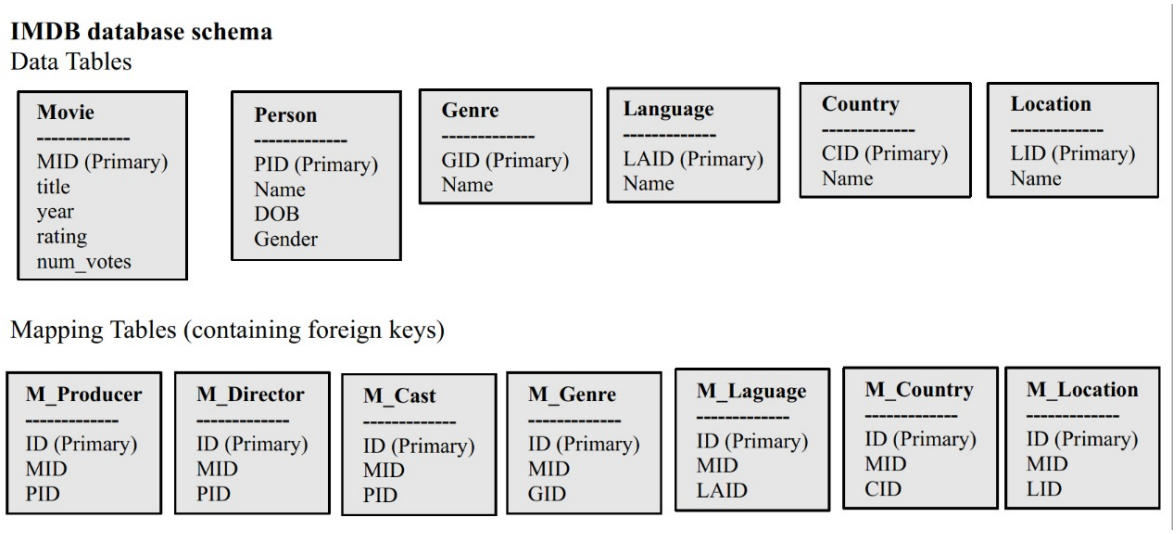
In [1]:

```
import sqlite3
import pandas as pd
import numpy as np
```

In [2]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[2]:



List all tables in database

In [3]:

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

In [4]:

```
table = pd.read_sql_query("SELECT name FROM sqlite_master WHERE type='table' ;", con)
table
```

Out[4]:

	name
0	Movie
1	Genre
2	Language
3	Country
4	Location
5	M_Location
6	M_Country
7	M_Language
8	M_Genre
9	Person
10	M_Producer
11	M_Director
12	M_Cast
13	PERSONS

#drop table cursor = con.cursor() cursor.execute('DROP TABLE PERSONS') con.commit()

In [35]:

```
P=pd.read_sql_query("SELECT * FROM PERSONS",con)
P.head()
```

Out[35]:

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

In [36]:

```
m=pd.read_sql_query("SELECT * FROM MOVIE",con)
```

In [14]:

```
m.head()
```

Out[14]:

	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

In [37]:

```
p=pd.read_sql_query("SELECT * FROM Person",con)
```

In [38]:

```
p.head()
```

Out[38]:

	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

In [17]:

```
g=pd.read_sql_query("SELECT * FROM Genre",con)
g.head()
```

Out[17]:

	index	Name	GID
0	0	Adventure, Drama, Fantasy	0
1	1	Action, Comedy, Crime	1
2	2	Action, Adventure, Fantasy	2
3	3	Action, Adventure, Sci-Fi	3
4	4	Drama, Horror, Thriller	4

In [18]:

```
l=pd.read_sql_query("SELECT * FROM Language",con)
l.head()
```

Out[18]:

	index	Name	LAID
0	0	English	0
1	1	Marathi	1
2	2	Hindi	2
3	3	Cantonese	3
4	4	Telugu	4

In [19]:

```
c=pd.read_sql_query("SELECT * FROM Country",con)
c.head()
```

Out[19]:

	index	Name	CID
0	0	UK	0
1	1	USA	1
2	2	India	2
3	3	Australia	3
4	4	Hong Kong	4

In [20]:

```
l=pd.read_sql_query("SELECT * FROM Location",con)
l.head(5)
```

Out[20]:

	index	Name	LID
0	0	Durban, South Africa	0
1	1	New York City, New York, USA	1
2	2	Cape Town Film Studios, Cape Town, Western Cap...	2
3	3	Pittsburgh, Pennsylvania, USA	3
4	4	Atlanta, Georgia, USA	4

In [23]:

```
mg=pd.read_sql_query("SELECT * FROM M_Genre",con)
mg.head(5)
```

Out[23]:

	index	MID	GID	ID
0	0	tt2388771	0	0
1	1	tt5164214	1	1
2	2	tt1365519	2	2
3	3	tt0848228	3	3
4	4	tt8239946	4	4

preprosessing Data

1.removing duplicates from person table

In [29]:

```
p=pd.read_sql_query("SELECT * FROM Person",con)
p.shape
```

Out[29]:

```
(38285, 4)
```

In [30]:

```
final=p.drop_duplicates(subset={"PID"}, keep='first')
final.shape[0]
```

Out[30]:

```
37566
```

In [31]:

```
cursor = con.cursor()
cursor.execute('CREATE TABLE PERSONS(PID varchar(50) ,Name varchar(50),Gender varchar(10));
con.commit()
```

In [32]:

```
x=list(final.iloc[0].values)
print(type(x))
print(x)
```

```
<class 'list'>
[0, 'nm0000288', ' Christian Bale', 'Male']
```

In [33]:

```
from tqdm import tqdm
cursor = con.cursor()
for i in  tqdm(range(final.shape[0])):
    x=list(final.iloc[i].values)
    cursor.execute('INSERT INTO Persons VALUES (?,?,?);',x[1:])
con.commit()
```

```
100%|████████████████████████████████████████████████████████████████
███| 37566/37566 [00:17<00:00, 2146.35it/s]
```

In [34]:

```
a=pd.read_sql_query('select PID ,COUNT(Name )from PERSONS GROUP BY PID ORDER BY COUNT(Name)
a.head()
```

Out[34]:

	PID	COUNT(Name)
0	None	0
1	nm0000002	1
2	nm0000027	1
3	nm0000039	1
4	nm0000042	1

2.Trim PID in M_cast

In [45]:

```
mc=pd.read_sql_query('select  *from M_cast  ',con)
mc.head()
```

Out[45]:

	index	MID	PID	ID
0	0	tt2388771	nm0000288	0
1	1	tt2388771	nm0000949	1
2	2	tt2388771	nm1212722	2
3	3	tt2388771	nm0365140	3
4	4	tt2388771	nm0785227	4

In [47]:

```
cursor = con.cursor()
cursor.execute('UPDATE M_Cast SET PID = REPLACE(PID, " ", "")')
con.commit()
```

3.remove romans characters in year from movie

In [49]:

```
cursor = con.cursor()
cursor.execute('UPDATE Movie SET year = REPLACE(year, "I", "");')
cursor.execute('UPDATE Movie SET year = REPLACE(year, "V", "");')
cursor.execute('UPDATE Movie SET year = REPLACE(year, "X", "");')
con.commit()
```

In [51]:

```
a=pd.read_sql_query('select  *from Movie where year like "I%"  ',con)
a.head()
```

Out[51]:

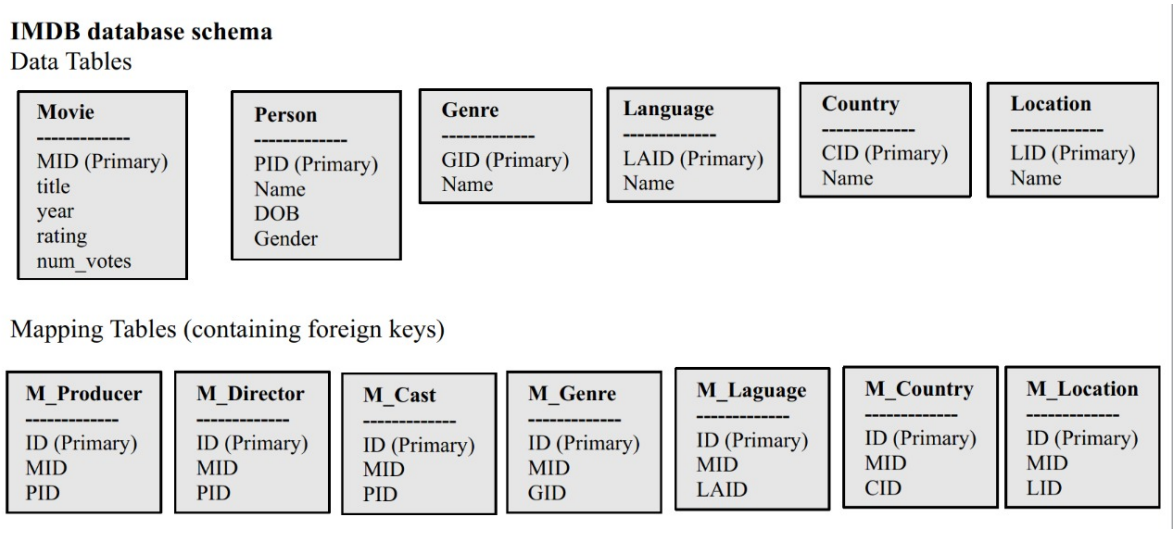
index	MID	title	year	rating	num_votes

Question 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 [3]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[3]:



In [41]:

```
output_1=pd.read_sql_query('''SELECT DISTINCT  p.name,m.title,m.year as \'Director\' FROM P
                              JOIN M_Director md on md.pid=p.pid
                              JOIN Movie m on m.mid=md.mid
                              JOIN M_Genre mg on mg.mid=m.mid
                              JOIN Genre g on g.gid=mg.gid
                              WHERE g.name LIKE"%comedy%" and m.year % 4 = 0''',con)
output_1.shape
```

Out[41]:

```
(246, 3)
```

In [42]:

```
output_1.head()
```

Out[42]:

	Name	title	Director
0	Milap Zaveri	Mastizaade	2016
1	Danny Leiner	Harold & Kumar Go to White Castle	2004
2	Anurag Kashyap	Gangs of Wasseypur	2012
3	Frank Coraci	Around the World in 80 Days	2004
4	Griffin Dunne	The Accidental Husband	2008

In [43]:

```
print(output_1)
```

```
                  Name                          title Director
0          Milap Zaveri                     Mastizaade     2016
1          Danny Leiner  Harold & Kumar Go to White Castle     2004
2        Anurag Kashyap             Gangs of Wasseypur     2012
3          Frank Coraci    Around the World in 80 Days     2004
4         Griffin Dunne         The Accidental Husband     2008
5           Anurag Basu                         Barfi!     2012
6       Gurinder Chadha              Bride & Prejudice     2004
7            Mike Judge  Beavis and Butt-Head Do America     1996
8          Abhinay Deo                       Blackmail   I 2018
9       Tarun Mansukhani                        Dostana     2008
10          Shakun Batra                  Kapoor & Sons     2016
11         Aditya Chopra            Rab Ne Bana Di Jodi     2008
12          Rohit Dhawan                        Dishoom     2016
13          Nitya Mehra                 Baar Baar Dekho     2016
14      Dibakar Banerjee          Oye Lucky! Lucky Oye!     2008
15          Umesh Shukla                OMG: Oh My God!     2012
16         Aditya Chopra                        Befikre     2016
17            Farah Khan                Happy New Year   I 2014
18          Karan Johar            Student of the Year     2012
19            Farah Khan                    Main Hoon Na     2004
20        Shoojit Sircar                     Vicky Donor     2012
21       Siddharth Anand                       Bang Bang   I 2014
22        Abbas Tyrewala        Jaane Tu... Ya Jaane Na     2008
23          Priyadarshan                     Hera Pheri     2000
24  Shubhashish Bhutiani                Hotel Salvation     2016
25          James Dodson      The Other End of the Line     2008
26          Homi Adajania                       Cocktail     2012
27          Joy Augustine                Tere Mere Sapne     1996
28          Mudassar Aziz              Happy Bhag Jayegi     2016
29          Gauri Shinde                English Vinglish     2012
..                   ...                           ...      ...
216         Anand Balraj       Daal Mein Kuch Kaala Hai     2012
217         Govind Menon              Kis Kis Ki Kismat     2004
218          Mohan Segal                      New Delhi     1956
219       Pankaj Parashar                 Ab Ayega Mazaa     1984
220        Tarun Majumdar                    Dadar Kirti     1980
221            Salim Raza                  Bach ke Zara     2008
222         Jabbar Patel               Ek Hota Vidushak     1992
223         Sanjay Chhel       Maan Gaye Mughall-E-Azam     2008
224  Sachin Kamlakar Khot                Ugly Aur Pagli     2008
225         David Dhawan                 Bol Radha Bol     1992
226            Kalpataru            Ghar Ghar Ki Kahani     1988
227          Vimal Kumar                  Suno Sasurjee     2004
228     Srinivas Bhashyam                   Paisa Vasool     2004
229      Ganapathy Bharat                        Hari Om     2004
230             Debu Sen                  Do Dooni Char     1968
231          Raj Kaushal               Shaadi Ka Laddoo     2004
232        Kabir Sadanand      Popcorn Khao! Mast Ho Jao     2004
233              Mehmood               Ginny Aur Johny     1976
234   Parvati Balagopalan                     Straight  II 2009
235       Basu Chatterjee               Lakhon Ki Baat     1984
236          Shankaraiya                    Khokababu     2012
237  Chandrakant Kulkarni              Meerabai Not Out     2008
238          Yograj Bhat                 Ranga S.S.L.C     2004
239          Deepak Anand            Yaad Rakhegi Duniya     1992
240          Vijaya Mehta                     Pestonjee     1988
```

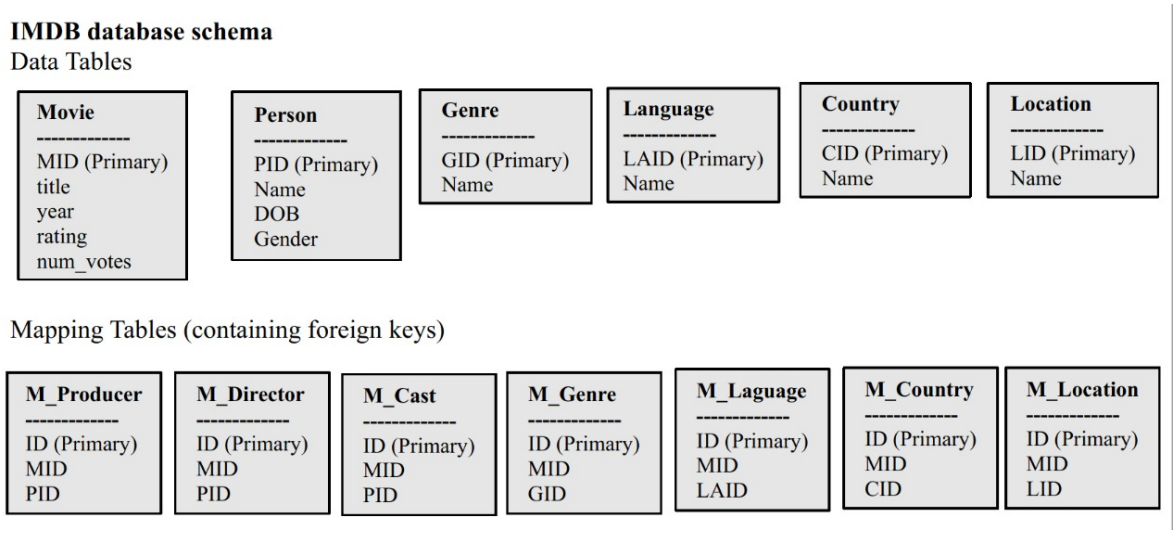
```
241   Siddharth Anand Kumar                Let's Enjoy     2004
242         Amma Rajasekhar                    Sathyam     2008
243           Oliver Paulus               Tandoori Love     2008
244             Raja Chanda                 Le Halua Le     2012
245        K.S. Prakash Rao           Raja Aur Rangeeli     1996

[246 rows x 3 columns]
```

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

In [52]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[52]:



In [55]:

```
output_2 = pd.read_sql_query('''
                             select p.name from PERSONS p
                             join M_Cast mc on p.pid = trim(mc.pid)
                             join Movie m on m.mid = mc.mid
                             where m.title=\'Anand\' and m.year = 1971
                             ''', con)
print(output_2)
```

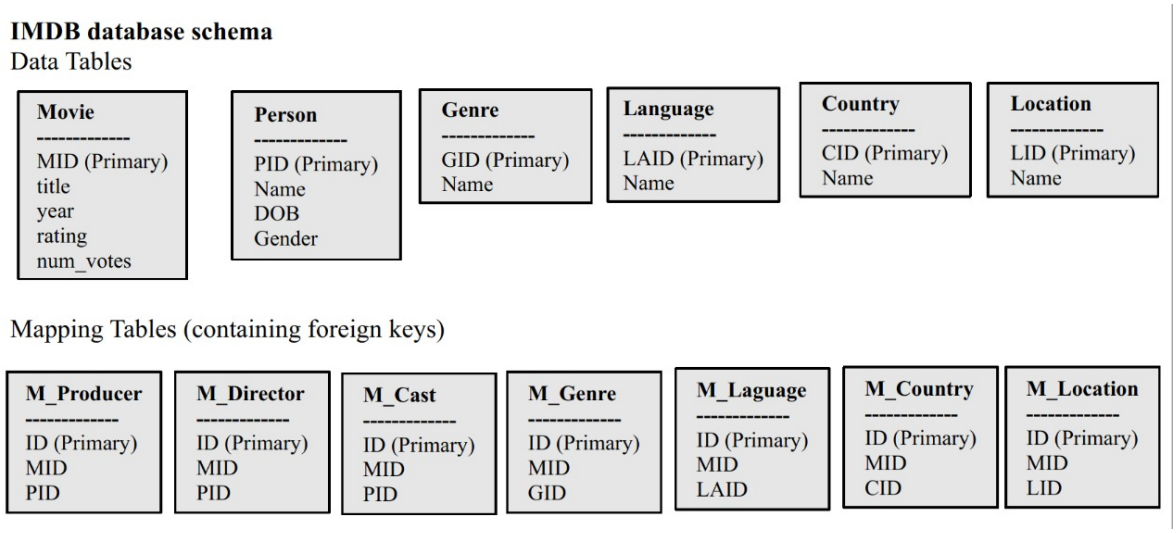
```
                Name
0    Amitabh Bachchan
1       Rajesh Khanna
2      Brahm Bhardwaj
3          Ramesh Deo
4           Seema Deo
5          Dev Kishan
6         Durga Khote
7       Lalita Kumari
8        Lalita Pawar
9        Atam Prakash
10      Sumita Sanyal
11     Asit Kumar Sen
12         Dara Singh
13      Johnny Walker
14          Moolchand
15       Gurnam Singh
16             Savita
```

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

In [56]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[56]:



In [13]:

```
output_3=pd.read_sql_query('''SELECT Name FROM Person WHERE TRIM(PID) IN (SELECT TRIM(PID)
INNER JOIN M_cast mc ON m.MID=mc.MID WHERE  PID IN (SELECT PID FROM MOVIE m INNER JOIN M_ca
WHERE TRIM(year)<'1970') and TRIM(year)>'1990')''',con)
print(output_3)
```

```
                  Name
0          Rishi Kapoor
1       Amitabh Bachchan
2                Asrani
3          Zohra Sehgal
4        Parikshat Sahni
5          Rakesh Sharma
6            Sanjay Dutt
7              Ric Young
8                  Yusuf
9         Suhasini Mulay
10           A.K. Hangal
11          Jeremy Child
12          Farida Jalal
13        Waheeda Rehman
14         Rajesh Khanna
15            Ramesh Deo
16             Seema Deo
17        Asit Kumar Sen
18        Brahm Bhardwaj
19          Lalita Pawar
20            Dara Singh
21         Johnny Walker
22             Moolchand
23            Saira Banu
24           Prem Chopra
25           Dina Pathak
26        Achala Sachdev
27             Shashikala
28    Mohandas K. Gandhi
29      Jawaharlal Nehru
..                   ...
289                 Uma
290              Ismail
291         Miss Firoza
292                Dube
293               Dolly
294             Shekhar
295              Poonam
296        Jamila Massey
297          K.R. Vijaya
298               Sethi
299         Suryakantham
300           Sunil Dutt
301         Subhash Ghai
302           Feroz Khan
303       Rajendra Kumar
304              Mehmood
305          Manoj Kumar
306            Dev Anand
307                Sachin
308           Prayag Raj
309       Randhir Kapoor
310     Naseeruddin Shah
```

```
311          Hema Malini
312        Shashi Kapoor
313        Shammi Kapoor
314          Vinod Mehra
315          Deven Verma
316       Master Bhagwan
317         Rishi Kapoor
318               Asrani

[319 rows x 1 columns]
```

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 [63]:

```
output_4=pd.read_sql_query('''SELECT Name,COUNT(MID) c FROM M_Director md  JOIN PERSON p ON
                              GROUP BY Name HAVING c>=10 ORDER BY c DESC''',con)
print(output_4)
```

```
                         Name   c
0                 David Dhawan  39
1                 David Dhawan  39
2                 Mahesh Bhatt  36
3                 Mahesh Bhatt  36
4              Ram Gopal Varma  30
5                 Priyadarshan  30
6              Ram Gopal Varma  30
7                 Vikram Bhatt  29
8                 Vikram Bhatt  29
9          Hrishikesh Mukherjee  27
10         Hrishikesh Mukherjee  27
11                  Yash Chopra  21
12                  Yash Chopra  21
13             Basu Chatterjee  19
14              Shakti Samanta  19
15             Basu Chatterjee  19
16              Shakti Samanta  19
17                Subhash Ghai  18
18                Subhash Ghai  18
19    Abbas Alibhai Burmawalla  17
20               Shyam Benegal  17
21    Abbas Alibhai Burmawalla  17
22            Rama Rao Tatineni  17
23               Shyam Benegal  17
24                      Gulzar  16
25              Manmohan Desai  16
26                Raj N. Sippy  16
27                      Gulzar  16
28              Manmohan Desai  16
29                Raj N. Sippy  16
..                          ...  ..
81                Sanjay Gupta  11
82             Govind Nihalani  11
83                 Ketan Mehta  11
84                  Mohit Suri  11
85               Nasir Hussain  11
86           Pramod Chakravorty  11
87                Sanjay Gupta  11
88                   Bimal Roy  10
89                Hansal Mehta  10
90               J. Om Prakash  10
91                  J.P. Dutta  10
92                Mehul Kumar  10
93                  N. Chandra  10
94                  Raj Kapoor  10
95               Sudhir Mishra  10
96             Tigmanshu Dhulia  10
97              Vishal Bhardwaj  10
98                    Bimal Roy  10
99                Hansal Mehta  10
100              J. Om Prakash  10
101                 J.P. Dutta  10
102                 K. Bapaiah  10
103         K. Muralimohana Rao  10
```

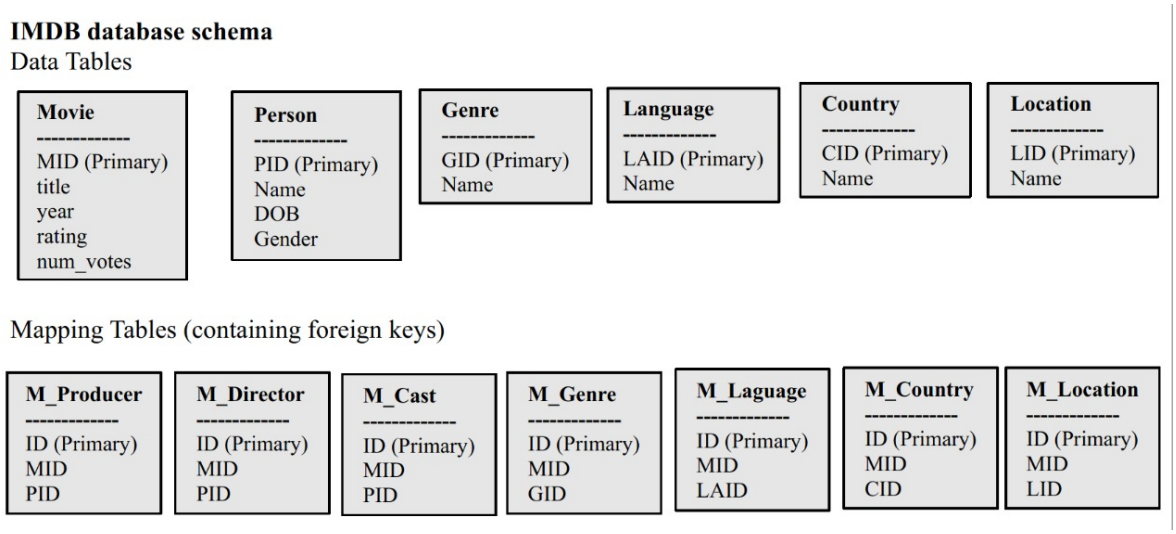
```
104              Mehul Kumar  10
105               N. Chandra  10
106           Pankaj Parashar  10
107                Raj Kapoor  10
108             Sudhir Mishra  10
109          Tigmanshu Dhulia  10
110           Vishal Bhardwaj  10

[111 rows x 2 columns]
```

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

In [64]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[64]:



In [61]:

```
output_5=pd.read_sql_query('''SELECT YEAR,COUNT(MID) as female_movies from Movie WHERE MID
                           (SELECT MID FROM M_Cast mc JOIN PERSONS p ON p.PID=TRIM(mc.PID)
                           (SELECT MID FROM M_Cast mc JOIN PERSONS p ON p.PID=TRIM(mc.PID)
                           AND MID NOT IN (SELECT MID FROM M_Cast mc jOIN PERSON p ON p.PI
                           GROUP BY YEAR  ORDER BY YEAR DESc ''',con)
output_5.shape
```

Out[61]:

```
(4, 2)
```

In [62]:

```
print(output_5)
```

```
    year  female_movies
0   2000              1
1   1999              1
2   1939              1
3   2018              1
```

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 [51]:

```
output_5b=pd.read_sql_query('''SELECT year,(SELECT  COUNT(MID)  from Movie WHERE MID IN
                  (SELECT MID FROM M_Cast mc JOIN PERSONS p ON mc.PID=p.PID WHERE MID IN
                  (SELECT MID FROM M_Cast mc JOIN PERSONS p ON mc.PID=p.PID WHERE Gender =
                  and MID NOT IN(SELECT  MID FROM M_Cast mc JOIN PERSONS p ON mc.PID=p.PID
                  GROUP BY year)*100/(COUNT(Movie.MID)*1.0) AS PERCENTAGE,COUNT(Movie.MID)
                  (SELECT  year from MOVIE  WHERE MID IN (SELECT MID FROM M_Cast mc JOIN P
                  (SELECT MID FROM M_Cast mc JOIN PERSONS p ON mc.PID=p.PID WHERE Gender =
                  and MID NOT IN(SELECT  MID FROM M_Cast mc JOIN PERSONS p ON mc.PID=p.PID
                  group by year ORDER BY year DESC''',con)
output_5b.shape
```

Out[51]:

```
(4, 3)
```

In [52]:

```
print(output_5b)
```

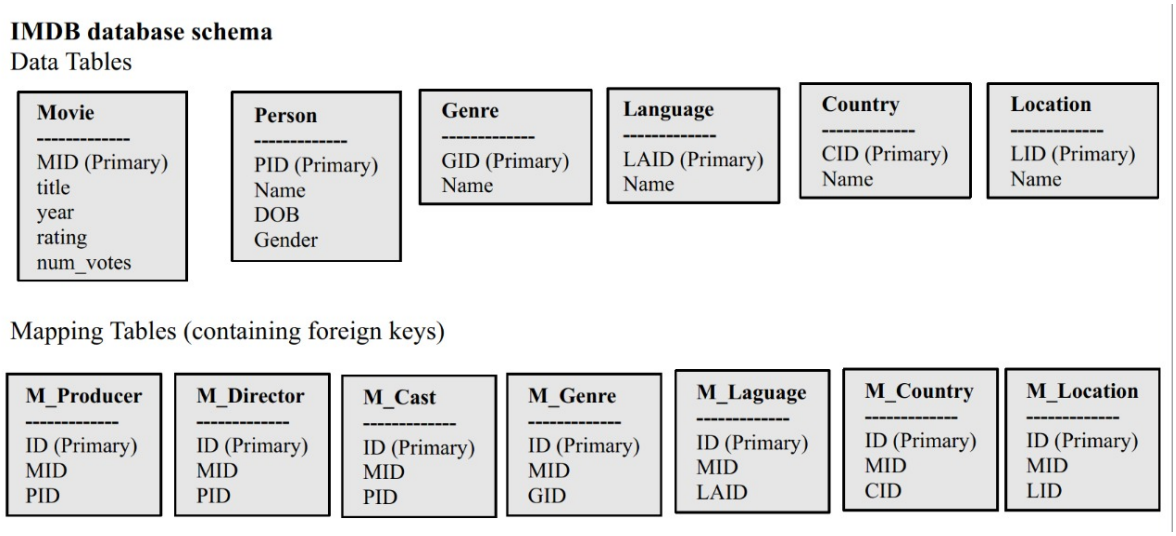
```
    year  PERCENTAGE  TOTAL_MOVIES
0   2000    1.562500            64
1   1999    1.515152            66
2   1939   50.000000             2
3   2018    9.090909            11
```

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 [43]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[43]:



In [48]:

```
output_6=pd.read_sql_query('''SELECT  m.title Movie_name,count(DISTINCT(mc.PID)) Cast_size
                              mc.MID=m.MID GROUP BY m.MID ORDER BY Cast_size DESC''',con  )
output_6.size
```

Out[48]:

```
6950
```

In [49]:

```
output_6.head()
```

Out[49]:

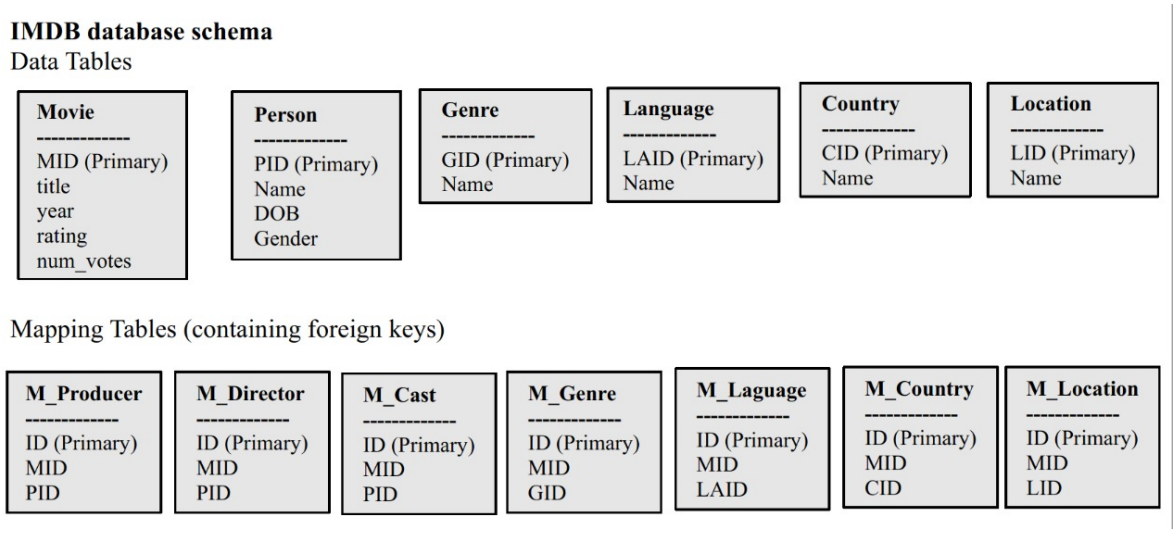
	Movie_name	Cast_size
0	Ocean's Eight	238
1	Apaharan	233
2	Gold	215
3	My Name Is Khan	213
4	Captain America: Civil War	191

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 [65]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[65]:



In [24]:

```
#https://stackoverflow.com/questions/51609285/sql-query-for-find-the-decade-with-the-larges
output_7 = pd.read_sql_query('''SELECT d.year Start, d.year+9 End, count(*) no_of_films FRO
                            (SELECT DISTINCT year from Movie) d JOIN Movie m ON m.year >= S
                            GROUP BY End ORDER BY no_of_films desc LIMIT 1''',con)
print(output_7)
```

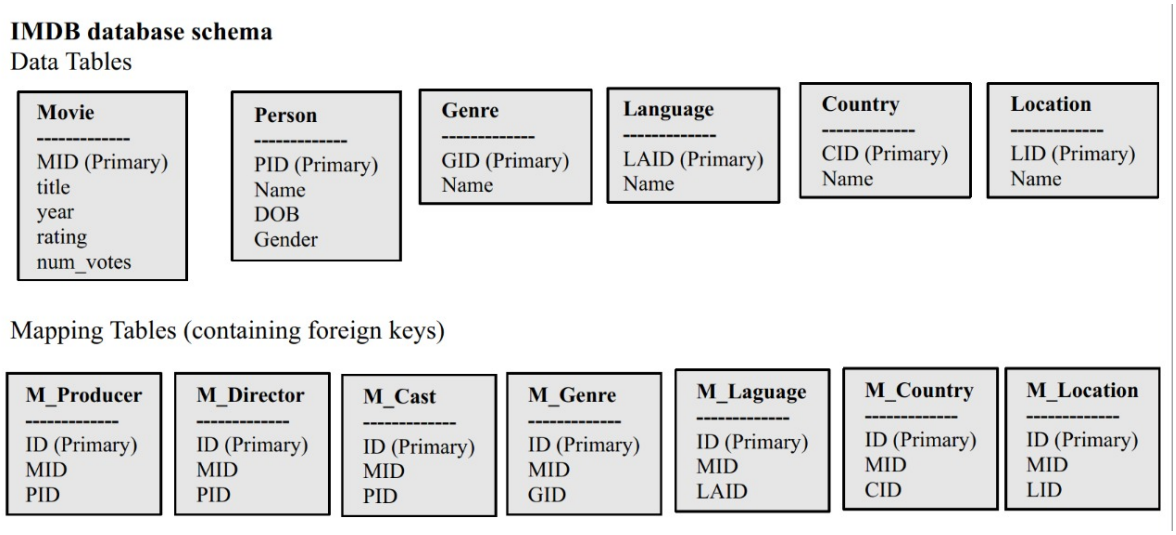
```
  Start   End  no_of_films
0  2009  2018         4560
```

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 [91]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[91]:



In [95]:

```
# https://www.coursehero.com/file/p7mfaba/15-Find-the-actors-who-were-never-unemployed-for-
output_8=pd.read_sql_query('''select Name as Actor from Person where PID not in (select dis
                              c1 natural join Movie as m1  where exists(select MID from M_
                             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)
output_8
```

Out[95]:

	Actor
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	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
20	Roshan Jayesh Patel
21	T'khai Phillips
22	Sachin Soni
23	Hridhay Somera
24	Ethaniel Jaden Moonsamy
25	Gareth Ryan Benjamin
26	Nirvayesh Chakravorty Thanendra
27	Adiyan Ahmed Choudhury
28	Amara Motala
29	Diyara Prakash
...	...

	Actor
38255	Sandip Ray
38256	S.V. Krishna Reddy
38257	R.K. Selvamani
38258	Amma Rajasekhar
38259	Rahat Kazmi
38260	Rohit Gupta
38261	Bela Negi
38262	Sanjay Talreja
38263	Rajatesh Nayyar
38264	Murali Nair
38265	Pryas Gupta
38266	Shivamani
38267	Oliver Paulus
38268	Vishal Inamdar
38269	Kumar Shahani
38270	Ka-Fai Wai
38271	Avtandil Varsimashvili
38272	G. Ram Prasad
38273	Raja Chanda
38274	Deepak Ramteke
38275	Srinivas Sunderrajan
38276	Kamika Verma
38277	Dhorairaj Bhagavan
38278	Nasir Shaikh
38279	Abbas
38280	Kannan
38281	Adrian Fulle
38282	Gulshan Kumar
38283	Iqbal
38284	Sushma Shiromani

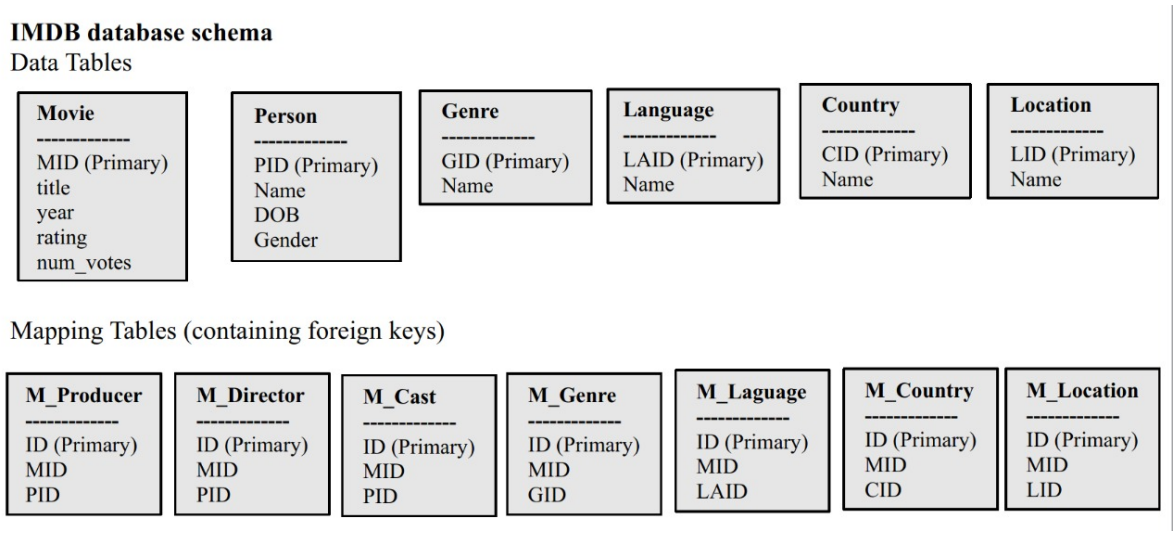
38285 rows × 1 columns

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

In [96]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[96]:



In [104]:

```
output_9 = pd.read_sql_query('''SELECT DISTINCT  Actor, Count(*) Movies_with_YashChopra
 FROM(SELECT DISTINCT p1.Name as Director, m1.title as Movie
 FROM Person p1 Inner Join M_Director md on TRIM(md.PID)=p1.PID
 Inner Join Movie m1 on TRIM(md.MID)=m1.MID and  p1.Name LIKE 'Yash%' Group By p1.Name, m1.
 Inner Join (SELECT DISTINCT p2.Name as Actor,m2.title as 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 Group By p2.Name, m2.title) t2 on t1.Movie=t2.M
 Group By t2.Actor Order By Movies_with_YashChopra DESC''',con)
output_9.shape
```

Out[104]:

```
(514, 2)
```

In [105]:

```
print(output_9)
```

```
                Actor  Movies_with_YashChopra
0          Jagdish Raj                      11
1      Manmohan Krishna                     10
2      Manmohan Krishna                     10
3             Iftekhar                       9
4           Madan Puri                       8
5          Vikas Anand                       8
6          Anupam Kher                       7
7        Shashi Kapoor                       7
8          Anupam Kher                       7
9        Shashi Kapoor                       7
10    Amitabh Bachchan                       6
11       Rakhee Gulzar                       5
12      Waheeda Rehman                       5
13      Achala Sachdev                       4
14         Deven Verma                       4
15         Hema Malini                       4
16         Neetu Singh                       4
17            Ravikant                       4
18        Rishi Kapoor                       4
19      Shah Rukh Khan                       4
20         Deven Verma                       4
21         Hema Malini                       4
22        Rishi Kapoor                       4
23         A.K. Hangal                       3
24         Anil Kapoor                       3
25         Annu Kapoor                       3
26       Leela Chitnis                       3
27        Mohan Sherry                       3
28     Parikshat Sahni                       3
29         Prem Chopra                       3
..                 ...                     ...
484         Uttam Sodi                       1
485       Varun Thajur                       1
486       Varun Thakur                       1
487      Varun Vardhan                       1
488        Vic Waghorn                       1
489        Vicky Ahuja                       1
490    Vicky K. Foster                       1
491       Vinay Sharma                       1
492      Vinita Sharma                       1
493         Vinod Negi                       1
494         Vinod Raut                       1
495    Virat Raj Gupta                       1
496  Vishal Om Prakash                       1
497     Yashodra Katju                       1
498         Yasin Khan                       1
499       Zohra Sehgal                       1
500         Aamir Khan                       1
501       Arjun Sablok                       1
502         Aziz Mirza                       1
503          Dev Anand                       1
504         Feroz Khan                       1
505            Mehmood                       1
506       Puneet Issar                       1
507        Raman Kumar                       1
508       Rani Mukerji                       1
```

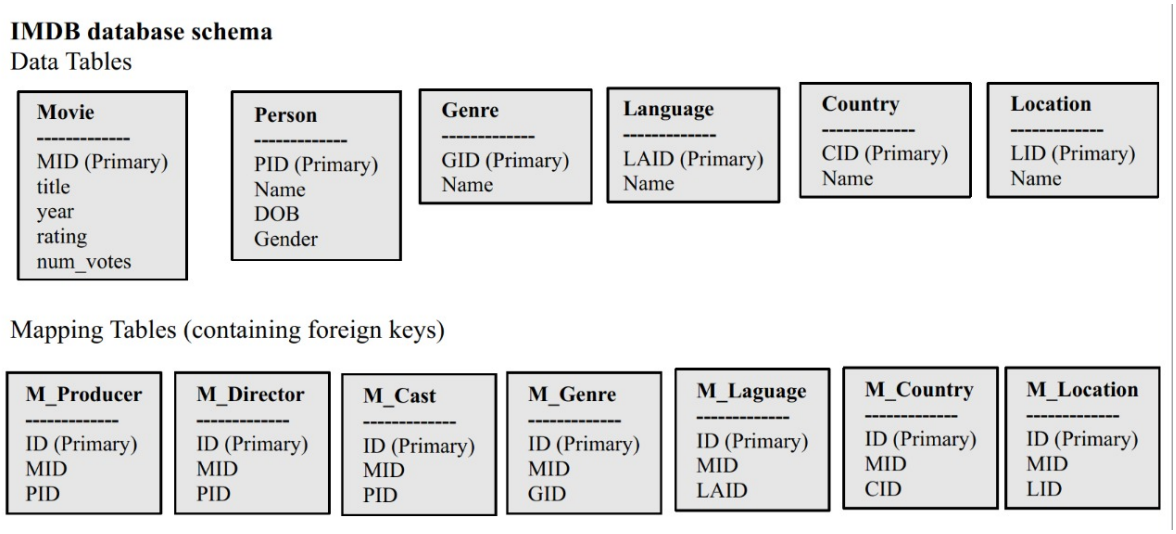
```
509       Romesh Sharma                1
510              Sachin                1
511          Sajid Khan                1
512          Sunny Deol                1
513         Tinnu Verma                1

[514 rows x 2 columns]
```

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

In [106]:

```
from IPython.display import Image
Image("db_schema.jpeg",width=1200, height=300)
```

Out[106]:



In [118]:

```
#https://www.coursehero.com/file/p7mfaba/15-Find-the-actors-who-were-never-unemployed-for-m
output_10= pd.read_sql_query('''SELECT DISTINCT TRIM(name) Name
 FROM Person p INNER JOIN M_Cast mc on p.PID = TRIM(mc.PID) INNER JOIN Movie m ON m.MID = m
 and m.title in (SELECT DISTINCT title FROM Person p3 INNER JOIN M_Cast mc3 on p3.PID = TRI
 INNER JOIN Movie m3 ON m3.MID = mc3.MID AND p3.Name IN (SELECT DISTINCT Name FROM Person p
 INNER JOIN Movie m2 ON m2.MID = mc2.MID AND TRIM(p2.Name)!='Shah Rukh Khan' AND m2.title I
 (SELECT DISTINCT title FROM Person p3 INNER JOIN M_Cast mc3 ON p3.PID = TRIM(mc3.PID) AND
 INNER JOIN Movie m3 ON m3.MID = mc3.MID))) ORDER BY Name''',con)

output_10.shape
```

Out[118]:

```
(16165, 1)
```

In [119]:

```
print(output_10)
```

```
                          Name
0      'Musafir' Radio Performing
1                   A'Ali de Sousa
2                  A. Abdul Hameed
3                        A. Darpan
4                        A. Gabibi
5                          A. Khan
6                       A. Kukereja
7                        A. Lakshmi
8                       A. Narsimha
9                      A. Prabhakar
10                    A. Ravi Verma
11                    A. Shalomayev
12                        A. Sharma
13                      A.A. Deepak
14                        A.A. Khan
15                      A.C. Murali
16                      A.C. Sarkar
17                       A.D. Singh
18                      A.G. Poddar
19                       A.H. Shore
20                      A.K. Hangal
21                       A.K. Raina
22                        A.K. Rana
23                      A.R. Basha
24                  A.R. Manikandan
25                        A.R. Rama
26                        A.R. Rana
27                           A.R.S.
28                      A.S. Duggal
29                     A.V. Iyenger
...                             ...
16135                   Zia Ahmed
16136                 Zia Ur-Rehman
16137                Zibrail Ansari
16138                         Zippy
16139              Zivile Matikiene
16140                    Zoa Morani
16141                  Zoe Woodruff
16142                          Zoeb
16143                   Zoha Tapia
16144                  Zohra Sehgal
16145       Zongra Tulku Rinpoche
16146                  Zoran Korach
16147                Zorawar Shukla
16148                   Zoya Afroz
16149                 Zoya Merchant
16150                     Zoya Shah
16151                       Zubaida
16152                   Zubair Khan
16153                        Zubeda
16154                   Zubeda Khan
16155                   Zubeen Garg
16156                       Zubeida
16157                 Zubin Jauhari
16158                         Zufin
16159                   Zul Vellani
```

```
16160                Zulfi Sayed
16161          Zulkhumor Muminova
16162            Zurab Kapianidze
16163                  Zuri Echea
16164               Zuzanna Zajac

[16165 rows x 1 columns]
```

In [ ]: