SOCIAL MEDIA MOVIE DATABASE

INFO 6210 Project

Anisha Ganguly
NUID: 001408910

Fan Ji
NUID: 001350642

Zhouwei Wang
NUID: 001443886

Table of contents

1. ## <u>Abstract</u>
In this project, we're creating a database to store social media news about our domain, which is movies. We're attempting to model, gather and clean data and integrate it with social media data (Twitter, Instagram and YouTube, etc.) about our domain in accordance with movies. Each domain has entities that represent consumer, producer and company and thing. We've modelled movie consumers as movie fans or movie goers, movie producers as movie directors/ actors like Christopher Nolan and movie company as Warner Bros. Pictures, etc.

2. ## <u>Introduction</u>

When Jack spread his arms behind Rose standing on the edge of the deck in the Titanic, a million hearts swooned. When Darth Vader told Luke who his father was, the sigh of stunned disbelief filled thousands of theaters across the globe. Scared viewers refrained from taking a shower for days after watching Marion Crane's stabbing in Psycho. Movies have unimaginable ways of bonding with the audience. They entertain, inform, educate, even scare. They make them laugh and they make them cry. They enable viewers to identify with the characters or situations, or they take them to the world of surreal make-believe, to places where they have never been before. The sway of movies on different audiences is multi-dimensional. In the age of social media, it is much easier to stay connected with news about your favorite movies, actors or entertainment companies. Almost every blockbuster movies, actor, company or director have their own official social media account that they use to promote their movies or their brand. These social media accounts have millions of fans following them, commenting and liking their every post and generating hashtags to popularize their post. Everyday, there are millions of consumer posts sharing their every thought about their favorite movie, character, scene, actor, etc. With the overabundance of data in this world, it is hard to filter through the information that truly matters. Through this project, we've attempted to create a database that stores relevant information about movies and we've displayed this data by answering SQL queries.
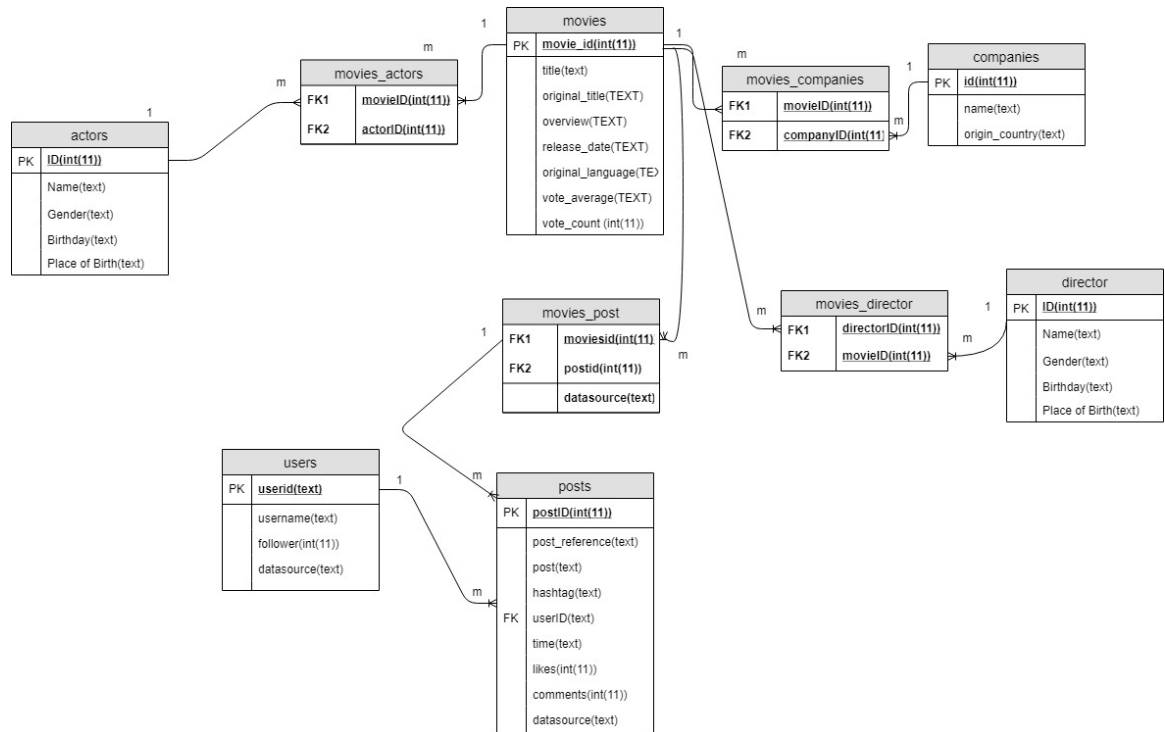
3. Dataset

The dataset we're using consists of the following tables:

- Movies: This table consists of movie details about various movies. The attributes are as follows-
  - Id: The unique id assigned to every movie. It is the primary key of this table.
  - Title: Represents the movie name or the title of the movie.
  - Original_title: Represents the title of the movie in its original language based on the country of origin of the movie.
  - Release_date: Represents the release date of the movie
  - Original_language: Represents the original language the movie was released in
  - vote_average: The voted average ratings of the movie
  - vote_count: The average count of the movie

- Companies: This dataset consists of details about the entertainment companies that make movies. The various attributes are as follows-
  - Id: The unique id assigned to every entertainment company name. It is the primary key of this table.
  - Name: Represents the company names.
  - Origin_country: Represents the country of origin of the companies.

- Movies_companies: This is the junction table joining movies and companies.

  - Movies_id: The movie id from the movie's table
  - companyID: The company id from the companies table

- directors: Consists of details about movie directors. The attributes are as follows-

  - id: The id that is assigned to uniquely identify each director
  - Name: The names of all the directors that directed the movies
  - Gender: The gender of the directors
  - Birthday: The birthday of the directors
  - Place_of_Birth: The place that the directors were born in

- Movies_directors: This is the junction table joining movies and the directors.

  - directorId: The director id from the director's table
  - movieID: The movie id from the movie's table

- actors: This table consists of information of the actors who acted in movies. The attributes are as follows –

- **ID:** The id assigned to every actor name to uniquely identify each row. This is the primary key of this table
- **Name:** The names of all the actors that acted in the movies
- **Gender:** The gender of the actors
- **Birthday:** The birthday of the actors
- **Place_of_Birth:** The place where the actors were born in

o **Movies_actors:** This is the junction table joining movies and the actors.

- **movieID:** The movie id from the movie's table
- **actorID:** The actor's id from the actor's table

o **Posts:** This table consists of all the posts gathered from the different social media's concatenated into one table. They represent consumer posts about movies The attributes of this table are as follows-

- **postID:** The id that uniquely identifies each social media posts of a consumer. This is the primary key in our table
- **post_reference:** The postid gathered from the social media api that is referencing the postid primary key
- **post:** The tweet, caption, comment's posted by users about the movies
- **hashtag:** The hashtag's used by the users to search for the movies or the hashtag's obtained from the posts
- **userID:** The unique id of the user who posted the posts
- **time:** The time at which the post was created or posted
- **likes:** The favorites or likes count a post received
- **comments:** The comment count of each post
- **datasource:** The attribute that distinguishes which source the social media data was retrieved from

o **movies_post:** The junction table between the movies table and the posts table

- **movieid:** The movie id from the movie's table
- **datasource:** The datasource of the social media

o **users:** The table that contains information about the user's who posted in the posts table. The attributes are as follows:

- **userid:** The user Id obtained from social media scraping that uniquely identifies the user who posted the posts. This is the primary key of this table
- **username:** The username of the user who posted the posts
- **followers:** The number of followers or subscribers a user has
- **datasource:** The social media datasource the details of the user was obtained from

4. Conceptual Model:

actors
- PK: ID(int(11))
- Name(text)
- Gender(text)
- Birthday(text)
- Place of Birth(text)

movies_actors
- FK1: movieID(int(11))
- FK2: actorID(int(11))

movies
- PK: movie_id(int(11))
- title(text)
- original_title(TEXT)
- overview(TEXT)
- release_date(TEXT)
- original_language(TE)
- vote_average(TEXT)
- vote_count (int(11))

movies_companies
- FK1: movieID(int(11))
- FK2: companyID(int(11)

companies
- PK: id(int(11))
- name(text)
- origin_country(text)

movies_post
- FK1: moviesid(int(11)
- FK2: postid(int(11))
- datasource(text)

movies_director
- FK1: directorID(int(11))
- FK2: movieID(int(11))

director
- PK: ID(int(11))
- Name(text)
- Gender(text)
- Birthday(text)
- Place of Birth(text)

users
- PK: userid(text)
- username(text)
- follower(int(11))
- datasource(text)

posts
- PK: postID(int(11))
- post_reference(text)
- post(text)
- hashtag(text)
- FK: userID(text)
- time(text)
- likes(int(11))
- comments(int(11))
- datasource(text)

5. Social Media for Analysis:

o 1)What are users saying about me?
Here we are taking 'me' as a movie name and seeing what people are saying about that movie name and extracting all the posts about 'Godfather'.

select * from post where post like '%Godfather%';

Output:

	postid	post_reference	post	hashtag	userID	time
▶	159	1120141301087862784	RT @NorbertElekes: Bechdel Test for top-rated IMDb movies ❌ The…	#The Shawshank Redemption	2876616999	2019-04-22 01:44
	161	1120343281911484417	RT @PaulineofDreams: Happy birthday to The Godfather of Light @ty…	#The Godfather	155991794	2019-04-22 15:07
	162	1120343202723012610	For the Godfather..	#The Godfather	763387438525997056	2019-04-22 15:06
	163	1120343182338871298	@Ayourb Happy birthday, Godfather. May Allah bless you with the best…	#The Godfather	455288787	2019-04-22 15:06
	164	1120343000838811648	RT @longhairwiteboy: Jennifer's Body better than The Godfather	#The Godfather	4017884532	2019-04-22 15:06
	166	1120342518045061123	😊 The Godfather house is on the market! https://t.co/AsAkDrFLG8 htt…	#The Godfather	58068584	2019-04-22 15:04
	167	1120342462785101824	"Dostluk ve para , zeytinyağı ve su gibidir." The Godfather 2	#The Godfather	754971443943895040	2019-04-22 15:03
	168	1120341910621757440	He should ask his aspiring godfather with the skyscraper cap 😊. https…	#The Godfather	173113740	2019-04-22 15:01
	169	1120341712726048769	@BillKristol After The Godfather, everyone wanted to sound/act like a…	#The Godfather	808874124093378566	2019-04-22 15:00
	170	1120341683743461378	'The mannerisms and style of the Marlon Brando portrayal in The Godf…	#The Godfather	9366237655434577712	2019-04-22 15:00
	171	1120341507934957569	RT @AmFriendsHU: A Visit With .@HebrewU's Raphael Mechoulam, t…	#The Godfather	1018522680671326209	2019-04-22 15:00
	172	1120341502150893568	John Muir - The Godfather of Conservation https://t.co/SZHlmkjiWl http…	#The Godfather	18826101	2019-04-22 15:00
	173	1120341499525439490	"I'm gonna make him an offer he can't refuse." The Godfather (1972)	#The Godfather	1068403263551217664	2019-04-22 15:00
	174	1120340773701062656	RT @Mofiobafemi: Happy Birthday, The Godfather @Ayourb, age with…	#The Godfather	257037100	2019-04-22 14:57
	176	1120340360234881025	Never hate your enemies, it affects your judgement. ~ The Godfather ~	#The Godfather	454713765	2019-04-22 14:55
	177	1120340303590903808	@ieffpearlman I'm with you on everything but The Godfather. One of th…	#The Godfather	498357863	2019-04-22 14:55

o 2)How viral are my posts? (Aggregation,Order By)
How viral a post is, depends on the number of likes and comments a post gets. Here, we're creating a Table to store the Viral Posts and we're calculating a viral post like viral = (0.3*Likes + 0.7* Comments) and then we're returning a column displaying how viral a post is and ordering it in descending order.

Select distinct post,hashtag,time,likes,comments,(0.3*likes + 0.7*comments) as viral,datasource

from post

order by viral DESC;

Output:

post	hashtag	time	likes	comments	viral	datasource
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv…	#Your Name.	2019-04-22 15:07:51	0	147700	103390.0	twitter
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv…	#Your Name.	2019-04-22 15:07:52	0	147700	103390.0	twitter
RT @halsey: my pink dream with my pink friends this is: #BoyWithLuv…	#My Friends	2019-04-22 15:08:03	0	128462	89923.4	twitter
RT @Bindilrwin: Every time I watch this clip my heart overflows with e…	#Life Is Bea…	2019-04-22 15:07:52	0	90961	63672.7	twitter
RT @BarackObama: The attacks on tourists and Easter worshippers in…	#Doctor Wh…	2019-04-21 15:39:10	0	89249	62474.3	twitter
RT @ashleymcnamara: My kid is gaming with friends so I brought him…	#My Friends	2019-04-22 15:08:02	0	74680	52276.0	twitter
RT @JamesJSChai: 30 years after Grave of the Fireflies was released,…	#Grave of th…	2019-04-22 11:14:54	0	39355	27548.5	twitter
RT @_Lone_Ranger21: Jon: My name, My Real name is Aegon Targar…	#Psycho	2019-04-22 15:07:06	0	30260	21182.0	twitter
RT @Milliennial: Since abortion is outlawed in 5 states (I believe) at firs…	#In a Heartb…	2019-04-22 15:07:56	0	28573	20001.1	twitter
RT @otome_kagura: 【キャンペーン11日目】事前登録受付中! #ザン…	#Ikiru	2019-04-22 14:41:02	0	25216	17651.2	twitter
RT @LogitechG: Happy Easter! 🐣 We're giving you the chance to win…	#The Dark K…	2019-04-22 15:03:19	0	20960	14672.0	twitter
RT @news_line_me: \ その場で当たる / フォロー&RTで「サー…	#Ikiru	2019-04-22 14:41:27	0	18361	12852.7	twitter
RT @Eyeb0nez_: I love my artists friends https://t.co/H3chacP9up	#My Friends	2019-04-22 15:08:04	0	17981	12586.7	twitter
RT @tonyhawk: Guy approaches me while standing in line at coffee sh…	#Your Name.	2019-04-22 15:07:08	0	17257	12079.9	twitter
RT @ikezaki_YEAH: 「KING OF PRISM by PrettyRhythm」応援して…	#Ikiru	2019-04-22 13:44:44	0	15937	11155.9	twitter
RT @torikameinu: 実は茶道師範でもあるルー大柴が、師範名の大柴宗…	#Ikiru	2019-04-22 14:54:05	0	13325	9327.5	twitter
RT @Palledo_namai: キック最高齢。週1ペースで約三ヶ月。はじめた…	#Harakiri	2019-04-22 13:19:56	0	13008	9105.6	twitter
RT @jross_x: Some advice from my experiences working in the ER: A…	#Your Name.	2019-04-22 15:07:25	0	12603	8822.1	twitter

Use the average viral from different data source to evaluate my post viral:

select avg(viral) as avgviral,datasource from viraltable group by datasource;

Output:

avgviral	datasource
766.98093	twitter
32.92667	instagram
4.15000	youtube

- 3)What posts are likely to be interesting to me?

select post from post where hashtag like "%Your name%";

Output:

post
@Mochispromise I heard search suggestion ban is fine.. probably mea…
There is freedom in Your name
RT @hiplikejibooty: IMAGINE BEING ABLE TO SEE BTS RIGHT NO…
RT @ryanitlab: Julián Castro was bashing Bernie by name weeks ago,…
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv…
Dear Ppl Who Only Come to My Stats to Disagree: Why Do You Do Th…
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv…
You guys are missing out im doing 5 20$ #giveaways #GiveawayAlert…
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv…
Here's a fun game your teen name is your least favorite soda for your fi…
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv…
@juacamole Hi, this is LaShelia. I'd be glad to check on your FedEx sh…
RT @fermin_marilou: Heavenly Father, we lift up rayers for protection…
RT @jross_x: Some advice from my experiences working in the ER: A…

- 4)What posts are like mine?(Subqueries)
Here, we are looking for posts about the movie we're interested in and also checking if a particular user is also talking about the movie we're interested in.

select post from post where hashtag in (select hashtag from post,user where post.userid=user.userid and username like "sam"

and post.post like '%Your Name%');

Output:

post	
@Mochispromise I heard search suggestion ban is fine.. probably mea...	
There is freedom in Your name	
RT @hiplikejibooty: IMAGINE BEING ABLE TO SEE BTS RIGHT NO...	
RT @ryanitlab: Julián Castro was bashing Bernie by name weeks ago,...	
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv...	
Dear Ppl Who Only Come to My Stats to Disagree: Why Do You Do Th...	
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv...	
You guys are missing out im doing 5 20$ #giveaways #GiveawayAlert...	
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv...	
Here's a fun game your teen name is your least favorite soda for your fi	

- 5)What users post like me?(Join)
  In this query, we're extracting all the username's and posts of the the user who are using hashtag's of the movie we like to see if the user post's like us.

  select p.post,p.hashtag,u.username

  from post p join user u on u.userid = p.userid

  where hashtag like "%Your name%";

  Output:

post	hashtag	username
@Mochispromise I heard search suggestion ban is fine.. probably mea...	#Your Name.	Cheezy~
There is freedom in Your name	#Your Name.	zild
RT @hiplikejibooty: IMAGINE BEING ABLE TO SEE BTS RIGHT NO...	#Your Name.	ᴶᴴᴵᴺᴵᴺᴳ ᴮᴿᴵᴳᴴᵀ...[BOYWITHLUV]
RT @ryanitlab: Julián Castro was bashing Bernie by name weeks ago,...	#Your Name.	Declan McElroy
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv...	#Your Name.	Alexandra Jean
Dear Ppl Who Only Come to My Stats to Disagree: Why Do You Do Th...	#Your Name.	Nana Jovaun Kwame
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv...	#Your Name.	celine
You guys are missing out im doing 5 20$ #giveaways #GiveawayAlert...	#Your Name.	Right
RT @RidiculousDak: Corals aren't plants they're animals. Imagine carv...	#Your Name.	Nisrin Wafer

- 6)Who should I be following?(Aggregation)
  The users who are popular are usually the users who a person should be following. Here, we're finding the most popular by taking the popularity as popularity = (0.2*likes + 0.2* comments + 0.6* Followers). After ordering them in descending order, the most popular users are found, and they are the one's a user is most likely to follow.

  select distinct u.username,p.post, (0.2* p.likes +0.2*p.comments + 0.6*u.follower) As popularity, p.datasource

  from post p join user u on p.userid = u.userid

order by datasource,popularity desc;

Output:

username	post	popularity	datasource
yrf	Raj and Simran met as strangers but redefined romance forever. ❤️ ...	613064.4	instagram
indozonemovie	#TheShawshankRedemption · · · Yuk, kirim kutipan movie favoritm...	367754.8	instagram
historyofthebatman	Evening Gothamites! Since we highlighted the 2011 award winning #vi...	171045.0	instagram
historyofthebatman	Evening Gothamites! I hope you all are having a purrr-fect #Caturday!...	171044.4	instagram
ghibli.movies	🎦: Spirited Away — When was the last time you've watched Spirited...	148930.0	instagram
ghibli.movies	🔥 Limited edition - not found in stores! 👉 Get yours in the link of our...	145784.6	instagram
kinedu	Why moms get NOTHING DONE... 😩 🎥@storyofthislife	145613.0	instagram
looperhq	Enlijah Wood and Andy Serkis filming one of their scenes in The Lord o...	109006.6	instagram
robertdenirodaily	Happy 72nd birthday to the great James Woods! 🎉🎊 ······ 📸 SWIP...	84795.2	instagram
amzingfantasy	***obvious reference*** - According to the Russo Brothers, Endgame w...	59715.4	instagram

- o 7)What topics are trending in my domian?
  We can find what hashtags are trending in our domain by taking the count of the hashtags and then displaying the list of hashtags in our domain

  select hashtag, count(*) as trending from post group by hashtag order by trending desc;

  Output:

hashtag	trending
	980
#GoodFellas	141
#Ikiru	133
#Psycho	127
#Harakiri	113
#Mommy	102
#City of God	100
#A Dog's Will	100
#Doctor Who: The Day of the Doctor	100
#Sherlock: The Final Problem	100

- o 8)What keywords/hashtags should I add to my post?(Group By, Order By,Having)
  Here, we will find the sum of popularity from the popularity table we created earlier that and display the hashtags that are most popular. To get the most trending hashtags we also need to consider the time attribute as we want the latest hashtag's so we have divided the popularity with the datediff function to get the latest post.

  select sum(popularity) as popularity,hashtag

from (select hashtag ,datasource,(0.5*likes+0.5*comments)/datediff('2019-04-23',time)as popularity from post) AS popularity_date

group by hashtag

having popularity > 10000

order by popularity DESC;

Output:

popularity	hashtag
1582405.50000	#Your Name.
118826.00000	#My Friends
92954.00000	#The Handmaiden
73589.00000	#In a Heartbeat
58625.00000	#A Dog's Will
53901.00000	#Life Is Beautiful
43420.72839	#Ikiru
27203.25000	#Grave of the Fireflies
24147.00000	#Doctor Who: The Day of the Doctor
20953.25000	#Psycho
18610.50000	#Seven Samurai
17995.00000	#The Dark Knight
12150.00000	#Mommy

o 9)Should I follow someone back?
   We are more likely to follow someone back who have pots similar to our interest and has a good amount of followers.

   select distinct u.username, u.follower

   from user u,post p

   where p.post like "%Harry%" and u.follower > 10000

   order by follower desc limit 10;

   Output:

username	follower	
The New York Times	43275734	
FBE	18844806	
IGN	11931562	
Looper	4217279	
SyrebralVibes	3374280	
Dorkly	2893849	
Wisecrack	2609040	
ACTUALLY HAPPENED	2071512	
GQ Magazine	1298093	
CinemaWins	1152121	

6.Use cases:

o 1)Find movies whose Average Rating/Vote_Average is above 7.5

Through this query we're finding all the movies that have an average rating above 7.5. We're using the > (greater than) operator in the where clause to achieve this.

- Code Snippet:

/* 1) Find movies whose Average Ratings is above 7.5 */

select movies.id, movies.title, movies.vote_average,release_date

from movies

where vote_average > 7.5;

- Output:

id	title	vote_average	release_date
19404	Dilwale Dulhania Le Jayenge	9	1995-10-20
278	The Shawshank Redemption	8.7	1994-09-23
238	The Godfather	8.6	1972-03-14
372058	Your Name.	8.6	2016-08-26
424	Schindler's List	8.5	1993-12-15
240	The Godfather: Part II	8.5	1974-12-20
129	Spirited Away	8.5	2001-07-20
497	The Green Mile	8.4	1999-12-10
637	Life Is Beautiful	8.4	1997-12-20
680	Pulp Fiction	8.4	1994-09-10

o 2)Find movies where the directors are female directors

Through this query, we are trying to find all the directors that are female directors. We will join the movies table with the movies_director table and then perform another join on the movies_director table with the directors table and then filter out the rows that have 'Female' as gender in the where clause.

- Code Snippet:

/* 2) Find movies where the directors are female directors */

select movies.id, movies.title, movies.vote_average, movies.vote_count,directors.ID, directors.Name,directors.Gender

from(movies

INNER JOIN movies_directors on movies.id = movies_directors.movieID

INNER JOIN directors on movies_directors.directorID = directors.ID)

where Gender = 'Female';

▪ Output:

id	title	vote_average	vote_count	ID	Name	Gender
339877	Loving Vincent	8.3	1156	1003237	Dorota Kobiela	Female
603	The Matrix	8.1	14035	9340	Lana Wachowski	Female
603	The Matrix	8.1	14035	9339	Lilly Wachowski	Female
310569	The Second Mother	8	340	550280	Anna Muylaert	Female
336804	Mustang	8	714	1128442	Deniz Gamze Ergüven	Female
435129	The Breadwinner	7.9	262	96677	Nora Twomey	Female
2011	Persepolis	7.9	887	20659	Marjane Satrapi	Female
71157	Polisse	7.8	641	64210	Maïwenn	Female
414425	Mudbound	7.6	574	472630	Dee Rees	Female
773	Little Miss Sunshine	7.6	3379	16060	Valerie Faris	Female

○ 3)Find all the actors whose place of birth is Japan.

In this query, we will find all the actors who are born in Japan. We will perform an inner join on the movies table with the movies_actor table and the perform another join on the actors table with the movie_actors table and then filter out the rows that have words like 'Japan' in the Place of Birth attribute in the where clause.

▪ Code Snippet:

/* 3) Find all the actors whose place of birth is Japan */

select movies.id, movies.title, actors.ID, actors.Name,actors.Place_of_Birth

from(movies

INNER JOIN movies_actors on movies.id = movies_actors.movieID

INNER JOIN actors on movies_actors.actorID = actors.ID)

where Place_of_Birth like '%Japan%';

▪ Output:

id	title	ID	Name	Place_of_Birth
372058	Your Name.	225730	Ryunosuke Kamiki	Saitama, Japan
372058	Your Name.	1369100	Mone Kamishiraishi	Kagoshima, Kagoshima Prefecture, Japan
372058	Your Name.	87637	Masami Nagasawa	Iwata, Shizuoka Prefecture, Japan
372058	Your Name.	91288	Etsuko Ichihara	Chiba, Chiba Prefecture, Japan
372058	Your Name.	1668610	Ryou Narita	Saitama Prefecture, Japan
129	Spirited Away	19588	Miyu Irino	Tokyo, Japan
129	Spirited Away	19589	Mari Natsuki	Tokyo - Japan
129	Spirited Away	119243	Bunta Sugawara	Sendai, Miyagi, Japan
12477	Grave of the Fireflies	72413	Tsutomu Tatsumi	Japan
12477	Grave of the Fireflies	72414	Ayano Shiraishi	Japan

o 4) Retrieve all the posts where hashtag is 'Psycho' from Instagram.

- Code Snippet:

/* 4) Retrieve all the posts where hashtag is 'Psycho' from Instagram */

select post.postid,post.post, post.hashtag, post.time, user.username,user.userid, post.datasource

from(post

inner join user on post.userid = user.userID)

where post.hashtag like '%Psycho%' and post.datasource = 'Instagram';

- Output:

postid	post	hashtag	time	username	userid	datasou
4440	👀 #funnymemes #edgy #dankmemes #offe...	#Psycho	2019-04-21 03:37:18	iam.satanjr	3955696313	instagram
4441	#psycho #fightme #bitch #dmme #mood #mus...	#Psycho	2019-04-21 03:36:36	confess_io_hub	12495191063	instagram
4442	#latergram #selfie #abe #bigeyes #alien #iph...	#Psycho	2019-04-21 03:36:19	mychl19	5694963846	instagram
4443	#memes #meme #momo #momos #armemes #...	#Psycho	2019-04-21 03:35:45	armemes_chistes	12722492981	instagram
4444	Sneak peak of psycho what y'all think doe 🤨 #...	#Psycho	2019-04-21 03:35:26	sweetz__monroe	221909806	instagram
4445	Evening vibe #naturephotography #nature #i...	#Psycho	2019-04-21 03:30:44	moon_catcher_sweetie	1968290338	instagram
4446	Futuristic Ninja #edit #psycho #neon #anime #...	#Psycho	2019-04-21 03:29:01	godly_otaku_soul	2100604899	instagram
4447	— money dont help with pain, where the drug...	#Psycho	2019-04-21 03:26:18	ath3ist	12303234079	instagram
4450	— insane	#Psycho	2019-04-21 03:22:29	ath3ist	12303234079	instagram

o 5) Retrieve all the posts where hashtag is 'Psycho' and the user followers is greater than 1000.

- Code Snippet:

/* 5) Retrieve all the posts where hashtag is 'Psycho' and the user followers is greater than 1000*/

select post.postid,post.post, post.hashtag, post.time, user.username,user.userid, user.follower, post.datasource

from(post

inner join user on post.userid = user.userID)

where post.hashtag like '%Psycho' and user.follower > 1000 order by user.follower desc;

- Output:

hashtag	time	username	userid	follower	datasource
_tous ... #Psycho	2019-04-22 15:04:26	🎏🔱 DP de la BAROLLIERE matricule 130 FR 🔱	2464135300	19028	twitter
⋺metin... #Psycho	2019-04-22 15:06:07	CİHAN #GS 74TR	4744788023	16853	twitter
⋺metin... #Psycho	2019-04-22 15:04:58	CİHAN #GS 74TR	4744788023	16853	twitter
⋺metin... #Psycho	2019-04-22 15:03:35	CİHAN #GS 74TR	4744788023	16853	twitter
Gangst... #Psycho	2019-04-21 02:18:26	cuzzaloink	195238810	9551	instagram
⋺metin... #Psycho	2019-04-22 15:03:51	Arda Ergün❤	1037848566419677186	8886	twitter
Gangst... #Psycho	2019-04-21 02:26:36	cripileaks	7845686006	7089	instagram
please... #Psycho	2019-04-21 02:47:14	vintageshitposting	6719736210	6522	instagram
snark #Psycho	2019-04-21 03:08:43	h_g_m_corner	8272060746	6309	instagram

- 6) Find the cast of movies "Goodfellas" and "The Godfather"
  - Code Snippet:

/* 6) Find the cast of movies "Goodfellas" and "The Godfather" */

select movies.title, actors.Name

from (movies

INNER JOIN movies_actors on movies.id = movies_actors.movieID

INNER JOIN actors on movies_actors.actorID = actors.ID)

where movies.title = 'Goodfellas' or movies.title = 'The Godfather';

  - Output:

title	Name
The Godfather	Marlon Brando
The Godfather	Al Pacino
The Godfather	James Caan
The Godfather	Robert Duvall
GoodFellas	Robert De Niro
GoodFellas	Joe Pesci
GoodFellas	Ray Liotta
GoodFellas	Paul Sorvino

- 7)Find the users whose followers number is greater than 1000

  return a list of users in different social media with datasource, follwers, userid and username

  - Code Snippet:

Select * from user where follower between 500000 and 5000000

and datasource = "twitter" order by follower desc limit 20;

- Output:

userid	username	follower	datasource
▶ 21701757	GQ Magazine	1298093	twitter
1162687218	APC Nigeria	947695	twitter
34245009	India TV	675404	twitter
31632905	mid-day	638762	twitter
2597554885	SpotboyE	516767	twitter

o 8) Find all the post with user whoser follower number are between 500 and 1000

reutrn a list of post, username,follower,likes and create time.

- Code Snippet:

select p.post,p.time,p.likes,u.username,u.follower, p.datasource

from post p join user u

on p.userID = u.userid

where u.follower between 500 and 1000;

- Output:

post	time	likes	username	follower	datasource
Kuch Kuch hota hai Khabi Kushi Khabi Gham Mujhse Dosti Karoge! D...	2019-04-22 07:44:42	2	zahraa 🍂	648	twitter
Voice of Lata Mangeshkar Ji in the background throughout the films in...	2019-04-21 16:16:38	0	AMRISHU	854	twitter
RT @Himayyy: Kabhi Khushi Kabhie Gham Dilwale Dulhania Le Jayen...	2019-04-20 07:07:08	0	make ur momma sad type.	858	twitter
RT @Himayyy: Kabhi Khushi Kabhie Gham Dilwale Dulhania Le Jayen...	2019-04-20 06:58:03	0	Kay🦋	823	twitter
Kabhi Khushi Kabhie Gham Dilwale Dulhania Le Jayenge Mohabbatei...	2019-04-20 06:29:48	26	Himali	544	twitter
RT @fvrx_: @SherylRiar omg this place looks like the scene in Dilwale...	2019-04-19 22:16:05	0	Sheryl Riar	859	twitter
RT @Koimoi: .@iamsrk & @AnupamPKher get into a Twitter bant...	2019-04-19 05:46:35	0	SOHAIL ❤️	754	twitter
RT @Koimoi: .@iamsrk & @AnupamPKher get into a Twitter bant...	2019-04-19 05:07:44	0	Nirmal Srkian 🇮🇳 Srkian follow 💯 follow back	542	twitter
@llqra_k Dilwale dulhania le jayenge	2019-04-18 20:58:29	0	Hazza did a Mazza 🌝	869	twitter
RT @Koimoi: .@iamsrk & @AnupamPKher get into a Twitter bant...	2019-04-18 16:04:05	0	rhea	571	twitter
RT @indiatvnews: . @AnupamPKher and @iamsrk have worked toget...	2019-04-18 11:36:11	0	ĞÅûŘÄV	548	twitter
RT @indiatvnews: . @AnupamPKher and @iamsrk have worked toget...	2019-04-18 10:44:47	0	Faisal Khan	895	twitter
Veteran actor #Anupam_Kher and superstar Shah Rukh Khan's jest ov...	2019-04-18 09:21:57	0	Suryaa Telugu News	974	twitter
@shinhwachj1998 Izlemediysen eğer Dilwale dulhania le jayenge, Jab...	2019-04-16 14:04:24	1	Ece ama ne demek ne belli	979	twitter
RT @lucamendo: Si quieren ver una buena peli en Netflix recomiendo...	2019-04-22 14:09:59	0	Rocío Casafus ♡	950	twitter

o 9) Find the first 30 post about movie Avatar with most likes and comments.

return a list of post , username,follower, datasource, likes and comments

- Code Snippet:

select p.post, p.datasource,p.likes,u.username,u.follower

from post p join user u

on p.userID = u.userid

where p.post like "%Avatar%"

order by p.likes desc;

- Output:

post	datasource	likes	username	follower
Day 2 of Sakura Con pt 1- I'm havin a blast!! Met some sick ass cospla...	instagram	29	tardisfied	198
Photos of Keith. #anime #onepiece #attackontitan #narutoshippuden #...	instagram	26	alex_anil_s	222
Final Day of #FanXSaltLake! . . . #FanXSpring19 #SaltLakeComicCon...	instagram	14	_theblondewitch_	307
The Shawshank Redemption (1994) vs. Avatar (2009) https://t.co/D024...	twitter	0	Virgo Phase (2017-32)	251
@Kylian_____ Avatar me la suda pero ponerla tan abajo huele y lo d...	twitter	0	G.M.	12

o 10) Find the movie with company in US.

return a list of movies with movie name,release date,vote,company name and company location

- Code Snippet:

Select m.title,m.release_date,m.vote_average,c.name,c.origin_country

from movies m join movies_companies mc on m.movie_id = mc.movieID join companise c on mc.companyID = c.id

where c.origin_country = "US";

- Output:

title	release_date	vote_average	name	origin_country
The Shawshank Redemption	1994-09-23	8.7	Castle Rock Entertainment	US
The Shawshank Redemption	1994-09-23	8.7	Warner Bros. Pictures	US
The Godfather	1972-03-14	8.6	Paramount	US
The Godfather	1972-03-14	8.6	Alfran Productions	US
Schindler's List	1993-12-15	8.5	Universal Pictures	US
Schindler's List	1993-12-15	8.5	Amblin Entertainment	US
The Godfather: Part II	1974-12-20	8.5	Paramount	US
The Godfather: Part II	1974-12-20	8.5	The Coppola Company	US
The Green Mile	1999-12-10	8.4	Castle Rock Entertainment	US
The Green Mile	1999-12-10	8.4	Darkwoods Productions	US
The Green Mile	1999-12-10	8.4	Warner Bros. Pictures	US
Life Is Beautiful	1997-12-20	8.4	Miramax	US
Pulp Fiction	1994-09-10	8.4	Miramax	US
Pulp Fiction	1994-09-10	8.4	A Band Apart	US
Green Book	2018-11-16	8.3	Participant Media	US

o   11) Great women actor

Find all the women actors who was born between 1955 and 1980, she acted movie with vote lager than 8.4 */

▪   Code Snippet:

select m.title,m.vote_average, a.*from actors a

left join movies_actors ma on a.ID=ma.actorID

left join movies m on ma.movieID = m.movie_id

where Birthday between "1955-01-01" and "1980-01-01"

and Gender = "Female" and title is not null

and vote_average > 8.4;

▪   Output:

title	vote_average	ID	Name	Gender	Birthday	Place of Birth
Dilwale Dulhania Le Jayenge	9	55061	Kajol	Female	1974-08-05	Mumbai, Maharashtra, India
Schindler's List	8.5	6692	Caroline Goodall	Female	1959-11-13	London, England, UK

o   12)find movies that released after 2000
    ▪   Code Snippet:

select id,title,release_date from movies where release_date between 2000 and 2020;

▪   Output:

id	title	release_date
372058	Your Name.	2016-08-26
129	Spirited Away	2001-07-20
40096	A Dog's Will	2000-09-15
155	The Dark Knight	2008-07-16
122	The Lord of the Rings: The Return of the King	2003-12-01
244786	Whiplash	2014-10-10
484468	The Wolf's Call	2019-02-20
517814	Capernaum	2018-10-06
432517	Sherlock: The Final Problem	2017-01-15
598	City of God	2002-02-05

o   13)find the post on the movies of company "Sony Pictures"
    ▪   Code Snippet:

select title as moviename,post,likes,comments

from post, movie_post,movies

where post.postid=movie_post.postid and movie_post.movieid=movies.id

and movie_post.movieid in

   (select distinct movieid

   from companise c,movies_companies mc

where c.id=mc.companyid and c.name="Paramount");

- Output:

	moviename	post	likes	comments
▶	The Godfather	RT @PaulineofDreams: Happy birthday to The ...	0	2
	The Godfather	For the Godfather..	0	0
	The Godfather	@Ayourb Happy birthday, Godfather. May Alla...	0	0
	The Godfather	RT @longhairwiteboy: Jennifer's Body better th...	0	325
	The Godfather	@CP_BaByBaBy @Freddykadiddle1 @CP_BaBy...	0	0
	The Godfather	☺ The Godfather house is on the market! https...	0	0
	The Godfather	"Dostluk ve para , zeytinyağı ve su gibidir." Th...	11	0
	The Godfather	He should ask his aspiring godfather with the sk...	0	0
	The Godfather	@BillKristol After The Godfather, everyone wan...	0	0
	The Godfather	'The mannerisms and style of the Marlon Brando...	0	0

- o 14)Get the most like post in the past 72hrs from "twitter"
  - Code Snippet:

select post,userid,hashtag,likes

from post

where datasource="twitter" and time between "2019-04-20" AND "2019-04-23"

order by likes desc

limit 10;

- Output:

	post	userid	hashtag	likes
▶	DEAD POETS SOCIETY (1989) Cinematography...	2172922045	#Dead Poets Society	336
	The Handmaiden (2016) is one of the most visu...	893627764108120064	#The Handmaiden	192
	I will take a doctors opinion regarding medicine ...	577671042	#Doctor Who: The Day of the Doctor	148
	Allow me to pile on - I love GOT - just shot a 90 ...	164354823	#The Godfather: Part II	147
	A faith shaped by Scripture will cause a Christia...	20794662	#City of God	137
	Mom wants to watch the green mile with me an...	409182259	#The Green Mile	74
	"I once saw a movie in which only police and mili...	3613555641	#Schindler's List	72
	This Forrest Gump easter egg... #MortalKomba...	3403061291	#Forrest Gump	70
	The character photo shoot of the Tenth Doctor ...	21329773	#Doctor Who: The Day of the Doctor	67
	The Green Mile is one of the greatest films ever ...	10192126166630177792	#The Green Mile	65

- 15)Get the company location which produce the movie which has high vote average(>8.0) and high vote count(>10,000)
  - Code Snippet:

select name,origin_country

from companise,movies,movies_companies

where companise.id=movies_companies.companyid and movies_companies.movieid=movies.id and vote_average>8 and vote_count>10000;

  - Output:

name	origin_country
Castle Rock Entertainment	US
Warner Bros. Pictures	US
Warner Bros. Pictures	US
Paramount	US
Miramax	US
A Band Apart	US
Jersey Films	
DC Entertainment	US
DC Comics	US
Legendary Entertainment	US

- 16)Get the user infomation whose post has more comments than likes
  - Code Snippet:

select distinct user.userid,username,follower,user.datasource

from post,user

where post.userid=user.userid and comments>likes

  - Output:

userid	username	follower	datasource
1016626898686894080	Neha≡	137	twitter
8471299671546 14273	Safiyyah	168	twitter
767345859193401344	Dinisha ✿	172	twitter
1656288236	make ur momma sad type.	858	twitter
431806491	Kay ♨	823	twitter
635388531	Come at me bro	185	twitter
608005118	Darshan Natha	344	twitter
45070259	Sheryl Riar	859	twitter
2991077426	Ronak Saraf	138	twitter
2750445959	SOHAIL ♥	754	twitter

7. **Stored Procedures:**

- 1) Show all user posts with movie title 'Mononoke'
  - Code Snippet:

CREATE DEFINER=`root`@`localhost` PROCEDURE `movie_post`()

BEGIN

select distinct movies.id, movies.title, post.post, post.likes, post.userID, post.comments

from (movies

inner join movie_post on movies.id = movie_post.movieid

inner join post on movie_post.datasource = post.datasource)

where movies.title like '%Mononoke%';

END

  - Output:

id	title	post	likes	userID	comments
128	Princess Mononoke	Princess Mononoke kesinlikle https://t.co/sJm1x...	1	407695428	0
128	Princess Mononoke	@_King_Widow They are Kodama from the stud...	1	1349016367	0
128	Princess Mononoke	RT @EliLuca1: Just saying The Princess Monono...	0	911883688689885186	8
128	Princess Mononoke	RT @AnimeSxnpai: 8-bit Princess Mononoke htt...	0	1971297277	2144
128	Princess Mononoke	Somebody showed me an auction listing of som...	1	8667772	0
128	Princess Mononoke	@AH_Michael Princess Mononoke is my fav stud...	0	435396593	0
128	Princess Mononoke	@munfairy Noragami, fruits basket, kill la kill, Ak...	0	937695094307569666	0
128	Princess Mononoke	RT @EliLuca1: Just saying The Princess Monono...	0	206291029	8

- 2) Show attributes of movies and attributes of social media whose vote_average is greater than 8.8.
  - Code Snippet:

CREATE DEFINER=`root`@`localhost` PROCEDURE `movie_rating`()

BEGIN

select movies.id, movies.title, movies.vote_average, post.post, post.likes, post.time

from (movies

inner join movie_post on movies.id = movie_post.movieid

inner join post on movie_post.datasource = post.datasource)

where movies.vote_average > 8.8;

END

- Output:

id	title	vote_average	post	likes	time
19404	Dilwale Dulhania Le Jayenge	9	Of course my @iamsrk heart would not be living...	0	2019-04-22 15:05:08
19404	Dilwale Dulhania Le Jayenge	9	Of course my @iamsrk heart would not be living...	0	2019-04-22 15:05:08
19404	Dilwale Dulhania Le Jayenge	9	Of course my @iamsrk heart would not be living...	0	2019-04-22 15:05:08
19404	Dilwale Dulhania Le Jayenge	9	Of course my @iamsrk heart would not be living...	0	2019-04-22 15:05:08
19404	Dilwale Dulhania Le Jayenge	9	Of course my @iamsrk heart would not be living...	0	2019-04-22 15:05:08
19404	Dilwale Dulhania Le Jayenge	9	Of course my @iamsrk heart would not be living...	0	2019-04-22 15:05:08
19404	Dilwale Dulhania Le Jayenge	9	Of course my @iamsrk heart would not be living...	0	2019-04-22 15:05:08
19404	Dilwale Dulhania Le Jayenge	9	Of course my @iamsrk heart would not be living...	0	2019-04-22 15:05:08

- o 3)Get Post By Name

By using this procedure, user can get all of the post of the movie which they pass movie name as parameter into the procedure.

- Code Snippet:

CREATE DEFINER=`root`@`localhost` PROCEDURE `getPostByName`(in moviename text)

BEGIN

      SELECT * FROM moviepost where title=moviename;

END

	title	post	likes	comments
▶	Your Name.	@Mochispromise I heard search suggestion ban...	0	0
	Your Name.	There is freedom in Your name	0	0
	Your Name.	RT @hiplikejibooty: IMAGINE BEING ABLE TO SE...	0	668
	Your Name.	RT @ryanitlab: Julián Castro was bashing Berni...	0	104
	Your Name.	RT @RidiculousDak: Corals aren't plants they're...	0	147700
	Your Name.	Dear Ppl Who Only Come to My Stats to Disagre...	0	0
	Your Name.	You guys are missing out im doing 5 20$ #givea...	0	0
	Your Name.	Here's a fun game your teen name is your least ...	0	0
	Your Name.	@juacamole Hi, this is LaShelia. I'd be glad to ch...	0	0
	Your Name.	RT @fermin_marilou: Heavenly Father, we lift u...	0	1

- o 4)Standardize Comments

Sp2 and Sp3 are used to get standardized likes and comments.

As we collected data from different social media social, the user amount varies from one media to another. So we need to standardize the likes and comments to make it consistent. The calculating method here is post.likes divided by the average likes from the post's data source.

- Code Snippet:

CREATE DEFINER=`root`@`localhost` PROCEDURE `standardizecomment`()

BEGIN

　　select postid, (comments)/`avg(comments)` as std from post,avgcomment where post.datasource=avgcomment.datasource having std>0 order by std desc ;

END

- Output:

postid	std
248	121.7537
288	121.7537
320	121.7537
321	121.7537
250	121.7537
266	121.7537
267	121.7537
275	121.7537
283	121.7537
291	121.7537

- 5)Standardize Likes
  - Code Snippet:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `standardizelike`()
```

BEGIN

　　select postid, (likes)/`avg(likes)` as std from post,avglikes where post.datasource=avglikes.datasource having std>0 order by std desc ;

END

- Output:

postid	std
3650	278.7688
4625	173.3937
3128	159.2964
2640	122.7910
51	122.7910
502	121.9613
2936	113.6646
3753	103.9783
601	61.3955
451	59.7362

- 6)Get post popularity by movie name

- Code Snippet:

```
->CREATE DEFINER=`root`@`localhost` PROCEDURE `get_popularity`(name text)

BEGIN

select post, hashtag,(likes + comments) as popularity from post

where hashtag = name

order by popularity;

END

->call 6210Project.get_popularity('forrest Gump');
```

- Output:

post	hashtag	popularity
I wish I had the confidence he did XD spot on hilarious	forrest gump	62
I wish I had the confidence he did XD spot on hilarious	forrest gump	62
Normal people: You should talk to your crush directly in order to get he…	forrest gump	46
Normal people: You should talk to your crush directly in order to get he…	forrest gump	46
I love this guy	forrest gump	42
I love this guy	forrest gump	42
This movie is so sad. The guy is so hilarious	forrest gump	41
This movie is so sad. The guy is so hilarious	forrest gump	41
She is so beautiful. Reminds me a bit of Irene Dunne…	forrest gump	33
She is so beautiful. Reminds me a bit of Irene Dunne…	forrest gump	33

8.Functions:

- 1)Function to find the most liked social media post.
    - Code Snippet:

CREATE DEFINER=`root`@`localhost` FUNCTION `most_likes`(likes int(11)) RETURNS text CHARSET utf8mb4

```
    DETERMINISTIC

BEGIN

declare l text;

        if likes > 100 then

    set l = 'Most Likes';

    elseif likes < 100 then

    set l = 'Less Likes';

    END IF;
```

- Output:

userID	post	most_likes(likes)
690886590	🎬Movie vs. Location . 1984 / 2019 „Once upon...	Most Likes
5410941866	#woodystrode #blackicon #onceuponatimeina...	Less Likes
2259282749	C'era una volta in America(1984) diretto da Ser...	Most Likes
186463823	Once upon a time in America - Es war einmal in ...	Less Likes
3630039416	A dica dessa sexta é um dos filmes mais incrívei...	Most Likes
1636040808	Assistindo mais uma vez ! Pela primeira vez est...	Less Likes
10226162150	I never ask for help but I am desperately beggi...	Less Likes
5937649638	A Dog's Will [O Auto da Compadecida] (2000) - ...	Less Likes

- 2)Function to find the most popular and less popular movies
    - Code Snippet:

CREATE DEFINER=`root`@`localhost` FUNCTION `movie_popularity`(vote_count int(11)) RETURNS text CHARSET utf8mb4

```
    DETERMINISTIC

BEGIN

declare c text;
```

```
        IF vote_count > 1000 then

    set c = 'Popular';

    ELSEIF vote_count < 1000 then

    set c = 'Less Popular';

    END IF;

RETURN (c);

END
```

- Output:

id	title	movie_popularity(vote_count)
244786	Whiplash	Popular
914	The Great Dictator	Popular
429	The Good, the Bad and the...	Popular
18491	Neon Genesis Evangelion: ...	Less Popular
901	City Lights	Less Popular
484468	The Wolf's Call	Less Popular
517814	Capernaum	Less Popular
452522	Twin Peaks	Less Popular

- 3) Get hot Movie By Location
  - Code Snippet:

```
->CREATE DEFINER=`root`@`localhost` FUNCTION `getHotMovieByLocation`(location
text) RETURNS int(11)

    DETERMINISTIC

BEGIN

Declare mvid int;

    Select movieid into mvid from movies_companies m,companise c where
m.companyid=c.id and origin_country=location order by popularityCal(movieid) DESC
limit 1;

RETURN mvid;

END
```

- Output:

```
->select moviedb.getHotMovieByLocation('US');
```

```
moviedb.getHotMovieByLocation('US')
▶  637
```

- o 4)Calculate popularity

  This function can help calculate the average popularity by movie id.

  Calculating method: Average popularity of the posts(Each post popularity is (0.3*likes+0.7*comments)) for that movie

  ▪ Code Snippet:

  CREATE DEFINER=`root`@`localhost` FUNCTION `popularityCal`(movieidIn int) RETURNS float

     DETERMINISTIC

  BEGIN

  declare popularity float;

  declare p float;

       select avg(popularityEach) into popularity from (select (0.3*likes+0.7*comments) as popularityEach,movieid from post p,movie_post where p.postid=movie_post.postid) as ss where movieid=movieidIn group by movieid;

  return (popularity);

  End

  ▪ Output:


```
moviedb.popularityCal(19404)
▶  51.89516067504883
```

- o 5)Input a movie id ,return movie name
  ▪ Code Snippet:

  CREATE DEFINER=`root`@`localhost` FUNCTION `get_name`(idnum Integer) RETURNS varchar(30) CHARSET utf8mb4

  DETERMINISTIC

  BEGIN

  declare name VARCHAR(30);

  SELECT title into name FROM movies

WHERE movies.movie_id = idnum;

RETURN name;

END

- Output:

select 6210Project.get_name(278);

6210Project.get_name(278)
The Shawshank Redemption

- 6)Input a score, return a movie which vote is larger than this score
  - Code Snippet:

CREATE DEFINER=`root`@`localhost` FUNCTION `get_highvoted`(score float) RETURNS varchar(50) CHARSET utf8mb4

    DETERMINISTIC

BEGIN

declare name varchar(50);

SELECT title into name FROM movies

WHERE movies.vote_average > score limit 1;

RETURN name;

END

- Output:

select 6210Project.get_highvoted(7.0);

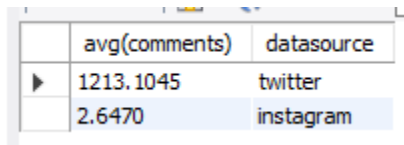| 6210Project.get_highvoted(7.0)	
Star Wars	

9.<u>Views:</u>

- o 1)By using this view, users can check the average comments count from different social media. Furthermore, users can use this view, compare it with the post comment and learn the comments level of that post.
  - Code Snippet:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `avgcomment` AS
    SELECT
        AVG(`post`.`comments`) AS `avg(comments)`,
        `post`.`datasource` AS `datasource`
    FROM
        `post`
        GROUP BY `post`.`datasource`
```

  - Output:

avg(comments)	datasource
1213.1045	twitter
2.6470	instagram

- o 2)By using this view, users can check the average likes count from different social media.
  - Code Snippet:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `avglikes` AS
    SELECT
```

AVG(`post`.`likes`) AS `avg(likes)`,

`post`.`datasource` AS `datasource`

FROM

`post`

GROUP BY `post`.`datasource`

- Output:

	avg(likes)	datasource
▶	1.2053	twitter
	103.5793	instagram

o 3)In this view, user can directly view the movie's names and their post content, likes and comments.
- Code Snippet:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `moviepost` AS
    SELECT DISTINCT
        `movies`.`title` AS `title`,
        `post`.`post` AS `post`,
        `post`.`likes` AS `likes`,
        `post`.`comments` AS `comments`
    FROM
        ((`movies`
        JOIN `movie_post`)
        JOIN `post`)
    WHERE
        ((`movie_post`.`movieid` = `movies`.`id`)
            AND (`movie_post`.`postid` = `post`.`postid`))
```

▪ Output:

title	post	likes	comments
Dilwale Dulhania Le Jayenge	Of course my @iamsrk heart would not be living...	0	0
Dilwale Dulhania Le Jayenge	Interesting #Facts about DDLJ . Aditya Chopr...	1	0
Dilwale Dulhania Le Jayenge	RT @AshikLatchman: 1. Parents wedding video ...	0	1
Dilwale Dulhania Le Jayenge	1. Parents wedding video (live and uncut) 2. Th...	3	1
Dilwale Dulhania Le Jayenge	Kuch Kuch hota hai Khabi Kushi Khabi Gham Muj...	2	0
Dilwale Dulhania Le Jayenge	Dilwale Dulhania Le Jayenge	Dialogue Status	...	0	0
Dilwale Dulhania Le Jayenge	Never did my eyebrows. Never will. The kajol in ...	0	0
Dilwale Dulhania Le Jayenge	📷 assyrianjalebi: Dilwale Dulhania Le Jayenge (...	0	0
Dilwale Dulhania Le Jayenge	Found my Bollywood field of yellow just like in Di...	5	0
Dilwale Dulhania Le Jayenge	Voice of Lata Mangeshkar Ji in the background t...	0	0

o 4)In this view, it combined the movie name and its hashtag on social media. User can use this to learn which tag is related to which movie, especially useful when the data source is from youtube.

▪ Code Snippet:

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `name-hashtag` AS

SELECT DISTINCT

`movies`.`title` AS `title`, `post`.`hashtag` AS `hashtag`

FROM

((`movies`

JOIN `movie_post`)

JOIN `post`)

WHERE

((`movie_post`.`movieid` = `movies`.`id`)

AND (`movie_post`.`postid` = `post`.`postid`))

▪ Output:

title	hashtag
Dilwale Dulhania Le Jayenge	#Dilwale Dulhania Le Jayenge
The Shawshank Redemption	#The Shawshank Redemption
The Godfather	#The Godfather
Your Name.	#Your Name.
Schindler's List	#Schindler's List
The Godfather: Part II	#The Godfather: Part II
Spirited Away	#Spirited Away
The Green Mile	#The Green Mile
Life Is Beautiful	#Life Is Beautiful
Twin Peaks	#Twin Peaks

o   5)In this view, user can view the vote_average and popularity calculated by social media likes and comments together. The popularityCal() is the defined function.
   ▪ Code Snippet:

CREATE

   ALGORITHM = UNDEFINED

   DEFINER = `root`@`localhost`

   SQL SECURITY DEFINER

VIEW `votevspopularity` AS

   SELECT

      `movies`.`id` AS `id`,

      `movies`.`title` AS `title`,

      POPULARITYCAL(`movies`.`id`) AS `popularity`,

      `movies`.`vote_average` AS `vote_average`

   FROM

      `movies`

      LIMIT 50

   ▪ Output:

	id	title	popularity	vote_average
▶	19404	Dilwale Dulhania Le Jayenge	51.8952	9
	278	The Shawshank Redemption	94.0511	8.7
	238	The Godfather	47.2977	8.6
	372058	Your Name.	16538.6	8.6
	424	Schindler's List	30.9321	8.5
	240	The Godfather: Part II	33.8948	8.5
	129	Spirited Away	84.0629	8.5
	497	The Green Mile	26.6616	8.4
	637	Life Is Beautiful	658.257	8.4
	680	Pulp Fiction	21.7394	8.4

- 6) View above average followers
  - Code Snippet:

CREATE

  ALGORITHM = UNDEFINED

  DEFINER = `root`@`localhost`

  SQL SECURITY DEFINER

VIEW `aboveavgfollowers` AS

  SELECT

    `user`.`userid` AS `userid`,

    `user`.`username` AS `username`,

    `user`.`follower` AS `follower`

  FROM

    `user`

  WHERE

    (`user`.`follower` > (SELECT

        AVG(`user`.`follower`)

      FROM

        `user`))

      ORDER BY `user`.`follower` DESC

  - Output:

userid	username	follower	
807095	The New York Times	43275734	
UC0v-tlzsn0QZwJnkiaUSJVQ	FBE	18844806	
UCKy1dAqELo0zrOtPkf0eTMw	IGN	11931562	
UCP1iRaFIS5EYjJBryFV9JPw	Looper	4217279	
UCi0LydWaEUy3Vx8flL29ebQ	SyrebralVibes	3374280	
UCHdos0HAIEhIMqUc9L3vh1w	Dorkly	2893849	
UC6-ymYjG0SU0jUWnWh9ZzEQ	Wisecrack	2609040	
UCGDV4-Gr3gyc-eGCgRtXknw	ACTUALLY HAPPENED	2071512	
21701757	GQ Magazine	1298093	
UCL8h3ri2WN_-IbviBIWtUcQ	CinemaWins	1152121	
432242744	yrf	1018160	
1162687218	APC Nigeria	947695	
UCk0UErv9b4Hn5ucNNjqD1UQ	kylelandry	730772	
34245009	India TV	675404	

- o 7)Aggregate ratings
  - ▪ Code Snippet:

CREATE

    ALGORITHM = UNDEFINED

    DEFINER = `root`@`localhost`

    SQL SECURITY DEFINER

VIEW `aggregate_ratings` AS

    SELECT

        AVG(`movies`.`vote_average`) AS `avg(movies.vote_average)`,

        AVG(`movies`.`vote_count`) AS `avg(movies.vote_count)`,

        MAX(`movies`.`vote_average`) AS `max(movies.vote_average)`,

        MIN(`movies`.`vote_average`) AS `min(movies.vote_average)`,

        MIN(`movies`.`vote_count`) AS `min(movies.vote_count)`,

        MAX(`movies`.`vote_count`) AS `max(movies.vote_count)`

    FROM

        `movies`

  - ▪ Output:

avg(movies.vote_average)	avg(movies.vote_count)	max(movies.vote_average)	min(movies.vote_average)	min(movies.vote_count)	max(movies.vote_count)
7.873209549071618	2299.1074	9	7.6	150	18852

- o 8) Influential user
  - Code Snippet:

  CREATE

      ALGORITHM = UNDEFINED

      DEFINER = `root`@`localhost`

      SQL SECURITY DEFINER

  VIEW `influencial_user` AS

      SELECT

          `user`.`userid` AS `userid`,

          `user`.`username` AS `username`,

          SUM(((`user`.`follower` + `post`.`likes`) + `post`.`comments`)) AS `totalinfluence`

      FROM

          (`user`

          JOIN `post` ON ((`user`.`userid` = `post`.`userID`)))

      GROUP BY `user`.`userid` , `user`.`username`

      ORDER BY `totalinfluence` DESC

  - Output:

userid	username	totalinfluence
807095	The New York Times	43275792
44129303	restaurant_babes2.0	1922121
21701757	GQ Magazine	1298106
8748704165	thegodfather_fangirl_6	1263888
8748494292	thegodfather_funny_6	1230864
8746508972	thegodfather_always23	1230224
297326313	historyofthebatman	1141718
432242744	yrf	1029002

- o 9) Viral table
  - Code Snippet:

      CREATE

```sql
ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `viraltable` AS

SELECT DISTINCT

    `post`.`post` AS `post`,

    `post`.`hashtag` AS `hashtag`,

    `post`.`time` AS `time`,

    `post`.`likes` AS `likes`,

    `post`.`comments` AS `comments`,

    ((0.3 * `post`.`likes`) + (0.7 * `post`.`comments`)) AS `viral`,

    `post`.`datasource` AS `datasource`

FROM

    `post`

ORDER BY `viral` DESC
```

- Output:

post	hashtag	time	likes	comments	viral	datasource
RT @RidiculousDak: Corals aren't plants they'r...	#Your Name.	2019-04-22 15:06:57	0	147700	103390.0	twitter
RT @RidiculousDak: Corals aren't plants they'r...	#Your Name.	2019-04-22 15:06:58	0	147700	103390.0	twitter
RT @RidiculousDak: Corals aren't plants they'r...	#Your Name.	2019-04-22 15:07:02	0	147700	103390.0	twitter
RT @RidiculousDak: Corals aren't plants they'r...	#Your Name.	2019-04-22 15:07:19	0	147700	103390.0	twitter
RT @RidiculousDak: Corals aren't plants they'r...	#Your Name.	2019-04-22 15:07:21	0	147700	103390.0	twitter
RT @RidiculousDak: Corals aren't plants they'r...	#Your Name.	2019-04-22 15:07:22	0	147700	103390.0	twitter
RT @RidiculousDak: Corals aren't plants they'r...	#Your Name.	2019-04-22 15:07:23	0	147700	103390.0	twitter
RT @RidiculousDak: Corals aren't plants they'r...	#Your Name.	2019-04-22 15:07:24	0	147700	103390.0	twitter

10. INDEXING

1) index hashtag

Select * from post where hashtag like '%YourName'

| 245 | 1; select * from post where hashtag li… | 50 row(s) returned | 0.0063 sec / 0.0071 s… |

Create Index hashtag ON post(hashtag(100))

| 247 | 1; select * from post where hashtag li… | 50 row(s) returned | 0.0052 sec / 0.0060… |

2) index follower

Select * from user where follower > 500

| 259 | 1; select * from user where follower >… | 495 row(s) returned | 0.0040 sec / 0.0028… |

create index follower on user(follower);

| 255 | 1; select * from user where follower >… | 2631 row(s) returned | 0.0013 sec / 0.0049… |

3) index vote

select * from movies where vote_average>8.5;

| 261 | 1; select * from movies where vote_av… | 4 row(s) returned | 0.00084 sec / 0.000… |

create index vote on movies(vote_average);

| 263 | 1; select * from movies where vote_av… | 4 row(s) returned | 0.00037 sec / 0.0000… |

11. Trigger

We attempted to create a trigger where if we try to delete a user, the post for this user will be deleted.

CREATE DEFINER=`root`@`localhost` TRIGGER `user_AFTER_DELETE` AFTER DELETE ON `user` FOR EACH ROW BEGIN

Delete from post

 where userID IN (select old.userid from deleted);

END

12.Approach to social media tagging:

Further to help us with the mess of social media tagging , we create some tables to store syntactic and semantic information about tagging manually.

- o Domain tags: Each post has one hashtag. We define all of the distinct hashtags as our domain tags.
- o Synonyms: We consider the tags that have the same letters but different uppercase and lowercase as synonyms and we do this table manually.
  Sample:

Tag	Synonyms
forrest gump	Forrest Gump(Award-Winning Work)
dark knight	The Dark Knight
The Godfather	The Godfather(film)
The Godfather	the godfather full movie
The Godfather	Godfather

- o Mis-spelling: Use mis-spelling words table from one public github domain.
  Sample:

athenian	athenean
imaginary	imagenary
continually	constinually
kleenex	klenex
stereotypes	sterotypes
medicine	medacine
atheist	athiest

- o Semantic information: Category the tags manually, including movie-name, movie-procedures, movie-sounds and so on.
  Sample:

Categories	Tags
movie name	forrest gump
movie songs	krimelide ca$$
movie actor	Edward Furlong
movie name	lotr the return of the kiing
producer	madman
movie name	Princess Mononoke
director	Hayao Miyazaki

13. Refrence:

- o Mis-spelling table:
  - https://github.com/brianhempel/fuzzy_tools
- o SQL:
  - https://dev.mysql.com/doc/
  - http://www.mysqltutorial.org/

  Scraping:
  - https://www.kaggle.com/tmdb/tmdb-movie-metadata
  - https://instaloader.github.io/index.html
  - https://docs.python.org/3.7/library/sqlite3.html
  - https://developers.google.com/youtube/v3/
  - https://developer.twitter.com/en/docs.html
- o UML:
  - https://www.draw.io/
- o Other
  - https://github.com/nikbearbrown/INFO_6210

14. Licence:

Copyright <2019> < Anisha Ganguly, Fan Ji, Zhouwei Wang>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.