In [1]:

```
import sqlite3
import pandas as pd
conn = sqlite3.connect('Db-IMDB.db')
c = conn.cursor()
```

In [47]:

```
c.execute('UPDATE Person SET PID = TRIM(PID)')
c.execute('UPDATE M_Cast SET PID = TRIM(PID)')
c.execute('UPDATE M_Cast SET MID = TRIM(MID)')
c.execute('UPDATE M_Director SET PID = TRIM(PID)')
c.execute('UPDATE M_Director SET MID = TRIM(MID)')
c.execute('UPDATE M_Genre SET MID = TRIM(MID)')
c.execute('UPDATE M_Genre SET GID = TRIM(GID)')
c.execute('UPDATE Movie SET MID = TRIM(MID)')
c.execute('UPDATE Genre SET GID = TRIM(GID)')
c.execute('UPDATE Person SET Name = TRIM(Name)')
c.execute('UPDATE M_Producer SET PID=TRIM(PID)')
c.execute('UPDATE Movie SET year =REPLACE(year, "I ", "")')
```

Out[47]:

```
<sqlite3.Cursor at 0x252da46ea40>
```

In [18]:

```
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 [26]:

```
result=pd.read_sql_query('SELECT DISTINCT Person.Name,  Movie.title,  Movie.year \
                          FROM Genre, Person, M_Director, M_Genre ,Movie \
                          WHERE M_Director.MID= Movie.MID AND M_Director.PID= Person.PID  \
                            AND Movie.year % 4=0 \
                           AND M_Genre.MID= M_Director.MID AND M_Genre.GID = \
                           Genre.GID AND Genre.Name LIKE "%Comedy%" ORDER BY Movie.year LIMIT 20',
conn)
result
```

Out[26]:

	Name	title	year
0	Satyen Bose	Jagriti	1956
1	Chetan Anand	Funtoosh	1956
2	Amit Mitra	Jagte Raho	1956
3	Mohan Segal	New Delhi	1956
4	R.K. Nayyar	Love in Simla	1960
5	Bimal Roy	Parakh	1960
6	S.U. Sunny	Kohinoor	1960
7	Ravindra Dave	Dulha Dulhan	1964
8	Kidar Nath Sharma	Chitralekha	1964
9	Shakti Samanta	Kashmir Ki Kali	1964
10	K. Shankar	Rajkumar	1964

	Name	title	year
12	Debu Sen	Do Boond Chal	1968
13	Lekh Tandon	Jhuk Gaya Aasman	1968
14	A. Bhimsingh	Sadhu Aur Shaitaan	1968
15	Jyoti Swaroop	Padosan	1968
16	Naresh Kumar	Gora Aur Kala	1972
17	S.S. Balan	Sanjog	1972
18	Ramesh Sippy	Seeta Aur Geeta	1972
19	Mukul Dutt	Raaste Kaa Patthar	1972

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

In [7]:

```
result=pd.read_sql_query('SELECT Person.Name FROM Person WHERE PID IN(SELECT PID FROM M_Cast JOIN
Movie WHERE \
                         Movie.MID = M_Cast.MID AND Movie.title="Anand" \
                         AND M_Cast.PID NOT IN(SELECT M_Cast.PID FROM \
                         M_Director, M_Cast WHERE M_Cast.PID = M_Director.PID)) \
                         ', conn)
result
```

Out[7]:

	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

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

In [28]:

```
result=pd.read_sql_query('SELECT Person.Name FROM Person \
WHERE PID IN(SELECT Person.PID FROM Person WHERE PID IN (SELECT M_Cast.PID FROM M_Cast JOIN Movie
WHERE \
                         Movie.MID = M_Cast.MID AND Movie.year<1970  \
                         AND M_Cast.PID NOT IN(SELECT M_Cast.PID FROM \
                         M_Director, M_Cast WHERE M_Cast.PID = M_Director.PID))) \
                         AND PID IN(SELECT M_Cast.PID FROM M_Cast JOIN Movie WHERE \
```

```
                        M_Director, M_Cast WHERE M_Cast.PID = M_Director.PID)) LIMIT 20 \
                        ', conn)
result
```

Out[28]:

	Name
0	Amitabh Bachchan
1	Zohra Sehgal
2	Parikshat Sahni
3	Rakesh Sharma
4	Sanjay Dutt
5	Ric Young
6	Yusuf
7	Suhasini Mulay
8	A.K. Hangal
9	Jeremy Child
10	Farida Jalal
11	Waheeda Rehman
12	Rajesh Khanna
13	Ramesh Deo
14	Seema Deo
15	Asit Kumar Sen
16	Lalita Kumari
17	Brahm Bhardwaj
18	Lalita Pawar
19	Dara Singh

1. 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 [10]:

```
result=pd.read_sql_query('SELECT Person.Name, COUNT(*) AS "No of Movies"\
                    FROM Person, M_Director WHERE Person.PID =M_Director.PID \
                    GROUP BY M_Director.PID HAVING "No of Movies" >4 \
                    ORDER BY COUNT(*) DESC LIMIT 20', conn)
result
```

Out[10]:

	Name	No of Movies
0	David Dhawan	78
1	Mahesh Bhatt	70
2	Ram Gopal Varma	60
3	Vikram Bhatt	58
4	Hrishikesh Mukherjee	54
5	Yash Chopra	42
6	Basu Chatterjee	38
7	Shakti Samanta	38

	Name	No of Movies
9	Shyam Benegal	34
10	Abbas Alibhai Burmawalla	34
11	Manmohan Desai	32
12	Gulzar	32
13	Raj N. Sippy	32
14	Raj Kanwar	30
15	Mahesh Manjrekar	30
16	Priyadarshan	30
17	Indra Kumar	28
18	Raj Khosla	28
19	Rahul Rawail	28

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

   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 [50]:

```
result=pd.read_sql_query('SELECT * FROM(SELECT M2.year,  \
(SELECT COUNT(*) FROM(SELECT  M1.title \
    FROM Person as P2 JOIN M_Cast ON P2.PID= M_Cast.PID JOIN \
    Movie AS M1 ON M_Cast.MID \
=M1.MID AND M1.year=M2.year AND M_Cast.PID NOT IN(SELECT M_Cast.PID FROM \
M_Director, M_Cast WHERE M_Cast.PID = M_Director.PID \
)AND M_Cast.PID NOT IN(SELECT M_Cast.PID FROM \
M_Producer, M_Cast WHERE M_Cast.PID = M_Producer.PID ) \
GROUP BY M1.MID  HAVING COUNT(CASE WHEN P2.Gender="Male" then 1 end)=0)) as "female movie count" \
FROM Movie AS M2 \
GROUP BY M2.year \
ORDER BY M2.year) AS T WHERE T."female movie count">0',conn)
result
```

Out[50]:

	year	female movie count
0	1964	1
1	1996	1
2	1999	1
3	2000	1
4	2001	2
5	2004	1
6	2009	1
7	2018	2

In [49]:

```
result=pd.read_sql_query('SELECT * , (((T."female movie count")*1.00)/((T."total
Movie")*1.00))*100.00 AS "Female Movie Percentage" FROM(SELECT  M2.year,  \
(SELECT COUNT(*) FROM(SELECT  * \
    FROM Person as P2 JOIN M_Cast ON P2.PID= M_Cast.PID JOIN \
    Movie AS M1 ON M_Cast.MID \
=M1.MID AND M1.year=M2.year AND M_Cast.PID NOT IN(SELECT M_Cast.PID FROM \
M_Director, M_Cast WHERE M_Cast.PID = M_Director.PID \
)AND M_Cast.PID NOT IN(SELECT M_Cast.PID FROM \
M_Producer, M_Cast WHERE M_Cast.PID = M_Producer.PID ) \
```

```
(SELECT COUNT(*) FROM Movie AS M5 WHERE M5.year=M2.year) AS "total Movie"  \
FROM Movie AS M2 \
 GROUP BY M2.year \
ORDER BY M2.year ) AS T WHERE T."female movie count">0',conn)
result
```

Out[49]:

	year	female movie count	total Movie	Female Movie Percentage
0	1964	1	15	6.666667
1	1996	1	60	1.666667
2	1999	1	66	1.515152
3	2000	1	64	1.562500
4	2001	2	73	2.739726
5	2004	1	103	0.970874
6	2009	1	109	0.917431
7	2018	2	103	1.941748

1. 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 [26]:

```
result=pd.read_sql_query('select Movie.title, COUNT(DISTINCT M_Cast.PID) as " \
Size" \
FROM Movie JOIN M_Cast ON Movie.MID= M_Cast.MID \
GROUP BY Movie.MID \
ORDER BY " \
Size" DESC LIMIT 20', conn)
result
```

Out[26]:

	title	Size
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
10	The Avengers	138
11	Housefull 3	129
12	Fan	127
13	Split Wide Open	126
14	Bajrangi Bhaijaan	124
15	Train Station	122
16	Daddy	121

	title	size
18	Octopussy	126
19	Dhoom:3	115

1. 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 [28]:

```
result=pd.read_sql_query('SELECT (CAST((Movie.year - (SELECT MIN(Movie.year) FROM Movie)) / 10 AS
INT)  * 10 \
    + (SELECT MIN(Movie.year) FROM Movie)) AS "Start",\
(CAST((Movie.year - (SELECT MIN(Movie.year) FROM Movie)) / 10 AS INT) * 10 \
+ (SELECT MIN(Movie.year) FROM Movie) + 9) AS "End", \
 COUNT(Movie.MID) AS "No of Movies" \
FROM Movie  \
    GROUP BY "Start" \
ORDER BY "End" ', conn)
result
```

Out[28]:

	Start	End	No of Movies
0	1	10	117
1	1931	1940	6
2	1941	1950	14
3	1951	1960	83
4	1961	1970	155
5	1971	1980	260
6	1981	1990	354
7	1991	2000	573
8	2001	2010	1012
9	2011	2020	901

1. 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 [8]:

```
result=pd.read_sql_query('SELECT  Person.Name \
FROM Person WHERE Person.PID NOT IN(SELECT DISTINCT(Person.PID) \
FROM M_Cast \
 AS M_C1 NATURAL JOIN Movie AS M1 \
WHERE EXISTS(SELECT M_C2.MID \
FROM M_Cast \
 AS M_C2 NATURAL JOIN Movie AS M2  \
WHERE M_C1.PID = M_C2.PID AND (M2.year - 3) > M1.year \
AND NOT EXISTS(SELECT M_C3.MID \
FROM M_Cast \
 AS M_C3 NATURAL JOIN Movie AS M3  \
WHERE M_C1.PID = M_C3.PID AND M1.year < M3.year \
AND M3.year < M2.year))) LIMIT 20', conn)
result
```

Out[8]:

	Name
0	Christian Bale

	Name
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	Keveshan Pillay
13	Louis Ashbourne Serkis
14	Moonsamy Narasigadu
15	Soobrie Govender
16	Gopal Singh
17	Kista Munsami
18	Mahomed Araf Cassim
19	Riaz Mansoor

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

In [4]:

```
result=pd.read_sql_query('select P1.Name, count(Movie.MID) as "No of Movies" \
 FROM Person AS P1 JOIN M_Cast ON M_Cast.PID=P1.PID JOIN Movie ON M_Cast.MID= Movie.MID \
JOIN M_Director ON Movie.MID = M_Director.MID \
JOIN Person AS P2 ON M_Director.PID= P2.PID \
WHERE P2.Name = "Yash Chopra" GROUP BY P1.PID \
HAVING COUNT(Movie.MID) > (SELECT COUNT(Movie.MID) \
FROM Person AS P3 JOIN M_Cast ON M_Cast.PID=P3.PID \
JOIN Movie ON M_Cast.MID=Movie.MID JOIN M_Director ON Movie.MID = M_Director.MID \
JOIN Person AS P4 on M_Director.PID = P4.PID \
WHERE P1.PID = P3.PID AND P4.Name != "Yash Chopra" \
GROUP BY P4.PID) \
ORDER BY "No of Movies" DESC', conn)
result
```

Out[4]:

	Name	No of Movies
0	Manmohan Krishna	20
1	Shashi Kapoor	14
2	Anupam Kher	14
3	Jagdish Raj	11
4	Iftekhar	9
5	Hema Malini	8
6	Vikas Anand	8
7	Madan Puri	8
8	Deven Verma	8
9	Amitabh Bachchan	6
10	Rakhee Gulzar	5
11	Waheeda Rehman	5

	Name	No of Movies
13	Rajesh Khanna	3
14	Prem Chopra	3
15	Sudha Chopra	3
16	Anil Kapoor	3
17	Annu Kapoor	3
18	Sanjeev Kumar	3
19	Yunus Parvez	3
20	Parikshat Sahni	3
21	Saul George	3
22	Surendra Rahi	3
23	Juhi Chawla	2
24	Saeed Jaffrey	2

1. 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 [3]:

```
result=pd.read_sql_query('SELECT Name FROM Person WHERE Name LIKE "S%K"',conn)
result
```

Out[3]:

	Name
0	Satish Kaushik
1	Saawan Kumar Tak
2	Samir Karnik
3	Shashank
4	Shiladitya Moulik
5	Saandesh Nayak
6	Sachin Bhowmick
7	S. Deepak
8	Sushil Malik

As Sharukh Khan is not present in Database so I will use Aamir Khan as Name

In [40]:

```
result=pd.read_sql_query('SELECT DISTINCT P4.Name  FROM Person AS P4 JOIN M_Cast AS M_C3 \
ON P4.PID =M_C3.PID JOIN Movie AS M4 ON M4.MID=M_C3.MID \
WHERE P4.PID NOT IN(SELECT M_Director.PID FROM M_Director) AND P4.Name !="Aamir Khan" \
AND M4.title IN(SELECT M3.title FROM Movie AS M3 JOIN M_Cast AS M_C2 ON \
M3.MID=M_C2.MID JOIN Person AS P2 ON P2.PID=M_C2.PID WHERE P2.Name IN (SELECT Person.Name FROM Per
son JOIN M_Cast ON \
Person.PID=M_Cast.PID JOIN Movie as M2 ON M2.MID= M_Cast.MID AND M2.MID IN(SELECT Movie.MID FROM M
ovie JOIN M_Cast \
ON Movie.MID = M_Cast.MID WHERE M_Cast.PID IN (SELECT Person.PID FROM M_Cast JOIN \
Person ON M_Cast.PID= Person.PID WHERE Person.NAME ="Aamir Khan")) AND Person.Name !="Aamir Khan"
\
AND Person.PID NOT IN(SELECT M_Director.PID FROM M_Director))) \
AND P4.Name NOT IN(SELECT P2.Name FROM Movie AS M3 JOIN M_Cast AS M_C2 ON \
M3.MID=M_C2.MID JOIN Person AS P2 ON P2.PID=M_C2.PID WHERE P2.Name IN (SELECT Person.Name FROM Per
```

```
ovie JOIN M_Cast \
ON Movie.MID = M_Cast.MID WHERE M_Cast.PID IN (SELECT Person.PID FROM M_Cast JOIN \
Person ON M_Cast.PID= Person.PID WHERE Person.NAME ="Aamir Khan")) AND Person.Name !="Aamir Khan"
\
AND Person.PID NOT IN(SELECT M_Director.PID FROM M_Director))) LIMIT 10',conn)
result
```

Out[40]:

	Name
0	Freida Pinto
1	Rohan Chand
2	Caroline Christl Long
3	Rajeev Pahuja
4	Michelle Santiago
5	Jandre le Roux
6	Raj Awasti
7	Michael Chapman
8	James Heron
9	Alex Jaep