A

Project Report on

Social Media Database Management System

Developed by

### KASHYAP GORASIYA(IT-037)

### GAUTAM GANDHI(IT-033)

**Guided By**

**Internal Guide: Prof. Archana N Vyas**

**Department of Information Technology Faculty of Technology**

**DD University**



**Department of Information Technology Faculty of Technology, Dharmsinh Desai University**

**College Road, Nadiad-387001 October-2019**

**ACKNOWLEDGEMENT**

We would like to give our sincere acknowledgement to everybody responsible for the successful completion of our project “SOCIAL MEDIA DATABASE MANAGEMENT SYSTEM”.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of this project.

We owe our deep gratitude to our project guide Prof. Archana Vyas, who took been interest on our project work and guided us all along till the completion of our project work by providing all the necessary help for developing a good Database System.

We would also like to thank all our lecturers.

Finally we convoy our acknowledgement to all our friends and family members who directly or indirectly associated with us in the successful completion of the project. We thank one and all.

TABLE OF CONTENTS

1. Certificate I
2. Acknowledgement II

[1. SYSTEM OVERVIEW .........................................................................................](#_TOC_250012)

* 1. [Current system 1](#_TOC_250011)
  2. [Objectives of the Proposed System 1](#_TOC_250010)
  3. Advantages of the Proposed system (over current) 1

1. [E-R DIAGRAM 6](#_TOC_250009)
2. [DATA DICTIONARY 7](#_TOC_250008)
3. [SCHEMA DIAGRAM 10](#_TOC_250007)

5. DATABASE IMPLEMENTION...........................................................................

* 1. [Create Schema 11](#_TOC_250006)
  2. Insert Data values 14
  3. Queries (Based on group by, having, constrain) 24
  4. Queries (joins, self joins, sub query)…………………………………………........28
  5. PL/SQL 34
  6. [Functions 35](#_TOC_250005)
  7. [Triggers 36](#_TOC_250004)
  8. [Cursors 40](#_TOC_250003)

1. [FUTURE ENHANCEMENTS OF THE SYSTEM 41](#_TOC_250001)
2. [BIBLIOGRAPHY 42](#_TOC_250000)

**DHARMSINH DESAI UNIVERSITY**

## NADIAD-387001, GUJARAT



**CERTIFICATE**

This is to certify that the project entitled “**Social Media Database Management System**” is a bonafide report of the work carried out by

1. **Mr. GANDHI GAUTAM ,** Student ID No : **19ITUOF015**
2. **Mr. GORASIYA KASHYAP** , Student ID No : **19ITUEF014**

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2020-2021.

### Prof. Archana N Vyas.

(Project Guide)

Department of Information Technology, Faculty of Technology,

Dharmsinh Desai University, Nadiad Date:

### Prof. Vipul Dabhi

Head , Department of Information Technology, Faculty of Technology,

Dharmsinh Desai University, Nadiad Date:

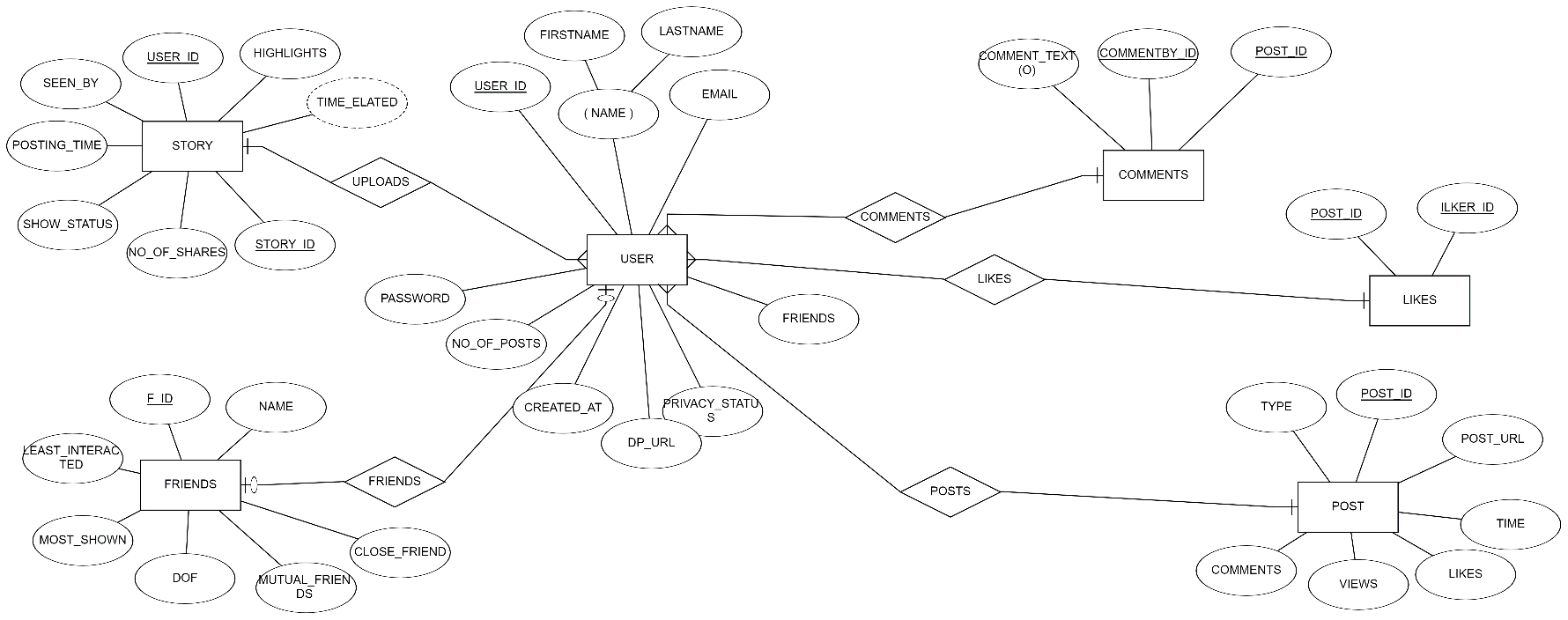
# SYSTEM OVERVIEW

1)Social Media Database System keeps track of almost everything a real life social media deals with. (eg: Facebook).

2)It keeps the track of users, their friends ,stories, posts and much more.

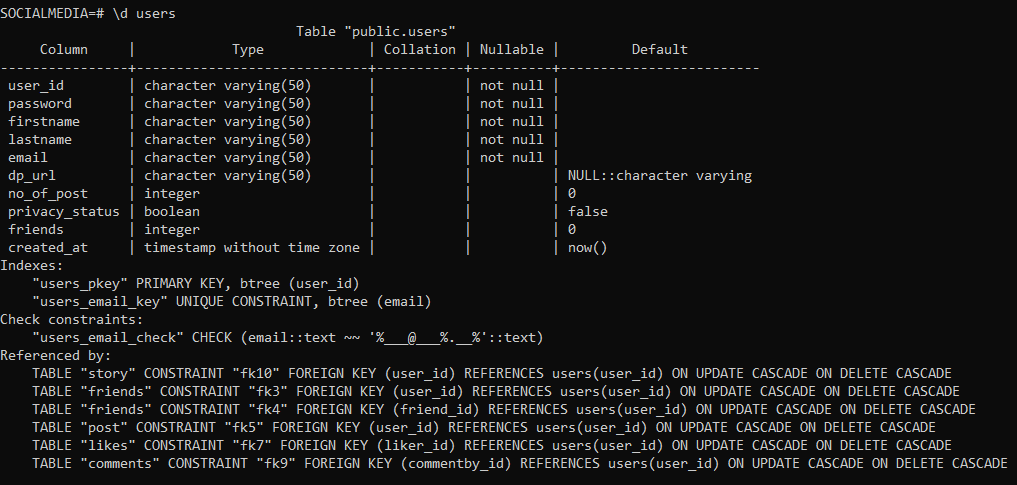
3)All the basic queries as well as complex pl/sql , cursors and triggers have been implemented in order to prove the effectiveness of the database Management system.

# 2.E-R DIAGRAM

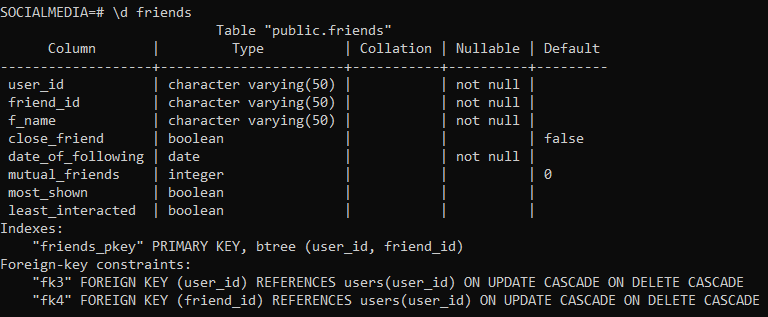


# DATA DICTIONARY

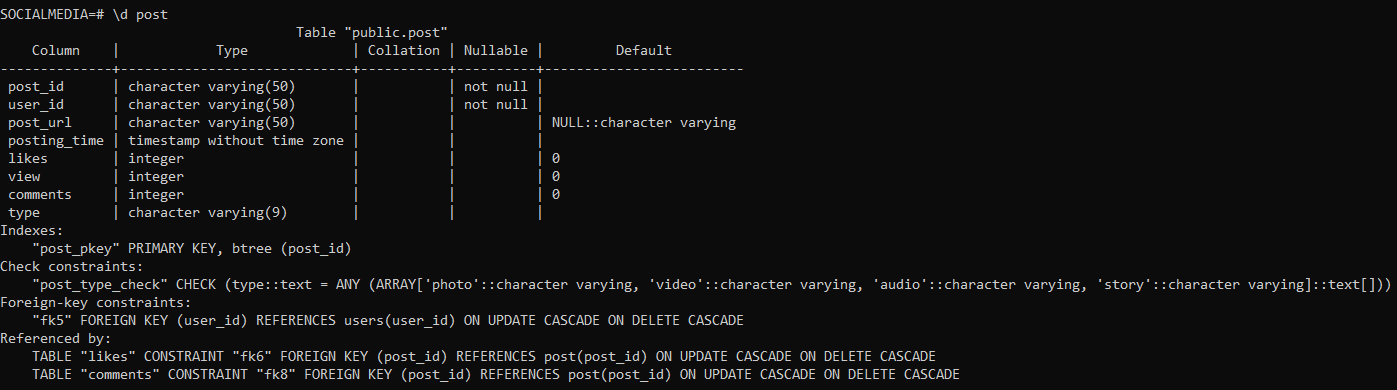
## USERS



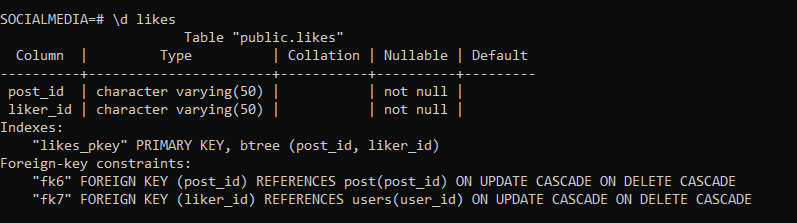
* 1. **FRIENDS**



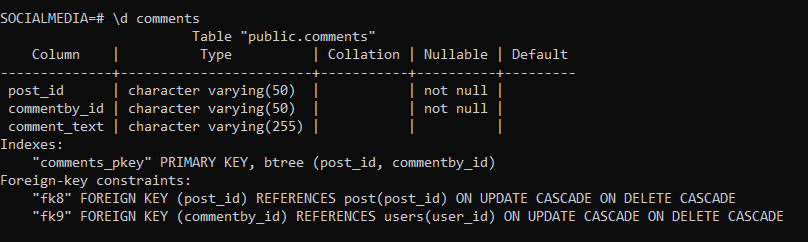
* 1. **POST**



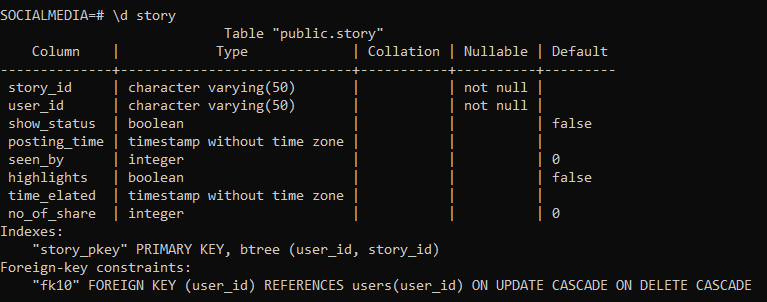
* 1. **LIKES**



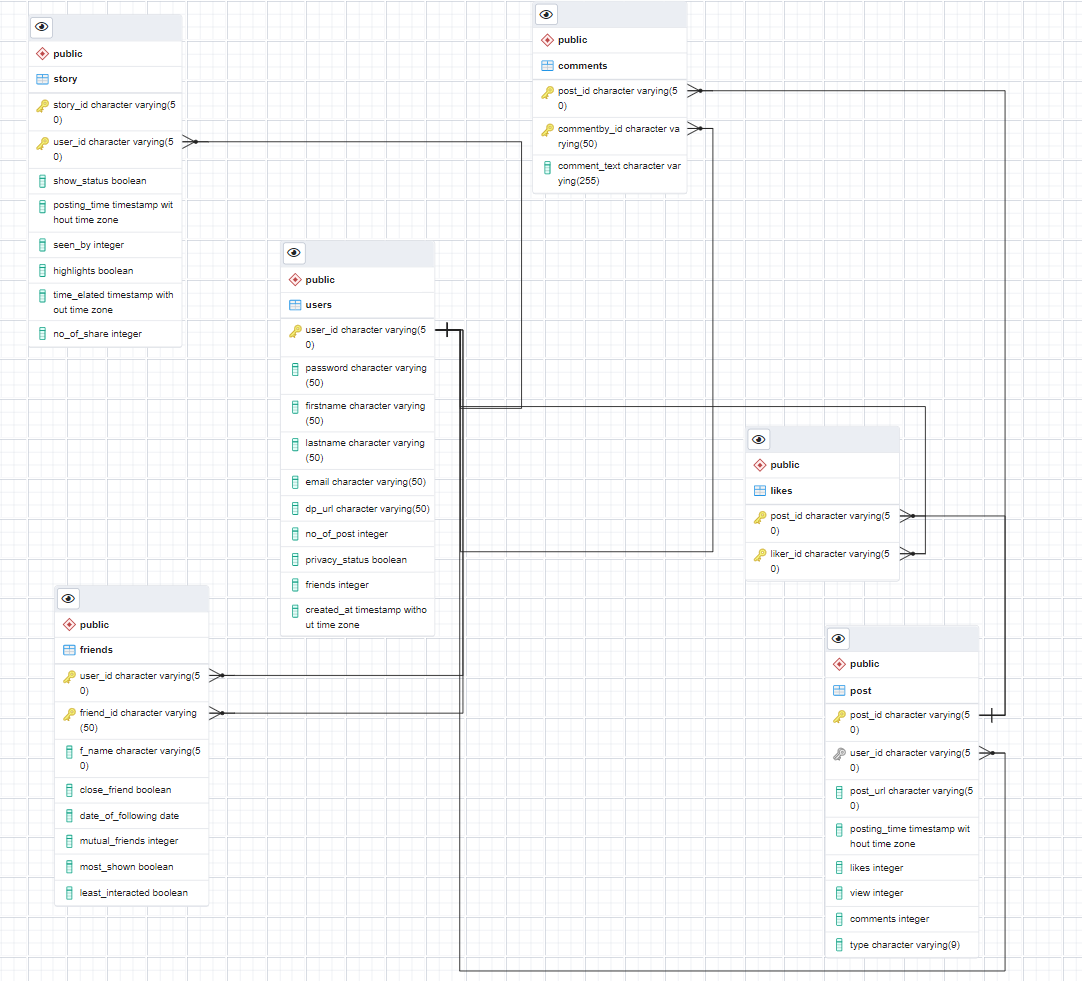
* 1. **COMMENT**



* 1. **STORY**



# SCHEMA DIAGRAM



1. **DATABASE IMPLEMENTATION**

## CREATE SCHEMA

### USERS

CREATE TABLE users (

user\_id VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL,

firstname VARCHAR(50) NOT NULL,

lastname VARCHAR(50) NOT NULL,

email VARCHAR(50) CHECK(email LIKE '%\_\_\_@\_\_\_%.\_\_%') UNIQUE NOT NULL,

dp\_url VARCHAR(50) DEFAULT NULL,

no\_of\_post INT DEFAULT 0,

privacy\_status BOOLEAN DEFAULT FALSE,

friends INT DEFAULT 0,

created\_at TIMESTAMP DEFAULT NOW(),

PRIMARY KEY(user\_id)

);

### FRIENDS

### CREATE TABLE friends(

### user\_id VARCHAR(50) NOT NULL,

### friend\_id VARCHAR(50) NOT NULL,

### f\_name VARCHAR(50)NOT NULL,

### close\_friend BOOLEAN DEFAULT FALSE,

### date\_of\_following DATE NOT NULL,

### mutual\_friends INT DEFAULT 0,

### most\_shown BOOLEAN,

### least\_interacted BOOLEAN,

### 

### PRIMARY KEY(user\_id,friend\_id),

### CONSTRAINT fk3 FOREIGN KEY(user\_id) REFERENCES users(user\_id) ON DELETE CASCADE ON UPDATE CASCADE,

### CONSTRAINT fk4 FOREIGN KEY(friend\_id) REFERENCES users(user\_id) ON DELETE CASCADE ON UPDATE CASCADE

### );

### POST

CREATE TABLE post(

post\_id VARCHAR(50) NOT NULL ,

user\_id VARCHAR(50) NOT NULL,

post\_url VARCHAR(50) NULL DEFAULT NULL ,

posting\_time TIMESTAMP ,

likes INT DEFAULT 0,

view INT DEFAULT 0,

comments INT DEFAULT 0,

type VARCHAR(9) check(type IN ('photo', 'video','audio','story')),

PRIMARY KEY(post\_id),

CONSTRAINT fk5 FOREIGN KEY(user\_id) REFERENCES users(user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

### LIKES

CREATE TABLE likes(

post\_id VARCHAR(50) NOT NULL,

liker\_id VARCHAR(50) NOT NULL,

PRIMARY KEY(post\_id,liker\_id),

CONSTRAINT fk6 FOREIGN KEY(post\_id) REFERENCES post(post\_id) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk7 FOREIGN KEY(liker\_id) REFERENCES users(user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

### COMMENTS

CREATE TABLE comments(

post\_id VARCHAR(50) NOT NULL,

commentby\_id VARCHAR(50) NOT NULL,

comment\_text VARCHAR(255) ,

PRIMARY KEY(post\_id,commentby\_id),

CONSTRAINT fk8 FOREIGN KEY(post\_id) REFERENCES post(post\_id) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk9 FOREIGN KEY(commentby\_id) REFERENCES users(user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

### STORY

CREATE TABLE story(

story\_id VARCHAR(50) NOT NULL,

user\_id VARCHAR(50) NOT NULL,

show\_status BOOLEAN DEFAULT FALSE,

posting\_time TIMESTAMP ,

seen\_by INT DEFAULT 0,

highlights BOOLEAN DEFAULT FALSE,

time\_elated TIMESTAMP ,

no\_of\_share INT DEFAULT 0,

PRIMARY KEY(user\_id,story\_id),

CONSTRAINT fk10 FOREIGN KEY(user\_id) REFERENCES users(user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

## INSERT DATA VALUE

### USERS

INSERT INTO users(user\_id , password , firstname , lastname , email , dp\_url , no\_of\_post , privacy\_status , friends , created\_at )VALUES

('USR01','Pass@USR1','Raj','Sharma','rajsharma@gmail.com','IC/DP/USRDP01.png',2,FALSE,4,'2010-06-02 01:10:25-03'),

('USR02','Pass@USR2','Yash','Varma','yashvarma@gmail.com','IC/DP/USRDP02.png',1,TRUE,4,'2019-07-22 05:05:25-02'),

('USR03','Pass@USR3','Pooja','Maheta','poojamaheta@gmail.com','IC/DP/USRDP03.png',null,FALSE,null,'2020-08-08 09:10:25-07'),

('USR04','Pass@USR4','Riya','Kotak','riyakotak@gmail.com','IC/DP/USRDP04.png',3,TRUE,4,'2011-05-22 10:02:25-01'),

('USR05','Pass@USR5','Parth','Bhatt','parthbhatt@gmail.com','IC/DP/USRDP05.png',null,FALSE,3,'2016-06-15 02:06:25-06'),

('USR06','Pass@USR6','Alice','Shah','aliceshah@gmail.com','IC/DP/USRDP06.png',null,TRUE,2,'2018-06-16 08:13:25-04'),

('USR07','Pass@USR7','Akash','Benarji','akashbenarji@gmail.com','IC/DP/USRDP07.png',1,FALSE,5,'2017-04-06 19:15:25-07'),

('USR08','Pass@USR8','Juli','Kapoor','julikapoor@gmail.com','IC/DP/USRDP08.png',1,FALSE,3,'2018-09-12 03:56:25-05'),

('USR09','Pass@USR9','Rahul','Agrawal','rahulagrawal@gmail.com','IC/DP/USRDP09.png',1,TRUE,2,'2017-06-07 04:10:25-09'),

('USR010','Pass@USR10','Kajal','Adani','kajaladani@gmail.com','IC/DP/USRDP010.png',null,FALSE,1,'2018-06-22 17:20:25-08'),

('USR011','Pass@USR11','Pratik','Gandhi','pratikgandhi@gmail.com','IC/DP/USRDP011.png',1,TRUE,null,'2019-06-04 19:57:25-07'),

('USR012','Pass@USR12','Jitu','Kumar','jitukumar@gmail.com','IC/DP/USRDP012.png',null,FALSE,1,'2019-12-25 16:27:25-07'),

('USR013','Pass@USR13','Vaibhav','Roshan','vaibhavroshan@gmail.com','IC/DP/USRDP013.png',null,TRUE,null,'2019-10-31 18:40:25-05'),

('USR014','Pass@USR14','Minal','Gada','minalgada@gmail.com','IC/DP/USRDP014.png',2,FALSE,null,'2017-06-22 06:51:25-04'),

('USR015','Pass@USR15','Vartika','Hathi','vartikahathi@gmail.com','IC/DP/USRDP015.png',null,TRUE,2,'2016-04-30 12:10:25-04'),

('USR016','Pass@USR16','Meena','Aiyar','meenaaiyar@gmail.com','IC/DP/USRDP016.png',null,FALSE,null,'2016-03-14 21:37:25-06'),

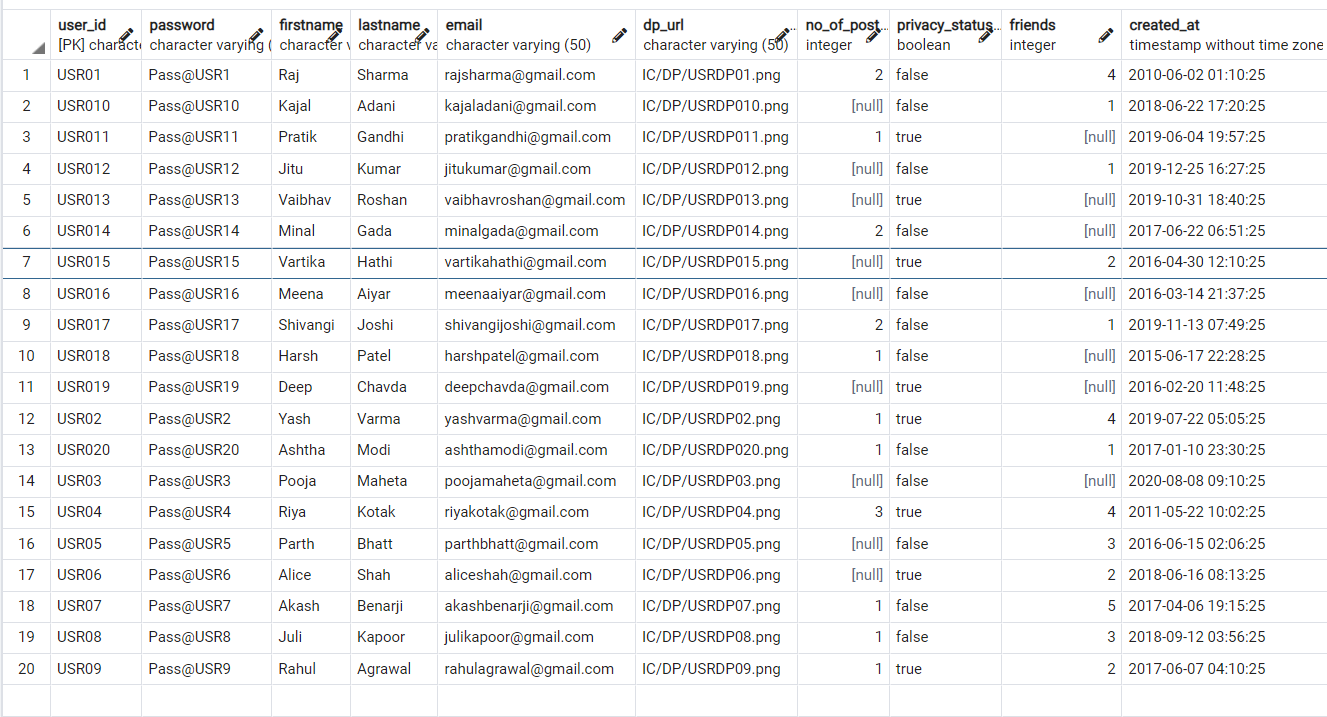
('USR017','Pass@USR17','Shivangi','Joshi','shivangijoshi@gmail.com','IC/DP/USRDP017.png',2,FALSE,1,'2019-11-13 07:49:25-07'),

('USR018','Pass@USR18','Harsh','Patel','harshpatel@gmail.com','IC/DP/USRDP018.png',1,FALSE,null,'2015-06-17 22:28:25-01'),

('USR019','Pass@USR19','Deep','Chavda','deepchavda@gmail.com','IC/DP/USRDP019.png',null,TRUE,null,'2016-02-20 11:48:25-08'),

('USR020','Pass@USR20','Ashtha','Modi','ashthamodi@gmail.com','IC/DP/USRDP020.png',1,FALSE,1,'2017-01-10 23:30:25-09')

;



### FRIENDS

INSERT INTO friends( user\_id , friend\_id , f\_name , close\_friend , date\_of\_following , mutual\_friends , most\_shown , least\_interacted)VALUES

('USR01','USR02','Yash Varma',TRUE,'2019-08-23',1,FALSE,TRUE),

('USR01','USR020','Ashtha Modi',FALSE,'2018-09-02',0,FALSE,TRUE),

('USR01','USR012','Jitu Kumar',FALSE,'2020-01-02',1,TRUE,FALSE),

('USR01','USR04','Riya Kotak',TRUE,'2012-03-15',1,TRUE,FALSE),

('USR02','USR01','Raj Sharma',TRUE,'2019-10-30',1,TRUE,FALSE),

('USR02','USR03','Pooja Maheta',FALSE,'2020-12-16',0,FALSE,TRUE),

('USR02','USR04','Riya Kotak',TRUE,'2020-06-07',0,FALSE,TRUE),

('USR02','USR010','Kajal Adani',FALSE,'2021-04-05',0,FALSE,TRUE),

('USR04','USR02','Yash Varma',FALSE,'2020-06-07',0,FALSE,TRUE),

('USR04','USR05','Parth Bhatt',FALSE,'2017-07-17',0,FALSE,TRUE),

('USR04','USR08','Parth Bhatt',FALSE,'2019-09-19',0,TRUE,FALSE),

('USR04','USR07','Akash Benarji',TRUE,'2018-12-02',1,TRUE,FALSE),

('USR05','USR01','Raj Sharma',FALSE,'2016-06-22',1,FALSE,TRUE),

('USR05','USR04','Riya Kotak',FALSE,'2016-06-22',0,FALSE,TRUE),

('USR05','USR010','Kajal Adani',TRUE,'2019-01-30',0,TRUE,FALSE),

('USR06','USR09','Rahul Agrawal',FALSE,'2018-09-19',0,TRUE,FALSE),

('USR06','USR011','Pratik Gandhi',FALSE,'2020-10-20',0,FALSE,TRUE),

('USR07','USR02','Yash Varma',TRUE,'2019-10-10',0,TRUE,FALSE),

('USR07','USR012','Jitu Kumar',TRUE,'2020-04-06',1,FALSE,TRUE),

('USR07','USR013','Vaibhav Roshan',FALSE,'2021-08-12',0,TRUE,FALSE),

('USR07','USR018','Kajal Adani',FALSE,'2017-04-06',1,FALSE,TRUE),

('USR07','USR020','Ashtha Modi',TRUE,'2017-11-02',0,FALSE,TRUE),

('USR08','USR01','Raj Sharma',FALSE,'2018-06-22',1,TRUE,FALSE),

('USR08','USR04','Riya Kotak',FALSE,'2019-09-09',0,FALSE,TRUE),

('USR08','USR016','Meena Aiyar',TRUE,'2020-06-20',0,FALSE,TRUE),

('USR09','USR05','Parth Bhatt',TRUE,'2016-06-06',0,TRUE,FALSE),

('USR09','USR014','Mina Gada',TRUE,'2018-06-16',0,TRUE,FALSE),

('USR010','USR06','Alice Shah',TRUE,'2020-01-01',0,TRUE,FALSE),

('USR012','USR02','Yash Varma',TRUE,'2019-12-25',0,FALSE,TRUE),

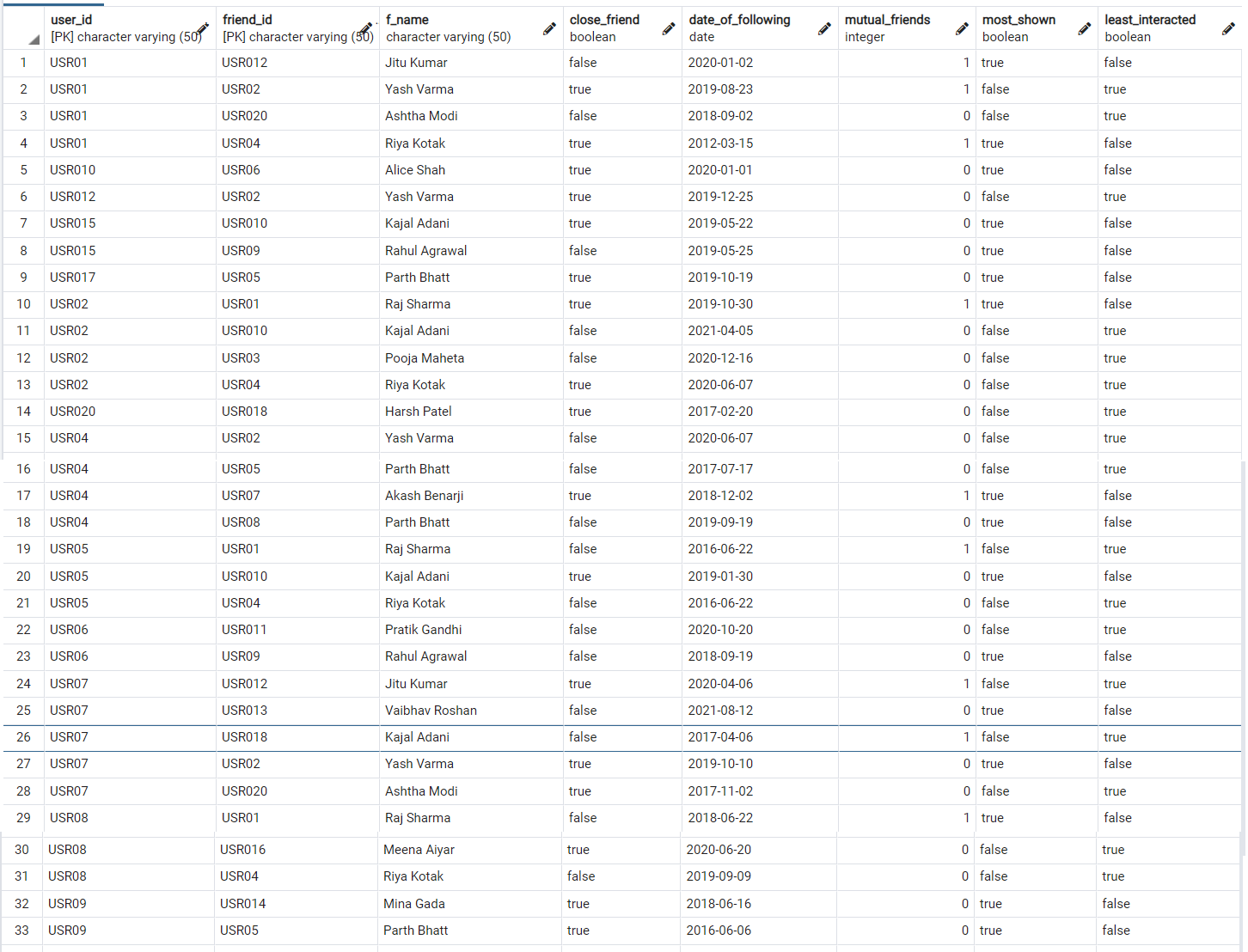
('USR015','USR010','Kajal Adani',TRUE,'2019-05-22',0,TRUE,FALSE),

('USR015','USR09','Rahul Agrawal',FALSE,'2019-05-25',0,TRUE,FALSE),

('USR017','USR05','Parth Bhatt',TRUE,'2019-10-19',0,TRUE,FALSE),

('USR020','USR018','Harsh Patel',TRUE,'2017-02-20',0,FALSE,TRUE)

;



### POST

INSERT INTO post( post\_id , user\_id , post\_url , posting\_time , likes , view , comments , type )VALUES

('USR01PST01','USR01','IC/POST/POSTUSR01/POST01USR01.png','2011-05-02 01:10:25-03',1,8,1,'photo'),

('USR01PST02','USR01','IC/POST/POSTUSR01/POST02USR01.png','2020-03-20 10:00:25-03',2,7,0,'video'),

('USR02PST01','USR02','IC/POST/POSTUSR02/POST01USR01.png','2019-07-23 05:01:03-02',1,6,1,'photo'),

('USR04PST01','USR04','IC/POST/POSTUSR04/POST01USR04.png','2012-05-22 11:059:20-01',1,5,0,'photo'),

('USR04PST02','USR04','IC/POST/POSTUSR04/POST02USR04.png','2015-11-02 12:010:05-01',2,4,1,'audio'),

('USR04PST03','USR04','IC/POST/POSTUSR04/POST03USR04.png','2019-12-20 13:02:00-01',1,3,0,'video'),

('USR07PST01','USR07','IC/POST/POSTUSR07/POST01USR07.png','2017-09-01 23:50:25-07',2,2,2,'photo'),

('USR08PST01','USR08','IC/POST/POSTUSR08/POST01USR08.png','2019-09-12 03:06:00-05',1,1,0,'photo'),

('USR09PST01','USR09','IC/POST/POSTUSR09/POST01USR09.png','2018-09-04 02:10:56-09',2,8,1,'photo'),

('USR11PST01','USR011','IC/POST/POSTUSR11/POST01USR11.png','2019-12-25 10:52:25-07',1,4,0,'video'),

('USR14PST01','USR014','IC/POST/POSTUSR14/POST01USR14.png','2018-06-22 06:52:26-04',2,7,1,'audio'),

('USR14PST02','USR014','IC/POST/POSTUSR14/POST02USR14.png','2020-06-22 06:50:27-04',1,8,0,'photo'),

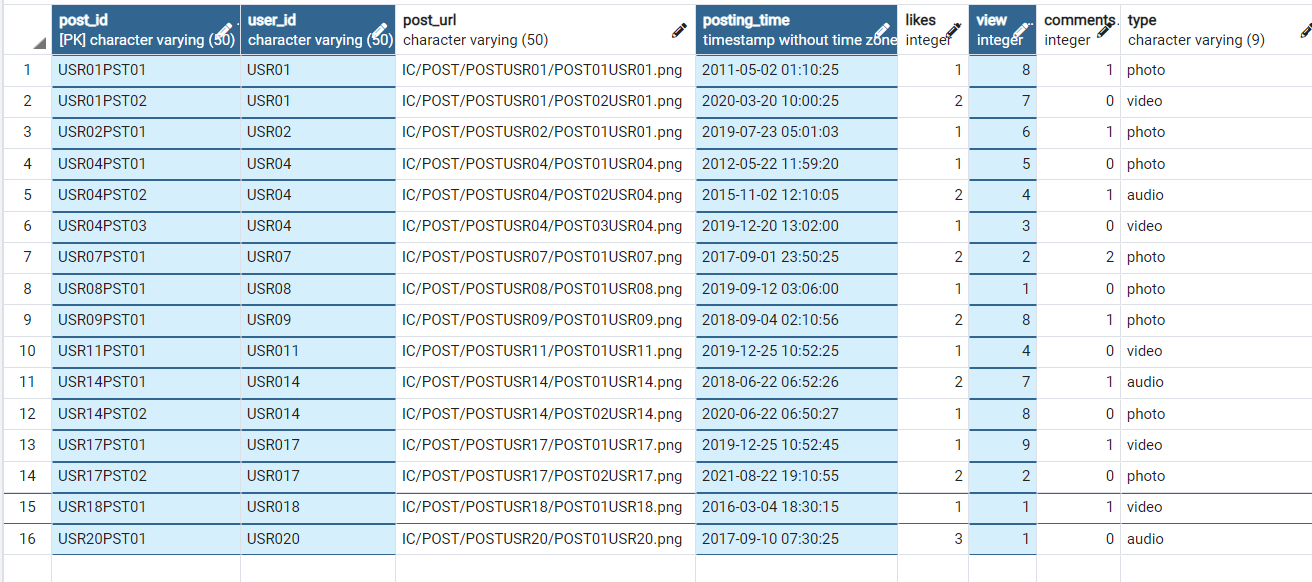
('USR17PST01','USR017','IC/POST/POSTUSR17/POST01USR17.png','2019-12-25 10:52:45-07',1,9,1,'video'),

('USR17PST02','USR017','IC/POST/POSTUSR17/POST02USR17.png','2021-08-22 19:10:55-07',2,2,0,'photo'),

('USR18PST01','USR018','IC/POST/POSTUSR18/POST01USR18.png','2016-03-04 18:30:15-08',1,1,1,'video'),

('USR20PST01','USR020','IC/POST/POSTUSR20/POST01USR20.png','2017-09-10 07:30:25-09',3,1,0,'audio')

;



### LIKES

INSERT INTO likes( post\_id , liker\_id )VALUES

('USR01PST01','USR01'),

('USR01PST02','USR011'),

('USR01PST02','USR014'),

('USR02PST01','USR01'),

('USR04PST01','USR02'),

('USR04PST02','USR07'),

('USR04PST02','USR08'),

('USR04PST03','USR05'),

('USR07PST01','USR011'),

('USR07PST01','USR012'),

('USR08PST01','USR013'),

('USR09PST01','USR014'),

('USR09PST01','USR05'),

('USR11PST01','USR016'),

('USR14PST01','USR017'),

('USR14PST01','USR018'),

('USR14PST02','USR015'),

('USR17PST01','USR016'),

('USR17PST02','USR01'),

('USR17PST02','USR017'),

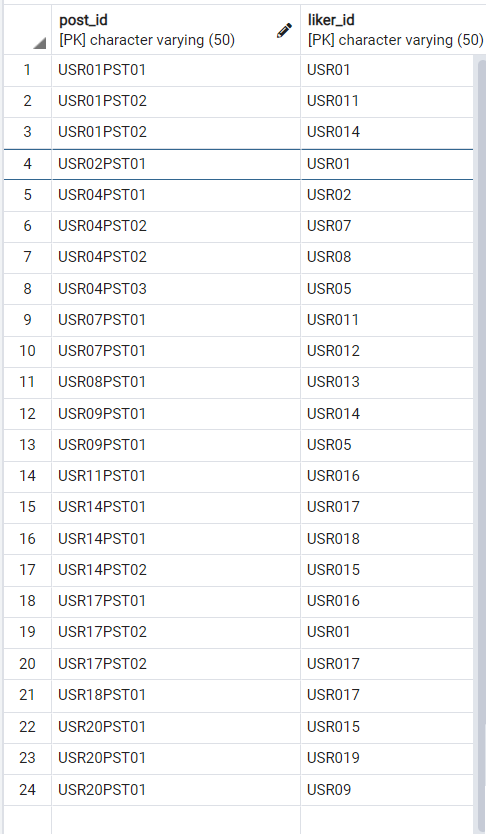
('USR18PST01','USR017'),

('USR20PST01','USR019'),

('USR20PST01','USR09'),

('USR20PST01','USR015')

;



### COMMENT

### INSERT INTO comments( post\_id, commentby\_id , comment\_text ) VALUES

### ('USR01PST01','USR020','WOWWWWWW'),

### ('USR02PST01','USR04','Awesome'),

### ('USR04PST02','USR08','Great'),

### ('USR07PST01','USR09','Good'),

### ('USR07PST01','USR020','Superb'),

### ('USR09PST01','USR05','Nice'),

### ('USR14PST01','USR011','Alas'),

### ('USR17PST01','USR012','Best wishes'),

### ('USR18PST01','USR019','Happy Birthday')

### ;

### 

### STORY

INSERT INTO story( story\_id , user\_id , show\_status , posting\_time , seen\_by , highlights, no\_of\_share )VALUES

('STORYUSR01','USR020',TRUE,'2021-06-22 04:10:25-07',3,TRUE,1),

('STORYUSR03','USR01',TRUE,'2021-06-22 06:10:34-07',9,TRUE,0),

('STORYUSR04','USR07',TRUE,'2021-06-22 23:10:10-07',7,FALSE,3),

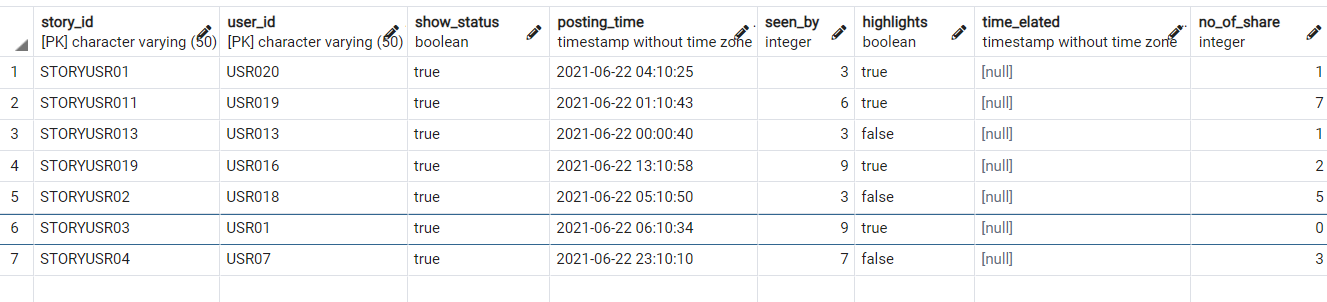
('STORYUSR02','USR018',TRUE,'2021-06-22 05:10:50-07',3,FALSE,5),

('STORYUSR011','USR019',TRUE,'2021-06-22 01:10:43-07',6,TRUE,7),

('STORYUSR019','USR016',TRUE,'2021-06-22 13:10:58-07',9,TRUE,2),

('STORYUSR013','USR013',TRUE,'2021-06-22 00:00:40-07',3,FALSE,1)

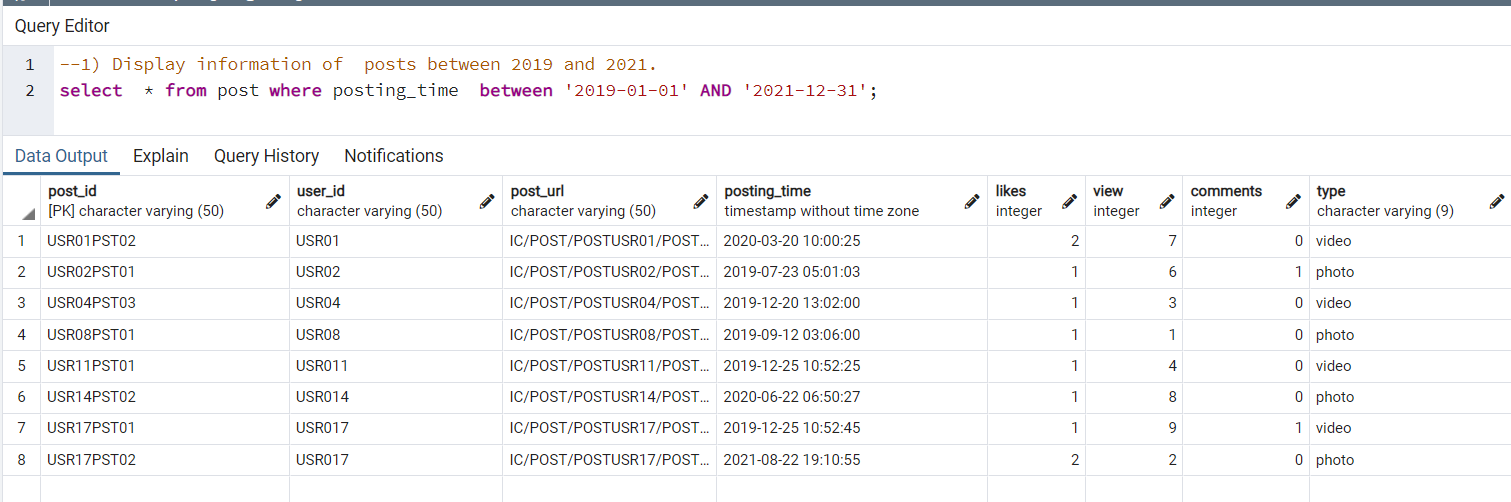
;



## BASIC QUERIES

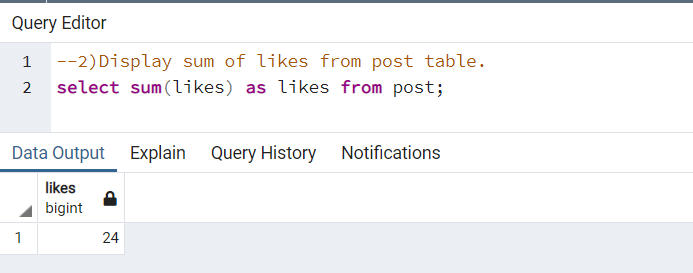
**1) Display information of posts between 2019 and 2021.**

select \* from post where posting\_time between '2019-01-01' AND '2021-12-31';



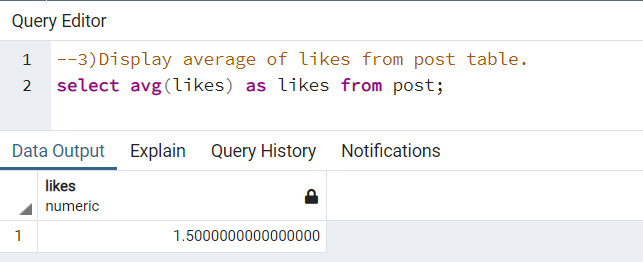
**2)Display sum of likes from post table.**

select sum(likes) as likes from post;



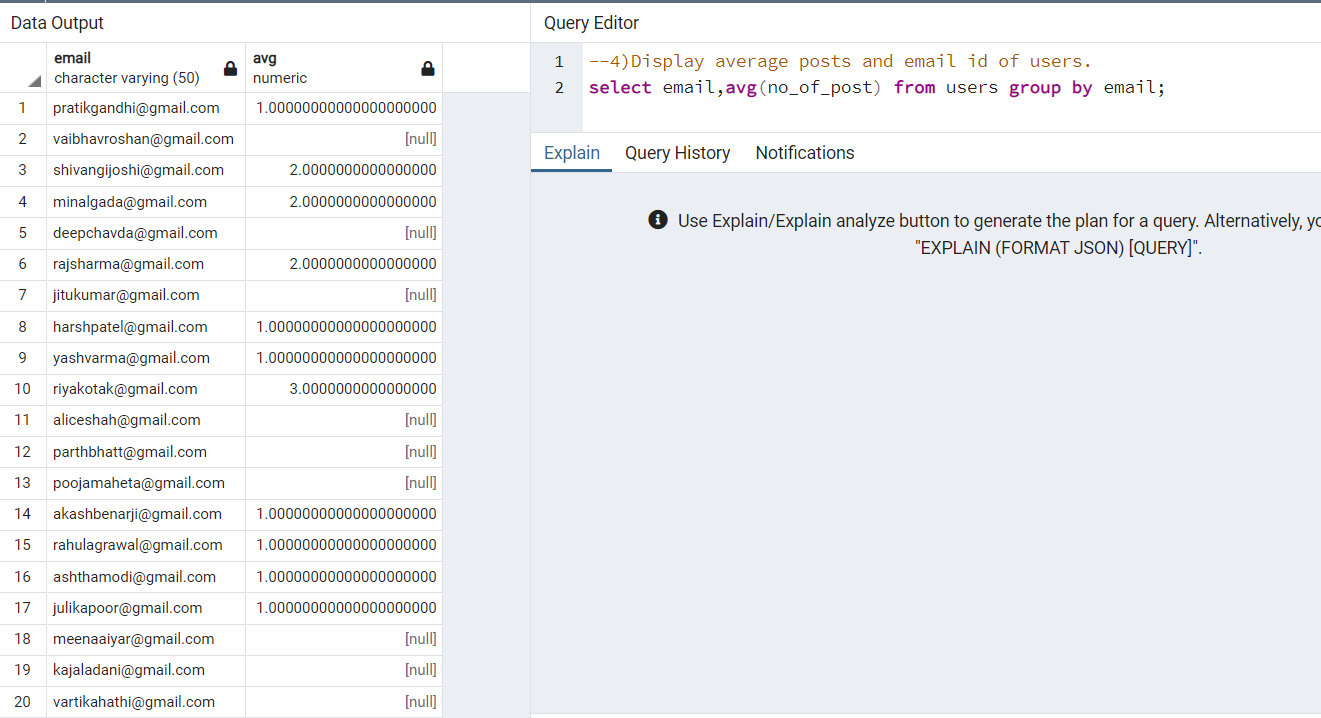
**3)Display average of likes from post table.**

select avg(likes) as likes from post;



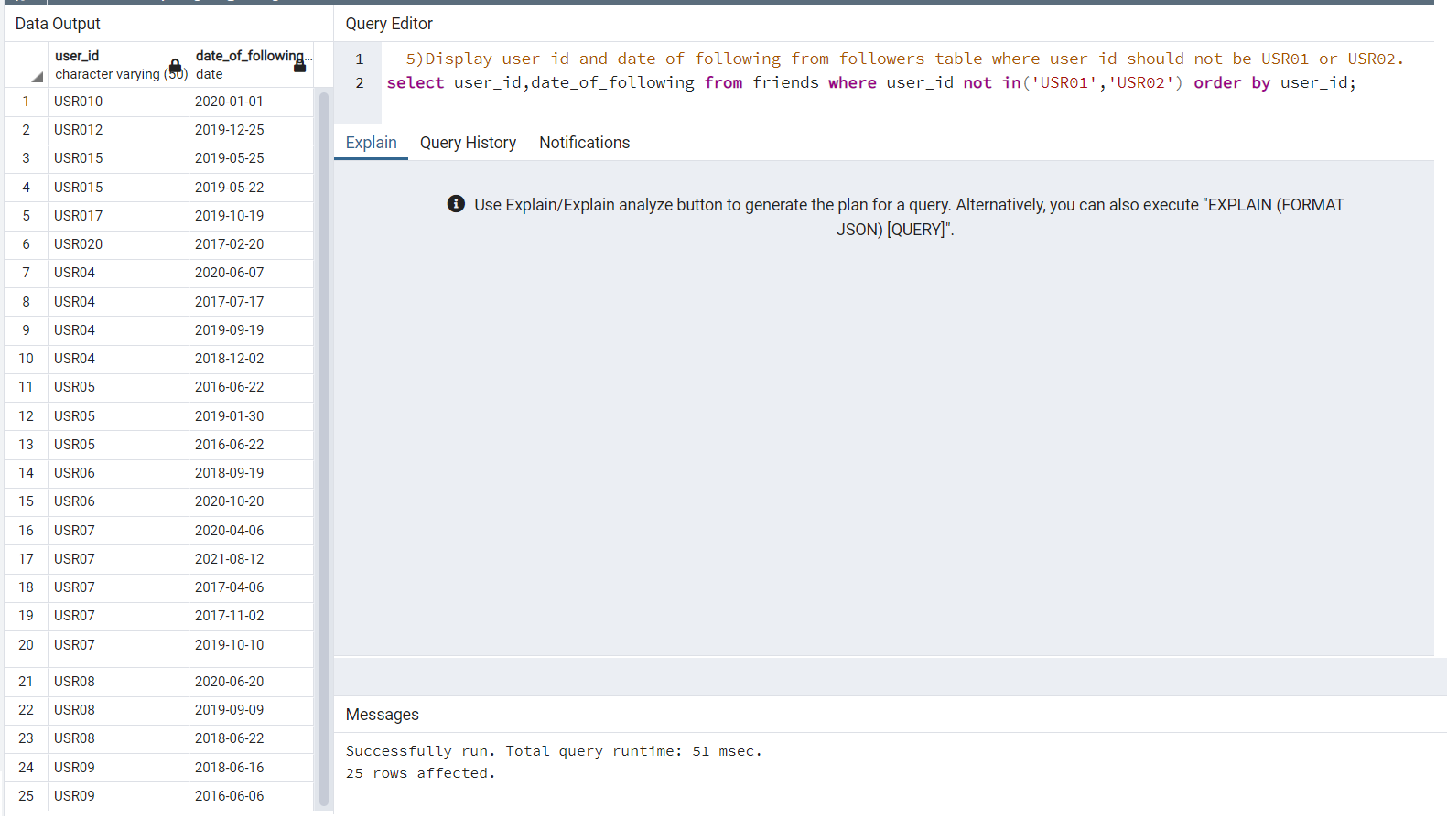
**4)Display average posts and email id of users.**

select email,avg(no\_of\_post) from users group by email;



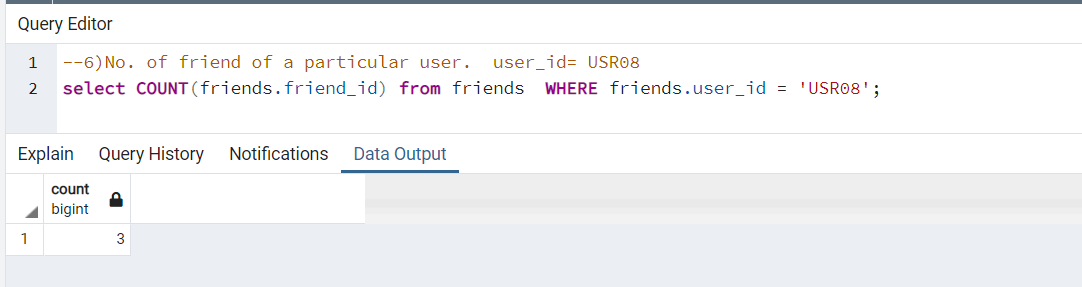
**5)Display user id and date of following from followers table where user id should not be USR01 or USR02.**

select user\_id,date\_of\_following from friends where user\_id not in('USR01','USR02') order by user\_id;



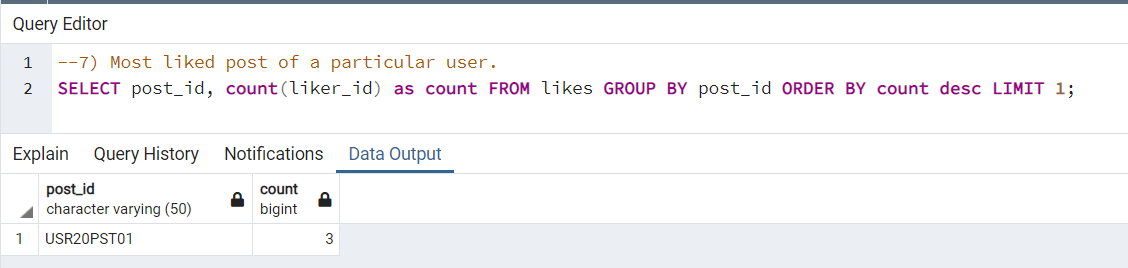
**6)No. of friend of a particular user. user\_id= USR08**

select COUNT(friends.friend\_id) from friends WHERE friends.user\_id = 'USR08';



**7) Most liked post of a particular user.**

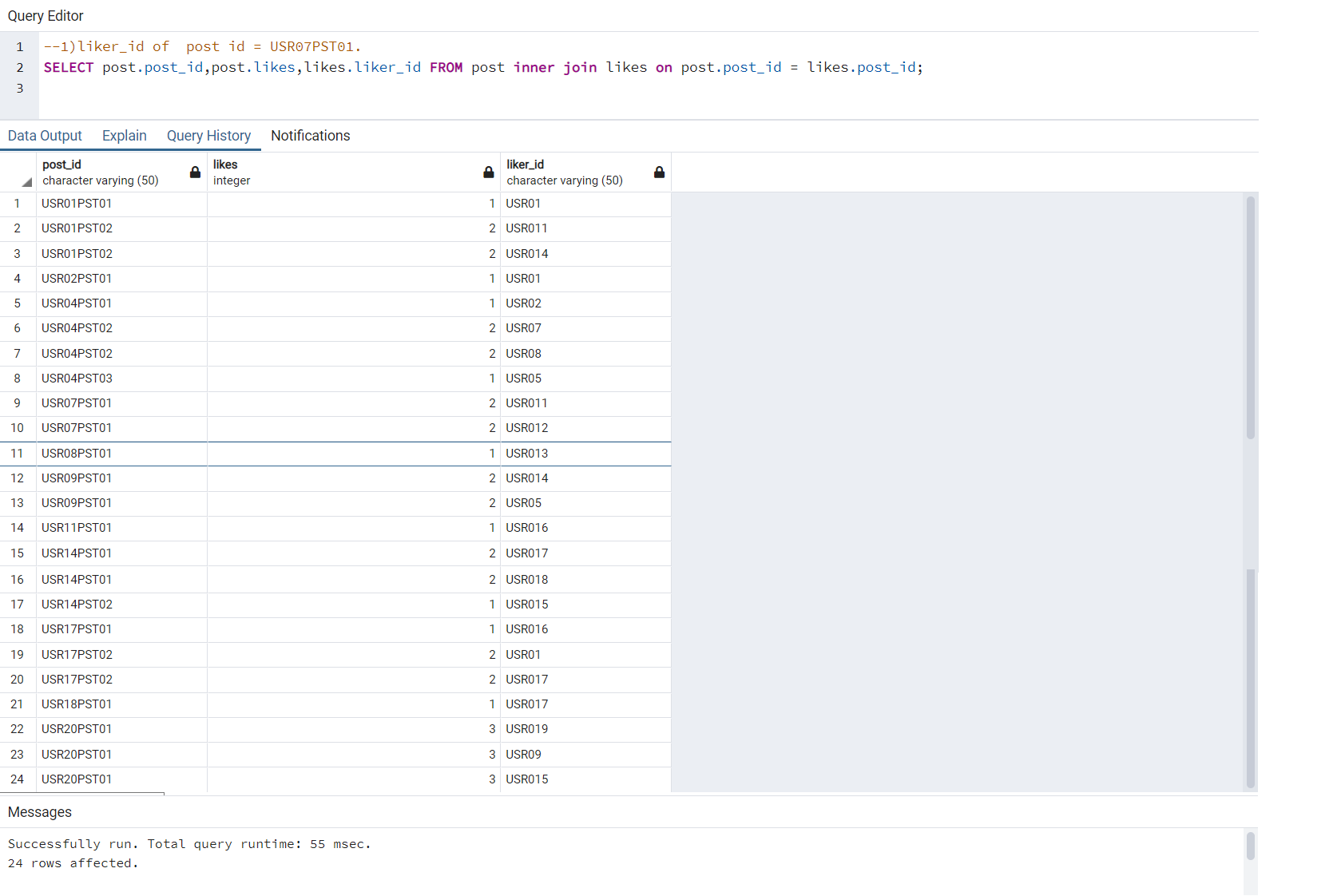
SELECT post\_id, count(liker\_id) as count FROM likes GROUP BY post\_id ORDER BY count desc LIMIT 1;



**5.4 Queries (self join & join & subqueries)**

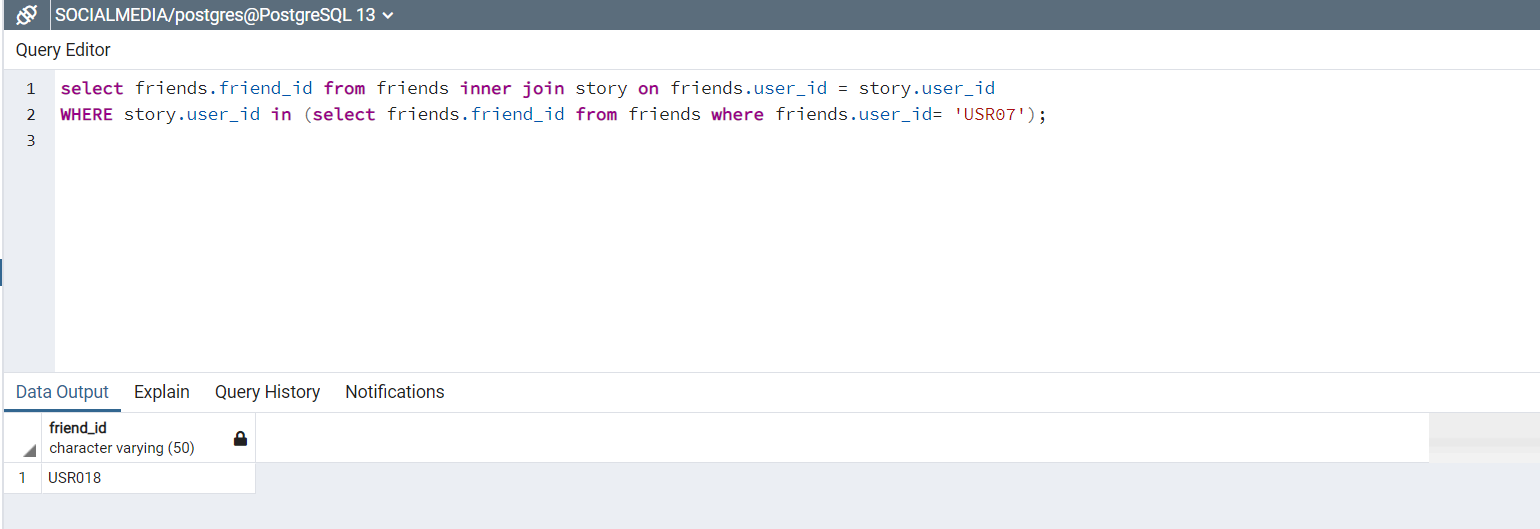
**1)liker\_id of post id = USR07PST01.**

SELECT post.post\_id,post.likes,likes.liker\_id FROM post inner join likes on post.post\_id = likes.post\_id;



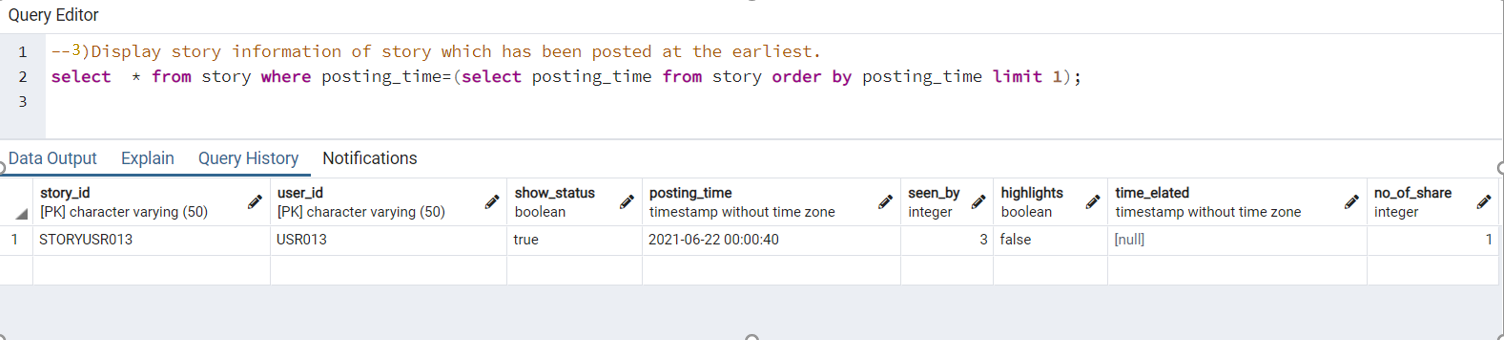
**2)Display name of friends of user='USR07' who had posted story?**

select friends.friend\_id from friends inner join story on friends.user\_id = story.user\_id WHERE story.user\_id in (select friends.friend\_id from friends where friends.user\_id= 'USR07');



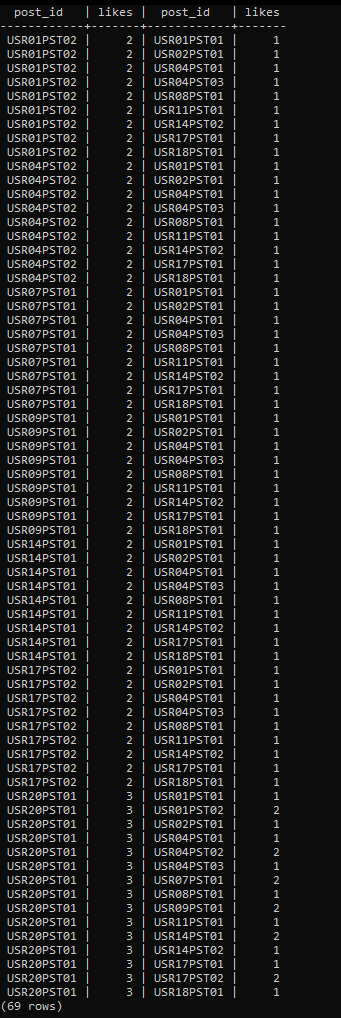
**3)Display story information of story which has been posted at the earliest.**

select \* from story where posting\_time=(select posting\_time from story order by posting\_time limit 1);

****

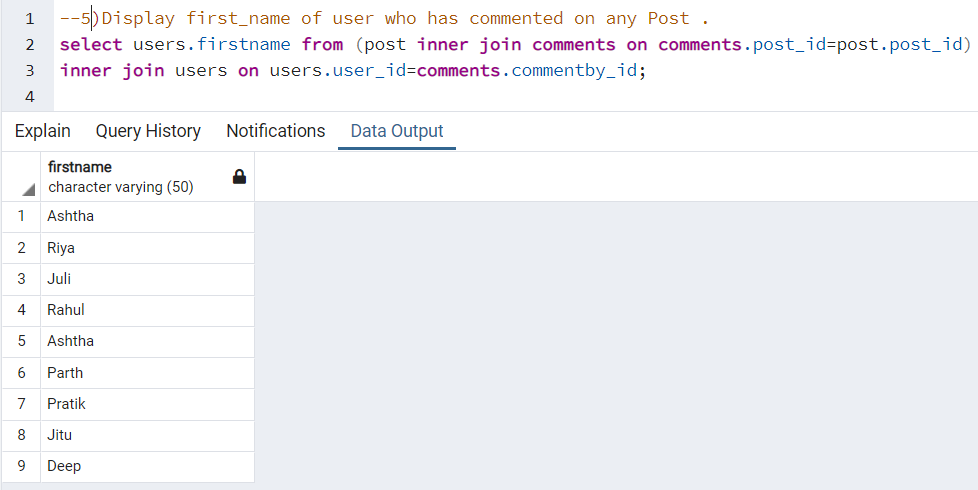
**4)Display post\_id ,likes from post table where no. of likes of one is greater than another.**

select a.post\_id,a.likes,b.post\_id,b.likes from post a, post b where a.likes>b.likes;



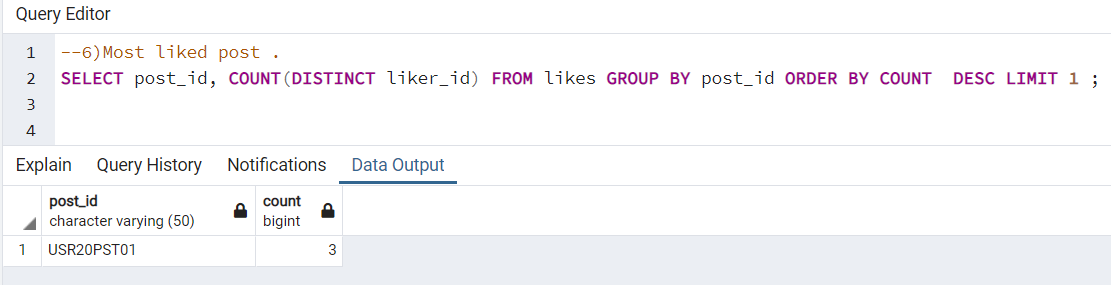
**5)Display first\_name of user who has commented on any Post .**

select users.firstname from (post inner join comments on comments.post\_id=post.post\_id) inner join users on users.user\_id=comments.commentby\_id;

****

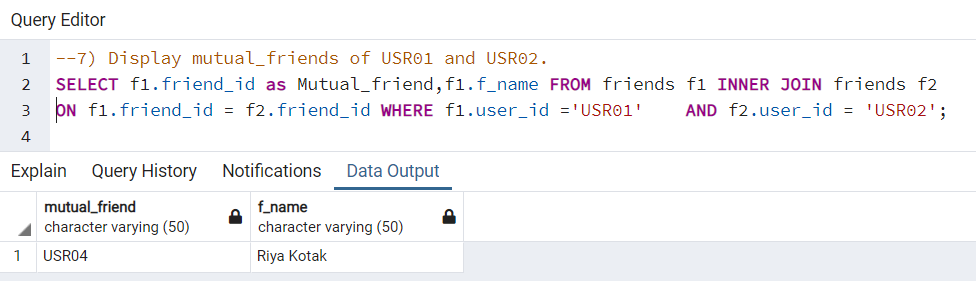
**6)Most liked post .**

SELECT post\_id, COUNT(DISTINCT liker\_id) FROM likes GROUP BY post\_id ORDER BY COUNT DESC LIMIT 1 ;



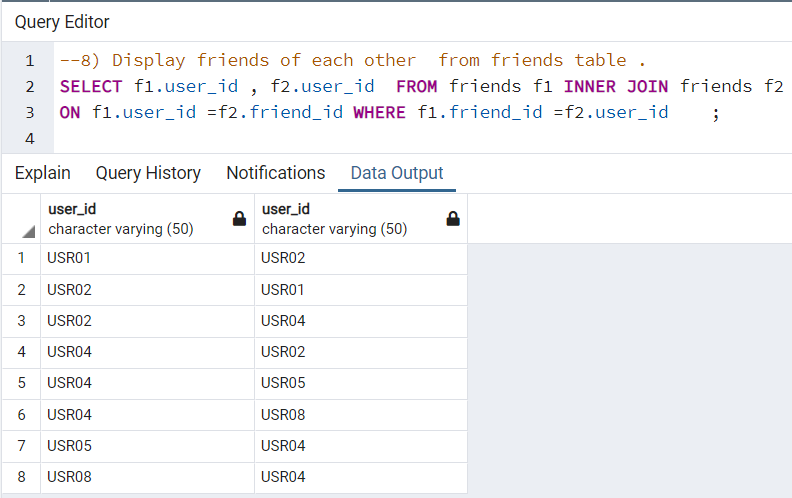
**7) Display mutual\_friends of USR01 and USR02.**

SELECT f1.friend\_id as Mutual\_friend,f1.f\_name FROM friends f1 INNER JOIN friends f2 ON f1.friend\_id = f2.friend\_id WHERE f1.user\_id ='USR01' AND f2.user\_id = 'USR02';



**8) Display friends of each other from friends table .**

SELECT f1.user\_id , f2.user\_id FROM friends f1 INNER JOIN friends f2 ON f1.user\_id =f2.friend\_id WHERE f1.friend\_id =f2.user\_id ;

****

**9)searching for users that aren't already friends.**

SELECT f2.user\_id ,COUNT(\*) as friends\_in\_common FROM friends f1 LEFT JOIN friends f2 ON f1.friend\_id = f2.friend\_id

WHERE f1.user\_id = 'USR07' GROUP BY f2.user\_id ORDER BY friends\_in\_common DESC ;

****

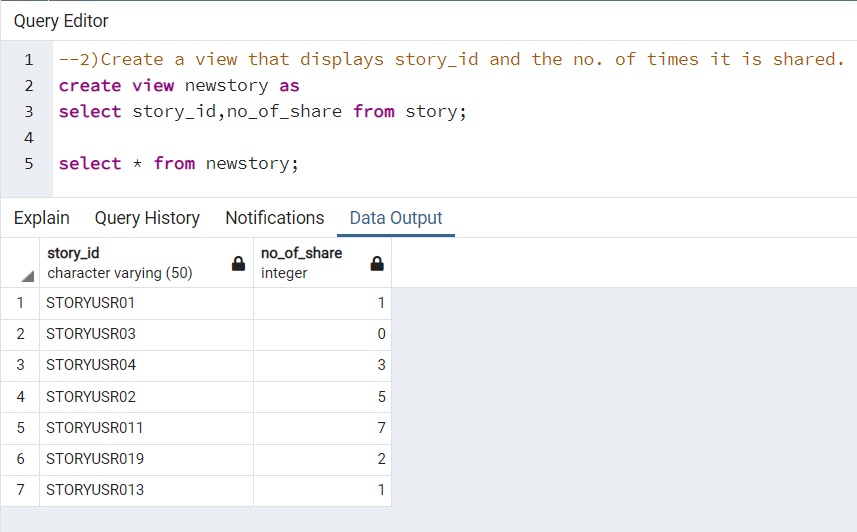
**5.5 PL/SQL**

**1)Create a view that displays story\_id and the no. of times it is shared.**

create view newstory as

select story\_id,no\_of\_share from story;

select \* from newstory;

****

**5.6 Functions**

**1)Create a function gives the total number of private accounts.**

create function get\_privacy\_status()

returns int

language plpgsql

as

$$

Declare

user\_count integer;

Begin

select count(\*)

into user\_count

from users

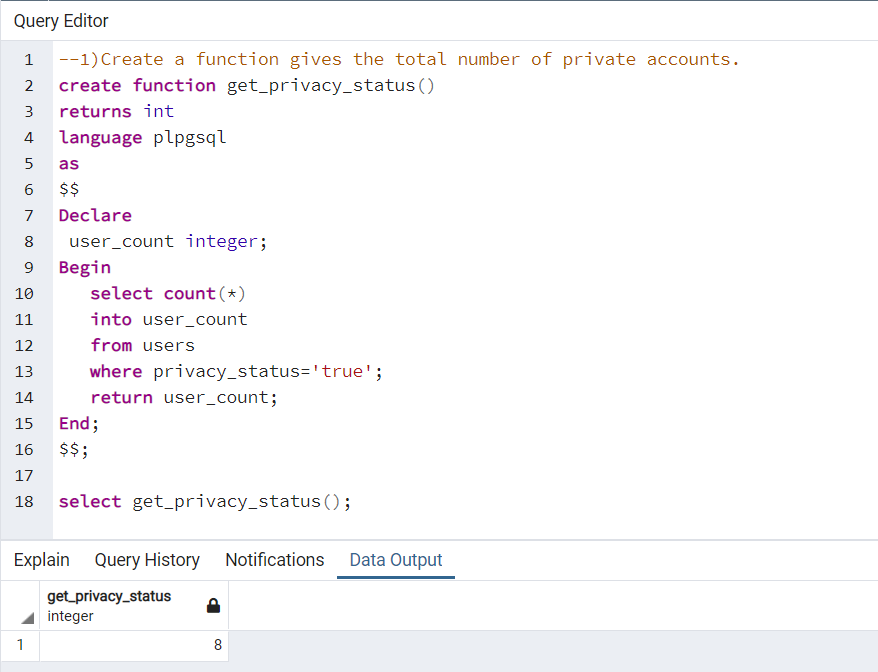
where privacy\_status='true';

return user\_count;

End;

$$;

select get\_privacy\_status();

****

**5.7 Triggers**

**1)Create a trigger which displays an error message if someone tries to enter invalid E-mail.**

create function check\_email() returns trigger as $$

BEGIN

if NEW.email LIKE '%\_\_\_@\_\_\_%.\_\_%' then

return NEW;

end if;

raise exception 'Your E-mail format must be like '\*\*@\*\*.\*\*';

END;

$$

LANGUAGE plpgsql;

create trigger email\_check

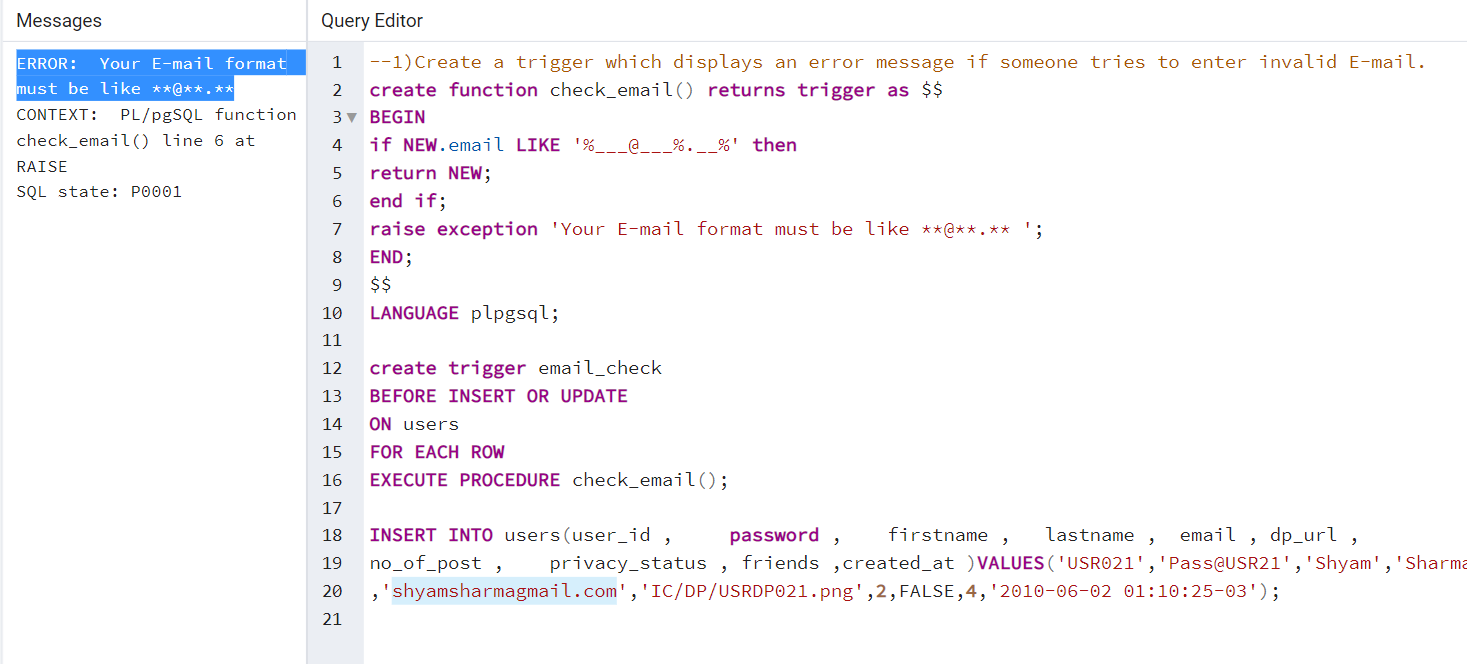
BEFORE INSERT OR UPDATE

ON users

FOR EACH ROW

EXECUTE PROCEDURE check\_email();

INSERT INTO users(user\_id , password , firstname , lastname , email , dp\_url ,no\_of\_post , privacy\_status , friends ,created\_at )VALUES('USR021','Pass@USR21','Shyam','Sharma','shyamsharmagmail.com','IC/DP/USRDP021.png',2,FALSE,4,'2010-06-02 01:10:25-03');



**2)Create a trigger which inserts the new tuples on insertion and old tuples on deletion into a new table called friend\_audit.**

create or replace function do\_friend\_audit()

returns trigger as $$

begin

if(TG\_OP='DELETE') then insert into friend\_audit

select 'D',now(),OLD.user\_id,OLD.f\_name;

return OLD;

elseif(TG\_OP='UPDATE') then insert into friend\_audit

select 'U',now(),NEW.user\_id,NEW.f\_name;

return NEW;

elseif(TG\_OP='INSERT') then insert into friend\_audit

select 'I',now(),NEW.user\_id,NEW.f\_name;

return NEW;

end if;

end;

$$

language plpgsql;

create trigger friend\_audit

after insert or update or delete on friends

for each row

execute procedure do\_friend\_audit();

create table friend\_audit(OP varchar(10) not null,

stamp timestamp not null,

user\_id varchar(10) not null,

friend\_name varchar(10) not null);

INSERT INTO users(user\_id , password , firstname , lastname , email , dp\_url , no\_of\_post , privacy\_status , friends , created\_at )VALUES

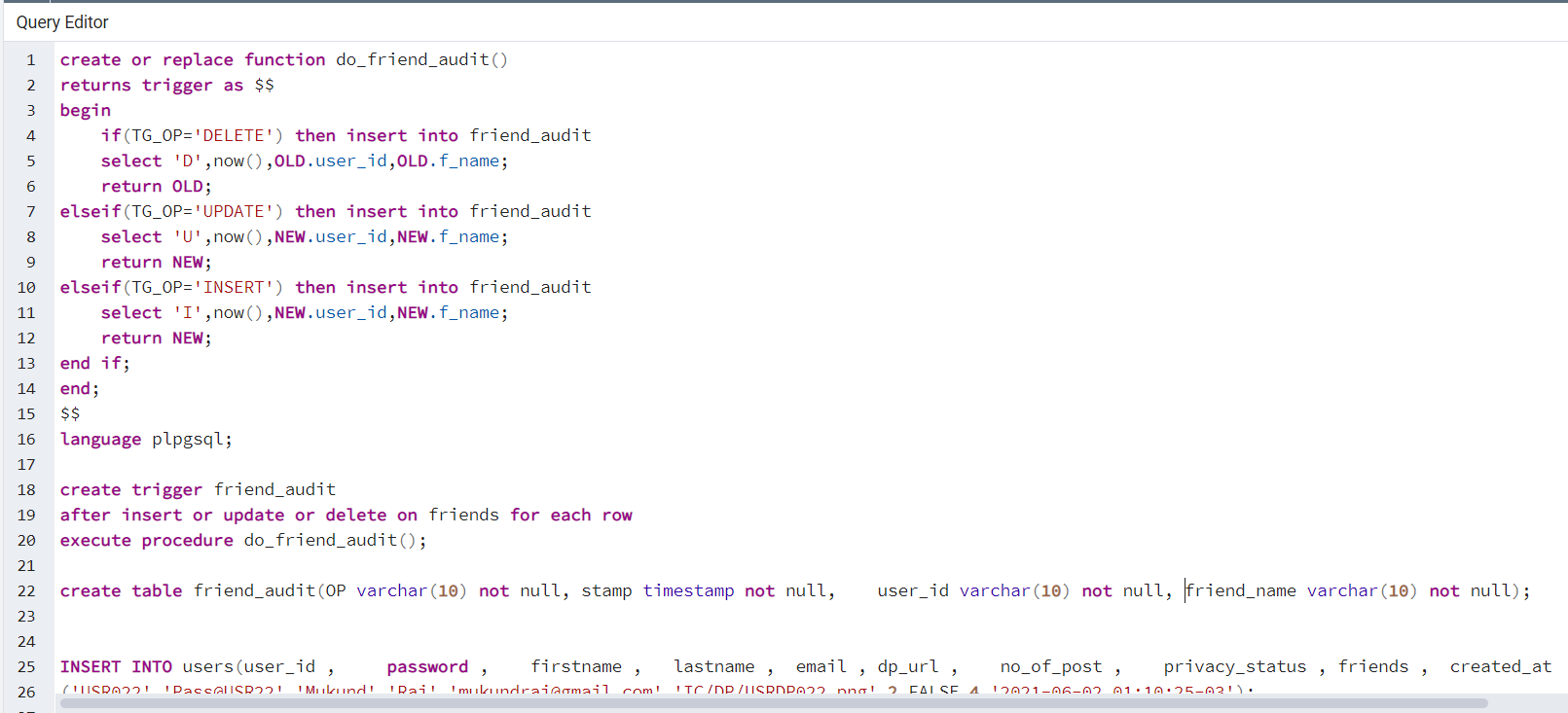
('USR022','Pass@USR22','Mukund','Raj','mukundraj@gmail.com','IC/DP/USRDP022.png',2,FALSE,4,'2021-06-02 01:10:25-03');

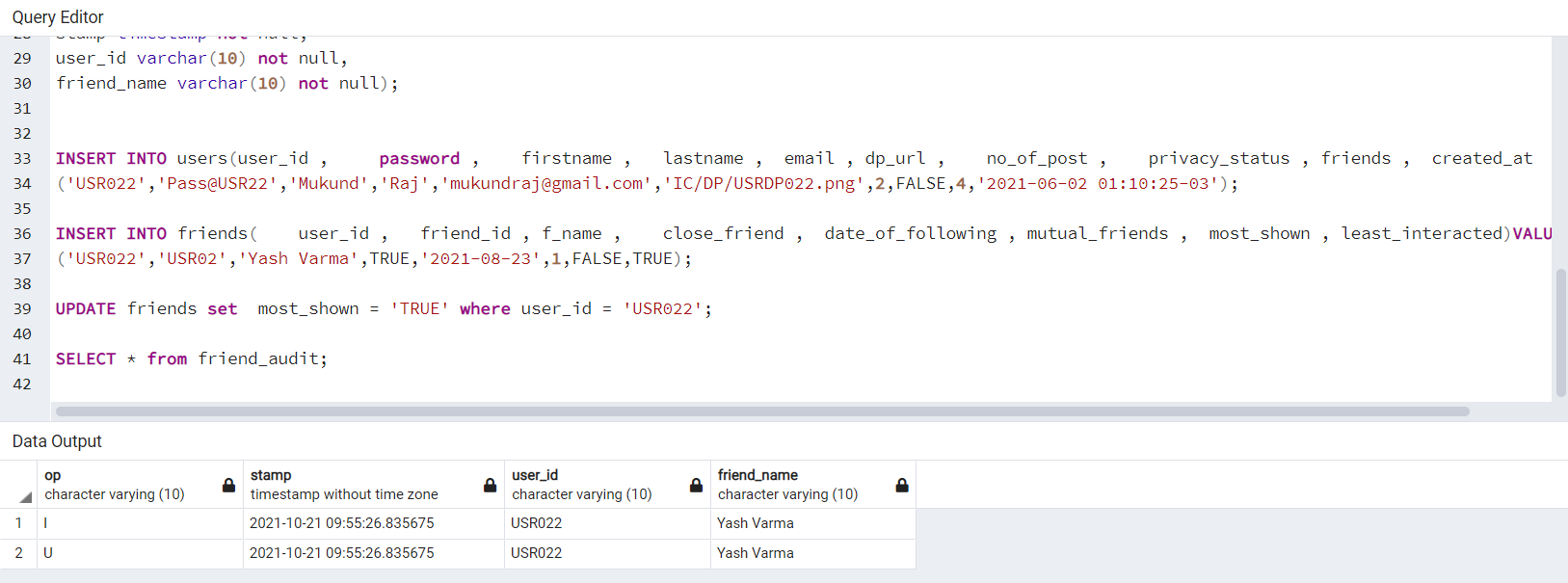
INSERT INTO friends( user\_id , friend\_id , f\_name , close\_friend , date\_of\_following , mutual\_friends , most\_shown , least\_interacted)VALUES

('USR022','USR02','Yash Varma',TRUE,'2021-08-23',1,FALSE,TRUE);

UPDATE friends set most\_shown = 'TRUE' where user\_id = 'USR022';

SELECT \* from friend\_audit;



****

**5.8 Cursor**

**1)Create a cursor which displays the the respective story’s information iff it is included in highlights. Otherwise it should not display anything.**

create or replace function get\_story\_info(str\_id varchar(50)) returns text as $$

declare name text default '';

storyrec record;

cur\_str cursor(str\_id varchar(50))

for select user\_id,story\_id

from story

where story.highlights='TRUE' and str\_id=story\_id;

begin

open cur\_str(str\_id);

fetch cur\_str into storyrec;

exit when not found;

name := storyrec.user\_id || ' , ' || storyrec.story\_id;

close cur\_str;

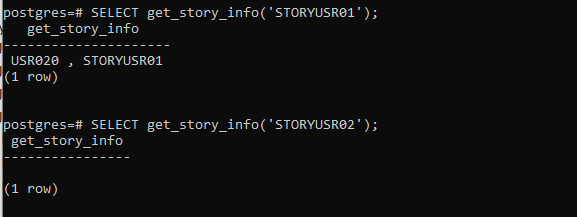
return name;

end; $$

language plpgsql;

SELECT get\_story\_info(‘STORYUSR01’);

SELECT get\_story\_info(‘STORYUSR02’);



# FUTURE ENHANCEMENTS OF THE SYSTEM

* + - * We will design Front-end Design in HTML , CSS , JavaScript and Develop Bank- end in Python.
      * For security purpose New Registration is done using OTP.
      * We will make database more consistent and We are making this database efficient and easy to implement with huge data capacity.
      * Methods and user data input will be lot easy after the implement of GUI.
      * We will also add some extra features so that the users can get answer for their complaints as fast as possible.

# BIBLIOGRAPHY

* + - * For the successful implementation of this project we referred to many websites and books.
      * We created the ER Diagram and Schema Diagram on “erdplus.com”.
      * Mostly we referred the online material for syntax of procedures, triggers, Exception and cursors.

### Reference book:

**Data Base System Concepts**

**-Henry F. Korth & A. Silberschatz 2nd Ed. McGraw-Hill 1991**

**Reference Websites:**

* + - * <https://www.stackoverflow.com/>
      * <https://erdplus.com>