mayankgupta9968@gmail.com_22

January 6, 2020

```python
[1]: # Import common packages and create database connection
     import pandas as pd
     import sqlite3 as db

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

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.

```python
[2]: %%time
     # List all the distinct directors who directed a 'Comedy' movie in a leap year.
     # citation https://stackoverflow.com/questions/6534788/check-for-leap-year
     # https://www.mathsisfun.com/leap-years.html
     result = pd.read_sql_query(
         '''
         SELECT DISTINCT trim(P.NAME) as director, M.title as movie, M.year, G.Name
         FROM Movie M
                 JOIN M_Director MD ON M.MID = MD.MID
                 JOIN Person P on trim(MD.PID) = trim(P.PID)
                 JOIN M_Genre MG on M.MID = MG.MID
                 JOIN Genre G on MG.GID = G.GID
         WHERE G.Name LIKE '%Comedy%'
           AND (((M.year % 4 = 0) AND (M.year % 100 != 0)) OR (M.year % 400 = 0))
         GROUP BY director
         ORDER BY director
         '''
         , conn);

     result
```

```
CPU times: user 26.5 s, sys: 135 ms, total: 26.6 s
Wall time: 31.2 s
```

```
[2]:              director                       movie  year  \
     0          A. Bhimsingh          Sadhu Aur Shaitaan  1968
     1        Abbas Tyrewala     Jaane Tu... Ya Jaane Na  2008
     2         Abhishek Jain            Kevi Rite Jaish   2012
```

3	Abhishek Sharma	Tere Bin Laden: Dead Or Alive	2016
4	Aditya Chopra	Rab Ne Bana Di Jodi	2008
5	Aditya Datt	Will You Marry Me	2012
6	Ajai Sinha	3 Bachelors	2012
7	Akashdeep	Santa Banta Pvt Ltd	2016
8	Amit Mitra	Jagte Raho	1956
9	Amma Rajasekhar	Sathyam	2008
10	Anand Balraj	Daal Mein Kuch Kaala Hai	2012
11	Anees Bazmee	Singh Is Kinng	2008
12	Anil Devgan	Raju Chacha	2000
13	Anil Senior	Dil Kabaddi	2008
14	Anurag Basu	Barfi!	2012
15	Anurag Kashyap	Gangs of Wasseypur	2012
16	Arbaaz Khan	Dabangg 2	2012
17	Ashish R. Mohan	Khiladi 786	2012
18	Ashwani Dhir	Son of Sardaar	2012
19	Ashwini Chaudhary	Jodi Breakers	2012
20	Aspi Irani	Garam Masala	1972
21	Aziz Mirza	Kismat Konnection	2008
22	Basu Chatterjee	Chhoti Si Baat	1976
23	Bhagyaraj	Mr. Bechara	1996
24	Bhappi Sonie	Brahmachari	1968
25	Bimal Roy	Parakh	1960
26	Bobby Kolli	Sardaar Gabbar Singh	2016
27	Brij	Victoria No. 203	1972
28	Chandrakant Kulkarni	Meerabai Not Out	2008
29	Chandrakant Singh	Rama Rama Kya Hai Dramaaa	2008
..	...	...	...
162	Shashi Ranjan	Dhoom Dadakka	2008
163	Shirish Kunder	Joker	2012
164	Shona Urvashi	Saas Bahu Aur Sensex	2008
165	Shoojit Sircar	Vicky Donor	2012
166	Shubhashish Bhutiani	Hotel Salvation	2016
167	Shyam Benegal	Welcome to Sajjanpur	2008
168	Shyam Ramsay	Purana Mandir	1984
169	Siddharth Anand	Bachna Ae Haseeno	2008
170	Siddharth Anand Kumar	Let's Enjoy	2004
171	Sohail Khan	Freaky Ali	2016
172	Sourabh Shrivastava	Oh, My God!!	2008
173	Srinivas Bhashyam	Paisa Vasool	2004
174	Subhash Ghai	Yuvvraaj	2008
175	Suhas Kadav	Motu Patlu: King of Kings	2016
176	Sunil K. Reddy	Thikka	2016
177	Sunil Subramani	Fuddu	2016
178	T.L.V. Prasad	Tauba Tauba	2004
179	Tarun Majumdar	Balika Badhu	1976
180	Tarun Mansukhani	Dostana	2008

181	Umesh Ghadge	Kyaa Kool Hain Hum 3	2016
182	Umesh Shukla	OMG: Oh My God!	2012
183	Vaibhav Misra	LoveShhuda	2016
184	Vickrant Mahajan	Challo Driver	2012
185	Vijay	Tutak Tutak Tutiya	2016
186	Vijay Anand	Ram Balram	1980
187	Vijay Sadanah	Ek Ladka Ek Ladki	1992
188	Vijaya Mehta	Pestonjee	1988
189	Vimal Kumar	Chhote Sarkar	1996
190	Vivek Sharma	Bhoothnath	2008
191	Yograj Bhat	Ranga S.S.L.C	2004

Name

0	Comedy, Thriller
1	Comedy, Drama, Romance
2	Comedy, Family
3	Comedy
4	Comedy, Drama, Music
5	Comedy
6	Comedy, Romance
7	Action, Comedy
8	Comedy, Drama
9	Action, Comedy
10	Comedy
11	Action, Comedy, Crime
12	Action, Comedy, Drama
13	Comedy, Drama
14	Comedy, Drama, Romance
15	Action, Comedy, Crime
16	Action, Comedy
17	Action, Comedy
18	Action, Comedy
19	Comedy, Musical, Romance
20	Comedy, Musical, Action
21	Comedy, Drama, Romance
22	Comedy, Romance
23	Comedy
24	Comedy, Drama, Musical
25	Comedy, Drama, Romance
26	Action, Comedy
27	Comedy, Drama, Musical
28	Comedy, Drama, Romance
29	Comedy, Drama, Romance
..	...
162	Comedy, Crime, Drama
163	Comedy, Family, Sci-Fi
164	Comedy, Drama

```
165            Comedy, Romance
166             Comedy, Drama
167             Comedy, Drama
168        Comedy, Drama, Horror
169       Comedy, Drama, Romance
170                    Comedy
171             Comedy, Sport
172             Comedy, Drama
173                    Comedy
174       Comedy, Drama, Romance
175                    Comedy
176            Action, Comedy
177             Comedy, Drama
178             Comedy, Drama
179       Comedy, Drama, Romance
180       Comedy, Drama, Romance
181                    Comedy
182      Comedy, Drama, Fantasy
183            Comedy, Romance
184                    Comedy
185            Comedy, Horror
186       Action, Comedy, Drama
187            Comedy, Romance
188             Comedy, Drama
189    Comedy, Romance, Thriller
190       Comedy, Drama, Family
191       Action, Comedy, Drama

[192 rows x 4 columns]
```

[3]:
```
%%time
# List all the directors who directed a 'Comedy' movie in a leap year. A
 →director can direct multiple movies in leap year.
# citation https://stackoverflow.com/questions/6534788/check-for-leap-year
# https://www.mathsisfun.com/leap-years.html
result = pd.read_sql_query(
    '''
    SELECT DISTINCT trim(P.NAME) as director, M.title as movie, M.year, G.Name
    FROM Movie M
            JOIN M_Director MD ON M.MID = MD.MID
            JOIN Person P on trim(MD.PID) = trim(P.PID)
            JOIN M_Genre MG on M.MID = MG.MID
            JOIN Genre G on MG.GID = G.GID
    WHERE G.Name LIKE '%Comedy%'
      AND (((M.year % 4 = 0) AND (M.year % 100 != 0)) OR (M.year % 400 = 0))
    ORDER BY director
    '''
```

```
      , conn);

result
```

CPU times: user 26.6 s, sys: 131 ms, total: 26.7 s
Wall time: 31.4 s

[3]:

	director	movie	year	\
0	A. Bhimsingh	Sadhu Aur Shaitaan	1968	
1	A. Bhimsingh	Joroo Ka Ghulam	1972	
2	Abbas Tyrewala	Jaane Tu... Ya Jaane Na	2008	
3	Abhishek Jain	Kevi Rite Jaish	2012	
4	Abhishek Sharma	Tere Bin Laden: Dead Or Alive	2016	
5	Aditya Chopra	Rab Ne Bana Di Jodi	2008	
6	Aditya Chopra	Befikre	2016	
7	Aditya Datt	Will You Marry Me	2012	
8	Ajai Sinha	3 Bachelors	2012	
9	Akashdeep	Santa Banta Pvt Ltd	2016	
10	Amit Mitra	Jagte Raho	1956	
11	Amma Rajasekhar	Sathyam	2008	
12	Anand Balraj	Daal Mein Kuch Kaala Hai	2012	
13	Anees Bazmee	Singh Is Kinng	2008	
14	Anil Devgan	Raju Chacha	2000	
15	Anil Senior	Dil Kabaddi	2008	
16	Anurag Basu	Barfi!	2012	
17	Anurag Kashyap	Gangs of Wasseypur	2012	
18	Arbaaz Khan	Dabangg 2	2012	
19	Ashish R. Mohan	Khiladi 786	2012	
20	Ashwani Dhir	Son of Sardaar	2012	
21	Ashwani Dhir	One Two Three	2008	
22	Ashwini Chaudhary	Jodi Breakers	2012	
23	Aspi Irani	Garam Masala	1972	
24	Aziz Mirza	Kismat Konnection	2008	
25	Aziz Mirza	Raju Ban Gaya Gentleman	1992	
26	Aziz Mirza	Phir Bhi Dil Hai Hindustani	2000	
27	Basu Chatterjee	Chhoti Si Baat	1976	
28	Basu Chatterjee	Apne Paraye	1980	
29	Basu Chatterjee	Man Pasand	1980	
..	...	...	...	
202	Shona Urvashi	Saas Bahu Aur Sensex	2008	
203	Shoojit Sircar	Vicky Donor	2012	
204	Shubhashish Bhutiani	Hotel Salvation	2016	
205	Shyam Benegal	Welcome to Sajjanpur	2008	
206	Shyam Ramsay	Purana Mandir	1984	
207	Siddharth Anand	Bachna Ae Haseeno	2008	
208	Siddharth Anand Kumar	Let's Enjoy	2004	
209	Sohail Khan	Freaky Ali	2016	

210	Sourabh Shrivastava	Oh, My God!!	2008
211	Srinivas Bhashyam	Paisa Vasool	2004
212	Subhash Ghai	Yuvvraaj	2008
213	Suhas Kadav	Motu Patlu: King of Kings	2016
214	Sunil K. Reddy	Thikka	2016
215	Sunil Subramani	Fuddu	2016
216	T.L.V. Prasad	Tauba Tauba	2004
217	Tarun Majumdar	Balika Badhu	1976
218	Tarun Majumdar	Dadar Kirti	1980
219	Tarun Mansukhani	Dostana	2008
220	Umesh Ghadge	Kyaa Kool Hain Hum 3	2016
221	Umesh Shukla	OMG: Oh My God!	2012
222	Vaibhav Misra	LoveShhuda	2016
223	Vickrant Mahajan	Challo Driver	2012
224	Vijay	Tutak Tutak Tutiya	2016
225	Vijay Anand	Ram Balram	1980
226	Vijay Sadanah	Ek Ladka Ek Ladki	1992
227	Vijaya Mehta	Pestonjee	1988
228	Vimal Kumar	Chhote Sarkar	1996
229	Vimal Kumar	Suno Sasurjee	2004
230	Vivek Sharma	Bhoothnath	2008
231	Yograj Bhat	Ranga S.S.L.C	2004

Name

0	Comedy, Thriller
1	Comedy, Family
2	Comedy, Drama, Romance
3	Comedy, Family
4	Comedy
5	Comedy, Drama, Music
6	Comedy, Drama, Romance
7	Comedy
8	Comedy, Romance
9	Action, Comedy
10	Comedy, Drama
11	Action, Comedy
12	Comedy
13	Action, Comedy, Crime
14	Action, Comedy, Drama
15	Comedy, Drama
16	Comedy, Drama, Romance
17	Action, Comedy, Crime
18	Action, Comedy
19	Action, Comedy
20	Action, Comedy
21	Comedy, Crime
22	Comedy, Musical, Romance

```
23      Comedy, Musical, Action
24       Comedy, Drama, Romance
25     Musical, Comedy, Romance
26       Comedy, Drama, Musical
27              Comedy, Romance
28        Drama, Family, Comedy
29                Comedy, Drama
..                          ...
202               Comedy, Drama
203             Comedy, Romance
204               Comedy, Drama
205               Comedy, Drama
206        Comedy, Drama, Horror
207       Comedy, Drama, Romance
208                      Comedy
209               Comedy, Sport
210               Comedy, Drama
211                      Comedy
212       Comedy, Drama, Romance
213                      Comedy
214              Action, Comedy
215               Comedy, Drama
216               Comedy, Drama
217       Comedy, Drama, Romance
218               Comedy, Drama
219       Comedy, Drama, Romance
220                      Comedy
221      Comedy, Drama, Fantasy
222             Comedy, Romance
223                      Comedy
224              Comedy, Horror
225       Action, Comedy, Drama
226             Comedy, Romance
227               Comedy, Drama
228    Comedy, Romance, Thriller
229     Action, Comedy, Musical
230        Comedy, Drama, Family
231       Action, Comedy, Drama

[232 rows x 4 columns]
```

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

```
[3]: %%time
result = pd.read_sql_query(
    '''
    SELECT p.Name FROM Movie m
    JOIN M_Cast mc ON m.MID=mc.MID
```

```
    JOIN Person p ON trim(mc.PID)=trim(p.PID)
    WHERE m.title='Anand' AND m.year=1971
    '''
    , conn)

result
```

```
CPU times: user 212 ms, sys: 8.56 ms, total: 221 ms
Wall time: 222 ms
```

[3]:
```
                 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
```

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

[5]:
```
%%time
result = pd.read_sql_query(
        '''
        SELECT DISTINCT trim(p.PID) as pid, p.Name
        FROM Movie m
                JOIN M_Cast mc ON m.MID = mc.MID
                JOIN Person p ON trim(mc.PID) = trim(p.PID)
        WHERE m.year > 1990
          AND trim(p.PID) IN (SELECT DISTINCT trim(p.PID) as pid
                                FROM Movie m
                                        JOIN M_Cast mc ON m.MID = mc.MID
                                        JOIN Person p ON trim(mc.PID) = trim(p.
 ↪PID)
                                WHERE m.year < 1970)
        GROUP BY trim(p.PID)
```

```
        ''', conn)
result
```

CPU times: user 1min 33s, sys: 616 ms, total: 1min 33s
Wall time: 1min 35s

[5]:
	pid	Name
0	nm0000821	Amitabh Bachchan
1	nm0003987	Mohandas K. Gandhi
2	nm0004292	Raj Kapoor
3	nm0004334	Rekha
4	nm0004429	Dharmendra
5	nm0004432	Prithviraj Kapoor
6	nm0004433	Shammi Kapoor
7	nm0004434	Shashi Kapoor
8	nm0004435	Rajesh Khanna
9	nm0004564	Hema Malini
10	nm0004569	Sanjay Dutt
11	nm0004570	Sunil Dutt
12	nm0006348	Rajendra Kumar
13	nm0006369	Ashok Kumar
14	nm0006370	Manoj Kumar
15	nm0007131	Subhash Ghai
16	nm0007147	Dev Anand
17	nm0007807	Abbas
18	nm0008248	Abdul
19	nm0013026	Jalal Agha
20	nm0014973	Ajay
21	nm0015000	Ajit
22	nm0019425	Haidar Ali
23	nm0019454	Mohammed Ali
24	nm0023935	Amar
25	nm0038019	Arun
26	nm0044284	Aziz
27	nm0045119	Aruna Irani
28	nm0045124	Baby Farida
29	nm0045136	Sarika
..	...	...
303	nm1486554	Miss Firoza
304	nm1531547	Gopal Dutt
305	nm1532631	Pardesi
306	nm1532843	Ghanshyam Rohera
307	nm1533531	Master Amar
308	nm1533565	Brij Bhushan
309	nm1538793	Baby Deepali
310	nm1548528	Chhaya
311	nm1566068	Satyendra Kapoor

```
312   nm1572166                   Jambura
313   nm1575949            Shamsher Singh
314   nm1595330             Rakesh Sharma
315   nm1630755                     Thapa
316   nm1634809                    Naseem
317   nm1651577             Ramesh Sharma
318   nm1654650          Pradeep Chaudhry
319   nm1664482                     Amrit
320   nm1739846                     Ameer
321   nm1747430              Rohit Chopra
322   nm1751480                    Samina
323   nm1755975                  Raj Dutt
324   nm1789260                    Merlyn
325   nm1927212                D.K. Sapru
326   nm2136994           Babbanlal Yadav
327   nm2147526                    Asrani
328   nm2983292               Mohamad Ali
329   nm4563339                      Dube
330   nm7390393            Aachi Manorama
331   nm9036653             Surendra Rahi
332   nm9045541                   Shirley

[333 rows x 2 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.

```python
[4]: %%time
result = pd.read_sql_query(
        '''
        SELECT p.Name, count(md.ID) movieCount FROM M_Director md JOIN Person p␣
    ↪ON md.PID=p.PID
        GROUP BY md.PID HAVING movieCount >= 10 ORDER BY movieCount DESC
        ''', conn)

result
```

```
CPU times: user 69.3 ms, sys: 2.19 ms, total: 71.4 ms
Wall time: 73.3 ms
```

```
[4]:                   Name  movieCount
     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
8	Subhash Ghai	36
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
20	Rajkumar Santoshi	28
21	Rakesh Roshan	26
22	Dev Anand	26
23	Vijay Anand	26
24	Harry Baweja	26
25	Anurag Kashyap	26
26	Ananth Narayan Mahadevan	26
27	K. Raghavendra Rao	26
28	Anees Bazmee	24
29	Guddu Dhanoa	24
..	...	...
126	Aziz Mirza	10
127	Saeed Akhtar Mirza	10
128	Mohan Babu	10
129	Shankar Mukherjee	10
130	Subodh Mukherji	10
131	K. Muralimohana Rao	10
132	Shashilal K. Nair	10
133	Amol Palekar	10
134	Pankaj Parashar	10
135	Bharat Rangachary	10
136	Ramanand Sagar	10
137	Mohan Segal	10
138	Bhappi Sonie	10
139	Lekh Tandon	10
140	Rakeysh Omprakash Mehra	10
141	Remo D'Souza	10
142	Sanjay Gadhvi	10
143	Prawaal Raman	10
144	Sujoy Ghosh	10
145	Pradeep Sarkar	10
146	Sajid Khan	10

```
147              Krishna D.K.         10
148           Subhash Kapoor         10
149           S.S. Rajamouli         10
150           Nishikant Kamat        10
151           Siddharth Anand        10
152          Dibakar Banerjee        10
153           Shoojit Sircar         10
154                 R. Balki         10
155            Neeraj Pandey         10

[156 rows x 2 columns]
```

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

```python
[2]: %%time
result = pd.read_sql_query(
          '''
          SELECT count(m.year) movie_count, m.year as movie_year
          FROM Movie m where m.MID not in
          (SELECT mc.MID FROM Person p JOIN M_Cast mc ON trim(p.PID)=trim(mc.
      ↪PID) WHERE p.Gender='Male')
          GROUP BY movie_year
          ORDER BY movie_count DESC
          ''', conn)
result
```

```
CPU times: user 4min 31s, sys: 1.39 s, total: 4min 32s
Wall time: 4min 37s
```

```
[2]:    movie_count movie_year
    0             1       1939
    1             1       1999
    2             1       2000
    3             1       2009
    4             1       2012
    5             1       2018
    6             1     I 2018
```

5b.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.

```python
[6]: %%time
result = pd.read_sql_query(
          '''
          SELECT y.allMov as 'movie_count', x.year as movie_year, ((x.
      ↪Movies_Cnt*100.0)/y.allMov) as Percent FROM
          (SELECT count(*) Movies_Cnt , m.year
          FROM Movie m where m.MID not in
```

```
              (SELECT mc.MID FROM Person p JOIN M_Cast mc ON trim(p.PID) =␣
    ↪trim(mc.PID) WHERE p.Gender='Male')
              GROUP BY m.year) x INNER JOIN
              (SELECT count(*) allMov, m.year
              FROM Movie m
              GROUP BY m.year) y on x.year=y.year
              ''', conn)
result
```

```
CPU times: user 4min 38s, sys: 1.37 s, total: 4min 39s
Wall time: 4min 50s
```

[6]:
	movie_count	movie_year	Percent
0	2	1939	50.000000
1	66	1999	1.515152
2	64	2000	1.562500
3	100	2009	1.000000
4	109	2012	0.917431
5	93	2018	1.075269
6	10	I 2018	10.000000

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.

[7]:
```
%%time
result = pd.read_sql_query(
            '''
            SELECT count(DISTINCT mc.PId) as cast_count, m.title FROM Movie m
            JOIN M_Cast mc ON m.MID=mc.MID
            JOIN Person p ON trim(mc.PID)=trim(p.PID)
            GROUP BY m.MID
            ORDER BY cast_count DESC limit 1
            ''', conn)
result
```

```
CPU times: user 9min 47s, sys: 2.57 s, total: 9min 49s
Wall time: 9min 58s
```

[7]:
	cast_count	title
0	238	Ocean's Eight

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.

```
[10]: %%time
      # citation https://stackoverflow.com/questions/25955049/
       →sql-how-to-sum-up-count-for-many-decades?rq=1
      # result = pd.read_sql_query(
      #             '''
      #             SELECT (ROUND(m.year / 10) * 10) AS Decade, COUNT(1) AS␣
       →total_movies
      #             FROM Movie m
      #             GROUP BY ROUND(m.year/ 10)
      #             ORDER BY total_movies DESC LIMIT 1
      #             ''', conn)
      # result

      result = pd.read_sql_query('''
              SELECT d_year.year AS start, d_year.year+9 AS end, count(1) AS␣
       →total_movies FROM
              (SELECT DISTINCT(year) FROM Movie) d_year
              JOIN Movie m WHERE m.year>=start AND m.year<=end
              GROUP BY end
              ORDER BY total_movies DESC
              LIMIT 1
              ''', conn)
      result
```

```
CPU times: user 47.7 ms, sys: 34.3 ms, total: 81.9 ms
Wall time: 99 ms
```

```
[10]:    start   end   total_movies
      0   2008   2017          1128
```

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).

```
[9]: %%time
     # citation https://stackoverflow.com/questions/57733454/
      →to-find-actors-who-were-never-unemployed-for-more-than-3-years-in-a-stretch
     # Here I am using window function (LEAD) that provides comparing current row␣
      →with next row
     result = pd.read_sql_query(
                 '''
                 SELECT *, (next_year - year) AS gap FROM (SELECT *
                 , LEAD(year, 1, 0) OVER (PARTITION BY Name ORDER BY year ASC) AS␣
      →next_year
                 FROM (SELECT p.Name, m.title, m.year FROM Movie m
                 JOIN M_Cast mc ON m.MID=mc.MID
                 JOIN Person p ON trim(mc.PID)=trim(p.PID)))
```

```
          WHERE gap <=3 and gap >=0
          GROUP BY Name
          ORDER BY Name ASC
          ''', conn)
result
```

CPU times: user 12min 30s, sys: 2.87 s, total: 12min 33s
Wall time: 13min 13s

[9]:   Name   title   year \
    0   'Ganja' Karuppu   Sandai Kozhi   2005
    1   A. Abdul Hameed   Prem Nagar   1974
    2   A.A. Premawathie   Water   I 2005
    3   A.K. Agnihotri   Qatl   1986
    4   A.K. Hangal   Teesri Kasam   1966
    5   A.R. Manikandan   Teen Patti   2010
    6   A.R. Murugadoss   7 Aum Arivu   2011
    7   A.R. Rama   Hello Hum Lallann Bol Rahe Hain   2010
    8   A.V.S. Subramanyam   Ghatotkachudu   1995
    9   Aachal Munjal   We Are Family   2010
    10   Aachi Manorama   Vazhve Mayam   1982
    11   Aadhi   Kochadaiiyaan   2014
    12   Aadil   Chamku   2008
    13   Aadil Chahal   Nautanki Saala!   2013
    14   Aadil Sharma   Dear Zindagi   2016
    15   Aahana Kumra   Sona Spa   2013
    16   Aahuthi Prasad   Ninne Pelladatha   1996
    17   Aakash   Karam   2005
    18   Aakash Dabhade   3 Idiots   2009
    19   Aakash Dahiya   Dev.D   2009
    20   Aakash Maheriya   Patang   2011
    21   Aakash Pandey   Shabd   2005
    22   Aalaap Majgavkar   Rangoon   II 2017
    23   Aalok Pradhan   Rishtey   2002
    24   Aamir Ahmed   One Night Stand   I 2016
    25   Aamir Ali Malik   Yeh Kya Ho Raha Hai?   2002
    26   Aamir Bashir   Armaan   2003
    27   Aamir Khan   Yaadon Ki Baaraat   1973
    28   Aamir Yaseen   Dishoom   2016
    29   Aamna Sharif   Aao Wish Karein   2009
    ...   .
    7953   Soni Razdan   36 Chowringhee Lane   1981
    7954   Subhash Ghai   Khal Nayak   1993
    7955   Sudeep   Chandu   2002
    7956   Sudhir Mishra   Sazaye Maut   1981
    7957   Suhasini   Waiting   III 2015
    7958   Sujoy Ghosh   Bluffmaster!   2005
```

15

```
7959          Sunil Dutt                    Mother India    1957
7960           Sunil Pal                 Phir Hera Pheri    2006
7961          Sunny Deol                         Betaab    1983
7962        Suparn Verma                    Go Goa Gone     2013
7963     Suzad Iqbal Khan                       Monsoon   I 2015
7964        Tahir Hussain                 Dil Deke Dekho    1959
7965    Tigmanshu Dhulia            Gangs of Wasseypur     2012
7966         Tinnu Anand                       Nayakan     1987
7967         Tinnu Verma                         Vansh     1992
7968        Umesh Ghadge      Raju Ban Gaya Gentleman     1992
7969        Umesh Shukla                   Yaar Gaddar     1994
7970      V. Ravichandran                      Maanikya    2014
7971          Vasan Bala                         Aamir     2008
7972    Vidhu Vinod Chopra       Jaane Bhi Do Yaaro      1983
7973         Vijay Anand                   Kala Bazar      1960
7974          Vijay Raaz                Bhopal Express     1999
7975        Vijay Sharma                        Guddi      1971
7976        Vijaya Mehta                      Kalyug       1981
7977          Vikas Bahl          Hasee Toh Phasee        2014
7978        Vikram Bhatt               Khamoshiyan        2015
7979         Vinod Mehra                 Lal Patthar       1971
7980          Yana Gupta                          Dum      2003
7981         Yash Chopra                   Veer-Zaara      2004
7982      Zeishan Quadri         Gangs of Wasseypur       2012

     next_year  gap
0         2008    3
1         1975    1
2            0    0
3         1989    3
4         1967    1
5         2010    0
6         2012    1
7         2011    1
8         1998    3
9         2011    1
10        1983    1
11        2016    2
12        2011    3
13        2014    1
14        2016    0
15        2016    3
16        1999    3
17        2008    3
18        2012    3
19        2009    0
20        2012    1
```

```
21      2008    3
22         0    0
23      2003    1
24         0    0
25      2005    3
26      2006    3
27      1974    1
28      2018    2
29      2009    0
...      ...   ...
7953    1983    2
7954    1995    2
7955    2002    0
7956    1983    2
7957       0    0
7958    2005    0
7959    1960    3
7960    2006    0
7961    1984    1
7962    2013    0
7963       0    0
7964    1961    2
7965    2012    0
7966    1988    1
7967    1993    1
7968    1995    3
7969    1996    2
7970    2015    1
7971    2009    1
7972    1985    2
7973    1963    3
7974    2000    1
7975    1973    2
7976    1984    3
7977    2015    1
7978    2015    0
7979    1972    1
7980    2004    1
7981    2007    3
7982    2014    2

[7983 rows x 5 columns]
```

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

```
[3]:  # %%time
      # The following query is correct but didn't give the results,
```

```
# Running below query gives "database or disk is full" error
# result = pd.read_sql_query(
#              '''
#              SELECT P1.PID, P1.Name, count(Movie.MID) AS movies_with_yc from␣
↪Person as P1
#                 JOIN M_Cast
#                 JOIN Movie
#                 JOIN M_Director ON (trim(Movie.MID) = trim(M_Director.MID))
#                 JOIN Person as P2 ON (trim(M_Director.PID) = trim(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
#                         JOIN Movie
#                         JOIN M_Director ON (trim(Movie.MID) = trim(M_Director.
↪MID))
#                         JOIN Person AS P4 ON (trim(M_Director.PID) = trim(P4.
↪PID))
#                         WHERE P1.PID = P3.PID AND P4.Name != 'Yash Chopra'
#                         GROUP BY P4.PID
#                     )
#                 ORDER BY movies_with_yc DESC;
#              ''', conn)
# result
```

```
[4]: %%time
result = pd.read_sql_query(
             '''
             SELECT Director, Actor, Count(1) AS Movies_with_YashChopra
             FROM
                 (
                     SELECT p.Name AS Director, m.title AS Movie
                     FROM Person p
                     JOIN M_Director md ON trim(md.PID)=trim(p.PID)
                     JOIN Movie m ON trim(md.MID)=m.MID and p.Name LIKE 'Yash%'
                     GROUP BY p.Name, m.title
                 ) t1
             JOIN
                 (
                     SELECT p.Name AS Actor, m.title AS Movie
                     FROM Person p
                     JOIN M_Cast mc ON trim(mc.PID)=trim(p.PID)
                     JOIN Movie m ON trim(mc.MID)=m.MID
                     GROUP BY p.Name, m.title
                 ) t2
             ON t1.Movie=t2.Movie
```

```
            GROUP BY t1.Director, t2.Actor
            ORDER By Movies_with_YashChopra DESC
            ''', conn)

result
```

CPU times: user 8min 5s, sys: 37.9 s, total: 8min 43s
Wall time: 9min 24s

[4]:

	Director	Actor	Movies_with_YashChopra
0	Yash Chopra	Jagdish Raj	11
1	Yash Chopra	Manmohan Krishna	10
2	Yash Chopra	Manmohan Krishna	10
3	Yash Chopra	Iftekhar	9
4	Yash Chopra	Madan Puri	8
5	Yash Chopra	Vikas Anand	8
6	Yash Chopra	Anupam Kher	7
7	Yash Chopra	Shashi Kapoor	7
8	Yash Chopra	Anupam Kher	7
9	Yash Chopra	Shashi Kapoor	7
10	Yash Chopra	Amitabh Bachchan	6
11	Yash Chopra	Rakhee Gulzar	5
12	Yash Chopra	Waheeda Rehman	5
13	Yash Chopra	Achala Sachdev	4
14	Yash Chopra	Deven Verma	4
15	Yash Chopra	Hema Malini	4
16	Yash Chopra	Neetu Singh	4
17	Yash Chopra	Ravikant	4
18	Yash Chopra	Rishi Kapoor	4
19	Yash Chopra	Shah Rukh Khan	4
20	Yash Chopra	Deven Verma	4
21	Yash Chopra	Hema Malini	4
22	Yash Chopra	Rishi Kapoor	4
23	Yash Chopra	A.K. Hangal	3
24	Yash Chopra	Anil Kapoor	3
25	Yash Chopra	Annu Kapoor	3
26	Yash Chopra	Leela Chitnis	3
27	Yash Chopra	Mohan Sherry	3
28	Yash Chopra	Parikshat Sahni	3
29	Yash Chopra	Prem Chopra	3
..	...	...	...
484	Yash Chopra	Uttam Sodi	1
485	Yash Chopra	Varun Thajur	1
486	Yash Chopra	Varun Thakur	1
487	Yash Chopra	Varun Vardhan	1
488	Yash Chopra	Vic Waghorn	1
489	Yash Chopra	Vicky Ahuja	1

```
490  Yash Chopra      Vicky K. Foster                    1
491  Yash Chopra       Vinay Sharma                      1
492  Yash Chopra      Vinita Sharma                      1
493  Yash Chopra        Vinod Negi                       1
494  Yash Chopra        Vinod Raut                       1
495  Yash Chopra     Virat Raj Gupta                     1
496  Yash Chopra   Vishal Om Prakash                     1
497  Yash Chopra     Yashodra Katju                      1
498  Yash Chopra         Yasin Khan                      1
499  Yash Chopra       Zohra Sehgal                      1
500  Yash Chopra         Aamir Khan                      1
501  Yash Chopra       Arjun Sablok                      1
502  Yash Chopra         Aziz Mirza                      1
503  Yash Chopra          Dev Anand                      1
504  Yash Chopra         Feroz Khan                      1
505  Yash Chopra           Mehmood                       1
506  Yash Chopra       Puneet Issar                      1
507  Yash Chopra       Raman Kumar                       1
508  Yash Chopra       Rani Mukerji                      1
509  Yash Chopra      Romesh Sharma                      1
510  Yash Chopra            Sachin                       1
511  Yash Chopra         Sajid Khan                      1
512  Yash Chopra         Sunny Deol                      1
513  Yash Chopra        Tinnu Verma                      1

[514 rows x 3 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 whoseShahrukh number is 2

```
[11]: %%time
      result = pd.read_sql_query(
              '''
              SELECT Name FROM Person WHERE trim(Name) LIKE '%shah rukh khan%'
              ''', conn)

      result
```

```
CPU times: user 20.2 ms, sys: 12.4 ms, total: 32.6 ms
Wall time: 50.4 ms
```

```
[11]:            Name
      0   Shah Rukh Khan
```

Using below steps we can get Shah Rukh Khan 2nd Degree Connection

- Logic to Build following Query
- Select movies in which Shah Rukh Khan worked
- Select Shah Rukh level 1 i.e. 1st Degree connection of Shah Rukh Khan
- Select movies in which Shah Rukh level 1 worked but exclude movies with Shah Rukh Khan
- Select Shah Rukh level 2 who worked in some movie with Shah Rukh level 1

```
[6]: %%time
result = pd.read_sql_query('''
            SELECT DISTINCT P.Name
            FROM Person p
                    JOIN M_Cast mc
                        ON trim(p.PID) = trim(mc.PID)
            WHERE mc.MID IN (SELECT mc.MID
                                FROM M_Cast mc
                                WHERE trim(mc.PID) IN (
                                    SELECT trim(p.PID) as pid
                                    FROM Person p
                                            JOIN M_Cast mc
                                                ON trim(p.PID) = trim(mc.PID)
                                    WHERE mc.MID IN (
                                        SELECT mc.MID
                                        FROM Person p
                                                JOIN M_Cast mc
                                                    ON trim(p.PID) = trim(mc.PID)
                                        WHERE trim(p.Name) LIKE '%shah rukh khan%'
                                    )
                                    AND trim(p.Name) NOT LIKE '%shah rukh khan%'
                                )
                                AND mc.MID NOT IN (SELECT mc.MID
                                                    FROM Person p
                                                            JOIN M_Cast mc
                                                                ON trim(p.PID)␣
↪= trim(mc.PID)
                                                    WHERE trim(p.Name) LIKE␣
↪'%shah rukh khan%'))

            ''', conn)

result
```

```
CPU times: user 17min 18s, sys: 37.5 s, total: 17min 55s
Wall time: 18min 20s
```

```
[6]:                    Name
     0        Alicia Vikander
     1          Dominic West
     2        Walton Goggins
```

3	Daniel Wu
4	Kristin Scott Thomas
5	Derek Jacobi
6	Alexandre Willaume
7	Tamer Burjaq
8	Adrian Collins
9	Keenan Arrison
10	Andrian Mazive
11	Milton Schorr
12	Hannah John-Kamen
13	Peter Waison
14	Samuel Mak
15	Sky Yang
16	Civic Chung
17	Josef Altin
18	Billy Postlethwaite
19	Roger Jean Nsengiyumva
20	Jaime Winstone
21	Michael Obiora
22	Shekhar Varma
23	Rekha John-Cheriyan
24	Antonio Aakeel
25	Maisy De Freitas
26	Emily Carey
27	Gordon Chow
28	Duncan Airlie James
29	Jandre le Roux
...	...
25939	Premji
25940	Kamu
25941	Monal
25942	Kanchan Mullick
25943	Sumit Ganguly
25944	Aindrila Banerjee
25945	Aritro Dutta Banik
25946	Soham Chakraborty
25947	Hiran Chatterjee
25948	Prasun Gayen
25949	Mallika Ghosh
25950	Subrata Guharoy
25951	Raju Majumdar
25952	Sudipta Majumdar
25953	Vishwesh Krishnamoorthy
25954	Sabitha Perara
25955	R. Sundarajan
25956	Ushma Rathod
25957	Zubeda Khan

```
25958                N. Sagar
25959            Habib Tanvar
25960       Mohd. Zahiruddin
25961          Muktha George
25962                Anjuman
25963           Dhruv Shetty
25964           Minoo Mehtab
25965        Hayley Cleghorn
25966        Nirvasha Jithoo
25967         Kamal Maharshi
25968          Mohini Manik

[25969 rows x 1 columns]
```

[ ]: