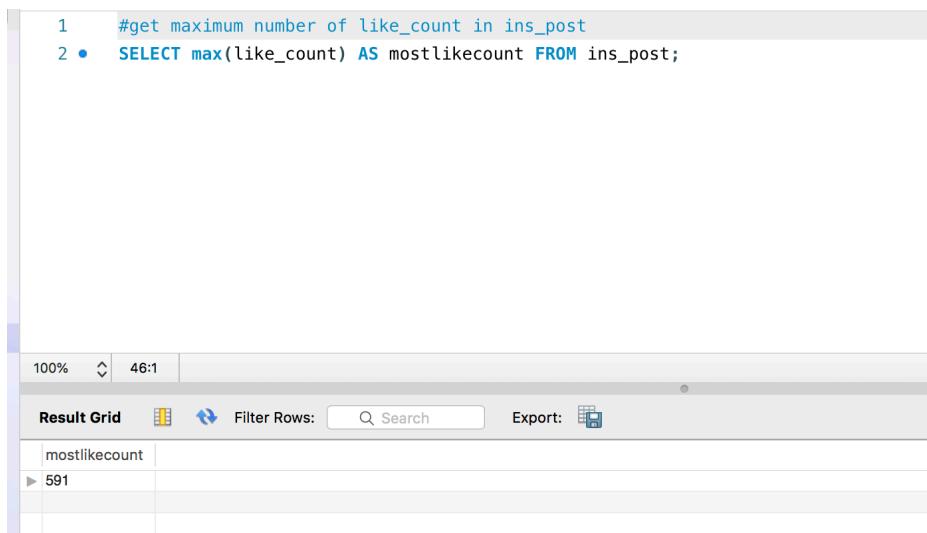


Use case

- 1.Find how many ‘likes’ can the most popular ins post got in our database (get maximum number of like_count in ins_post)



The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 #get maximum number of like_count in ins_post
2 • SELECT max(like_count) AS mostlikecount FROM ins_post;
```

The result grid shows one row with the value 591.

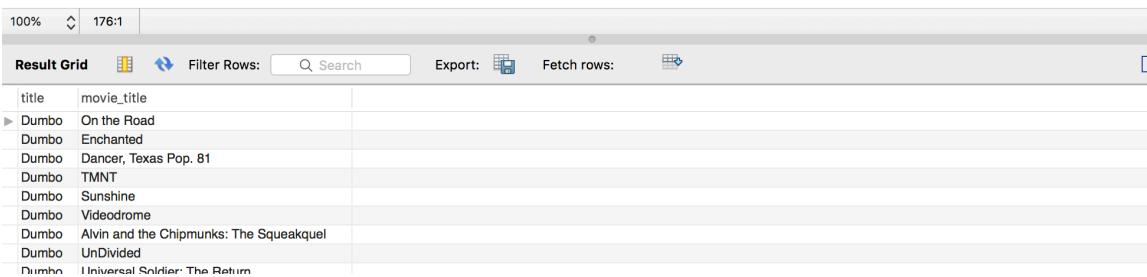
mostlikecount
591

- 2.Find all movies produced by a specific director:

(his/her new movie on the left, his/her old movie on the right)

```
SELECT new_movie.title,old_movie_final.movie_title FROM new_movie,old_movie_final
WHERE new_movie.director LIKE 'Tim Burton' OR old_movie_final.director_name LIKE 'Tim
Burton'
```

```
1 • movie,old_movie_final WHERE new_movie.director LIKE 'Tim Burton' OR old_movie_final.director_name LIKE 'Tim Burton'|
```



The screenshot shows the MySQL Workbench interface with a result grid. The query is:

```
1 movie,old_movie_final WHERE new_movie.director LIKE 'Tim Burton' OR old_movie_final.director_name LIKE 'Tim Burton'|
```

The result grid shows multiple rows of movie titles, all associated with the director 'Tim Burton'.

title	movie_title
Dumbo	On the Road
Dumbo	Enchanted
Dumbo	Dancer, Texas Pop. 81
Dumbo	TMNT
Dumbo	Sunshine
Dumbo	Videodrome
Dumbo	Alvin and the Chipmunks: The Squeakquel
Dumbo	UnDivided
Dumbo	Universal Soldier: The Return

3. get total number of like_count and comments_count

```
6     #get total number of like_count and comments_count
7 •  SELECT like_count,comments_count,(like_count+comments_count)
8     AS total_count FROM ins_post
```

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the code from step 3. The results grid displays the following data:

like_count	comments_count	total_count
41	1	42
41	1	42
41	1	42
82	9	91
82	9	91
82	9	91
82	9	91
82	9	91
82	9	91
82	9	91
82	9	91
82	9	91
82	9	91
82	9	91
44	3	47
44	3	47
44	3	47

4.get hashtags in ins_post that runtime 10%

```
1  #get hashtags in ins_post that runtime 10%
2 •  SELECT hashtags FROM ins_post
3  INNER JOIN new_movie ON post_id=new_movie_id
4  WHERE new_movie.runtime LIKE '10%'
```

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the code from step 4. The results grid displays the following data:

hashtags
▶ documentary
animals
youtuber
smallyoutuber
gaysofinstagram

5.#get movie_title that actor name start with Ta%

```
1 •   SELECT movie_title FROM old_movie_final WHERE main_actor LIKE 'Ta%'
```

100% 19:1

Result Grid Filter Rows: Search Export:

movie_title
Aquamarine
Anderson's Cross

Result Grid

6.Find all old Drama movies by joining genre table and old movie table

```
SELECT movie_title,genre FROM genre_recommend INNER JOIN old_movie_final ON old_movie_final.genre_id=genre_recommend.genre_id WHERE genre_recommend.genre LIKE 'Drama'
```

```
1 JOIN old_movie_final ON old_movie_final.genre_id=genre_recommend.genre_id WHERE genre_recommend.genre LIKE 'Drama'
```

100% 167:1

Result Grid Filter Rows: Search Export:

movie_title	genre
The Game of Their Lives	Drama
High Fidelity	Drama
I Think I Love My Wife	Drama
Meteor	Drama
Wall Street: Money Never Sleeps	Drama
Wonder Boys	Drama
The Blind Side	Drama
Get on Up	Drama
Faith Connections	Drama
Peaceful Warrior	Drama

7. Find the running time for different genre by joining 2 tables

```
SELECT genre_recommend.genre, runtime FROM genre_recommend INNER JOIN new_movie ON new_movie.new_movie_id=genre_recommend.new_movie_id
```

```
1 •  SELECT genre_recommend.genre, runtime FROM genre_recommend  
2   INNER JOIN new_movie ON new_movie.new_movie_id=genre_recommend.new_movie_id  
3
```

The screenshot shows the MySQL Workbench interface with a result grid titled "Result Grid". The grid displays the following data:

genre	runtime
Drama	83 min
Action	132 min
Adventure	104 min
Romance	83 min
Mystery	93 min
Comedy	139 min
Sci-Fi	123 min
Thriller	85 min
Musical	90 min
Crime	85 min
Fantasy	85 min

8. find actors of in-theater movies and find what movie they have joined before by joining 3 tables

```
SELECT new_movie.title, old_movie_final.main_actor, old_movie_final.movie_title FROM old_movie_final,new_movie,genre_recommend WHERE new_movie.new_movie_id = genre_recommend.new_movie_id AND genre_recommend.genre_id = old_movie_final.genre_id
```

The screenshot shows the MySQL Workbench interface with a result grid titled "Result Grid". The grid displays the following data:

title	main_actor	movie_title
Breakthrough	Elina Abai Kyzy	Queen of the Mountains
Breakthrough	Mick Jagger	Shine a Light
Breakthrough	Jacob Vargas	Heaven Is for Real
Breakthrough	Brian Blessed	Henry V
Breakthrough	Illeana Douglas	Alive
Breakthrough	Meryl Streep	Florence Foster Jenkins
Breakthrough	Martin Short	The Prince of Egypt
Breakthrough	Stockard Channing	Isn't She Great

9. Find the most popular hashtag, show them in discrete order

```
SELECT hashtags,sum(retweet_count) FROM news_trend GROUP BY hashtags ORDER BY sum(retweet_count) DESC
```

The screenshot shows a database query results grid. At the top, there is a code editor window with the following SQL query:

```
1 •  SELECT hashtags,sum(retweet_count) FROM news_trend GROUP BY hashtags ORDER BY sum(retweet_count) DESC
```

Below the code editor is the results grid interface. It has a header row with columns labeled "hashtags" and "sum(retweet_count)". The data rows are as follows:

hashtags	sum(retweet_count)
AvengersEndgame	17036
Tomatometer	10966
Fresh	7201
DontSpoilTheEndgame	5587
GameOfThrones	4405
CertifiedFresh	2743
HowToTrainYourDragon	2450
BOND25	2253

At the bottom left of the grid, it says "Result 2".

10. The most popular movies

```
SELECT movie_title,sum(retweet_count) FROM title_hashtags,new_movie,news_trend WHERE title_hashtags.movie_title=new_movie.title AND title_hashtags.hashtag=news_trend.hashtags GROUP BY movie_title ORDER BY sum(retweet_count) DESC
```

```

1  SELECT movie_title,sum(retweet_count)
2  FROM title_hashtags,new_movie,news_trend
3  WHERE title_hashtags.movie_title=new_movie.title
4  AND title_hashtags.hashtag=news_trend.hashtags
5  GROUP BY movie_title
6  ORDER BY sum(retweet_count) DESC

```

100% 33:6

Result Grid Filter Rows: Search Export:

movie_title	sum(retweet_count)
► How to Train Your Dragon: The Hidden World	2450
Us	1481
Captain Marvel	954
Under the Silver Lake	280
Breakthrough	240
The Curse of La Llorona	178
Little Woods	109
Wonder Park	88
Shazam!	70
Penguins	19
The Best of Enemies	6

Answer questions

1.What are people saying about me (somebody)?

```

1  #people's comments about me(black)
2 • SELECT context FROM ins_post WHERE hashtags LIKE 'black'

```

100% 57:2

Result Grid Filter Rows: Search Export:

context
► ... #666 #baphomet #black #blackmagic #blasphemy #beast #cult #da... Omgf 😱😱..he is so precious • • ignore tags: 😱 #colespouse #cole...

2.How viral are my post?

6885 retweet, so viral!!

```
4      #maximum retweet number
5 •  SELECT max(retweet_count) AS viral_post FROM news_trend|
```

Result Grid	
100%	57:5
Result Grid	Filter Rows: <input type="text"/> Search Export:
viral_post	
▶ 6885	

3.What posts are likely to be interesting to me?

```
6
7      #The movies have the actor Chris Evans are likely to be interesting to me
8 •  SELECT movie_title FROM old_movie_final WHERE main_actor LIKE 'Chris Evans'|
```

Result Grid	
100%	75:8
Result Grid	Filter Rows: <input type="text"/> Search Export:
movie_title	
▶ TMNT	
Sunshine	
Fantastic 4: Rise of the Silver Surfer	

4.What posts are like mine?

```

9
10   #news_tweet has the word 'awesome' like mine
11 •  SELECT * FROM news_trend WHERE news_tweet LIKE '%awesome%'
```

100% 59:11

Result Grid Filter Rows: Search Edit: Export/Import:

news_id	user_name	screen_name	time_of_tweet	retweet_count	news_tweet
1121415897237917697	DR Movie News	DRMovieNews1	Thu Apr 25 14:09:19 +0000 2019	13	Today! the #RoadToEndgame comes to an end...!!Join us as we...
1119269267072061441	Marcus Theatres	Marcus_Theatres	Fri Apr 19 15:59:22 +0000 2019	3	This is incredible way to go! Steve! I'm sure our #Dreamlou...
1117940670969405445	Marcus Theatres	Marcus_Theatres	Tue Apr 16 00:00:00 +0000 2019	1	We've spent the last few days soaking up all things @StarW...
► 1121415897237917697	DR Movie News	DRMovieNews1	Thu Apr 25 14:09:19 +0000 2019	13	Today! the #RoadToEndgame comes to an end...!!Join us as we...
1117940670969405445	Marcus Theatres	Marcus_Theatres	Tue Apr 16 00:00:00 +0000 2019	1	We've spent the last few days soaking up all things @StarW...
1099004286334164993	Animation Addict	animationaddict	Fri Feb 22 17:53:35 +0000 2019	9	Finaled my last shot on #ToyStory4 today! thank you @Cool...
1099004286334164993	Animation Addict	animationaddict	Fri Feb 22 17:53:35 +0000 2019	9	Finaled my last shot on #ToyStory4 today! thank you @Cool...
HULL	HULL	HULL	HULL	HULL	HULL

5.What users post like me?

```

12
13   #posts that time_of_tweet on Saturday are like mine
14 •  SELECT user_name FROM news_trend WHERE time_of_tweet LIKE '%sat%' OR '%wed%'
```

100% 61:14

Result Grid Filter Rows: Search Export:

user_name
Rotten Tomatoes
Fandango
Fandango
IMDb
Movies Anywhere
Animation Addict
AMC Theatres
20th Century Fox
AMC Theatres
news_trend 20

6. Who should I be following?

```
15
16     #I'll follow the account that retweet_count number >500
17 •  SELECT user_name FROM news_trend WHERE retweet_count >500
```

Result Grid Filter Rows: Export:

user_name
Hotten tomatoes
Rotten Tomatoes
DreamWorks An...
Warner Bros.
Rotten Tomatoes
IMDb
IMDb
Entertainment W...
Entertainment W...
Rotten Tomatoes
news_trend 21

Action Output Duration / Fetch Time

Time	Action	Response
------	--------	----------

7. What topics are trending in my domain?

```
18
19     # 7 Number of hashtag that mentioned most is trending
20 •  SELECT hashtags as title,COUNT(*) as number FROM ins_post GROUP BY hashtags
21
22
23
24
```

Result Grid Filter Rows: Export: Fetch rows:

title	number
penguins	5
disneynature	2
documentary	3
nature	3
animals	2
movies	4
film	3
youtuber	2
smallyoutuber	2
lgbt	1
lgbtq	1
gay	1
instagay	1
gaysofinstagram	1
bi	1
biboy	1
biguys	1

```

19      # 7 Number of hashtag that mentioned most is trending
20      #SELECT hashtags AS title,COUNT(*) AS number FROM ins_post GROUP BY hashtags
21      #SELECT MAX(numbers) FROM trendinghashtag
22 •   SELECT * FROM trendinghashtag WHERE numbers = 6
23

```

100% ▾ 1:22

Result Grid		Filter Rows:	Search	Export:
title	numbers			
disney	6			
petsemetary	6			
love	6			

8.What keywords/hashtags should I add to my post?

```
SELECT title,hashtag,retweet_count FROM title_hashtags,new_movie,news_trend
```

```
WHERE title_hashtags.movie_title=new_movie.title AND title_hashtags.hashtag=news_trend.hashtags
```

```

1      #8
2 •   SELECT title,hashtag,retweet_count
3     FROM title_hashtags,new_movie,news_trend
4     WHERE title_hashtags.movie_title=new_movie.title
5     AND title_hashtags.hashtag=news_trend.hashtags

```

100% ▾ 3:1

Result Grid			Filter Rows:	Search	Export:
title	hashtag	retweet_count			
Breakthrough	Breakthrough	14			
Breakthrough	BreakthroughMovie	16			
Breakthrough	BreakthroughMovie	45			
Breakthrough	BreakthroughMovie	7			
Breakthrough	BreakthroughMovie	23			
Breakthrough	BreakthroughMovie	19			
Breakthrough	BreakthroughMovie	11			
Breakthrough	BreakthroughMovie	11			
Breakthrough	BreakthroughMovie	20			
Breakthrough	BreakthroughMovie	11			
Breakthrough	BreakthroughMovie	4			

9.Should I follow somebody back?

```

25      #9
26 •   SELECT * FROM user_data WHERE number_of_tweets_in_24!=0 AND followers_num>100
27

```

100% ▾ 70:26

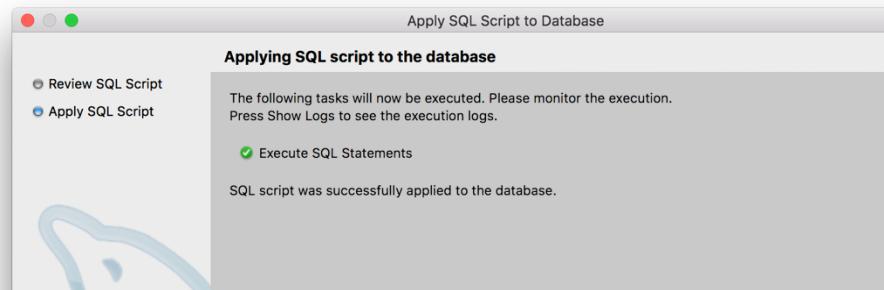
Result Grid Filter Rows: Search Export:

user_name	screen_name	followers_num	number_of_tweets_in_24	friends_count
mariana li-//	miraculousyeet	137	63	822
A loves MBB 🎉	brownspoetics	378	19	1616
CH-7	htdy_love_	372	18	111
Ryan Wooledge	Ryan26514438	179	13	1700
Nor Cal Mythos	NorCalMythos	4098	52	4438
Cynthia 💕	MixTape_Vol_1	568	106	967
Chiefsheart 🇺🇸 AU Hicccstrid	HaruTomio	935	45	879
Wani	LunziKawaihime	104	4	662

Function

1.get average number of retweet

```
1  CREATE DEFINER='root'@'localhost' FUNCTION `like_count`() RETURNS varchar(255) CHARSET utf8mb4
2    DETERMINISTIC
3  BEGIN
4    RETURN (SELECT AVG(retweet_count)FROM news_trend);
5  END
```



```
1 •  select xin.like_count();
2
```

A screenshot of the MySQL Workbench interface showing the results of the executed query. The results are displayed in a 'Result Grid' table. The table has one row with two columns. The first column contains the function call 'xin.like_count()' and the second column contains the result '98.771203155'. The table includes standard navigation controls at the top: zoom levels (100%, 1:1), a search bar, and export options.

2.Sum of retweet_count by search user_name

The screenshot shows the MySQL Workbench interface. On the left, a code editor displays a MySQL function definition:

```
1 CREATE DEFINER='root'@'localhost' FUNCTION `sum_retweet`retweet`(s text) RETURNS int(255)
2   READS SQL DATA
3   DETERMINISTIC
4   BEGIN
5     declare f int(255);
6     select sum(retweet_count) into f from news_trend
7     where s = user_name
8     group by user_name;
9
10    RETURN f;
11  END
```

On the right, a progress dialog titled "Apply SQL Script to Database" shows the status of the execution:

Applying SQL script to the database

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

Execute SQL Statements

SQL script was successfully applied to the database.

In the bottom panel, a results grid shows the output of the query:

Result Grid	Filter Rows:	Search	Export:
xin.sum_retweet'retwee('IMDb')			
19801			

3. number of count_id when user_name is IMDb

The screenshot shows the MySQL Workbench interface. On the left, a code editor displays the SQL script for creating a function:

```
1 CREATE DEFINER='root'@'localhost' FUNCTION `find_countid`(i text ) RETURNS varchar(255) CHARSET utf8mb4
2   READS SQL DATA
3   DETERMINISTIC
4   BEGIN
5     declare result varchar(255);
6     select count(news_id) into result
7     from news_trend
8     where news_trend.user_name = i;
9     RETURN result;
10    END
```

To the right, a modal window titled "Apply SQL Script to Database" is open, showing the status of the task:

Applying SQL script to the database
The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.
Execute SQL Statements
SQL script was successfully applied to the database.

The screenshot shows the MySQL Workbench interface. On the left, a code editor displays the SQL query to execute the function:

```
1 • |select xin.find_countid('IMDb');|
2
```

On the right, the "Result Grid" tab is active, showing the output of the query:

xin.find_countid('IMDb')
176

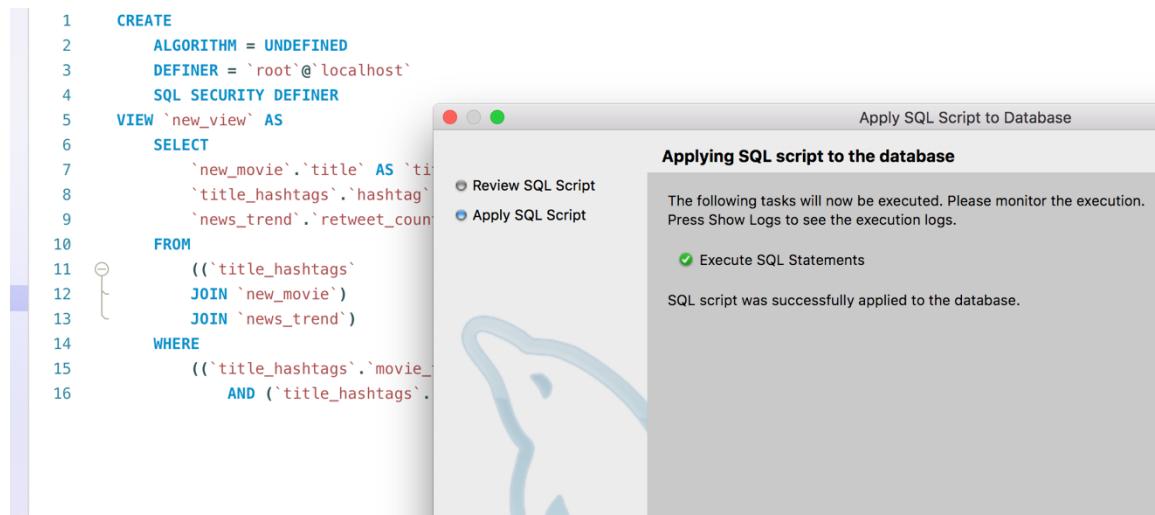
4. class of retweet_count



```
1 • CREATE DEFINER='root'@'localhost' FUNCTION `class`(retweet_count int ) RETURNS varchar(255) CHARSET utf8mb4
2     deterministic
3     begin
4         declare rankretweet varchar(255)
5
6         if retweet_count >50 then
7             set rankretweet = 'popular';
8         elseif(retweet_count <=50) then
9             set rankretweet = 'unpopular';
10        end if;
11        return(rankretweet)
12    end
```

Views

View1



The screenshot shows the MySQL Workbench interface. On the left, a code editor displays the SQL script for creating a view:

```
1 CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5 VIEW `new_view` AS
6     SELECT
7         `new_movie`.`title` AS `ti
8         `title_hashtags`.`hashtag`
9         `news_trend`.`retweet_coun
10    FROM
11        ((`title_hashtags`
12        JOIN `new_movie`)
13        JOIN `news_trend`)
14    WHERE
15        ((`title_hashtags`.`movie_` AND (`title_hashtags`.
```

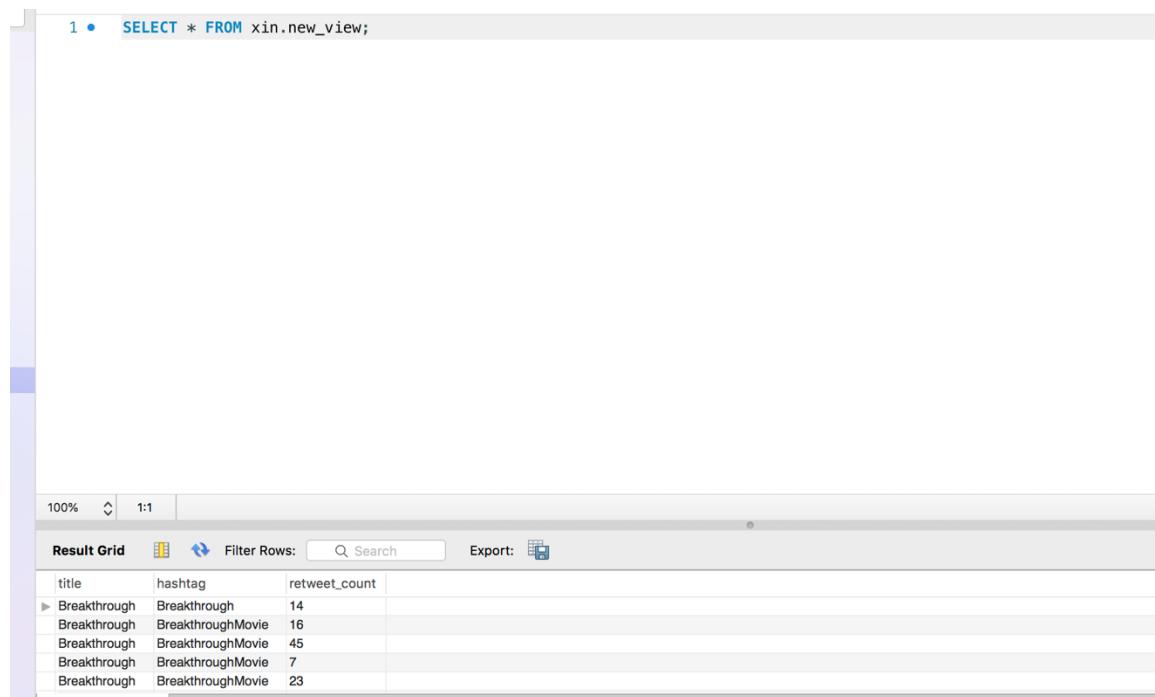
On the right, a modal window titled "Apply SQL Script to Database" is open, showing the status of the task:

Applying SQL script to the database

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

Execute SQL Statements

SQL script was successfully applied to the database.



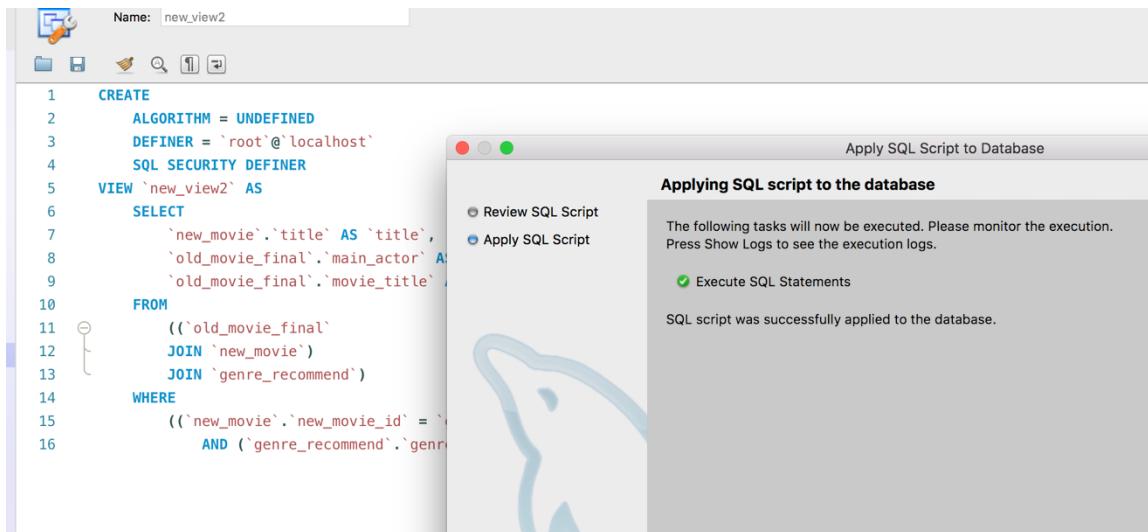
The screenshot shows the MySQL Workbench interface with the results of a query:

```
1 •  SELECT * FROM xin.new_view;
```

The results are displayed in a grid:

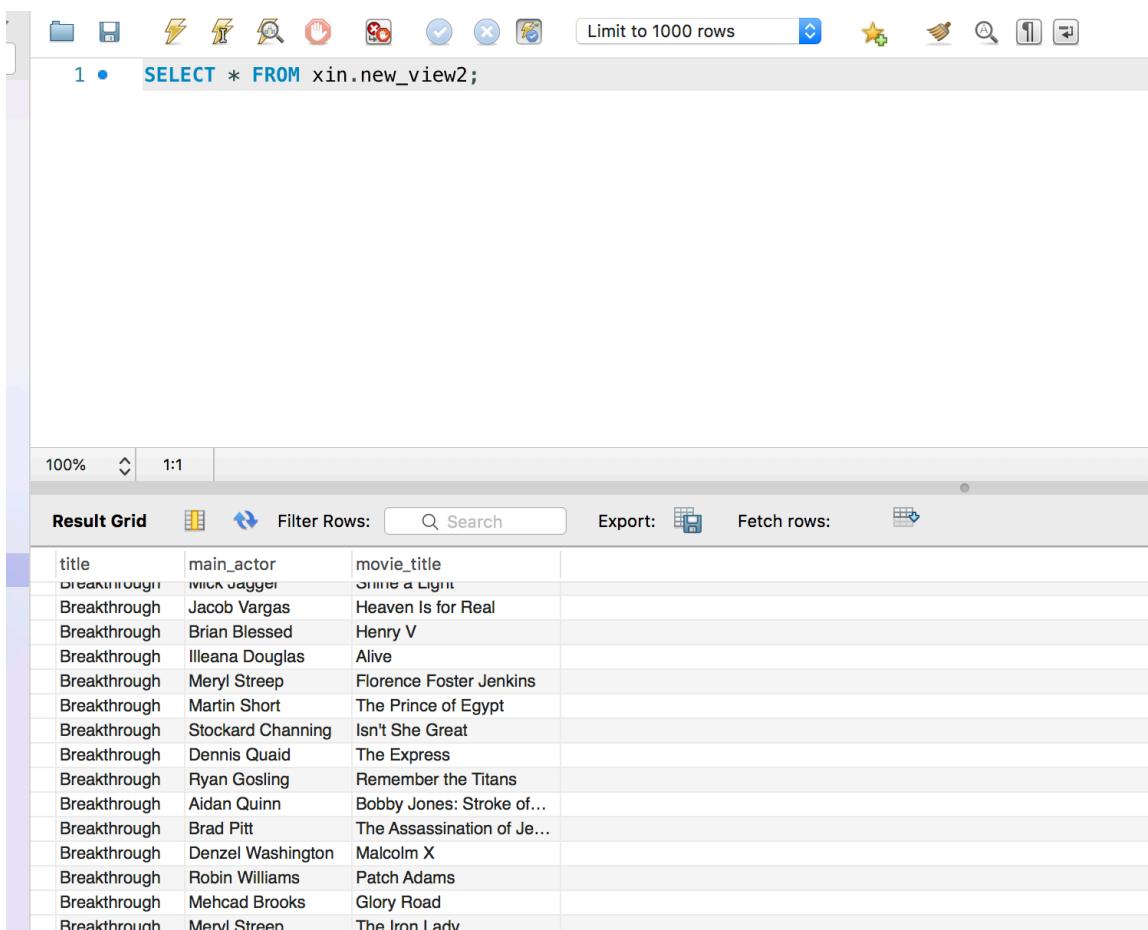
title	hashtag	retweet_count
▶ Breakthrough	Breakthrough	14
▶ Breakthrough	BreakthroughMovie	16
▶ Breakthrough	BreakthroughMovie	45
▶ Breakthrough	BreakthroughMovie	7
▶ Breakthrough	BreakthroughMovie	23

View2



```
Name: new_view2

1 CREATE
2   ALGORITHM = UNDEFINED
3   DEFINER = 'root'@'localhost'
4   SQL SECURITY DEFINER
5   VIEW `new_view2` AS
6     SELECT
7       `new_movie`.`title` AS `title`,
8       `old_movie_final`.`main_actor` AS `main_actor`,
9       `old_movie_final`.`movie_title` AS `movie_title`
10    FROM
11      ((`old_movie_final`
12        JOIN `new_movie`)
13        JOIN `genre_recommend`)
14   WHERE
15     ((`new_movie`.`new_movie_id` = `old_movie_final`.`id`
16       AND (`genre_recommend`.`genre` = `new_movie`.`genre`))
```



```
1 •  SELECT * FROM xin.new_view2;
```

Result Grid

title	main_actor	movie_title
Breakthrough	Mike Jayger	Saints & Sinners
Breakthrough	Jacob Vargas	Heaven Is for Real
Breakthrough	Brian Blessed	Henry V
Breakthrough	Ileana Douglas	Alive
Breakthrough	Meryl Streep	Florence Foster Jenkins
Breakthrough	Martin Short	The Prince of Egypt
Breakthrough	Stockard Channing	Isn't She Great
Breakthrough	Dennis Quaid	The Express
Breakthrough	Ryan Gosling	Remember the Titans
Breakthrough	Aidan Quinn	Bobby Jones: Stroke of Genius
Breakthrough	Brad Pitt	The Assassination of Jesse James by the Coward Robert Ford
Breakthrough	Denzel Washington	Malcolm X
Breakthrough	Robin Williams	Patch Adams
Breakthrough	Mehcad Brooks	Glory Road
Breakthrough	Mervl Street	The Iron Lady

View3

The screenshot shows the MySQL Workbench interface. In the top left, there's a code editor window titled "Name: new_view3" containing the SQL script for creating a view:

```
1 CREATE
2   ALGORITHM = UNDEFINED
3   DEFINER = `root`@`localhost`
4   SQL SECURITY DEFINER
5   VIEW `new_view3` AS
6     SELECT
7       `genre_recommend`.`genre` AS `g`
8       `new_movie`.`runtime` AS `r`
9     FROM
10    (`genre_recommend`
11     JOIN `new_movie` ON ((`new_`
```

In the top right, a modal dialog titled "Apply SQL Script to Database" is open, showing the status: "Applying SQL script to the database". It includes options to "Review SQL Script" or "Apply SQL Script" and a message indicating the script was successfully applied.

The screenshot shows the MySQL Workbench interface with a query editor window. The query being run is:

```
1 • SELECT * FROM xin.new_view3;
```

Below the query, the results are displayed in a "Result Grid" table:

genre	runtime
Action	102 min
Adventure	104 min
Romance	83 min
Mystery	93 min
Comedy	139 min
Sci-Fi	123 min
Thriller	85 min
Musical	90 min
Crime	85 min
Fantasy	85 min
Horror	101 min
Family	104 min
Biography	116 min

View4

The screenshot shows the MySQL Workbench interface. On the left, a SQL editor window displays the following SQL code:

```
1 CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `new_view4` AS
6         SELECT
7             `news_trend`.`user_name` AS `user_`
8         FROM
9             `news_trend`
10        WHERE
11            (`news_trend`.`retweet_count` > 50)
```

On the right, a modal dialog titled "Apply SQL Script to Database" is open, showing the results of the execution:

Applying SQL script to the database

Review SQL Script
 Apply SQL Script

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

Execute SQL Statements

SQL script was successfully applied to the database.

1 • **SELECT * FROM xin.new_view4;**

Execute the selected portion of the script or everything, if there is no selection

100% 1:1

Result Grid

user_name

Rotten Tomatoes

Rotten Tomatoes

Rotten Tomatoes