

**Data Management and Database Design**  
**INFO 6210**

**Assignment 3**

**Group Members**  
**HAN WU**  
**JIANYU HU**

## **Abstract**

*This assignment mainly contains 4 parts. First is perform database normalization. Normalization is performed with its 1NF, 2NF and 3NF. After normalization, indexes are created for improve the performance. 5 function and 5 procedures are also designed and created to fulfill the requirement of this assignment*

## **1. 1st Form of Normalization**

### **YouTube:**

Table 1: items

Primary key is items\_\_id and there are no multiple values in single attribute.

Therefor 1NF is satisfied

Table 2: URL

Primary key is items\_\_snippet\_\_thumbnails\_\_1\_\_url and there are no multiple values in single attribute.

Therefor 1NF is satisfied

Table 3: Localized

Primary key is items\_\_snippet\_\_localized\_\_title and there are no multiple values in single attribute.

Therefor 1NF is satisfied

Table 4: Channel

Primary key is items\_\_snippet\_\_channelId and there are no multiple values in single attribute.

Therefor 1NF is satisfied

### **Twitter:**

Table 1: Tweet

Primary key are user\_\_name and id and there are no multiple values in single attribute.

Therefor 1NF is satisfied

Table 2: media

Primary key is media\_\_id and there are no multiple values in single attribute.

Therefor 1NF is satisfied

Table 3: usermentions

Primary key is user\_\_mentions\_\_id and there are no multiple values in single attribute.

Therefor 1NF is satisfied

Table 4: urls

Primary key is links and there are no multiple values in single attribute.

Therefor 1NF is satisfied

## **2. 2nd Form of Normalization**

### **YouTube:**

Table 1: items

There is no calculated value in this entity.

partial dependency does not exist for this entity.

Therefor 2NF is satisfied.

this entity.

Therefore 2NF is satisfied.

Table 2: URL

There is no calculated value in this entity.

partial dependency does not exist for this entity.

Therefore 2NF is satisfied.

Table 3: Localized

There is no calculated value in this entity.

partial dependency does not exist for this entity.

Therefore 2NF is satisfied.

Table 4: Channel

There is no calculated value in this entity.

partial dependency does not exist for this entity.

Therefore 2NF is satisfied.

### Twitter:

Table 1: Tweet

There exists two primary keys which might cause the partition dependency

Therefore 2NF is not satisfied.

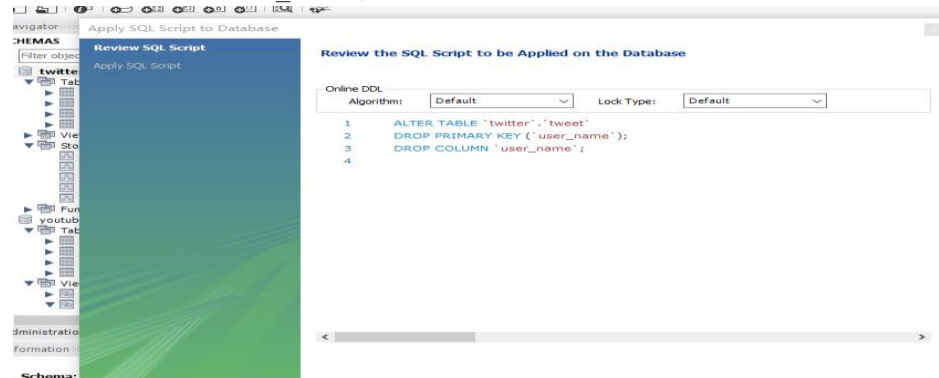
Modify:

Drop the primary key user\_name for it would not affect the statistic

*ALTER TABLE `twitter`.`tweet`*

*DROP PRIMARY KEY (`user\_name`);*

*DROP COLUMN `user\_name`;*



No partial dependency for this entity.

After modification 2NF is satisfied

Table 2: media

There is no calculated value in this entity.

partial dependency does not exist for this entity.

Therefore 2NF is satisfied.

Table 3: usermentions

There is no calculated value in this entity.

partial dependency does not exist for this entity.

Therefore 2NF is satisfied.

Table 4: urls

There is no calculated value in this entity.

partial dependency does not exist for this entity.  
Therefor 2NF is satisfied.

### 3. 3rd Form of Normalization

#### YouTube:

Table 1: items

There is no transitive dependency with in this entity Therefor 3NF is satisfied.

Table 2: URL

There is no transitive dependency with in this entity Therefor 3NF is satisfied.

Table 3: Localized

There is no transitive dependency with in this entity Therefor 3NF is satisfied.

Table 4: Channel

There is no transitive dependency with in this entity Therefor 3NF is satisfied.

#### Twitter:

Table 1: Tweet

There is no transitive dependency with in this entity Therefor 3NF is satisfied.

Table 2: media

There is no transitive dependency with in this entity Therefor 3NF is satisfied.

Table 3: usermentions

There is no transitive dependency with in this entity Therefor 3NF is satisfied.

Table 4: urls

There is no transitive dependency with in this entity Therefor 3NF is satisfied.

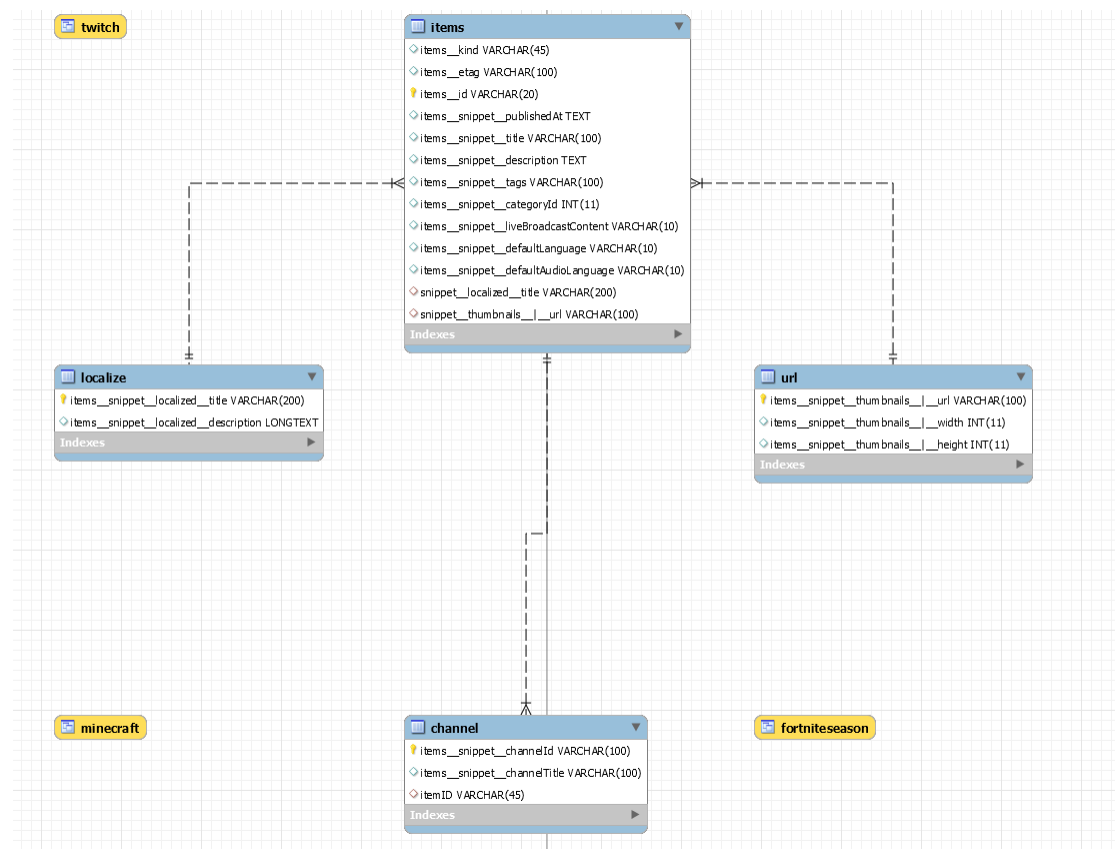


Figure1: YouTube EER Diagram

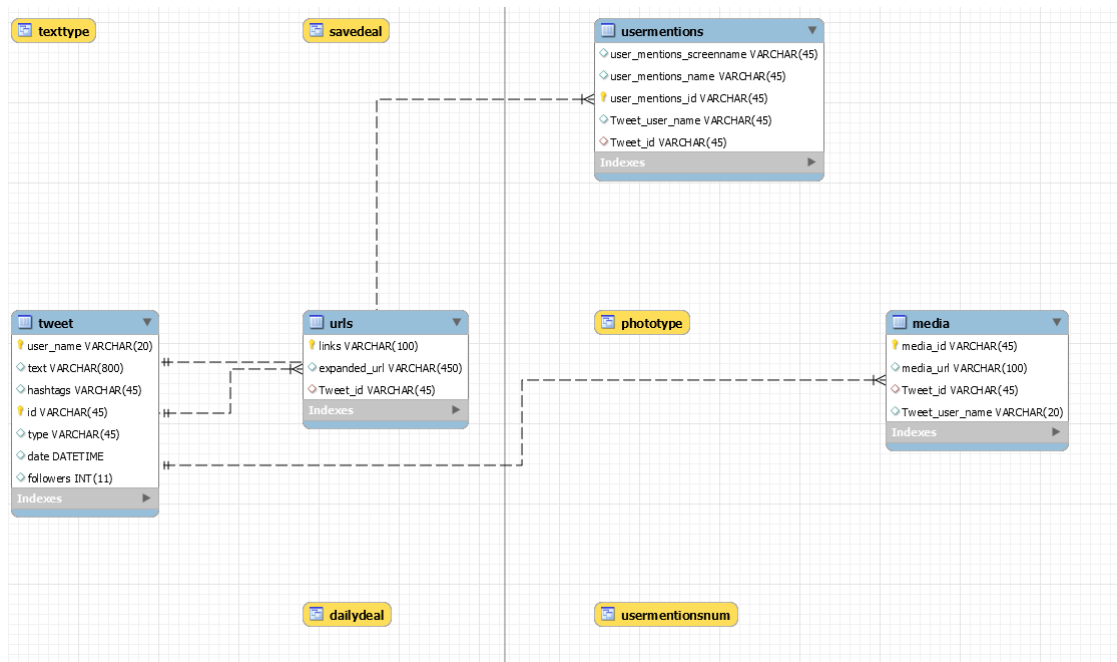


Figure2: Twitter EER Diagram

## 4. Views Created

### YouTube

CREATE OR REPLACE VIEW youtube.Minecraft AS

SELECT \* FROM youtube.items WHERE items\_\_snippet\_\_title LIKE '%Minecraft%';

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane shows a tree view with 'youtube' selected. The main pane displays a SQL query: `SELECT * FROM youtube.minecraft;`. Below the query, the 'Result Grid' shows a table with 5 columns: items\_\_kind, items\_\_etag, items\_\_id, items\_\_snippet\_\_publishedAt, and items. The table contains 7 rows of data, all with 'youtube#video' as the kind.

items__kind	items__etag	items__id	items__snippet__publishedAt	items
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/w505le_VjI...	25aXXJTVcCM	2019-02-21T21:27:59.000Z	Never
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/3nv6P0vvE...	BtuzB3HjiYc	2019-02-23T10:15:00.000Z	Monst
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/AVIaBGWb...	KDjy7ma0c4	2019-02-24T00:27:56.000Z	Minec
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/hJE2_5Oc7...	PBff9EFH90c	2019-02-23T19:41:59.000Z	I SnAq
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/ffTX7L0vBV...	pl8cagt9gK0	2019-02-19T02:46:26.000Z	Playin
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/r6kFxxRPe...	tUjLs6AdumM	2019-02-23T15:05:56.000Z	I Four
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/up-JV2H85t...	Yl0gPzl6YSU	2019-02-25T06:14:33.000Z	Monst

CREATE OR REPLACE VIEW youtube.Fortnite AS

SELECT \* FROM youtube.items

HAVING items\_\_snippet\_\_tags LIKE '%fortnite%';

The screenshot shows a database management tool with a 'Navigator' pane on the left and a main query editor on the right. The 'Navigator' pane shows a tree structure of databases: sakila, sys, twitter, and youtube. The 'youtube' database is expanded, showing tables (channel, items, localize, url) and views (childfriendly, fortnite). The 'fortnite' view is selected. The main query editor shows the following SQL query:

```
1 • SELECT * FROM youtube.fortnite;
```

The 'Result Grid' shows the following data:

items__kind	items__etag	items__id	items__snippet__publishedAt	iter
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/nIK-CvMsY...	_skHagRaXsk	2019-02-18T16:36:18.000Z	Cizz
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/gBQzHvboi...	0JX7tXvivPg	2019-02-19T00:17:58.000Z	Jooq
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/KZQzWm63...	0QGnoax6e_E	2019-02-26T19:00:00.000Z	I pa
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/MjurBkcBTS...	8ucDvWuCrA	2019-02-20T22:50:10.000Z	Top
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/6Cz9ukwTF...	9h4a1aETYCs	2019-02-27T00:08:00.000Z	Seai
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/ButdVILbfz...	9V86MOJR-Rk	2019-02-26T01:36:12.000Z	I spi
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/KrpdKqLGIS...	biLHbkuakwU	2019-02-25T03:24:05.000Z	NEV
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/y3qwcPxuK...	BjMz6cX5Kcg	2019-02-22T20:00:05.000Z	We
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/AEGrcqwFE...	d5IT38GGU-U	2019-02-23T18:30:37.000Z	Fort
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/qPHxVuV1o...	f4ScBIFxsBY	2019-02-25T19:48:17.000Z	USII
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/dhDnYiem...	nkftHkk1rk	2019-02-25T19:01:11.000Z	Fres

CREATE OR REPLACE VIEW youtube.childfriendly AS

SELECT \* FROM youtube.items HAVING items\_\_snippet\_\_tags = 'child friendly';

The screenshot shows the same database management tool with the 'childfriendly' view selected in the 'Navigator' pane. The main query editor shows the following SQL query:

```
1 • SELECT * FROM youtube.childfriendly;
```

The 'Result Grid' shows the following data:

items__kind	items__etag	items__id	items__snippet__publishedAt	items__snip
youtube#video	XpPGQXPnxQJhLgs6enD_n8JR4Qk/K4sGpzhS9...	IWC_mQGKPkU	2019-02-21T19:26:14.000Z	BUILD Your

CREATE OR REPLACE VIEW youtube.twitch AS

SELECT \* FROM youtube.items HAVING items\_\_snippet\_\_tags = 'twitch';

Navigator: fortnteseason childfriendly minecraft twitch fortnite

**SCHEMAS**

Filter objects

- sakila
- sys
- twitter
- youtube
  - Tables
    - channel
    - items
    - localize
    - url
  - Views
    - childfriendly
    - fortnite
    - fortnteseason
    - minecraft**
    - twitch
      - items\_\_kind
      - items\_\_etag
      - items\_\_id
      - items\_\_snippet\_
      - items\_\_snippet\_
      - items\_\_snippet\_
      - items\_\_snippet\_
      - items\_\_snippet\_
      - items\_\_snippet\_
      - items\_\_snippet\_

Administration Schemas

1 • SELECT \* FROM youtube.twitch;

Result Grid

items__kind	items__etag	items__id	items__snippet_publicis
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/uMjfxOVfH...	OSMgs5Gu4Bc	2019-02-25T17:00:04.0

CREATE OR REPLACE VIEW youtube.Fortnteseason AS

SELECT \* FROM youtube.items

WHERE items\_\_snippet\_\_tags = 'fortnite' AND items\_\_snippet\_\_title LIKE '%season%';

Navigator: fortnteseason childfriendly minecraft twitch fortnite

**SCHEMAS**

Filter objects

- sakila
- sys
- twitter
- youtube
  - Tables
    - channel
    - items
    - localize
    - url
  - Views
    - childfriendly
    - fortnite
    - fortnteseason**
    - fetching...
    - minecraft
    - twitch
  - Stored Procedures
  - Functions

1 • SELECT \* FROM youtube.fortnteseason;

Result Grid

items__kind	items__etag	items__id	items__snippet_publishedAt	items__snip
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/6Cz9ukwTF...	9h4a1aETyCs	2019-02-27T00:08:00.000Z	Season 8 Vo
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/KTX8ofdee...	LdDbsCAe_dI	2019-02-16T23:04:45.000Z	7 Things Yo
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/WJE8NR5IP...	LJWY_UdRMnQ	2019-02-22T16:30:00.000Z	Fortnite   Se
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/K1w5RxcO...	pIIScSVt0hs	2019-02-27T23:58:14.000Z	FORTNITE S
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/gR8PDs1pN...	T3Jh6Qa7MGA	2019-02-23T16:17:27.000Z	"NEW" SEA
youtube#video	"XpPGQXPnxQJhLgs6enD_n8JR4Qk/J3ebKvIP1C...	z7awoPkHhM	2019-02-25T01:51:09.000Z	Season 8 LI

## Twitter

CREATE OR REPLACE view twitter.PhotoType AS

SELECT COUNT(\*) FROM twitter.tweet WHERE type = "photo";

Navigator: SCHEMAS

Filter objects

- sakila
- sys
- twitter
  - Tables
    - media
    - tweet
    - urls
    - usermentions
  - Views
    - dailydeal
      - COUNT(\*)
    - phototype
      - COUNT(\*)
    - savedeal
      - user\_name
      - text
      - hashtags
      - id
      - type
      - date

dailydeal phototype savedeal texttype usermentionsnum

1 • SELECT \* FROM twitter.phototype;

Result Grid

COUNT(*)
23

CREATE OR REPLACE view twitter.TextType AS  
 SELECT COUNT(\*) FROM twitter.tweet WHERE type = "text";

Navigator: SCHEMAS

Filter objects

- sakila
- sys
- twitter
  - Tables
    - media
    - tweet
    - urls
    - usermentions
  - Views
    - dailydeal
      - COUNT(\*)
    - phototype
      - COUNT(\*)
    - savedeal
      - user\_name
      - text
      - hashtags
      - id
      - type
      - date
      - followers
    - texttype
      - COUNT(\*)
    - usermentionsnum
      - COUNT(\*)

dailydeal phototype savedeal texttype usermentionsnum

1 • SELECT \* FROM twitter.texttype;

Result Grid

COUNT(*)
35

CREATE OR REPLACE view twitter.UserMentionsNum AS  
 SELECT COUNT(\*) FROM twitter.usermentions  
 WHERE user\_mentions\_name IS NOT NULL;



Navigator: Schemas

Filter objects

- sakila
- sys
- twitter
  - Tables
    - media
    - tweet
    - urls
    - usermentions
  - Views
    - dailydeal
      - COUNT(\*)
    - phototype
      - COUNT(\*)
    - savedeal
      - user\_name
      - text
      - hashtags
      - id
      - type
      - date
      - followers
    - texttype
      - COUNT(\*)
    - usermentionsnum
      - COUNT(\*)
  - Stored Procedures

Administration Schemas

Information: mentionsnum 1 x

dailydeal phototype savedeal texttype usermentionsnum x

Limit to 1000 rows

1 • SELECT \* FROM twitter.usermentionsnum;

Result Grid

COUNT(*)
14

Filter Rows: Export: Wrap Cell Content:

CREATE OR REPLACE view twitter.DailyDeal AS

SELECT COUNT(\*) FROM twitter.tweet WHERE text LIKE '%DAILY DEAL%';

Navigator: Schemas

Filter objects

- sakila
- sys
- twitter
  - Tables
    - media
    - tweet
    - urls
    - usermentions
  - Views
    - dailydeal
      - COUNT(\*)
    - phototype
      - COUNT(\*)
    - savedeal
      - user\_name
      - text
      - hashtags
      - id
      - type
      - date
      - followers
    - texttype
      - COUNT(\*)
    - usermentionsnum
      - COUNT(\*)
  - Stored Procedures

Administration Schemas

Information: mentionsnum 1 x

dailydeal x

Limit to 1000 rows

1 • SELECT \* FROM twitter.dailydeal;

Result Grid

COUNT(*)
9

Filter Rows: Export: Wrap Cell Co

```
CREATE OR REPLACE view twiiter.SaveDeal AS
SELECT * FROM twitter.tweet WHERE text LIKE '%save%';
```

The screenshot shows a database management tool interface. On the left is a tree view of the database schema, including tables like 'media', 'tweet', 'urls', 'usermentions', and views like 'dailydeal', 'phototype', and 'savedeal'. The 'savedeal' view is selected. The main area displays a SQL query: `SELECT * FROM twitter.savedeal;`. Below the query, a 'Result Grid' shows the data returned by the query. The grid has columns: user\_name, text, hashtags, id, type, and date. The data rows show various deals from 'steam\_games' with details like 'WEEKEND DEAL \_\_ Save 50% on TEKKEN 7, Lear...', 'DAILY DEAL \_\_ Save 25% on Disgaea 5 Complet...', etc.

user_name	text	hashtags	id	type	date
steam_games	WEEKEND DEAL __ Save 50% on TEKKEN 7, Lear...	steamdeals	#1101293479290843136	text	2019-4
steam_games	DAILY DEAL __ Save 25% on Disgaea 5 Complet...	steamdeals	#1101628748678324224	photo	2019-4
steam_games	DAILY DEAL __ Save up to 75% on the Warham...	steamdeals	#1101632534092238850	photo	2019-4
steam_games	NOW AVAILABLE __ Dawn of Man - Save 15%,...		#110163278488229248	text	2019-4
steam_games	DAILY DEAL __ Save 70% on Offworld Trading ...	steamdeals	#1101912554933227520	photo	2019-4
steam_games	DAILY DEAL __ Save 50% on Tyranny, Learn m...	steamdeals	#1102274942521946113	text	2019-4
steam_games	DAILY DEAL __ Save 85% on Chivalry: Medieva...	steamdeals	#1102274943264317440	photo	2019-4
steam_games	DAILY DEAL __ Save 60% on RUINER, Learn mo...	steamdeals	#1102645317814476800	text	2019-4
steam_games	DAILY DEAL __ Save 33% on Yakuza 0, Learn m...	steamdeals	#1102645620320296960	text	2019-4
steam_games	MIDWEEK MADNESS __ Save 25% on X4, Learn ...		#1102995568081555456	photo	2019-4
steam_games	MIDWEEK MADNESS __ Save up to 75% on the		#1102995568081555456	photo	2019-4

## 5. Indexes Created

### YouTube

```
CREATE INDEX idx_items__id ON youtube.items(items__id);
```

```
CREATE INDEX idx_items__snippet__tags ON youtube.items(items__snippet__tags);
```

```
CREATE INDEX idx_items__snippet__title ON youtube.items(items__snippet__title);
```

```
CREATE INDEX idx_items__snippet__channelId ON
youtube.channel(items__snippet__channelId);
```

```
CREATE INDEX idx_items__snippet__localized__title ON
youtube.localize(items__snippet__localized__title);
```

### Twitter

```
CREATE INDEX idx_id ON twitter.tweet(id);
```

```
CREATE INDEX idx_user_name ON twitter.tweet(user_name);
```

```
CREATE INDEX idx_date ON twitter.tweet(date);
```

```
CREATE INDEX idx_media_id ON twitter.media(media_id);
```

```
CREATE INDEX idx_user_mentions_id ON twitter.usermentions(user_mentions_id);
```

## 6. Functions Created

### YouTube:

```
DELIMITER //
```

```
CREATE FUNCTION Counttag(tags TEXT)
```

```
RETURNS INT
```

```
BEGIN
```

```
DECLARE count INT;
```

```
SELECT COUNT(*)
```

```
INTO count
```

```
FROM youtube.items
```

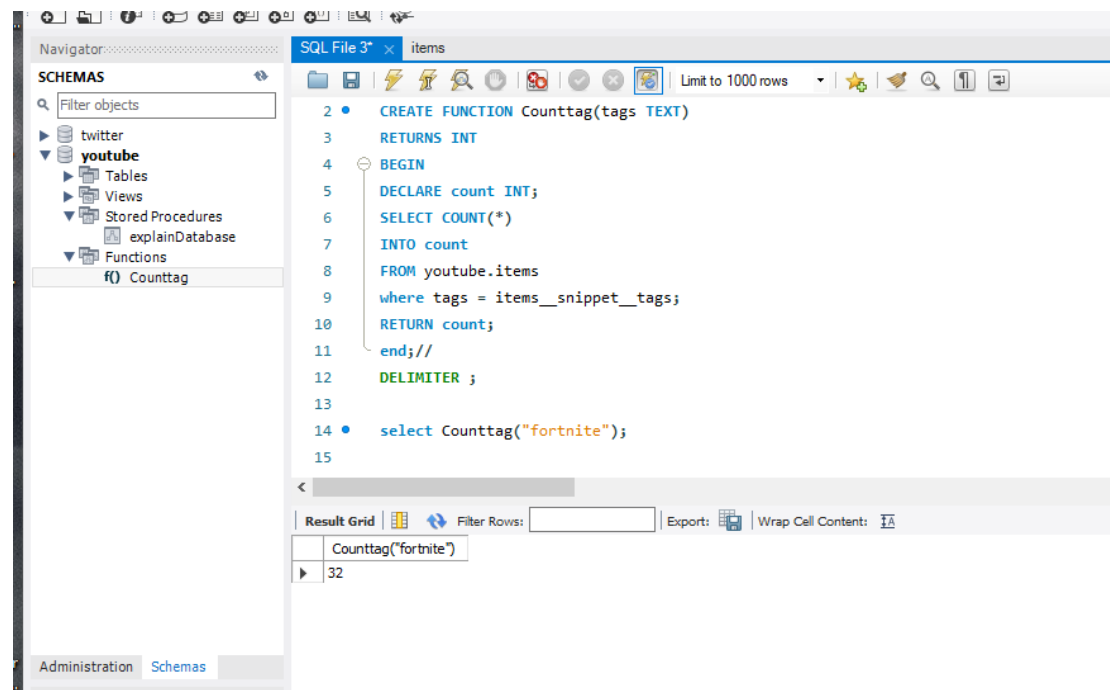
```
where tags = items__snippet__tags;
```

```
RETURN count;
```

```
end; //
```

```
DELIMITER ;
```

```
select Counttag("fortnite");
```



```
DELIMITER //
```

```
CREATE FUNCTION Cal_cate_ID(cateid Int)
```

```
RETURNS INT
```

```
BEGIN
```

```
DECLARE ct_cateid INT;
```

```

SELECT COUNT(*)
INTO ct_cateid
FROM youtube.items
where cateid = items__snippet__categoryId;
RETURN ct_cateid;
end;
DELIMITER ;

```

```
select Cal_cate_ID(20);
```

The screenshot shows a SQL IDE interface with a 'Navigator' pane on the left and a main editor pane. The 'Navigator' pane shows a tree structure with 'twitter' and 'youtube' schemas. Under 'youtube', there are 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Functions' folder is expanded, showing a function named 'f() Counttag'. The main editor pane shows the SQL code for creating a function 'Cal\_cate\_ID' and its execution. The code is as follows:

```

16 DELIMITER //
17 • CREATE FUNCTION Cal_cate_ID(cateid Int)
18 RETURNS INT
19 BEGIN
20 DECLARE ct_cateid INT;
21 SELECT COUNT(*)
22 INTO ct_cateid
23 FROM youtube.items
24 where cateid = items__snippet__categoryId;
25 RETURN ct_cateid;
26 end;
27 DELIMITER ;
28
29 • select Cal_cate_ID(20);

```

Below the code editor, there is a 'Result Grid' section. It shows the execution of the function 'Cal\_cate\_ID(20)' with the result '123'.

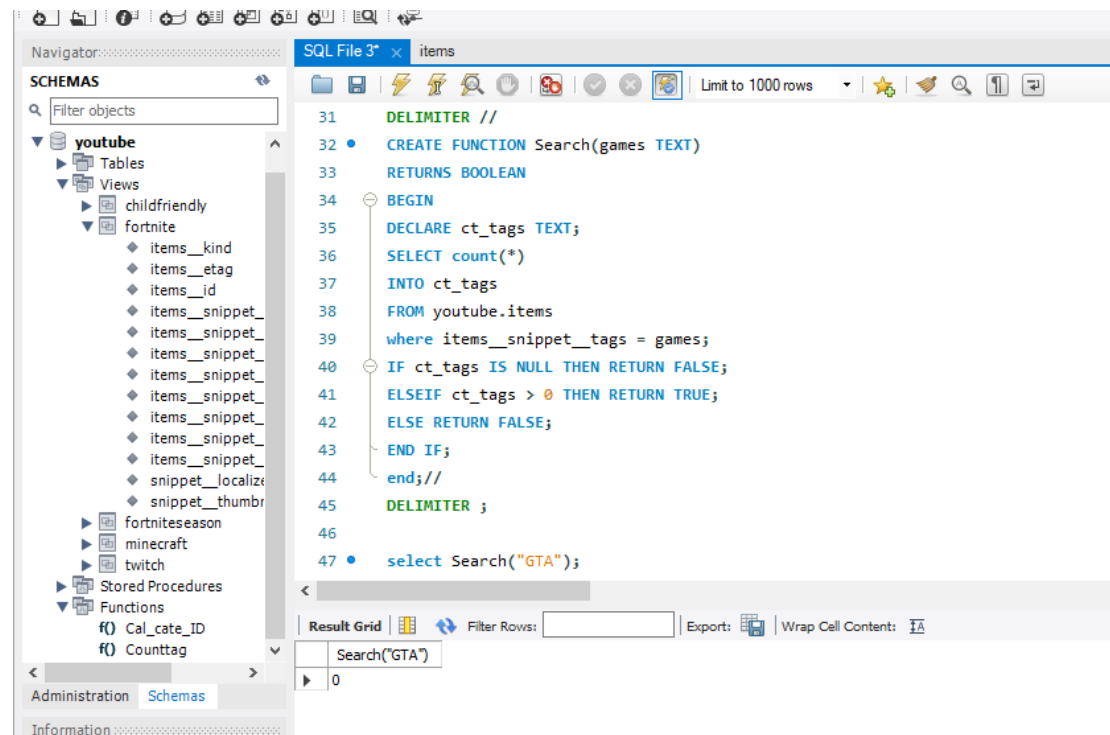
Cal_cate_ID(20)
123

```

DELIMITER //
CREATE FUNCTION Search(games TEXT)
RETURNS BOOLEAN
BEGIN
DECLARE ct_tags TEXT;
SELECT count(*)
INTO ct_tags
FROM youtube.items
where items__snippet__tags = games;
IF ct_tags IS NULL THEN RETURN FALSE;
ELSEIF ct_tags > 0 THEN RETURN TRUE;
ELSE RETURN FALSE;
END IF;

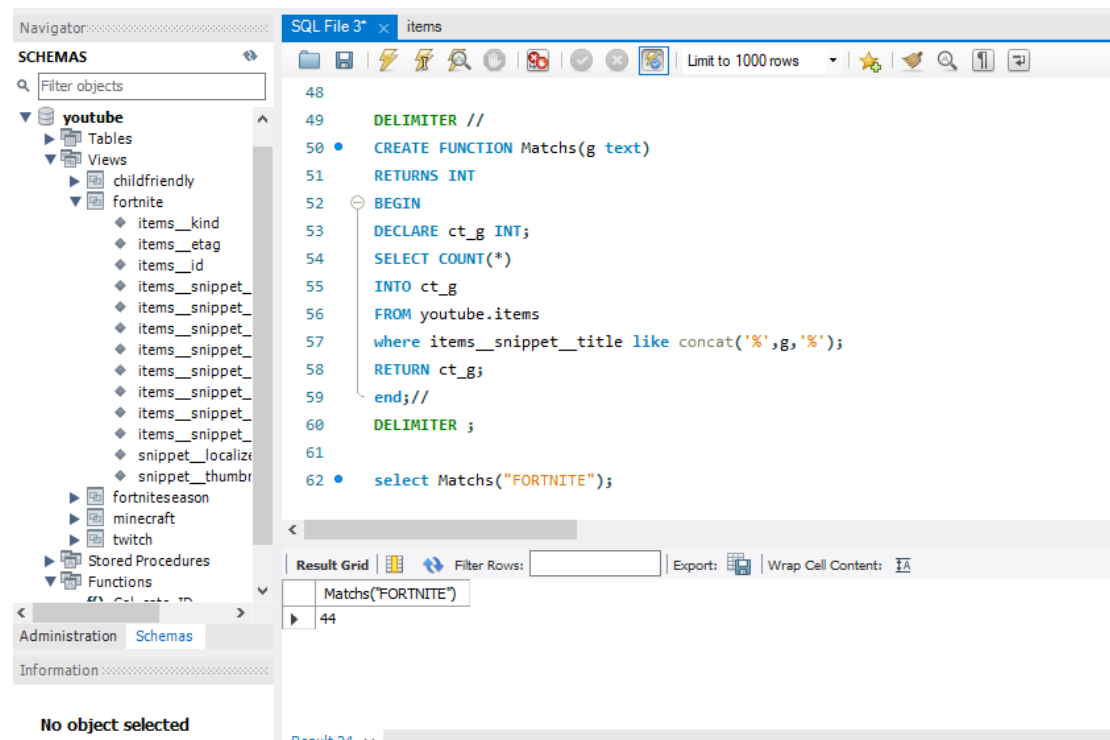
```

DELIMITER ;



```
DELIMITER //
CREATE FUNCTION Matches(g text)
RETURNS INT
BEGIN
DECLARE ct_g INT;
SELECT COUNT(*)
INTO ct_g
FROM youtube.items
where items__snippet__title like concat('%',g,'%');
RETURN ct_g;
end;//
DELIMITER ;
```

```
select Matches("FORTNITE");
```

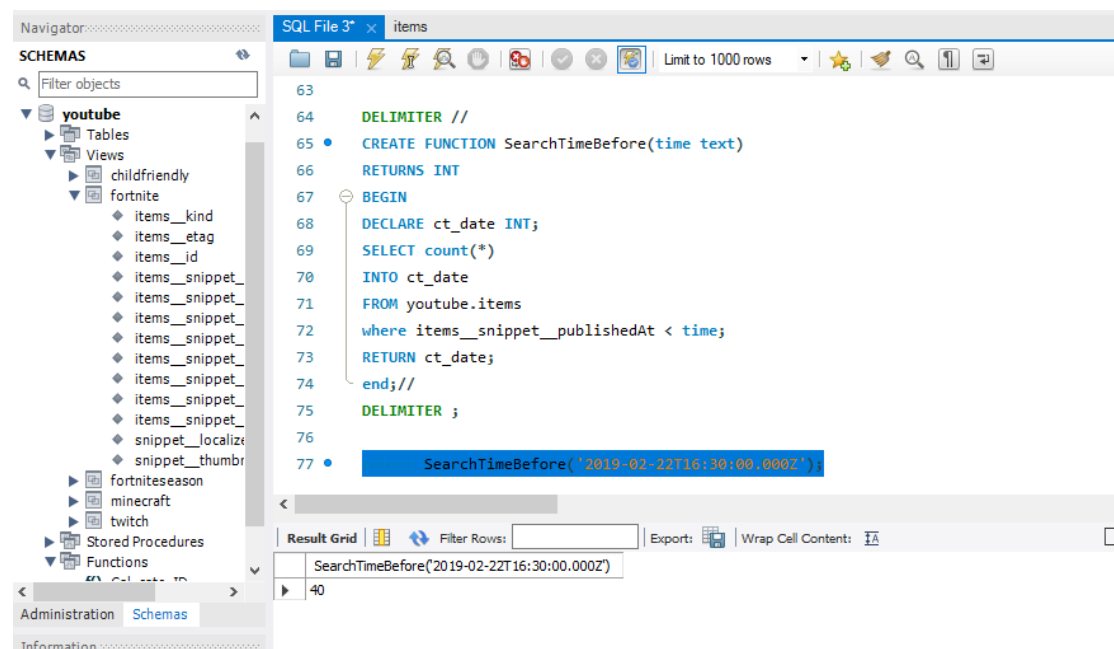


```

DELIMITER //
CREATE FUNCTION SearchTimeBefore(time text)
RETURNS INT
BEGIN
DECLARE ct_date INT;
SELECT count(*)
INTO ct_date
FROM youtube.items
where items__snippet__publishedAt < time;
RETURN ct_date;
end;
DELIMITER ;

select SearchTimeBefore('2019-02-22T16:30:00.000Z');

```



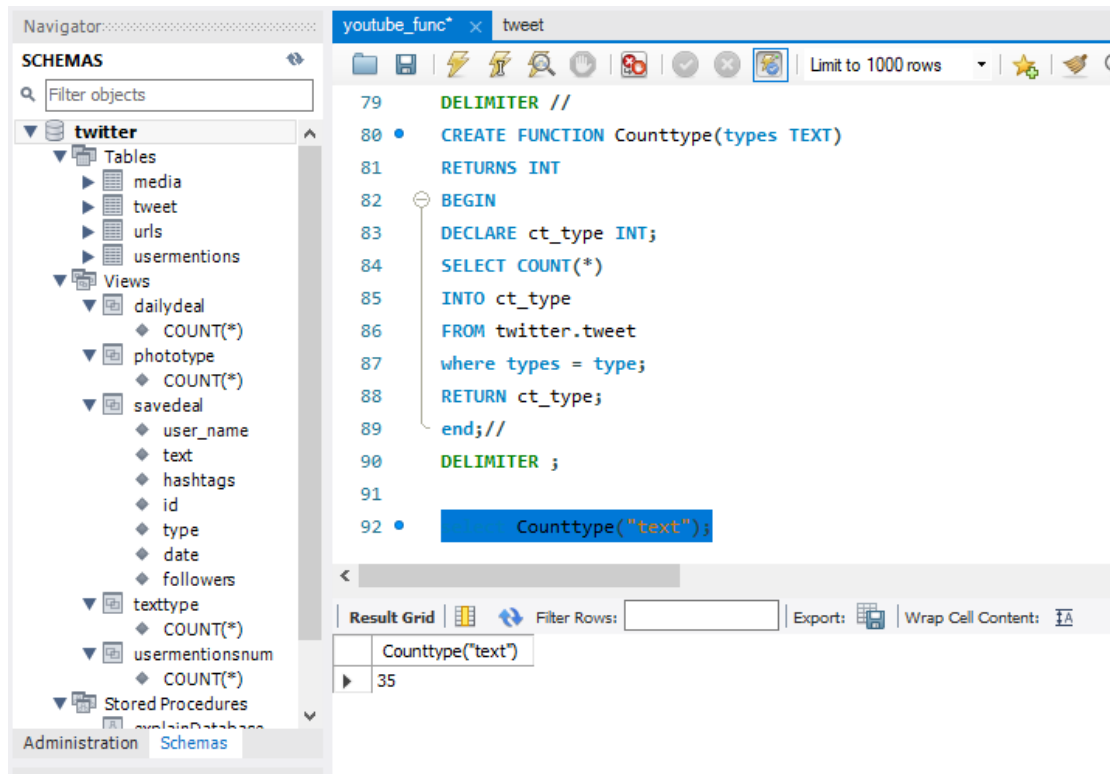
### Twitter:

```

DELIMITER //
CREATE FUNCTION Counttype(types TEXT)
RETURNS INT
BEGIN
DECLARE ct_type INT;
SELECT COUNT(*)
INTO ct_type
FROM twitter.tweet
where types = type;
RETURN ct_type;
end;//
DELIMITER ;

select Counttype("text");

```



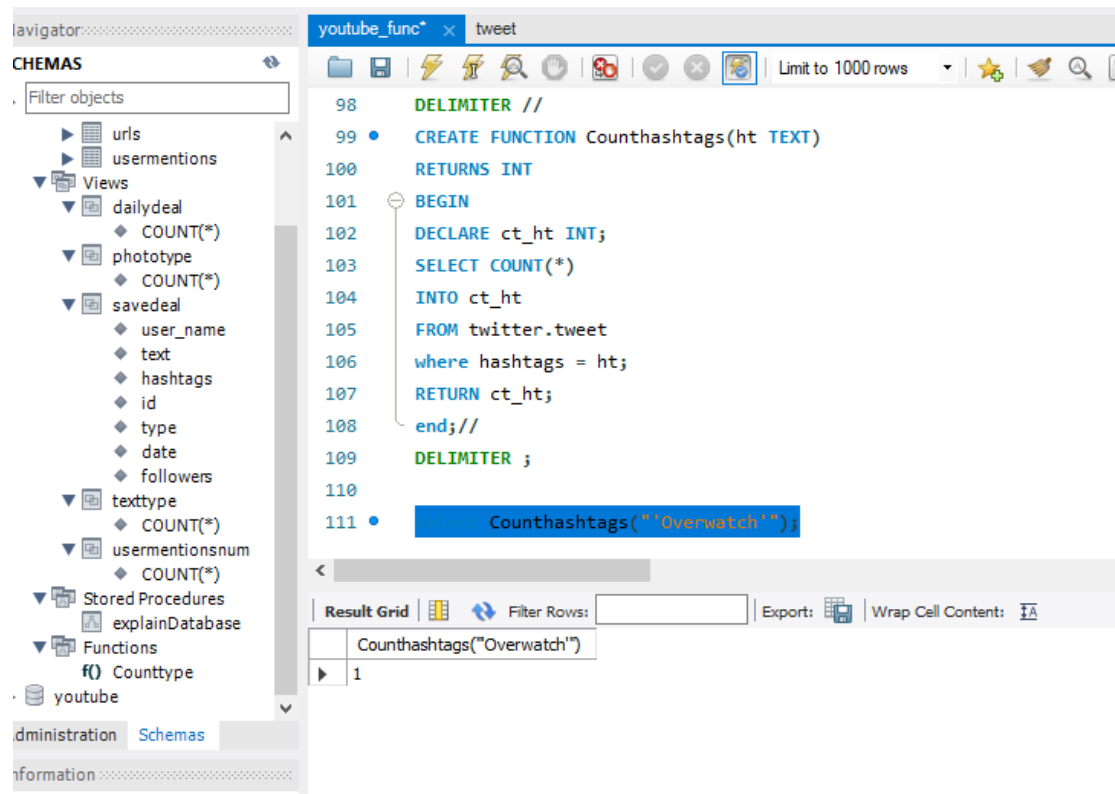
```

DELIMITER //
CREATE FUNCTION Counthashtags(ht TEXT)
RETURNS INT
BEGIN
DECLARE ct_ht INT;
SELECT COUNT(*)
INTO ct_ht
FROM twitter.tweet
where hashtags = ht;
RETURN ct_ht;
end;

select Counthashtags("Overwatch");

```





```

DELIMITER //
CREATE FUNCTION Searches(user TEXT)
RETURNS BOOLEAN
BEGIN
DECLARE ct_user TEXT;
SELECT count(*)
INTO ct_user
FROM twitter.tweet
where user_name = user;
IF ct_user IS NULL THEN RETURN FALSE;
ELSEIF ct_user > 0 THEN RETURN TRUE;
ELSE RETURN FALSE;
END IF;
end; //
DELIMITER ;

select Searches('steam_games');

```

The screenshot shows a database management interface with a left sidebar containing a tree view of database objects (schemas, views, tables, etc.). The main area is a SQL script editor with a toolbar at the top. The script defines a function 'Searchs' that takes a user name as input and returns a boolean value based on the count of mentions in the 'tweet' table. The script is as follows:

```

114 CREATE FUNCTION Searchs(user TEXT)
115 RETURNS BOOLEAN
116 BEGIN
117     DECLARE ct_user TEXT;
118     SELECT count(*)
119     INTO ct_user
120     FROM twitter.tweet
121     where user_name = user;
122     IF ct_user IS NULL THEN RETURN FALSE;
123     ELSEIF ct_user > 0 THEN RETURN TRUE;
124     ELSE RETURN FALSE;
125 END IF;
126 end; //
127 DELIMITER ;
128
129 Searchs('steam_games');

```

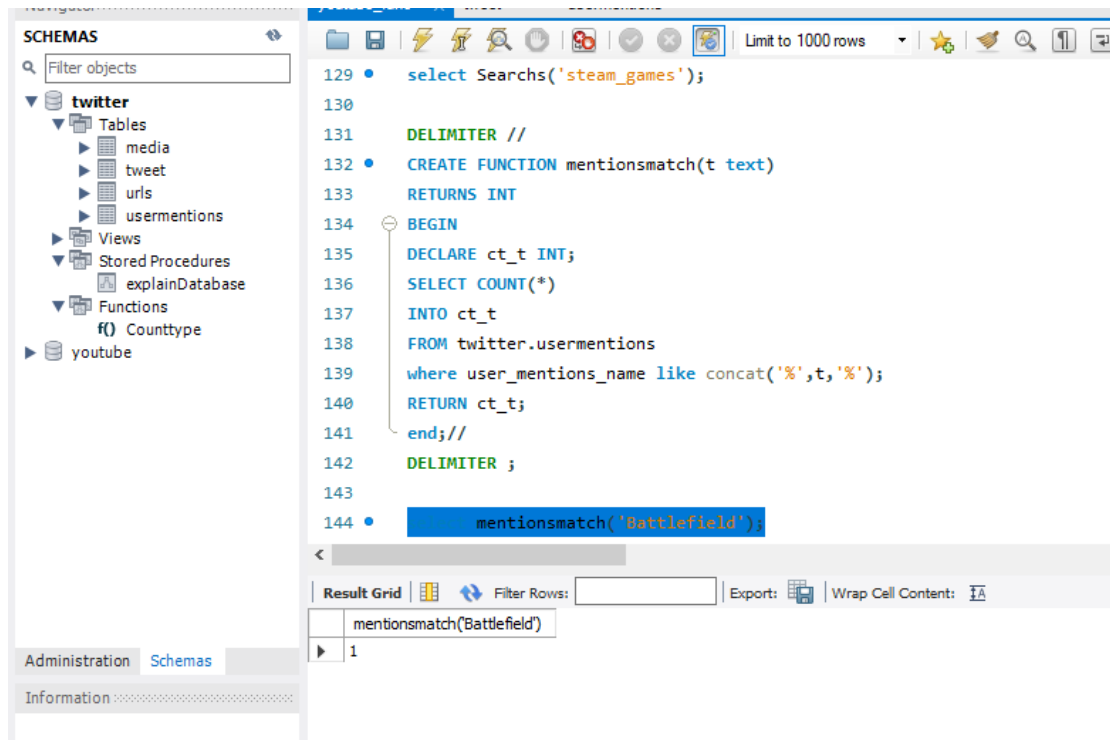
Below the script editor, there is a 'Result Grid' section. It shows a single row with the value '1' for the query 'Searchs('steam\_games')'.

```

DELIMITER //
CREATE FUNCTION mentionsmatch(t text)
RETURNS INT
BEGIN
DECLARE ct_t INT;
SELECT COUNT(*)
INTO ct_t
FROM twitter.usermentions
where user_mentions_name like concat('%',t,'%');
RETURN ct_t;
end; //
DELIMITER ;

select mentionsmatch('Battlefield');

```

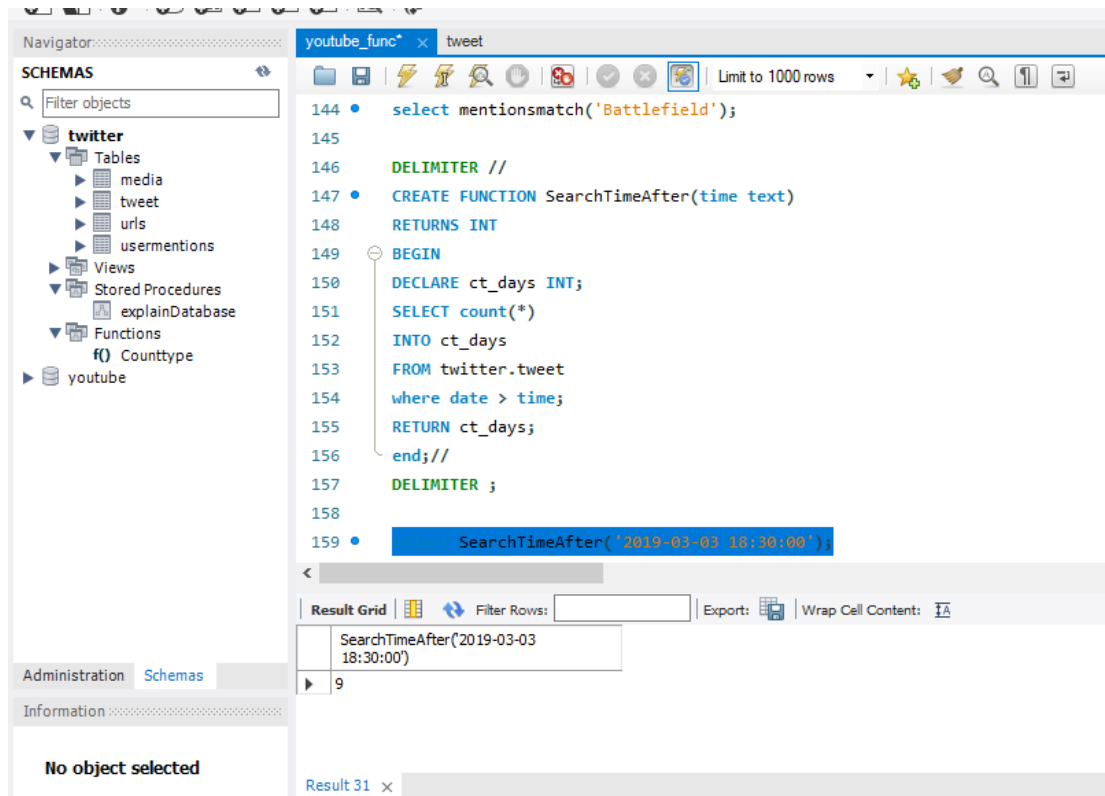


```

DELIMITER //
CREATE FUNCTION SearchTimeAfter(time text)
RETURNS INT
BEGIN
DECLARE ct_days INT;
SELECT count(*)
INTO ct_days
FROM twitter.tweet
where date > time;
RETURN ct_days;
end;//
DELIMITER ;

```

```
select SearchTimeAfter('2019-03-03 18:30:00');
```



## 7. Procedures Created

### YouTube:

DELIMITER //

CREATE PROCEDURE explainDatabase()

BEGIN

SHOW DATABASES;

SHOW TABLES;

SELECT i.TABLE\_NAME, i.COLUMN\_NAME, i.IS\_NULLABLE, i.DATA\_TYPE,  
i.PRIVILEGES

FROM information\_schema.columns i

WHERE table\_schema = 'youtube';

END;//

DELIMITER ;

Navigator: Schemas

Filter objects

- twitter
- youtube
  - Tables
    - channel
    - items
    - localize
    - url
  - Views
  - Stored Procedures
    - Cal\_cate\_ID
    - explainDatabase
    - FilterPX
    - SearchTimeBetween
    - titleMatch
  - Functions

Administration Schemas

Information

Procedure: explainDatabase

SQL File 4\* titleMatch SearchTimeBetween FilterPX Cal\_cate\_ID explainDatabase

Limit to 1000 rows

```

1 • call youtube.explainDatabase();
2

```

Result Grid

TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	PRIVILEGES
channel	items_snippet_channelId	NO	varchar	select,insert,update,references
channel	items_snippet_channelTitle	YES	varchar	select,insert,update,references
channel	itemID	YES	varchar	select,insert,update,references
childfriendly	items_kind	YES	varchar	select,insert,update,references
childfriendly	items_etag	YES	varchar	select,insert,update,references
childfriendly	items_id	NO	varchar	select,insert,update,references
childfriendly	items_snippet_publishedAt	YES	text	select,insert,update,references
childfriendly	items_snippet_title	YES	varchar	select,insert,update,references
childfriendly	items_snippet_description	YES	text	select,insert,update,references
childfriendly	items_snippet_tags	YES	varchar	select,insert,update,references
childfriendly	items_snippet_categoryId	YES	int	select,insert,update,references
childfriendly	items_snippet_liveBroadc...	YES	varchar	select,insert,update,references

Result 1 Result 2 Result 3

Output

```

DELIMITER //
CREATE PROCEDURE titleMatch(g TEXT)
BEGIN
SELECT *
FROM youtube.items
where items__snippet__title like concat('%',g,'%');
END;//
DELIMITER ;

```

Navigator: CHEMAS

- Filter objects
- twitter
- youtube
  - Tables
    - channel
    - items
    - localize
    - url
  - Views
  - Stored Procedures
    - Cal\_cate\_ID
    - explainDatabase
    - FilterPX
    - SearchTimeBetween
    - titleMatch
  - Functions

SQL File 4\* titleMatch SearchTimeBetween url FilterPX Cal\_cate\_ID

```

1 • call youtube.titleMatch('Rainbow Six');
2

```

Result Grid

items__snippet__publishedAt	items__snippet__title	items__snippet__description
2019-02-20T16:00:01.000Z	When GIRLS Play Rainbow Six Siege	This video is sponsored by https://clutchchairz...
2019-02-16T20:22:11.000Z	Rainbow Six Siege: The Hammer and the Scalpel...	Rainbow is an international special operations u...
2019-02-24T20:45:02.000Z	Upside Down Rainbow Six Siege Moments	Leave a like if you enjoyed! Thanks :D\nSubscri...
2019-02-21T19:15:02.000Z	HOW TO USE MOZZIE - Rainbow Six Siege (Bur...	▲Subscribe & Become a Nugget! ► http://bit.l...

Unable to retrieve node

DELIMITER //

CREATE PROCEDURE SearchTimeBetween(time1 text,time2 text)

BEGIN

SELECT \*

FROM youtube.items

where items\_\_snippet\_\_publishedAt between time1 and time2;

end;//

DELIMITER ;

Navigator: SCHEMAS

- Filter objects
- twitter
- youtube
  - Tables
    - channel
    - items
    - localize
    - url
  - Views
  - Stored Procedures
    - Cal\_cate\_ID
    - explainDatabase
    - FilterPX
    - SearchTimeBetween
    - titleMatch
  - Functions

SQL File 4\* titleMatch SearchTimeBetween url FilterPX Cal\_cate\_ID

```

1 • call youtube.SearchTimeBetween('2019-02-21T02:14:13.000Z', '2019-02-23T02:14:13.000Z');
2

```

Result Grid

items_id	items__snippet__publishedAt	items__snippet__title	items__snippet__description
_n8JR4Qk/lu0nsgigAh...	2019-02-21T18:30:09.000Z	Aladdin Deck Enhancer   Gaming Historian	
_n8JR4Qk/w505le_Vji...	2019-02-21T21:27:59.000Z	Never Break In To My Wife's Impossible Minecra...	
_n8JR4Qk/2Bax929vo...	2019-02-21T20:07:47.000Z	r/entitledparents #10   REDDIT STORY COMPIL...	
_n8JR4Qk/4PXCXvR...	2019-02-21T15:22:00.000Z	NOOB vs PRO vs HACKER - Fortnite Battle Royale	
_n8JR4Qk/Ect8CZT_I...	2019-02-22T11:14:34.000Z	Clash Royale: Mini P.E.K.K.A - Pancakes Every...	
_n8JR4Qk/y3qwcPxxK...	2019-02-22T20:00:05.000Z	We tried Cizzorz Deathrun 3.0 and discovered s...	
_n8JR4Qk/heDPFu0Nh...	2019-02-22T21:03:37.000Z	GTA 5 - 250 STAR WANTED LEVEL!! (Can We Es...	
_n8JR4Qk/0Ff6-aQiny...	2019-02-21T19:09:51.000Z	1,000,000 SUBSCRIBERS (best of 2018)	
_n8JR4Qk/g8htIuu-HV...	2019-02-22T20:03:05.000Z	TACTICAL FRIED CHICKEN   World of Tanks	
_n8JR4Qk/_jZ-ZaTSL...	2019-02-23T02:13:19.000Z	Can You Beat Shovel Knight Without Using the ...	
_n8JR4Qk/XfQ73NyEY...	2019-02-21T21:48:57.000Z	I Copied My Haters Outfits But Made Them Cut...	

Procedure:

```

DELIMITER //
CREATE PROCEDURE FilterPX(px_w INT,px_h INT)
BEGIN
SELECT *
FROM youtube.url
where px_w >= width AND px_h >= height;
end;//
DELIMITER ;

```

The screenshot shows a SQL IDE interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure. Under the 'youtube' schema, 'Stored Procedures' are listed, and 'FilterPX' is selected. The top pane shows the SQL code: `1 • call youtube.FilterPX(500, 120);` and `2`. The bottom pane shows the 'Result Grid' with 10 rows of data. The columns are 'items\_\_snippet\_\_thumbnails\_\_url', 'width', and 'height'. The data shows various YouTube video URLs with a width of 120 and a height of 90.

items__snippet__thumbnails__url	width	height
https://i.ytimg.com/vi/0-Se6alTkD4/default.jpg	120	90
https://i.ytimg.com/vi/0JX7bXivPg/default.jpg	120	90
https://i.ytimg.com/vi/0kM4IB7swmA/default.jpg	120	90
https://i.ytimg.com/vi/0QGnoax6e_E/default.jpg	120	90
https://i.ytimg.com/vi/0V9H5bXai_Y/default.jpg	120	90
https://i.ytimg.com/vi/1ZY5QlMxqNU/default.jpg	120	90
https://i.ytimg.com/vi/25aXXJTVcCM/default.jpg	120	90
https://i.ytimg.com/vi/2o3YE3u1OT4/default.jpg	120	90
https://i.ytimg.com/vi/3USYN8kHkG0/default.jpg	120	90
https://i.ytimg.com/vi/4dZQpG9KAtk/default.jpg	120	90
https://i.ytimg.com/vi/4NvXbI6GG0/default.jpg	120	90
https://i.ytimg.com/vi/5_zOCGmbcyE/default.jpg	120	90

```

DELIMITER //
CREATE PROCEDURE Cal_cate_ID()
BEGIN
DECLARE ct_cateid INT;
SELECT items__snippet__categoryId,COUNT(*)
FROM youtube.items
group by items__snippet__categoryId;
end;//
DELIMITER ;

```

Navigator: Filter objects

**SCHEMAS**

- twitter
  - channel
  - items
  - localize
  - url
- youtube**
  - Tables
    - channel
    - items
    - localize
    - url
  - Views
  - Stored Procedures
    - Cal\_cate\_ID**
    - explainDatabase
    - FilterPX
    - SearchTimeBetween
    - titleMatch
- Functions

Administration Schemas

Information

Procedure: **Cal\_cate\_ID**

SQL File 4\* titleMatch SearchTimeBetween FilterPX Cal\_cate\_ID x

Limit to 1000 rows

```
1 • call youtube.Cal_cate_ID();
2
```

Result Grid Filter Rows: Export: Wrap Cell Content: [F1](#)

items_snippet_categoryId	COUNT(*)
20	123
24	5
17	1
23	3
1	2
28	1

Result 1 x

## Twitter:

DELIMITER //

CREATE PROCEDURE explainDatabase()

BEGIN

SHOW DATABASES;

SHOW TABLES;

SELECT i.TABLE\_NAME, i.COLUMN\_NAME, i.IS\_NULLABLE, i.DATA\_TYPE,  
i.PRIVILEGES

FROM information\_schema.columns i

WHERE table\_schema = 'twitter';

END;//

DELIMITER ;



Navigator: SCHEMAS

Filter objects

- dailydeal
  - COUNT(\*)
- phototype
  - COUNT(\*)
- savedeal
  - user\_name
  - text
  - hashtags
  - id
  - type
  - date
  - followers
- texttype
  - COUNT(\*)
- usermentionsnum
  - COUNT(\*)
- Stored Procedures
  - Cal\_followers
  - Cal\_hashtags
  - explainDatabase
  - SearchDateBetween
  - tweetMatch
- Functions
  - youtube

Administration Schemas

Information

Procedure: explainDatabase

SQL File 4\* tweet Cal\_followers Cal\_hashtags SearchDateBetween tweetMatch explainDatabase

Limit to 1000 rows

```
1 • call twitter.explainDatabase();
2
```

Result Grid

TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	PRIVILEGES
dailydeal	COUNT(*)	NO	bigint	select,insert,update,references
media	media_id	NO	varchar	select,insert,update,references
media	media_url	YES	varchar	select,insert,update,references
media	Tweet_id	YES	varchar	select,insert,update,references
media	Tweet_user_name	YES	varchar	select,insert,update,references
phototype	COUNT(*)	NO	bigint	select,insert,update,references
savedeal	user_name	NO	varchar	select,insert,update,references
savedeal	text	YES	varchar	select,insert,update,references
savedeal	hashtags	YES	varchar	select,insert,update,references
savedeal	id	NO	varchar	select,insert,update,references
savedeal	type	YES	varchar	select,insert,update,references
savedeal	date	YES	datetime	select,insert,update,references

Result 1 Result 2 Result 3 x

```
DELIMITER //
CREATE PROCEDURE Cal_followers()
BEGIN
select followers, COUNT(*)
FROM twitter.tweet
group by followers;
end;//
DELIMITER ;
```

Navigator: SCHEMAS

Filter objects

- dailydeal
  - COUNT(\*)
- phototype
  - COUNT(\*)
- savedeal
  - user\_name
  - text
  - hashtags
  - id
  - type
  - date
  - followers
- texttype
  - COUNT(\*)
- usermentionsnum
  - COUNT(\*)
- Stored Procedures
  - Cal\_followers
  - Cal\_hashtags
  - explainDatabase
  - SearchDateBetween
  - tweetMatch
- Functions
  - youtube

Administration Schemas

Information

Procedure: Cal\_followers

SQL File 4\* tweet Cal\_followers Cal\_hashtags SearchDateBetween tweetMatch

Limit to 1000 rows

```
1 • call twitter.Cal_followers();
2
```

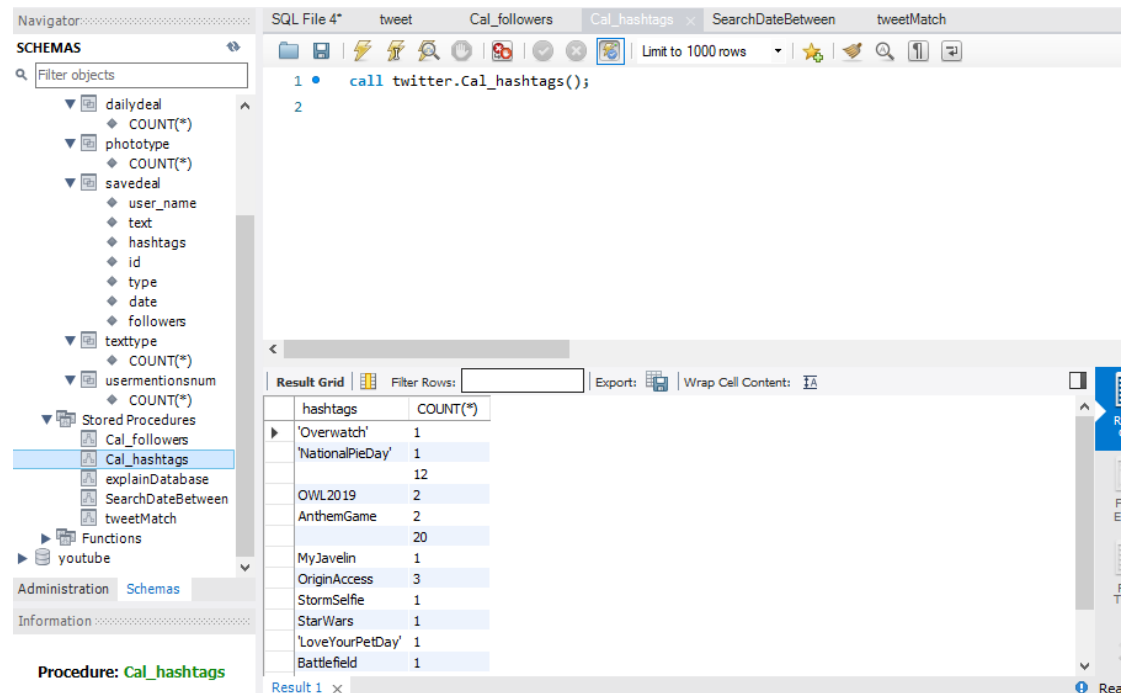
Result Grid

followers	COUNT(*)
1204084	20
733272	20
4987785	18

```

DELIMITER //
CREATE PROCEDURE Cal_hashtags()
BEGIN
select hashtags, COUNT(*)
FROM twitter.tweet
group by hashtags;
end;//
DELIMITER ;

```



Procedure: **Cal\_hashtags**

hashtags	COUNT(*)
'Overwatch'	1
'NationalPieDay'	1
	12
OWL2019	2
AnthemGame	2
	20
MyJavelin	1
OriginAccess	3
StormSelfie	1
StarWars	1
'LoveYourPetDay'	1
Battlefield	1

```

DELIMITER //
CREATE PROCEDURE SearchDateBetween(time1 text,time2 text)
BEGIN
SELECT *
FROM twitter.tweet
where date between time1 and time2;
end;//
DELIMITER ;

```

Navigator: SCHEMAS

Filter objects

- dailydeal
  - COUNT(\*)
- phototype
  - COUNT(\*)
- savedeal
  - user\_name
  - text
  - hashtags
  - id
  - type
  - date
  - followers
- texttype
  - COUNT(\*)
- usermentionsnum
  - COUNT(\*)
- Stored Procedures
  - Cal\_followers
  - Cal\_hashtags
  - explainDatabase
  - SearchDateBetween
  - tweetMatch
- Functions
  - youtube

Administration Schemas

Information

Unable to retrieve node description.

SQL File 4\* tweet Cal\_followers Cal\_hashtags SearchDateBetween tweetMatch

Limit to 1000 rows

1 • twitter.SearchDateBetween('2019-01-20 18:09:00', '2019-01-24 18:09:00');

2

Result Grid

	hashtags	id	type	date	followers
ewmercer: Some days, you wake up...	'Overwatch'	#1088136765796306944	text	2019-01-23 18:09:00	1204084
tionalPieDay, y'all! __ https://t.co/h...	'NationalPieDay'	#1088149374566318080	photo	2019-01-23 19:00:00	1204084
013213 @Vrykerion stay tuned for ...		#1088150122658033664	text	2019-01-23 19:03:00	1204084
from now! https://t.co/6Ahxj3FY		#1088493358236938240	text	2019-01-24 17:46:00	1204084

Result 3 x

Read On

```

DELIMITER //
CREATE PROCEDURE tweetMatch(g TEXT)
BEGIN
SELECT *
FROM twitter.tweet
where text like concat('%',g,'%');
END;//
DELIMITER ;

```

Navigator: SCHEMAS

Filter objects

- dailydeal
  - COUNT(\*)
- phototype
  - COUNT(\*)
- savedeal
  - user\_name
  - text
  - hashtags
  - id
  - type
  - date
  - followers
- texttype
  - COUNT(\*)
- usermentionsnum
  - COUNT(\*)
- Stored Procedures
  - Cal\_followers
  - Cal\_hashtags
  - explainDatabase
  - SearchDateBetween
  - tweetMatch
- Functions
  - youtube

Administration Schemas

Information

SQL File 4\* tweet Cal\_followers Cal\_hashtags SearchDateBetween tweetMatch

Limit to 1000 rows

1 • call twitter.tweetMatch('Overwatch');

2

Result Grid

	user_name	text	hashtags	id	type	date
Blizzard_Ent	RT @matthewmercer:	Some days, you wake up...	'Overwatch'	#1088136765796306944	text	2019-01-23
Blizzard_Ent	RT @PlayOverwatch:	Ring in the Year of the Pi...		#1088514296013762560	text	2019-01-24
Blizzard_Ent	RT @BAFTAGames:	We are coming to you LIVE ...		#1090353171002646530	text	2019-01-29
Blizzard_Ent	RT @overwatchleague:	2019 @Twitch All-Access...	OWL2019	#1091058078521339904	text	2019-01-31
Blizzard_Ent	RT @overwatchleague:	Community Countdown...		#1093558257531129863	text	2019-02-07
Blizzard_Ent	RT @overwatchleague:	Today is the day. #OW...	OWL2019	#1096148798223646722	text	2019-02-14
Blizzard_Ent	RT @PlayOverwatch:	Bon voyage! ____ Your jo...		#1097968930646343680	photo	2019-02-19
Blizzard_Ent	RT @PlayOverwatch:	__ Unauthorized Access ...		#1098660162540326913	text	2019-02-21
Blizzard_Ent	RT @PlayOverwatch:	Mobilize alongside the co...		#1100204605516869632	text	2019-02-26