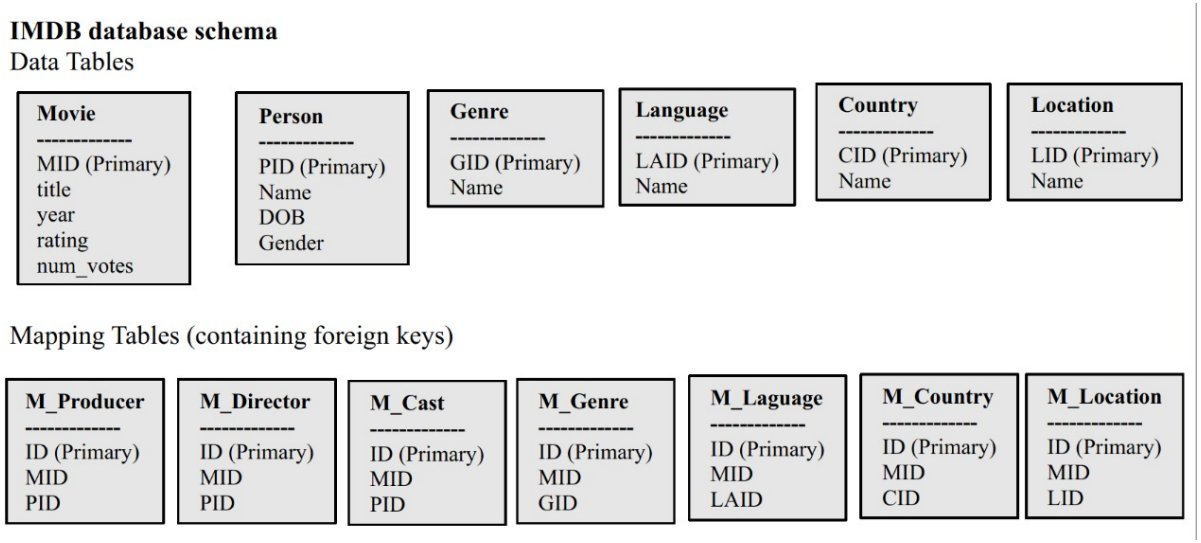
In [1]:
```
import sqlite3
import pandas as pd

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

In [2]:
```
from IPython.display import Image
Image(filename='db_schema.jpeg')
```

Out[2]:



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 [56]:
```
Movie = pd.read_sql_query("select Title as Title,Year as Year from Movi
e where MID in \
                          (select MID from M_Director where PID in \
                          (select PID from M_Director where MID in \
                          (Select MID from Movie where (year%4=0) and
 (year%100!=0) or (year%400=0) and MID in \
```

```
                    (Select MID from M_Genre where GID in (Select GID fro
m Genre where Name like '%Comedy%'))))) ",con)

Movie.head(5)
```

Out[56]:

	Title	Year
0	The Avengers	2012
1	Kedarnath	2018
2	Captain America: Civil War	2016
3	Andhadhun	2018
4	Lion	2016

In [57]:
```
Director = pd.read_sql_query("select Name from Person where PID in \
                              (select distinct(PID) from M_Director wher
e MID in \
                              (Select MID from Movie where (year%4=0) an
d (year%100!=0) or (year%400=0) and MID in \
                    (Select MID from M_Genre where GID in (Select GID fro
m Genre where Name like '%Comedy%')))) ",con)

Director.head(5)
```

Out[57]:

	Name
0	Griffin Dunne
1	Mahesh Manjrekar
2	Asrani
3	Madonna
4	Gurinder Chadha

In [58]:
```
Output=pd.read_sql_query("select p.Name,a.title as Movie,a.year as Yea
r,c.Name as Genre \
                          from Movie a , M_Director b,Genre c,M_Genre d
 ,Person p \
                          on a.MID = d.MID and \
                          a.MID = b.MID and \
                          c.Name like '%Comedy%' and b.PID=p.PID \
                          and a.year%4=0 group by p.Name,a.title",con)

Output
```

Out[58]:

	Name	Movie	Year	Genre
0	A. Bhimsingh	Aadmi	1968	Comedy, Horror, Musical
1	A. Bhimsingh	Joroo Ka Ghulam	1972	Comedy, Horror, Musical
2	A. Bhimsingh	Sadhu Aur Shaitaan	1968	Comedy, Horror, Musical
3	A. Muthu	Tera Jadoo Chal Gayaa	2000	Comedy, Horror, Musical
4	A.R. Murugadoss	Akira	I 2016	Comedy, Horror, Musical
...	...	...	...	...
1558	Yash Chopra	Vijay	1988	Comedy, Horror, Musical
1559	Yogesh Ishwar	Aaghaaz	2000	Comedy, Horror, Musical
1560	Yograj Bhat	Ranga S.S.L.C	2004	Comedy, Horror, Musical
1561	Yûgô Sakô	The Prince of Light	2000	Comedy, Horror, Musical
1562	Zaigham Imam	Alif	I 2017	Comedy, Horror, Musical

1563 rows × 4 columns

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

In [76]:
```
Output=pd.read_sql_query("""select name as Actor from person where trim
```

```
(pid) in
                                (select trim(pid) from m_cast where mid in
                                 (select mid from movie where trim(title)
='Anand')) """,con)
Output
```

Out[76]:

	Actor
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 [96]:
```
Output=pd.read_sql_query("""select name as Actor from person where trim
(pid) in
                                (select trim(pid) from m_cast where mid i
n
                                   (select mid from movie m where m.year
 > 1990)
                                   and pid in (select pid from m_cast whe
re mid in
                                           (select mid from movie n where
 n.year < 1970))) """,con)
Output
```

Out[96]:

	Actor
0	Rishi Kapoor
1	Amitabh Bachchan
2	Asrani
3	Zohra Sehgal
4	Parikshat Sahni
...	...
348	Vinod Mehra
349	Deven Verma
350	Master Bhagwan
351	Rishi Kapoor
352	Asrani

353 rows × 1 columns

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 [98]:
```
Output=pd.read_sql_query("select p.Name As Actor,count(*) as Count from
 Person p, M_Director md on md.PID=p.PID group by\
                    p.PID,p.Name having count(*) > 10 order by count des
c",con)
Output
```

Out[98]:

	Actor	Count
0	David Dhawan	39
1	David Dhawan	39
2	Mahesh Bhatt	35
3	Mahesh Bhatt	35
4	Priyadarshan	30
...	...	...
83	Ketan Mehta	11
84	Govind Nihalani	11
85	Govind Nihalani	11
86	Mohit Suri	11
87	Mohit Suri	11

88 rows × 2 columns

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

In [15]:
```
Output=pd.read_sql_query("""select movie.year as Year,count(*) as Count
```

```
 from movie \
                                WHERE not exists \
                                (select * from m_cast,person where person.g
ender='Male' and m_cast.mid=movie.mid \
                                and m_cast.pid=person.pid  ) group by movi
e.year""", con)
Output
```

Out[15]:

	Year	Count
0	1939	1
1	1999	1
2	2000	1
3	2009	1
4	2012	1
5	2018	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. 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 [20]:
```
Output=pd.read_sql_query("""select female_count.year as Year,((female_c
ount.Total_movies_with_only_female)*100)/total_count.Total as Percentag
e
                                from
                                ((select movie.year as Year,count(*) as Tot
al_movies_with_only_female from movie
                                WHERE not exists
                                ( select * from m_cast,person
                                where m_cast.mid=movie.mid and m_cast.pid=p
erson.pid and person.gender='Male' )
                                group by movie.year) female_count,
```

```
                                (select movie.year,count(*) as Total from m
ovie group by movie.year) total_count)
                                where female_count.year=total_count.year"""
, con)
Output
```

Out[20]:

	Year	Percentage
0	1931	100
1	1936	100
2	1939	100
3	1941	100
4	1943	100
...	...	...
120	IV 2011	100
121	IV 2017	100
122	V 2015	100
123	VI 2015	100
124	XVII 2016	100

125 rows × 2 columns

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 [122]:
```
Output=pd.read_sql_query("select m.title as Movie,count(distinct(mc.PI
D)) as Count \
                            from Movie m join M_Cast mc on mc.MID = m.MID\
```

```
                    group by m.MID order by Count desc limit 1",con)
Output
```

Out[122]:

	Movie	Count
0	Ocean's Eight	238

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 [127]:
```
Output=pd.read_sql_query("select d.year as Start, d.year+9 as End, coun
t(*)\
                    as Count from (select distinct year from Movie) d \
                    join Movie m on m.year>=start and m.year<= end \
                    group by end order by Count desc\
                    limit 1",con)
Output
```

Out[127]:

	Start	End	Count
0	2008	2017	1128

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 [129]:
```
Output=pd.read_sql_query("select Name as Actor 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 joi
n 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<m
2.year)))",con)
Output
```

Out[129]:

	Actor
0	Christian Bale
1	Cate Blanchett
2	Benedict Cumberbatch
3	Naomie Harris
4	Andy Serkis
...	...
38280	Kannan
38281	Adrian Fulle
38282	Gulshan Kumar
38283	Iqbal
38284	Sushma Shiromani

38285 rows × 1 columns

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

In [27]:
```
Output=pd.read_sql_query("select Director, Actor, Count(*) as Movies_wi
th_YashChopra \
                          from(select p1.Name as Director, m1.title as M
ovie \
                          from Person p1 Inner Join M_Director md on TRI
M(md.PID)=p1.PID \
                          Inner Join Movie m1 on TRIM(md.MID)=m1.MID and
```

```
 p1.Name LIKE 'Yash%' Group By p1.Name, m1.title) t1 \
                            Inner Join (select p2.Name as Actor,m2.title a
s Movie from Person p2 \
                            Inner Join M_Cast mc on TRIM(mc.PID)=p2.PID \
                            Inner Join Movie m2 on TRIM(mc.MID)=m2.MID Gro
up By p2.Name, m2.title) t2 on t1.Movie=t2.Movie \
                            Group By t1.Director, t2.Actor Order By Movies
_with_YashChopra DESC",con)
Output
```

Out[27]:

	trim(name)
0	'Musafir' Radio Performing
1	A'Ali de Sousa
2	A. Abdul Hameed
3	A. Darpan
4	A. Gabibi
...	...
16160	Zulfi Sayed
16161	Zulkhumor Muminova
16162	Zurab Kapianidze
16163	Zuri Echea
16164	Zuzanna Zajac

16165 rows × 1 columns

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 [29]:
```
#http://www.mysqltutorial.org/mysql-inner-join.aspx
Output=  pd.read_sql_query("""SELECT count(distinct trim(name)) as Name
 from person p
                                inner join m_cast c on p.pid=trim(c.pid)
                                inner join movie m on m.mid=c.mid and tri
m(p.name)!='Shah Rukh Khan'
                                and m.title in
                                (SELECT distinct title from person p3
                                inner join m_cast c3 on p3.pid=trim(c3.pi
d) and trim(p3.name) = p3.name
                                inner join movie m3 on m3.mid=c3.mid and
 p3.name in
                                (SELECT distinct name from person p2
                                inner join m_cast c2 on p2.pid=trim(c2.pi
d)
                                inner join movie m2 on m2.mid=c2.mid and
 trim(p2.name)!='Shah Rukh Khan'
                                and m2.title in
                                (SELECT distinct title from person p3
                                inner join m_cast c3 on p3.pid=trim(c3.pi
d) and trim(p3.name) = 'Shah Rukh Khan'
                                inner join movie m3 on m3.mid=c3.mid)))
                                order by name""", con)

Output
```

Out[29]:

	Name
0	16165