In [1]:

```
import sqlite3
import pandas as pd

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

In [2]:

```
sql = """ UPDATE genre SET name=trim(name) """
cur = conn.cursor()
cur.execute(sql)
conn.commit()
```

1. 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]:

```
query = pd.read_sql_query('''select distinct p.name, m.title, m.year from person p, movie m
    (select md.pid as pid, mg.mid as mid from m_director md join m_genre mg on mg.mid=md.mi
    where mg.gid IN
    (select g.gid from genre g where g.name like "%Comedy%")) as temp on temp.pid=p.pid and
    having m.year%4==0 and m.year%100!=0 or m.year%400==0''', conn)
query
```

216	Griffin Dunne	The Accidental Husband	2008
217	James Dodson	The Other End of the Line	2008
218	Sunil K. Reddy	Thikka	2016
219	Kunal Kohli	Thoda Pyaar Thoda Magic	2008
220	Guddu Dhanoa	Tu Chor Main Sipahi	1996
221	Milind Dhaimade	Tu Hai Mera Sunday	2016
222	Vijay	Tutak Tutak Tutiya	2016
223	Sachin Kamlakar Khot	Ugly Aur Pagli	2008
224	Shoojit Sircar	Vicky Donor	2012
225	Brij	Victoria No. 203	1972
226	Shyam Benegal	Welcome to Sajjanpur	2008
227	Aditya Datt	Will You Marry Me	2012

In [5]:

```
sql = """ UPDATE m_cast SET pid=trim(pid) """
cur = conn.cursor()
cur.execute(sql)
conn.commit()
```

2. List the names of all the actors who played in the movie 'Anand' (1971)

In [6]:

```
df = pd.read_sql_query('''SELECT name from Person where pid IN
                          (SELECT pid FROM m_cast where mid IN
                          (SELECT mid from movie where title = 'Anand'));''', conn)
```

In [7]:

```
df
```

3	Ramesh Deo
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

In [7]:

```
sql = '''UPDATE Movie SET year=SUBSTR(year,-4) WHERE year LIKE '% %' '''
cur = conn.cursor()
cur.execute(sql)
conn.commit()
```

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

In [8]:

```
movie = pd.read_sql_query('''SELECT name from person where pid IN
                          (SELECT pid FROM m_cast where mid IN
                          (SELECT mid FROM movie WHERE year NOT BETWEEN 1970 AND 1990))''',
movie
```

29981	Bharat Dabholkar
29982	Jyothika
29983	Marion Rodrigues
29984	Yograj Bhat
29985	Sadhu Kokila
29986	Parvin Dabas
29987	Vineet Khetrapal
29988	C. Jenner Jose
29989	Mukesh Asopa
29990	Rahat Kazmi
29991	Srinivas Sunderrajan
29992	Abbas
29993	Iqbal

4. 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 [11]:

```
query = '''
    select a.name, count(b.mid) num_movies from person a left join m_director b
    on a.pid=b.pid group by a.pid having num_movies>10 order by num_movies desc
'''
result = pd.read_sql_query(query, conn)
result
```

98	Mohan Kumar	12
99	Jag Mundhra	12
100	K. Ravi Shankar	12
101	Shantaram Rajaram Vankudre	12
102	Babbar Subhash	12
103	Deepak Tijori	12
104	Onir	12
105	Vipul Amrutlal Shah	12
106	Kabir Khan	12
107	Shaad Ali	12
108	Samir Karnik	12
109	A.R. Murugadoss	12
110	Vivek Agnihotri	12

5. a. For each year count the number of movies in that year that had only feamle actors

In [11]:

```
fr = pd.read_sql_query('''select z.year, count(*)
from Movie z
where not exists (select *
                  from Person x, M_Cast xy
                  where x.PID = xy.PID and xy.MID = z.MID and x.Gender!='Female')
group by z.year''', conn)
fr
```

Out[11]:

	year	count(*)
0	1939	1
1	1999	1
2	2000	1
3	2009	1
4	2012	1
5	2018	2

5. 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 [12]:

```
perc_query = pd.read_sql_query('''select a.year, a.c*100.00/b.c as percentage, b.c as total
from (select z.year, count(*) as c
      from Movie z
      where not exists (select *
                        from Person x, M_Cast xy
                        where x.PID = xy.PID and xy.MID = z.MID and x.Gender!='Female')
      group by z.year) a,
     (select z.year, count(*) as c from Movie z group by z.year) b
where a.year=b.year
order by a.year''', conn)
perc_query
```

Out[12]:

	year	percentage	total_overall
0	1939	50.000000	2
1	1999	1.515152	66
2	2000	1.562500	64
3	2009	0.909091	110
4	2012	0.900901	111
5	2018	1.923077	104

6. 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 [39]:

```
l_cast_query = pd.read_sql_query('''select x.title, count(distinct xy.PID) as c
from Movie x, M_Cast xy
where x.MID = xy.MID
group by x.MID, x.title
having not exists (select uv.MID
                   from M_Cast uv
                   group by uv.MID
                   having count(distinct uv.PID) > count(distinct xy.PID))''', conn)
l_cast_query
```

Out[39]:

	title	c
0	Ocean's Eight	238

7. 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 [13]:

```
decade_query = pd.read_sql_query('''select y.year, count(*)
from (select distinct x.year from Movie x) y,
     Movie z
where y.year <= z.year and z.year < y.year+10
group by y.year
having not exists (select y1.year
                   from (select distinct x1.year from Movie x1) y1, Movie z1
                   where y1.year <= z1.year and z1.year < y1.year+10
                   group by y1.year
                   having count(z1.MID) > count(z.MID))''', conn)
decade_query
```

Out[13]:

	year	count(*)
0	2008	1205

8. Find the actors that were never unemployed for more than 3 years at a stretch. (Assume that the actors remain unemployed between two consecutive movies).

In [18]:

```
unemployed_query = pd.read_sql_query('''select Name from Person where PID
    not in(select distinct(PID)from M_Cast as C1 natural join Movie as
    M1 where exists(select MID from M_Cast as C2 natural join Movie as M2 where
    C1.PID = C2.PID and (M2.year - 3) > M1.year and
    not exists(select MID from M_Cast as C3 natural join
    Movie as M3 where C1.PID = C3.PID and M1.year < M3.year and M3.year < M2.year)))''', cc
unemployed_query
```

	Name
0	Christian Bale
1	Cate Blanchett
2	Benedict Cumberbatch
3	Naomie Harris
4	Andy Serkis
5	Peter Mullan
6	Jack Reynor
7	Eddie Marsan
8	Tom Hollander
9	Matthew Rhys
10	Freida Pinto
11	Rohan Chand
12	Kaveshan Pillay

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

In [ ]:

```

```

10. The Shahrukh number of an actor is the length of the shortest path between the actor andShahrukh Khan in the "co-acting" graph. That is, Shahrukh Khan has Shahrukh number 0; allactors 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 [44]:

```
sql = """ UPDATE Person SET Name=trim(Name) """
cur = conn.cursor()
cur.execute(sql)
conn.commit()
```

In [45]:

```
shahrukh_queryy = pd.read_sql_query('''select  count(distinct c2.PID)
from Person a0, M_Cast c0, M_Cast c1a, M_Cast c1b, M_Cast c2
where  a0.Name = 'Shah Rukh Khan'
   AND a0.PID   = c0.PID
   AND c0.MID  = c1a.MID
   AND c1b.PID = c1a.PID
   AND c1b.MID = c2.MID
   AND c2.PID NOT IN
        (select d1.PID
         from  Person b0, M_Cast d0, M_Cast d1
         where b0.Name = 'Shah Rukh Khan'
           AND b0.PID  = d0.PID
           AND d0.MID = d1.MID
        )''', conn)
shahrukh_queryy
```

Out[45]:

	count(distinct c2.PID)
0	25698