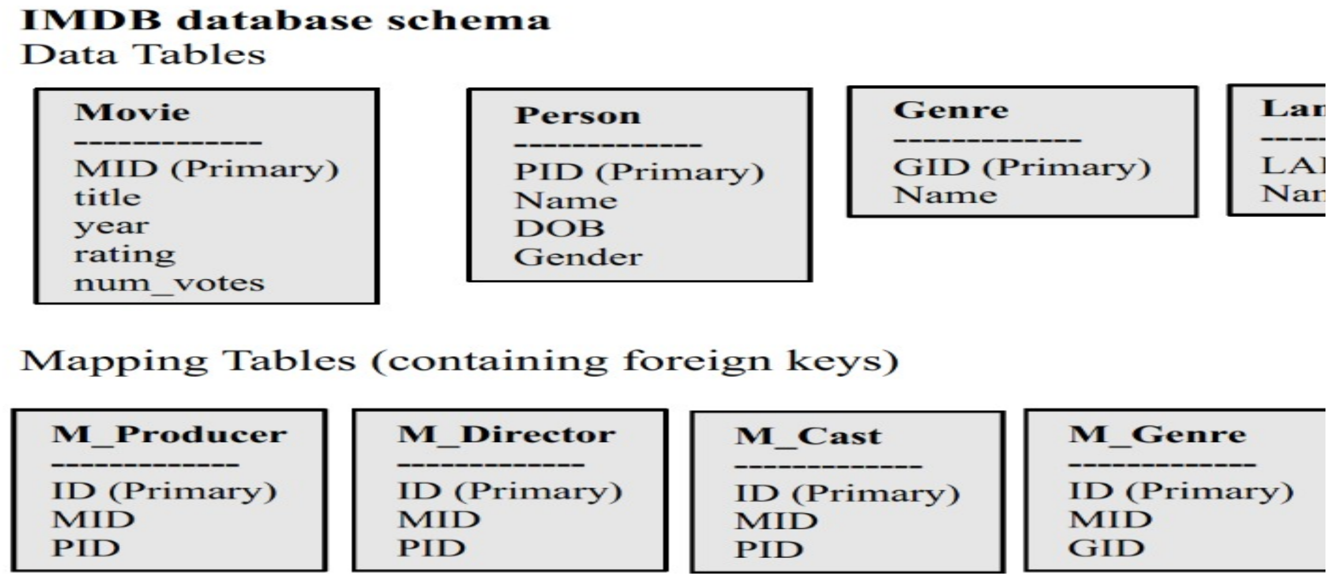
```
from google.colab import drive
drive.mount('/content/drive')
```

SQL Assignment on IMDB dataset

1. Load libraries

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

```
from IPython.display import Image
Image("/content/drive/My Drive/SQL Assignment/db_schema.jpeg",width=1200, height=300)
```



IMDB database schema

Data Tables

Movie

MID (Primary)
title
year
rating
num_votes

Person

PID (Primary)
Name
DOB
Gender

Genre

GID (Primary)
Name

Lar

LAI
Nar

Mapping Tables (containing foreign keys)

M_Producer

ID (Primary)
MID
PID

M_Director

ID (Primary)
MID
PID

M_Cast

ID (Primary)
MID
PID

M_Genre

ID (Primary)
MID
GID

2. Establishing connection to database file

```
conn = sqlite3.connect('/content/drive/My Drive/SQL Assignment/Db-IMDB.db')
```

3. List all tables in database

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

	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

4. Assignment Questions

- List all the directors who directed a 'Comedy' movie in a leap year. (You need to check that the g Your query should return director name, the movie name, and the year.

```
ans = pd.read_sql_query("SELECT p.Name Director_name,a.title Movie,a.year Year,c.Name 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
                          AND a.year%4=0 group by p.Name,a.title",conn)
ans
```

	Director_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

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

```
ans = pd.read_sql_query("SELECT Name Actor from Person p JOIN M_Cast c ON TRIM(p.PID) = TRIM(
                          (SELECT MID from Movie WHERE title = 'Anand')",conn)
ans
```

	Actor
0	Rajesh Khanna
1	Amitabh Bachchan
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

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

```
#source: https://stackoverflow.com/questions/29617880/sql-list-actors-who-acted-in-a-film-bef
ans = pd.read_sql_query("SELECT name Actor FROM Person WHERE TRIM(PID) IN \
                          (SELECT TRIM(PID) FROM M_Cast WHERE MID IN \
                          (SELECT MID FROM Movie m WHERE m.year > 1990) \
                          AND PID IN (SELECT PID FROM M_Cast WHERE MID IN \
                          (SELECT MID FROM Movie n WHERE n.year < 1970))",conn)
ans
```

```
                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
```

- List all directors who directed 10 movies or more, in descending order of the number of movies and the number of movies each of them directed

```
ans = pd.read_sql_query("SELECT DISTINCT p.Name Director,COUNT(*) number_of_movies FROM Perso
                          JOIN M_Director d on TRIM(p.PID) = TRIM(d.PID) \
                          GROUP BY TRIM(d.PID) HAVING COUNT(*) >=10 ORDER BY number_of_movies
ans
```

	Director	number_of_movies
0	David Dhawan	78
1	Mahesh Bhatt	70
2	Ram Gopal Varma	60
3	Vikram Bhatt	58
4	Hrishikesh Mukherjee	54
...	...	...
151	Siddharth Anand	10
152	Dibakar Banerjee	10
153	Shoojit Sircar	10
154	R. Balki	10
155	Neeraj Pandey	10

156 rows × 2 columns

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

```
#source: https://stackoverflow.com/questions/57743348/sql-query-imdb-data-to-count-the-total-
ans = pd.read_sql_query("SELECT movie.year Year,count(*) Count FROM Movie \
                          WHERE NOT EXISTS \
                          (SELECT * FROM M_Cast,Person WHERE person.gender='Male' and M_Cast.
                          and M_Cast.PID = person.PID ) GROUP BY movie.year",conn)
ans
```

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

125 rows × 2 columns

> - Now include a small change: report for each year the percentage of movies in that year witl
>   total number of movies made that year. For example, one answer will be: 1990 31.81 13522
>   were 13,522 movies, and 31.81% had only female actors. You do not need to round your an

```
#source: https://stackoverflow.com/questions/57743348/sql-query-imdb-data-to-count-the-total-
ans = pd.read_sql_query("SELECT female_count.year Year,((female_count.Total_movies_with_only_
                        ((SELECT movie.year Year,count(*) Total_movies_with_only_female_l
                        ( SELECT * FROM M_Cast,person WHERE M_Cast.mid = movie.MID and M_
                        GROUP BY movie.year) female_count, \
                        (SELECT movie.year,count(*) as Total FROM movie group by movie.ye
                        WHERE female_count.year=total_count.year",conn)
ans
```

```
            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
```

- Find the film(s) with the largest cast. Return the movie title and the size of the cast. By "cast siz that played in that movie: if an actor played multiple roles, or if it simply occurs multiple times ir

```
 ans = pd.read_sql_query("SELECT m.title Movie_Name,count(distinct(c.PID)) Cast_Size FROM Mov
                           ON c.MID = m.MID GROUP BY m.MID ORDER BY Cast_Size desc",conn)
ans
```

	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
...	...	...
3470	Subah Subah	1
3471	Chaar Sahibzaade 2: Rise of Banda Singh Bahadur	1
3472	Vaibhav Sethia: Don't	1
3473	Yeh Hai Malegaon Ka Superman	0
3474	The Wish Fish	0

3475 rows × 2 columns

- A decade is a sequence of 10 consecutive years. For example, say in your database you have m the first decade is 1965, 1966, ..., 1974; the second one is 1967, 1968, ..., 1976 and so on. Find t films and the total number of films in D.

```
 ans = pd.read_sql_query("SELECT d.year Start, d.year+9 End, count(*) no_of_films FROM \
                            (SELECT DISTINCT year from Movie) d JOIN Movie m ON m.year >= Sta
                            GROUP BY End ORDER BY no_of_films desc LIMIT 1",conn)
ans
```

	Start	End	no_of_films
0	2008	2017	1128

- Find the actors that were never unemployed for more than 3 years at a stretch. (Assume that th consecutive movies).

```
 #SOURCE: GITHUB
 ans = 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 join 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<m2.year)))",conn)
ans
```

```
ans
```

	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

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

```
 ans = 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.ti
 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.Mov
 Group By t2.Actor Order By Movies_with_YashChopra DESC",conn)
ans
```

```
                Actor  Movies_with_YashChopra
  0       Jagdish Raj                      11
  1   Manmohan Krishna                     10
  2   Manmohan Krishna                     10
  3          Iftekhar                       9
  4        Madan Puri                       8
 ...              ...                     ...
509    Romesh Sharma                        1
510           Sachin                        1
511       Sajid Khan                        1
512       Sunny Deol                        1
513      Tinnu Verma                        1

514 rows × 2 columns
```

- The Shahrukh number of an actor is the length of the shortest path between the actor and Shah Shahrukh Khan has Shahrukh number 0; all actors who acted in the same film as Shahrukh have in the same film as some actor with Shahrukh number 1 have Shahrukh number 2, etc. Return al

```
ans = pd.read_sql_query("SELECT DISTINCT TRIM(name) 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 m.title in (SELECT DISTINCT title FROM Person p3 INNER JOIN M_Cast c3 on p3.PID = TRIM(c
INNER JOIN Movie m3 ON m3.MID = c3.MID AND p3.Name IN (SELECT DISTINCT Name FROM Person p2 I
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.PID) AND TRIM
INNER JOIN Movie m3 ON m3.MID = c3.MID))) ORDER BY Name",conn)
```

```
ans
```

```
                          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
```