Mounting the drive

In [0]:

```
!kill -9 -1
```

In [1]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [2]:

```
!pwd
!ls
```

```
/content
drive  sample_data
```

In [3]:

```
import os
PATH = os.getcwd()
print(PATH)
```

```
/content
```

In [4]:

```
data_path = PATH + '/drive/My Drive/AAIC/Case Studies/SQL/'
data_path
```

Out[4]:

```
'/content/drive/My Drive/AAIC/Case Studies/SQL/'
```

In [0]:

```
import warnings
warnings.filterwarnings("ignore")
import pandas as pd
import sqlite3
```

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 [0]:

```
conn = sqlite3.connect(data_path + "Db-IMDB.db")
```

In [0]:

```
result = pd.read_sql_query("select distinct trim(p.name), m.title, m.year from movie m join
M_Director md on m.mid=md.mid join M_Genre mg on m.mid=mg.mid\
                           join Genre g on mg.gid=g.gid join Person p on md.pid=p.pid where g.name
like '%Comedy%' and\
                           ((m.year%4=0 and not m.year%100=0) or m.year%400=0);", conn)
result
```

Out[0]:

	trim(p.name)	title	year
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
...	...	...	...
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 × 3 columns

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

In [0]:

```
result = pd.read_sql_query("select p.name from Person p join M_Cast mc on trim(p.pid)=trim(mc.pid)
join movie m on mc.mid=m.mid where m.title='Anand';",conn)
result
```

Out[0]:

	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 and in a film after 1990. (That is: < 1970 and > 1990.)

In [17]:

```
pd.read_sql_query("select distinct p.name from Person p where p.name in \
```

```
                    (select p2.name from Person p2 join M_Cast mc on p2.pid=trim(mc.pid)\
                    join Movie m on mc.mid=m.mid where m.year<1970) and\
                    p.name in (select p3.name from Person p3 join M_Cast mc2 on p3.pid=trim(mc2.pid)\
                    join Movie m2 on mc2.mid=m2.mid where m2.year>1990)",conn)
```

Out[17]:

	Name
0	Rishi Kapoor
1	Rajesh Kumar
2	Anand Tiwari
3	Amitabh Bachchan
4	Asrani
...	...
468	Vinod Mehra
469	Deven Verma
470	Master Bhagwan
471	Rishi Kapoor
472	Asrani

473 rows × 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 [0]:

```
pd.read_sql_query("select distinct trim(p.name) as Names,count(p.name) name_count from Person p jo
in M_Director md on p.pid=md.pid join movie m on md.mid=m.mid\
                    where p.pid in(select trim(pid) from M_Director group by trim(pid) having count(*
)>=10) group by Names order by name_count desc",conn)
```

Out[0]:

	Names	name_count
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	Abbas Alibhai Burmawalla	34
10	Shyam Benegal	34
11	Gulzar	32
12	Manmohan Desai	32
13	Raj N. Sippy	32
14	Mahesh Manjrekar	30
15	Priyadarshan	30
16	Raj Kanwar	30
17	Indra Kumar	28
18	Rahul Rawail	28

	Names	name_count
19	Raj Khosla	28
20	Rajkumar Santoshi	28
21	Ananth Narayan Mahadevan	26
22	Anurag Kashyap	26
23	Dev Anand	26
24	Harry Baweja	26
25	K. Raghavendra Rao	26
26	Rakesh Roshan	26
27	Vijay Anand	26
28	Anees Bazmee	24
29	Anil Sharma	24
30	Guddu Dhanoa	24
31	Madhur Bhandarkar	24
32	Nagesh Kukunoor	24
33	Prakash Jha	24
34	Prakash Mehra	24
35	Rohit Shetty	24
36	Satish Kaushik	24
37	Umesh Mehra	24
38	Govind Nihalani	22
39	Ketan Mehta	22
40	Mohit Suri	22
41	Nasir Hussain	22
42	Pramod Chakravorty	22
43	Sanjay Gupta	22
44	Bimal Roy	20
45	Hansal Mehta	20
46	J. Om Prakash	20
47	J.P. Dutta	20
48	Mehul Kumar	20
49	N. Chandra	20
50	Raj Kapoor	20
51	Sudhir Mishra	20
52	Tigmanshu Dhulia	20
53	Vishal Bhardwaj	20
54	Rama Rao Tatineni	17
55	K. Bapaiah	10
56	K. Muralimohana Rao	10
57	Pankaj Parashar	10

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

In [0]:

```
pd.read_sql_query("select substr(m.year,-4,4) Year, count(*) NoOfMovies from movie m where not exi
sts (select * from Person p, M_Cast mc where trim(p.pid) = trim(mc.pid) and mc.mid = m.mid and p.g
ender not like 'F') group by Year;",conn)
```

Out[0]:

	Year	NoOfMovies

	Year	NoOfMovies
0	2009	1
1	2012	1
2	2018	1

5. 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 [0]:

```
pd.read_sql_query("select table1.year_table1, table1.year_count*100.00/table2.year_count as Percen
tage, table2.year_count as TotalMovies from (select\
      substr(m.year,-4,4) as year_table1, count(*) as year_count from movie m where not exists (se
lect * from person p, m_cast mc where trim(p.pid) = trim(mc.pid)\
      and mc.mid = m.mid and p.gender!='F') group by year_table1) table1 , (select substr(m.year,-
4,4) as year_table2, count(*) as year_count from movie m group by year_table2)\
      table2 where table1.year_table1=table2.year_table2;",conn)
```

Out[0]:

	year_table1	Percentage	TotalMovies
0	2009	0.909091	110
1	2012	0.900901	111
2	2018	0.961538	104

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 [0]:

```
pd.read_sql_query("select m.title Movie_Name, count(distinct mc.PID) as Cast_Size from Movie m joi
n M_Cast mc on m.mid=mc.mid group by m.mid order by Cast_Size desc limit 1;",conn)
```

Out[0]:

	Movie_Name	Cast_Size
0	Ocean's Eight	238

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 [0]:

```
pd.read_sql_query("select table1.year as Starting_Year, table1.year+9 Ending_Year,  count(*) as Nu
mber_of_movies from \
                  (select distinct m.year from movie m) table1, movie m1 where table1.year <= m1.ye
ar\
                  and m1.year < table1.year+10 group by table1.year having not exists (select table
2.year from (select distinct m2.year from movie m2) table2, movie m3 \
                  where table2.year <= m3.year and m3.year < table2.year+10 group by table2.year ha
ving count(m3.mid) > count(m1.mid));",conn)
```

Out[0]:

	Starting_Year	Ending_Year	Number_of_movies
0	2008	2017	1128

This means that the decade D started from 2008-2017

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 [0]:

```
pd.read_sql_query("select p1.Name from Person p1 where p1.pid not in (select distinct(mc1.pid) fro
m M_Cast as mc1 join Movie as m1 on mc1.mid=m1.mid where\
      exists(select mc2.mid from M_Cast as mc2 join Movie as m2 on mc2.mid=m2.mid where
trim(mc1.PID) = trim(mc2.PID) and (substr(m2.year,-4,4) - 3) > substr(m1.year,-4,4) and\
      not exists(select trim(mc3.MID) from M_Cast as mc3 join Movie as m3 on mc3.mid=m3.mid where
trim(mc1.PID) = trim(mc3.PID) and substr(m1.year,-4,4) < substr(m3.year,-4,4) and\
      substr(m3.year,-4,4) < substr(m2.year,-4,4))));",conn)
```

Out[0]:

	Name
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

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

In [0]:

```
pd.read_sql_query("select table1.name Name,table1.no_of_dirs NumberOfMovies from \
                          (select p1.name,mc.mid,m.title,count(p2.name) no_of_dirs from Person p
1 join M_cast mc on p1.pid = trim(mc.pid)\
                           join Movie m on m.mid=mc.mid join M_director md on md.mid=m.mid join
person p2 on md.pid=p2.pid where p2.name like 'Yash Chopra'\
                           group by p1.name having count(p2.name)> (select max(no_of_dirs) from
(select p3.name ,count(trim(p4.name)) no_of_dirs from Person p3\
                           join M_cast mc2 on p3.pid = trim(mc2.pid) join Movie m2 on
m2.mid=mc2.mid join M_director md2 on md2.mid=m2.mid\
                           join person p4 on md2.pid=p4.pid where trim(p4.name) not like 'Yash
Chopra' group by p3.name,trim(p4.name) ) table2 group by table2.name)) table1",conn)
```

Out[0]:

	Name	NumberOfMovies
0	A.K. Hangal	3
1	Achala Sachdev	4
2	Amarinder Sodhi	2
3	Amitabh Bachchan	6
4	Ananth Narayan Mahadevan	2
...	...	...
83	Rajendra Kumar	2

	Name	NumberOfMovies
84	Rishi Kapoor	4
85	Shashi Kapoor	7
86	Sunil Dutt	2
87	Yash Chopra	2

88 rows × 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 [0]:

```
pd.read_sql_query("select p.Name from person p where p.Name like '%shah rukh khan%'",conn)
```

Out[0]:

	Name
0	Shah Rukh Khan

In [0]:

```
pd.read_sql_query("select distinct trim(p4.name) ActorNames from Person p4\
                                join M_cast mc4 on p4.pid=trim(mc4.pid)\
                                join Movie m4 on m4.mid=mc4.mid and p4.name not like '%Shah Rukh
Khan%'\
                                and m4.title in (select distinct m3.title from Person p3\
                                join M_cast mc3 on p3.pid=trim(mc3.pid) and trim(p3.name) = p3.name \
                                join Movie m3 on m3.mid=mc3.mid and p3.name in\
                                (select distinct p2.name from Person p2\
                                join M_cast mc2 on p2.pid=trim(mc2.pid)\
                                join movie m2 on m2.mid=mc2.mid and p2.name not like '%Shah Rukh
Khan%'\
                                and m2.title in (select distinct m1.title from Person p1\
                                join M_cast mc1 on p1.pid=trim(mc1.pid) and p1.name like '%Shah Rukh
Khan%' \
                                join Movie m1 on m1.mid=mc1.mid)));",conn)
```

Out[0]:

	ActorNames
0	Sheeba Chaddha
1	Rishi Kapoor
2	Jeetendra Shastri
3	Yuvraj Sachdev
4	Nirmal Rishi
...	...
16160	Dhruv Shetty
16161	Mohini Manik
16162	Kamal Maharshi
16163	Hayley Cleghorn
16164	Nirvasha Jithoo

16165 rows × 1 columns