishna	10
2	Iftekhar	9
3	Shashi Kapoor	7
4	Rakhee Gulzar	5
5	Waheeda Rehman	5
6	Ravikant	4
7	Achala Sachdev	4
8	Neetu Singh	4
9	Leela Chitnis	3

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 [40]:

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

query = """
            select trim(name) actors
            from person
            where pid in
                (
                    select distinct trim(pid)
                    from m_cast where mid in
                    (
                        select trim(mid)
                        from m_cast
                        where trim(pid)
                        in (
                            select distinct trim(pid)
                            from m_cast
                            where mid in
                                (
                                    select mid
                                    from m_cast
                                    where trim(pid) in
                                        (
                                            select trim(pid)
                                            from person where trim(name) = "Shah Rukh Khan"
                                        )
                                )
                            )
                    )
            and trim(pid)
            not in
             (
               select distinct trim(pid)
               from m_cast where mid in
                 (
                     select mid
                     from m_cast
                     where trim(pid)
                     in
                      (
                          select trim(pid)
                          from person
                          where trim(name) = "Shah Rukh Khan"
                        )
                    )
                )
            and trim(pid)
            not in
            (
             select trim(pid)
             from person
             where trim(name) = "Shah Rukh Khan"
             )
        )
       """


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

```
(25698, 1)
Wall time: 581 ms
```

Out[40]:

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

In [ ]: