Out[]:	join movie using(MID) where Person.PID='nm2147526' order by year ''' , conn) index PID Name Gender index MID PID ID index title year rating num_votes 1 171 nm2147526 Asrani Male 51705 tt0244776 nm2147526 51705 1889 Satyakam 1969 8.2 570 2 1171 nm2147526 Asrani Male 37425 tt0067164 nm2147526 37425 1245 Guddi 1971 7.4 820
	3 1171 nm2147526 Asrani Male 62684 tt0067421 nm2147526 62684 2406 Mere Apne 1971 7.6 426 4 37974 nm2147526 Asrani Male 37425 tt0067164 nm2147526 37425 1245 Guddi 1971 7.4 820
In []:	372 rows × 13 columns
In []: Out[]:	PID Name 0 nm0438501 Rishi Kapoor 1 nm0000821 Amitabh Bachchan 2 nm2147526 Asrani 3 nm0782247 Zohra Sehgal
	4 nm0756378 Parikshat Sahni 295 nm0542469 Manish 296 nm1171301 Vijayalaxmi 297 nm0214350 Manohar Deepak 298 nm1533565 Brij Bhushan 299 nm0705549 Radha
In []: In []:	
In []: Out[]:	Name movie_count 0 David Dhawan 39 1 Mahesh Bhatt 35 2 Priyadarshan 30 3 Ram Gopal Varma 30
	4 Vikram Bhatt 29 5 Hrishikesh Mukherjee 27 6 Yash Chopra 21 7 Basu Chatterjee 19 8 Shakti Samanta 19 9 Subhash Ghai 18 10 Shyam Benegal 17
	11Abbas Alibhai Burmawalla1712Rama Rao Tatineni1713Manmohan Desai1614Gulzar1615Raj N. Sippy1616Raj Kanwar1517Mahesh Manjrekar15
	18 Indra Kumar 14 19 Raj Khosla 14 20 Rahul Rawail 14 21 Rajkumar Santoshi 14 22 Rakesh Roshan 13 23 Dev Anand 13 24 Vijay Anand 13
	 Harry Baweja 13 Anurag Kashyap 13 Ananth Narayan Mahadevan 13 K. Raghavendra Rao 13 Anees Bazmee 12 Guddu Dhanoa 12 Prakash Jha 12
	32 Satish Kaushik 12 33 Nagesh Kukunoor 12 34 Prakash Mehra 12 35 Umesh Mehra 12 36 Anil Sharma 12 37 Madhur Bhandarkar 12 38 Rohit Shetty 12
	39 Pramod Chakravorty 11 40 Sanjay Gupta 11 41 Nasir Hussain 11 42 Ketan Mehta 11 43 Govind Nihalani 11 44 Mohit Suri 11 45 Raj Kapoor 10 46 K. Bapaiah 10
	47 Vishal Bhardwaj 10 48 N. Chandra 10 49 Tigmanshu Dhulia 10 50 J.P. Dutta 10 51 Mehul Kumar 10 52 Hansal Mehta 10 53 Sudhir Mishra 10
In []: In []:	
In []: Out[]:	<pre>istinct MID from M_cast where MID NOT in(</pre>
	1 1936 3 2 1939 2 3 1941 1 4 1943 1 73 2014 126 74 2015 119
In []:	75 2016 129 76 2017 126 77 2018 104 78 rows × 2 columns pd.read_sql_query('''
	select distinct mid from M_Cast where mid not in (select mid from M_Cast join Person on Person.pid = M_Cast.pid where Person.gender = 'Male'))aa on aa.mid = Movie.mid group by Movie.year
Out[]:	year movie_count 0 1939
In []: In []:	5 2018 2 5(b) pd.read_sql_query(''' select cc.year,cc.movie_count as INT,cc.total as INT,(cc.mo vie_count*100.0/cc.total*1.0) as percentage from (Select mv2.year,bb.movie_count,count(*) as total from Movie as mv2 join (select year, count(*) as movie_count t from Movie as mv1
	<pre>join (select distinct mid from M_Cast as m1 where mid not in (select mid from M_Cast as m2 join Person as p1 on p1.pid = m2.pid where p1.gender = 'Male'))aa on aa.mid = mv1.mid group by mv1.year)bb on mv2.year=bb.year group by(mv2.year))</pre>
Out[]:	year INT INT percentage 0 1939
In []: In []:	4 2012 1 111 0.900901 5 2018 2 104 1.923077
In []: Out[]: In []: In []:	title cast_count O Ocean's Eight 238
Out[]:	<pre>count(*) as total_movies from (select distinct year from movie) aa join movie m on m.year >= aa.year and m.year < aa.year + 10 group by aa.year order by count(*) desc limit 1 ''',conn)</pre> start_year end_year total_movies
In []: In []:	<pre>pd.read_sql_query(''' Select *from(select aa.year as start_year, aa.year + 9 as end_year,</pre>
Out[]:	<pre>on m.year >= aa.year and m.year < aa.year + 10 group by aa.year order by count(*) desc limit 2) order by total_movies limit 1 ''', conn)</pre> start_year end_year total_movies 0 2009 2018 1202
In []: Out[]:	<pre>pd.read_sql_query("select max(movie_count), my_year from(select count(*) as movie_count, (year -1)/10 as my_year from movie group by my_year) ", conn) max(movie_count) my_year 0 1047 200</pre>
In []: In []:	<pre>pd.read_sql_query('''select * from M_cast join movie using(MID) where PID='nm5951787' ''',c onn)</pre>
Out[]:	
	index MID PID ID index title year rating num_votes 0 310 tt1365519 nm5951787 310 2 Tomb Raider 2018 6.4 142585 1 6129 tt4559006 nm5951787 6129 128 Ae Dil Hai Mushkil 2016 5.8 17713 2 7702 tt7275232 nm5951787 7702 170 Welcome to New York 2018 2.3 316 3 7998 tt1562871 nm5951787 7998 176 Ra.One 2011 4.8 34920 4 8676 tt7607940 nm5951787 8676 195 Namaste England 2018 1.3 1621
	index MID PID ID index title year rating num_votes 0 310 tt1365519 nm5951787 310 2 Tomb Raider 2018 6.4 142585 1 6129 tt4559006 nm5951787 6129 128 Ae Dil Hai Mushkil 2016 5.8 17713 2 7702 tt7275232 nm5951787 7702 170 Welcome to New York 2018 2.3 316 3 7998 tt1562871 nm5951787 7998 176 Ra.One 2011 4.8 34920
	index MID PID ID index title year rating num_votes 0 310 tt1365519 nm5951787 310 2 Tomb Raider 2018 6.4 142585 1 6129 tt4559006 nm5951787 702 170 Welcome to New York 2018 2.3 316 3 7998 tt1562871 nm5951787 7998 176 Ra.One 2011 4.8 34920 4 8676 tt7607940 nm5951787 8676 195 Namaste England 2018 1.3 1621 5 9698 tt3469244 nm5951787 9698 224 Phantom 2015 5.8 7505 6 10613 tt3495026 nm5951787 10613 242 Fan 2016 7.2 37639 7 12527 tt6272828 nm5951787 12718 306 Judwaa 2 2017 5.6 2830 8 12718 tt64565
	Index MID PID Index Itile Year rating num_votes
Out[]:	MID PID ID index MID PID ID index MID PID ID index MID PID ID index MID MISSISTER 310 2 Tomb Raider 2018 6.4 142565
Out[]:	
Out[]:	1
Out[]:	MID
Out[]: Out[]:	Mile
Out[]: Out[]:	1
Out[]: Out[]: In []:	10 10 10 10 10 10 10 10
Out[]: Out[]: In []:	Mathematical Math
Out[]: Out[]: Out[]:	March Marc
Out[]: Out[]: Out[]:	Mail
Out[]: Out[]: Out[]: In []: In []:	March Marc
Out[]: Out[]: Out[]: In []: In []:	The continue
Out[]: Out[]: Out[]: Out[]:	Mart
Out[]: Out[]: Out[]: In []: In []:	March Marc
Out[]: Out[]: Out[]: In []: In []:	Marie Mari
Out[]: Out[]: Out[]: Out[]: Out[]:	March Marc
Out[]: Out[]: Out[]: Out[]: Out[]:	Mart
In []: Out[]: In []: Out[]: In []:	March 1988
In []: Out[]: In []: Out[]: In []:	March Marc
In []: Out[]: In []: Out[]: In []:	The column
Out[]: Out[]: Out[]: Out[]: Out[]: Out[]:	Mart
Out[]: Out[]: Out[]: Out[]: Out[]: Out[]:	Mart
Out[]: Out[]: Out[]: Out[]: Out[]: Out[]:	March Marc
Out[]: Out[]: Out[]: Out[]: Out[]:	March Marc
<pre>Out[]: Out[]: Out[]: Out[]: Out[]: Out[]: Out[]: </pre>	March Marc
<pre>Out[]: Out[]: In []: In []: Out[]: Out[]: In []: Out[]: In []: Out[]: O</pre>	Mate
In []: In []: Out[]: Out[]: Out[]: In []: Out[]: In []: Out[]:	Mate

not in(select MID from Person join M_cast using(PID) where Name='Shah R $\,$

and PID not in (select PID from M_cast where PID!='nm0451321' and MID

(select Distinct MID from Person join M_cast using(PID) where Name='Sha

ukh Khan' group by(MID)))

Name

Freida Pinto nm2951768

Rohan Chand nm4575116

Damian Young nm0949433

Waris Ahluwalia nm1753302

Minoo Mehtab nm1506519

Hayley Cleghorn nm2371614

Nirvasha Jithoo nm2675737

Kamal Maharshi nm2370589

Mohini Manik nm1866356

4 Caroline Christl Long nm6467532

''', conn)

in

Out[]:

In []:

h Rukh Khan'))

0

1

3

2569325694

25695

25696

25697

25698 rows × 2 columns

In []: import os

ly

In []: import sqlite3

import pandas as pd

In []: from google.colab import drive

drive.mount('/content/drive')

Enter your authorization code:

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

cursor.execute('UPDATE Movie SET title = LTRIM(title);')

cursor.execute('UPDATE Movie SET year = RTRIM(LTRIM(year));')
cursor.execute('UPDATE Movie SET rating = RTRIM(LTRIM(rating));')

cursor.execute('UPDATE M_Producer SET pid = RTRIM(LTRIM(pid));')
cursor.execute('UPDATE M_Producer SET mid = RTRIM(LTRIM(mid));')

cursor.execute('UPDATE M_Director SET pid = RTRIM(LTRIM(pid));')
cursor.execute('UPDATE M_Director SET mid = RTRIM(LTRIM(mid));')

cursor.execute('UPDATE M_Cast SET pid = RTRIM(LTRIM(pid));')
cursor.execute('UPDATE M_Cast SET mid = RTRIM(LTRIM(mid));')

cursor.execute('UPDATE M_Genre SET gid = RTRIM(LTRIM(gid));')
cursor.execute('UPDATE M_Genre SET mid = RTRIM(LTRIM(mid));')

cursor.execute('UPDATE Genre SET gid = RTRIM(LTRIM(gid));')
cursor.execute('UPDATE Genre SET name = RTRIM(LTRIM(name));')

e.name LIKE'%comedy%'and movie.year % 4 = 0",conn)

Danny Leiner 2004 Harold & Kumar Go to White Castle

Name year

Milap Zaveri 2016

Anurag Kashyap 2012

Frank Coraci 2004

Griffin Dunne 2008

Amma Rajasekhar 2008

Oliver Paulus 2008

Raja Chanda 2012

K.S. Prakash Rao 1996

Name

0 Amitabh Bachchan

Rajesh Khanna Brahm Bhardwaj

Ramesh Deo

227 Siddharth Anand Kumar 2004

...

cursor.execute('UPDATE Person SET name = RTRIM(LTRIM(name));')
cursor.execute('UPDATE Person SET pid = RTRIM(LTRIM(pid));')

cursor.execute('UPDATE Person SET gender = RTRIM(LTRIM(gender));')

In []: df2=pd.read_sql_query("select Person1.Name, movie.year, movie.title from Person1 join M_Direct

or using(PID) join movie using(MID) join m_genre using(MID) join Genre using(GID) where Genr

title

Mastizaade

Let's Enjoy

Sathyam

Tandoori Love

Le Halua Le

Raja Aur Rangeeli

In []: df2=pd.read_sql_query("select Person.Name from Person join M_Cast on Person.PID=TRIM(m_cast.
PID) join movie using(MID) where movie.title='Anand' and movie.year=1971 ",conn)

Gangs of Wasseypur

Around the World in 80 Days

The Accidental Husband

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 ", "");')

cursor.execute('UPDATE Movie SET num_votes = RTRIM(LTRIM(num_votes));')

Mounted at /content/drive

import pandas as pd

In []: cursor = conn.cursor()

Create the connection

Out[]: <sqlite3.Cursor at 0x7f7a2111f9d0>

In []: Q1

In []: df2

1

2

3

4

...

228

229

230

231

1

3

In []: Q2

In []: df2

Out[]:

232 rows × 3 columns

Out[]:

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803 -6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3ao auth%3a2.0%3aoob&response_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fd ocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fgrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readon