In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
import sqlite3 as db

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

In [3]:

```
result = pd.read_sql_query('SELECT * FROM MOVIE', conn)
```

In [4]:

```
result
```

Out[4]:

	index	MID	title	year	rating	num_votes
0	0	tt2388771	Mowgli	2018	6.6	21967
1	1	tt5164214	Ocean's Eight	2018	6.2	110861
2	2	tt1365519	Tomb Raider	2018	6.4	142585
3	3	tt0848228	The Avengers	2012	8.1	1137529
4	4	tt8239946	Tumbbad	2018	8.5	7483
...	...	...	...	...	...	...
3470	3470	tt0090611	Allah-Rakha	1986	6.2	96
3471	3471	tt0106270	Anari	1993	4.7	301
3472	3472	tt0852989	Come December	2006	5.7	57
3473	3473	tt0375882	Kala Jigar	1939	3.3	174
3474	3474	tt0375890	Kanoon	1994	3.2	103

3475 rows × 6 columns

Question 1

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 [9]:

```
Question1 = pd.read_sql_query("SELECT DISTINCT p.Name, m.title, m.year FROM  Person p, Movie m whe
re (PID, MID) IN (Select trim(PID), MID from M_Director where MID IN ( Select MID from M_Genre whe
re GID IN ( SELECT GID FROM Genre where Genre.Name LIKE '%Comedy%'))) and m.year%4 = 0;", conn)
Question1
```

Out[9]:

	Name	title	year
0	Milap Zaveri	Mastizaade	2016
1	Milap Zaveri	Mastizaade	2016
2	Danny Leiner	Harold & Kumar Go to White Castle	2004
3	Danny Leiner	Harold & Kumar Go to White Castle	2004

	Name	title	year
4	Anurag Kashyap	Gangs of Wasseypur	2012
...	...	...	...
410	Siddharth Anand Kumar	Let's Enjoy	2004
411	Amma Rajasekhar	Sathyam	2008
412	Oliver Paulus	Tandoori Love	2008
413	Raja Chanda	Le Halua Le	2012
414	K.S. Prakash Rao	Raja Aur Rangeeli	1996

415 rows × 3 columns

Question 2

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

In [16]:

```
Question2 = pd.read_sql_query("SELECT Name FROM Person where PID IN (SELECT trim(PID) FROM M_Cast
where MID IN (SELECT MID FROM Movie where title = 'Anand' and year = '1971'))", conn)
Question2
```

Out[16]:

	Name
0	Amitabh Bachchan
1	Rajesh Khanna
2	Sumita Sanyal
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
16	Moolchand

Question 3

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

In [361]:

```
Query3 ="select DISTINCT trim(a.Name) as Name from person a \
inner join M_Cast c1 on trim(c1.PID) = a.PID \
inner join M_Cast c2 on trim(c2.PID) = a.PID \
inner join Movie m1 on c1.mid = m1.mid \
inner join Movie m2 on c2.mid = m2.mid \
where m1.year >1990 and m2.year <1970 \
```

```
"
Question3 = pd.read_sql_query(Query3, conn)
Question3
```

Out[361]:

	Name
0	Rishi Kapoor
1	Amitabh Bachchan
2	Asrani
3	Zohra Sehgal
4	Rakesh Sharma
...	...
328	Manish
329	Vijayalaxmi
330	Manohar Deepak
331	Brij Bhushan
332	Radha

333 rows × 1 columns

Question 4

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 [333]:

```
Query4 = " select p.name, vw.movie_count from Person p \
join (select md.pid, count(*) as movie_count from M_Director md \
group by md.pid \
having count(*) > 10 \
)vw on p.pid = vw.pid \
Order by movie_count DESC \
"

Question4 = pd.read_sql_query(Query4, conn)
Question4
```

Out[333]:

	Name	movie_count
0	David Dhawan	39
1	David Dhawan	39
2	Mahesh Bhatt	35
3	Mahesh Bhatt	35
4	Ram Gopal Varma	30
...	...	...
83	Mohit Suri	11
84	Ketan Mehta	11
85	Pramod Chakravorty	11
86	Govind Nihalani	11
87	Nasir Hussain	11

88 rows × 2 columns

Question 5

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

In [309]:

```
Query5 = " select m.year, count(*) as movie_count from Movie m \
join (select distinct mid from M_Cast where mid not in ( \
select mc.mid from M_Cast mc  \
join Person p on p.pid = trim(mc.pid) \
where p.gender = 'Male'))vw on vw.mid = m.mid \
group by m.year \
"
Question5a = pd.read_sql_query(Query5, conn)
Question5a
```

Out[309]:

	year	movie_count
0	1939	1
1	1999	1
2	2000	1
3	2009	1
4	2012	1
5	2018	1
6	I 2018	1

Year data is corrupted with Roman numbers added. Else 2018 count has to be 2

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.

In [310]:

```
Query5b = " select m.year, count(m.mid) as total_movie_count, i_vw.female_only_cast_movie_count, (
i_vw.female_only_cast_movie_count*100/(count(m.mid)*1.0)) as percentage_female_only_cast_movie fro
m Movie m  \
left join (select m.year, count(*) as female_only_cast_movie_count from Movie m \
join ( select distinct mid from M_Cast  \
where mid not in  ( select mc.mid from M_Cast mc \
join Person p on p.pid = trim(mc.pid) \
where p.gender = 'Male' ) )vw on vw.mid = m.mid \
group by m.year)i_vw on m.year = i_vw.year \
group by m.year \
"
Question5b = pd.read_sql_query(Query5b, conn)
Question5b
```

Out[310]:

	year	total_movie_count	female_only_cast_movie_count	percentage_female_only_cast_movie
0	1931	1	NaN	NaN
1	1936	3	NaN	NaN
2	1939	2	1.0	50.0
3	1941	1	NaN	NaN
4	1943	1	NaN	NaN
...	...	...	...	...
120	IV 2011	1	NaN	NaN
121	IV 2017	1	NaN	NaN
122	V 2015	1	NaN	NaN

	year	total_movie_count	female_only_cast_movie_count	percentage_female_only_cast_movie
123	VI 2015	1	NaN	NaN
124	XVII 2016	1	NaN	NaN

125 rows × 4 columns

Year data is corrupted with Roman numbers added.

Question 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 [204]:

```
Query6 = "select vw.mid, m.title, max(vw.cast_count) as cast_size from ( \
select count(*) as cast_count, mid \
from M_Cast group by mid ) vw \
join Movie m on m.mid = vw.mid \
"

Question6 = pd.read_sql_query(Query6, conn)
Question6
```

Out[204]:

	mid	title	cast_size
0	tt5164214	Ocean's Eight	238

Question 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 [205]:

```
Query7 = "SELECT Decade, max(movie_counts) from  ( \
SELECT Decade, count(*) as movie_counts from ( \
SELECT m.year, vw.min_year, (((m.year-vw.min_year)/10)+1) as Decade from Movie m \
JOIN ( select min(year) as min_year from Movie ) vw on 1 = 1 ) i_vw \
GROUP BY Decade \
)o_vw \
"

Question7 = pd.read_sql_query(Query7, conn)
Question7
```

Out[205]:

	Decade	max(movie_counts)
0	8	1012

Question 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 [327]:

```
Query8 = "SELECT em.PID,em.Name, (em.next_year - em.year) as gap from( \
SELECT i_unem.pid, i_unem.name, i_unem.title, i_unem.year, LEAD(i_unem.year, 1, 0) OVER (PARTITION
BY i_unem.name ORDER BY i_unem.year ASC) AS next_year from ( \
(SELECT distinct trim(pid) as pid, trim(name) as name from Person  )p \
JOIN(SELECT distinct trim(mid) as mid, trim(pid) as pid from M_Cast )mc on p.pid = mc.pid \
JOIN(SELECT trim(mid) as mid, trim(title) as title, trim(year) as year from Movie)m on m.mid = mc.
mid)i_unem )em \
Group By em.Name \
Having em.next_year > 0 and (em.next_year - em.year) < 3 \
"

Question8 = pd.read_sql_query(Query8, conn)
Question8
```

Out[327]:

	pid	name	gap
0	nm1869655	A. Abdul Hameed	1
1	nm0359845	A.K. Hangal	1
2	nm3901254	A.R. Manikandan	0
3	nm1436693	A.R. Murugadoss	1
4	nm4563111	A.R. Rama	1
...	...	...	...
4314	nm3035273	Ziyah Vastani	2
4315	nm9096966	Zoya Hussain	1
4316	nm1796288	Zubair Khan	1
4317	nm0958276	Zubeen Garg	1
4318	nm1302631	Zulfi Sayed	1

4319 rows × 3 columns

Question9

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

In [207]:

```
Query9 = "select a.*, b.Yash_Chopra_Directed_Movies from ( \
select distinct trim(p.pid) as Actor_Id, \
trim(p.name) as Actor_Name,count(distinct m.mid) as Non_Yash_Chopra_Directed_Movies from Person p
\
join M_Cast mc on trim(mc.pid) = p.pid \
join Movie m on m.mid = mc.mid \
join M_Director md on md.mid = m.mid \
join Person p1 on p1.pid = trim(md.pid) \
where trim(p1.name) != \'Yash Chopra\' \
group by trim(p.pid))a \
left join ( \
select distinct trim(p.pid) as Actor_Id,  trim(p.name) as Actor_Name, trim(p1.name) as
Director_Name, count(distinct m.mid) as Yash_Chopra_Directed_Movies from Person p \
join M_Cast mc on trim(mc.pid) = p.pid \
join Movie m on m.mid = mc.mid \
join M_Director md on md.mid = m.mid \
join Person p1 on p1.pid = trim(md.pid) \
where trim(p1.name) = \'Yash Chopra\' \
group by trim(p.pid))b on a.Actor_Id = b.Actor_Id \
where b.Yash_Chopra_Directed_Movies > a.Non_Yash_Chopra_Directed_Movies \
"

Question9 = pd.read_sql_query(Query9, conn)
Question9
```

Out[207]:

	Actor_Id	Actor_Name	Non_Yash_Chopra_Directed_Movies	Yash_Chopra_Directed_Movies
0	nm0007181	Yash Chopra	1	2
1	nm1767604	Ashok Verma	1	2
2	nm3163800	Nazir	1	2

Question 10

10. 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.

This is no Shahrukh Khan in the actors data, Question has to be Shah Rukh Khan

In [370]:

```
Query10a = "Select Name as SRK_Nmuber_2 from Person where PID IN (Select trim(PID) from M_Cast whe
re MID IN ( Select MID from M_Cast where trim(PID) IN (Select trim(PID) as PID from M_Cast where M
ID IN ( SELECT m.MID FROM Movie m \
INNER JOIN M_Cast c ON m.MID = c.MID \
INNER JOIN person p ON trim(c.PID) = trim(p.PID) \
where trim(Name) = 'Shah Rukh Khan')))) \
"
Question10a = pd.read_sql_query(Query10a, conn)
Question10a
```

Out[370]:

	SRK_Nmuber_2
0	Freida Pinto
1	Rohan Chand
2	Damian Young
3	Waris Ahluwalia
4	Caroline Christl Long
...	...
28438	Rahat Kazmi
28439	Srinivas Sunderrajan
28440	Abbas
28441	Gulshan Kumar
28442	Sushma Shiromani

28443 rows × 1 columns