**Mingle:An Office Networking Platform**

Team:

Gauri Sarode: gss383

Joby Joy: jj2196

Mingle is an application that allows all the employee to connect and update each other with their existing projects, form teams to contribute to projects, update their skills and interest in their profiles, add certifications, share pictures and location of the conferences,events and socialize with other employee across the firm.

Following are the functionalities of the system:

**Create/Register a Mingle profile:**

All employee’s of the organization are allowed to create a profile on Mingle. The employee can make the profile public or private upon his discretion. Once the employee deletes his profile, all the data corresponding to the employee profile will be deleted.

**Create or Update Posts on the Wall:**

Employees can create and update posts on their wall. The posts can be anything specific to the employee. Examples of which can be all the courses that he is takes, his views on modern technologies, hotlinks to his github repository, twitter handle etc. He can also create post specific to the project and provide his team explicitly to view the content.

**Network and build relationships:**

Mingle allows an employee to network and collaborate with other employees by sending a connect request. An employee can also use Mingle to form teams and collaborate with the team members in project activities. Connection can be team, friends or distinct friends given the scenario.

**Upload Multimedia Contents:**

Employees can post pictures, videos etc of presentation, conferences, training resources with Mingle providing the provision to either share it across team or across the organization.

**Adding Location details and Comments :**

Employees can post their location details on Mingle that allows others to know where you are. This feature can be used when the employee attends a event and would like to showcase his presence across the organizations. Mingle allows an employee to comment on post and also allows him to make the comment restricted to Team or Public .

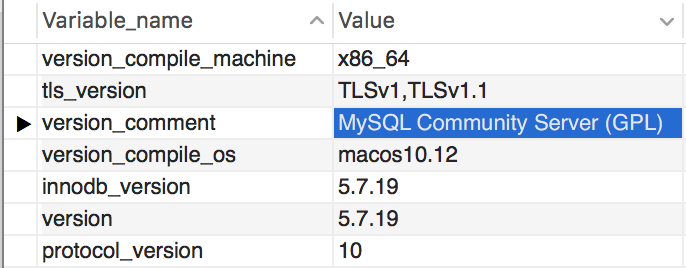
**Likes:**

Employees can like posts, comment or multimedia contents posted by an employee.

**Database Requirements:**

Mingle will be designed as a relational database application which will be using MySQL as its database server.

*MySQL version:*



*Entity Relationship Diagram:*

*Assumptions:*

* Every employee is assigned a level between A to D where A refers to Junior associates, B refers to senior associates, C refers to managers, D refers to CEO, President, Trustee. By default a new employee is assigned the level A following which the HR assigns the corresponding level based on their skills and experience.
* Access id are allowed at the profile and post level.
* Access id’s can be S: Self, T: Team, F: Friends, FOF : Friend and Friends of Friends, P: Public.
* Realation\_type in the relationship table indicates the type of friendship, F: Friends, FOF: Friend of Friends, T: Team.

*Relational Schema:*

* employee(**ssn,**email\_id)
* signin(ssn,password)
* registered\_employee(**profile\_name**,ssn,first\_name,last\_name,address,designation,skills,interests,manager,level,access\_id,email\_id,profile\_pic,gender)
* post(**post\_id**,post\_date,post\_title,post\_desc,access\_id)
* wall(post\_id,profile\_name)
* relationship(sender\_name,receiver\_name,relation\_type,request\_time,request\_status)
* multimedia\_content(**multimedia\_id**,post\_id,multimedia\_name,multimedia\_type, multimedia\_content,access\_id)
* comment(**comment\_id**,post\_id,desc,comment\_date,commentor\_name,access\_id)
* location(**loc\_id,**post\_id,name,description,latitude,longitude,access\_id)
* likes(**like\_id**,post\_id,comment\_id,multimedia\_id,loc\_id,viewer\_name,like\_date)

*Foreign Keys:*

* ssn in signin table is a foreign key referencing the ssn in the employee table.
* ssn in the registered\_employee table is a foreign key referencing the ssn in employee table.
* post\_id,profile\_name in wall table references the post\_id in post and profile\_name in the registered\_employee table respectively.
* In multimedia\_connent table post\_id is the foreign key referencing table post.
* In comment table post\_id is the foreign key referencing table post and commentor\_name is the foreign key referencing table registered\_employee.

* In like table likes viewer\_name is the foreign key referencing table registered\_employee and post\_id is the foreign key referencing table post and multimedia\_id is the foreign key referencing table multimedia\_content and loc\_id is the foreign key referencing table location.
* In relationship table sender\_name references table registered\_employee.

**DDL Statements:**

create database Mingle;

use Mingle;

drop table if exists employee;

#Create employee table

create table employee(

ssn int primary key, email\_id varchar(20) not null

);

#Create sign in table

drop table if exists signin;

create table signin

(

ssn int unique,

password varchar(50) not null,

foreign key (ssn) References employee(ssn)

on delete cascade

);

#Create registered\_employee table

drop table if exists registered\_employee;

create table registered\_employee

(

profile\_name varchar(30) primary key,

ssn int unique,

first\_name varchar(20) not null ,

last\_name varchar(20) not null,

address varchar (255),

designation varchar(20),

skills varchar(255),

interests varchar(255),

manager varchar(20),

level enum('A','B','C','D') default 'A',

email\_id varchar(50),

profile\_pic blob,

gender enum('Male','Female'),

access\_id enum('P','T','F','S','FOF') default 'S',

foreign key (ssn) References employee(ssn)

on delete cascade

);

#Create post

drop table if exists post;

create table post

(

post\_id varchar(30),

post\_time timestamp not null,

post\_title varchar(150),

post\_desc varchar(255),

access\_id varchar(5) default 'S',

primary key (post\_id)

);

#Create wall(post\_id,profile\_name)

drop table if exists wall;

create table wall

(

post\_id varchar(20),

profile\_name varchar(30),

foreign key (profile\_name) references registered\_employee(profile\_name)

on delete cascade

on update cascade,

foreign key (post\_id) references post(post\_id)

on delete cascade

on update cascade

);

#Create relationship table

drop table if exists relationship;

create table relationship

(

sender\_name varchar(30),

receiver\_name varchar(30),

relation\_type varchar(20),

request\_time timestamp,

friendship\_status varchar(20) NOT NULL,

foreign key (sender\_name) references registered\_employee(profile\_name)

on delete cascade

on update cascade,

foreign key (receiver\_name) references registered\_employee(profile\_name)

on delete cascade

on update cascade,

primary key(sender\_name,receiver\_name)

);

#Create multimedia table

drop table if exists multimedia\_content;

create table multimedia\_content

(

multimedia\_id varchar(10) primary key,

post\_id varchar(20),

multimedia\_name varchar(20),

Multimedia\_type varchar(20),

multimedia\_content blob,

access\_id varchar(5) default 'S',

foreign key (post\_id) references post(post\_id)

on delete cascade

on update cascade

);

#Create table Comment

drop table if exists comment;

create table comment

(

comment\_id varchar(20) primary key,

post\_id varchar(20),

comment\_desc varchar(255),

comment\_date datetime,

commentor\_name varchar(30),

access\_id varchar(5) default 'S',

foreign key (commentor\_name) references registered\_employee(profile\_name)

on delete cascade

on update cascade,

foreign key (post\_id) references post(post\_id)

on delete cascade

on update cascade

);

#Create location table

drop table if exists location;

create table location

(

loc\_id varchar(20) primary key,

post\_id varchar(20),

name varchar(100),

description varchar(255),

latitude varchar(30),

longitude varchar(30),

access\_id varchar(5) default 'S',

foreign key (post\_id) references post(post\_id)

on delete cascade

on update cascade

);

#Create like table

drop table if exists likes;

create table likes

(

like\_id varchar(20) primary key,

post\_id varchar(30),

multimedia\_id varchar(10),

comment\_id varchar(20),

loc\_id varchar(20),

viewer\_name varchar(20),

like\_time datetime,

foreign key (post\_id) references post(post\_id) on delete cascade on update cascade,

foreign key (multimedia\_id) references multimedia\_content(multimedia\_id) on delete cascade on update cascade,

foreign key (comment\_id) references comment(comment\_id) on delete cascade on update cascade,

foreign key (loc\_id) references location(loc\_id) on delete cascade on update cascade,

foreign key (viewer\_name) references registered\_employee(profile\_name) on delete cascade on update cascade

);

**Procedures :**

DELIMITER $$

#Procedure 1- Insert into employee table

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_employee`(

IN e\_ssn int,

IN e\_email varchar(20)

)

BEGIN

INSERT INTO employee

(

ssn,

email\_id

) Values

(

e\_ssn,

e\_email

);

END $$

#Procedure 2: insert into signin table

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_signin\_details`(

IN e\_ssn int,

IN e\_password varchar(50)

)

BEGIN

INSERT INTO signin

(

ssn,

password

) Values

(

e\_ssn,

e\_password

);

END $$

#Procedure 3: for inserting registered user

DROP PROCEDURE IF EXISTS `insert\_registered\_employee` $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_registered\_employee`(

IN profile\_name varchar(30),IN ssn int,

IN first\_name varchar(20),IN last\_name varchar(20),IN address varchar(255),IN designation varchar(20),

IN skills varchar(255),IN interests varchar(255),IN manager varchar(20),

IN level enum('A','B','C','D'),IN email\_id varchar(50),

IN profile\_pic blob,IN gender enum('Male','Female'),

IN access\_id enum('P','T','F','S','FOF')

)

BEGIN

INSERT INTO registered\_employee

(

profile\_name,ssn,first\_name, last\_name,address,designation,skills,

interests,manager,level, email\_id, profile\_pic,gender,access\_id

) Values

(

profile\_name,ssn,first\_name,last\_name,address,designation,skills,interests,manager,level,

email\_id,profile\_pic,gender,access\_id

);

END $$

#Procedure 4: for inserting post

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_post`(

IN post\_id varchar(30), IN post\_time timestamp,

IN post\_title varchar(150),IN post\_desc varchar(255),

IN access\_id varchar(3)

)

BEGIN insert into post

(

post\_id,post\_time,post\_title,post\_desc,access\_id

) values

(post\_id,post\_time,post\_title,post\_desc,access\_id

); END $$

#Procedure 5: saving records to wall

CREATE DEFINER=`root`@`localhost` PROCEDURE `add\_to\_wall`(

IN post\_id varchar(20),

IN profile\_name varchar(30)

)

BEGIN insert into wall

(

post\_id ,

profile\_name

)

values(post\_id, profile\_name

);

END $$

#Procedure 6 request to connect with colleagues

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_relationship`(

p\_sender\_name varchar(30),

p\_receiver\_name varchar(20),

p\_relation varchar(20),

p\_time\_request\_sent timestamp,

p\_request\_status enum('Accepted','Declined')

)

BEGIN

insert into relationship (

sender\_name , receiver\_name , relation\_type,

request\_time, request\_status

)

Values

(p\_sender\_name , p\_receiver\_name , p\_relation , p\_time\_request\_sent, p\_request\_status

);

END $$

#Procedure 7 insert multimedia contents

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_multimedia`(

IN p\_multimedia\_id varchar(10),

IN p\_post\_id varchar(20),

IN p\_multimedia\_name varchar(20),

IN p\_multimedia\_type varchar(20),

IN p\_multimedia\_content blob,

IN p\_access\_id varchar(5)

)

BEGIN insert into multimedia\_content

(

multimedia\_id, post\_id, multimedia\_name,

multimedia\_type,

multimedia\_content, access\_id

) values

(

p\_multimedia\_id, p\_post\_id, p\_multimedia\_name, p\_multimedia\_type,

p\_multimedia\_content, p\_access\_id

); END $$

#Procedure 8 insert comments

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_comments`(

IN comment\_id varchar(20),IN post\_id varchar(20),

IN comment\_desc varchar(255), IN comment\_date datetime,

IN commentor\_name varchar(30),

IN access\_id varchar(5)

)

BEGIN insert into comment

(

comment\_id,

post\_id, comment\_desc, comment\_date, commentor\_name,access\_id

)

values(

comment\_id, post\_id,

comment\_desc, comment\_date, commentor\_name, access\_id

); END $$

#Procedure 9: inserts location

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_locations`(

IN loc\_id varchar(20),IN post\_id varchar(20),

IN lname varchar(100),In description varchar(255),

IN latitude varchar(30), IN longitude varchar(30),

IN access\_id varchar(5)

)

BEGIN insert into location

(

loc\_id, post\_id,name,description, latitude,longitude,access\_id

)

values (

loc\_id,post\_id,lname,

description,latitude,longitude,access\_id

); END $$

#Procedure 10:inserts likes

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_likes`(

IN p\_like\_id varchar(20),IN post\_id varchar(30),IN p\_multimedia\_id varchar(10),

IN p\_comment\_id varchar(20), IN p\_loc\_id varchar(20),

IN p\_viewer varchar(20), IN p\_like\_time datetime

)

BEGIN insert into likes (

like\_id, post\_id,

multimedia\_id,comment\_id,loc\_id,

viewer\_name,like\_time

) values

(

p\_like\_id,post\_id,p\_multimedia\_id,

p\_comment\_id,p\_loc\_id,

p\_viewer,p\_like\_time

); END $$

#Procedure 11: Read from registered employee table

CREATE DEFINER=`root`@`localhost` PROCEDURE `registered\_employee\_acc`()

BEGIN select \* from registered\_employee;

END $$

#Procedure 12 :Read posts

CREATE DEFINER=`root`@`localhost` PROCEDURE `read\_posts`()

BEGIN select \* from post;

END $$

#Procedure 13 : Read wall

CREATE DEFINER=`root`@`localhost` PROCEDURE `read\_wall `()

BEGIN select \* from wall;

END $$

#Procedure 14: Read relationships

CREATE DEFINER=`root`@`localhost` PROCEDURE `view\_relationship`()

BEGIN select \* from relationship;

END $$

#Procedure 15: View Multimedia Contents

CREATE DEFINER=`root`@`localhost` PROCEDURE `multimedia\_contents`()

BEGIN select \* from multimedia\_content;

END $$

#Procedure 16

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

BEGIN select \* from comment;

END $$

#Procedure 17: View Location

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

BEGIN select \* from location;

END $$

# Procedure 18: View likes

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

BEGIN select \* from likes;

END $$

#Procedure 19: View employee table

CREATE DEFINER=`root`@`localhost` PROCEDURE `employee\_acc`()

BEGIN select \* from employee;

END $$

#Procedure 20: View sign\_in table

CREATE DEFINER=`root`@`localhost` PROCEDURE `employee\_signin`()

BEGIN select \* from signin;

END $$

DELIMITER ;

**Update Procedure:**

DELIMITER $$

DROP PROCEDURE IF EXISTS `update\_post\_table` $$

# Update Post Table

CREATE DEFINER=`root`@`localhost` PROCEDURE `update\_post\_table`

(

IN post\_id varchar(30), IN post\_time timestamp,

IN access\_id varchar(3)

)

BEGIN

update post

set post\_time= post\_time,access\_id = access\_id

where post\_id = post\_id;

END $$

DELIMITER ;

**Functions:**

DELIMITER $$

#Function: To check for a new or an existing employee

DROP FUNCTION IF EXISTS `validate\_user` $$

CREATE DEFINER=`root`@`localhost` FUNCTION `validate\_user`(

e\_ssn int

)

RETURNS varchar(50)

BEGIN

DECLARE check\_employee varchar(50);

select ssn into check\_employee from signin where ssn=e\_ssn;

if check\_employee!='NULL' Then

return ("Employee subscribed to Mingle");

else

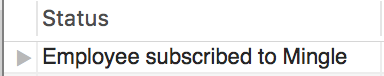
return ("New user registration. You have to sign up first!!");

end if;

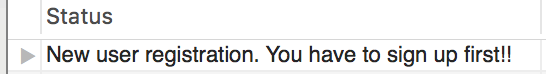
END $$

DELIMITER ;

select validate\_user(742347663) as Status;



select validate\_user(542347663) as Status;



**Test Data.**

Creating test data as part of the test framework for Mingle.

*Insert into employee*

call insert\_employee(742347663,'jj2196@nyu.edu');

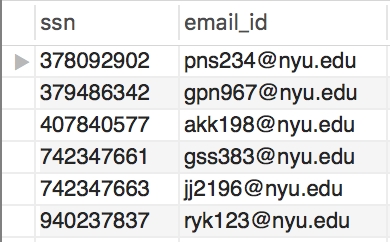
call insert\_employee(742347661,'gss383@nyu.edu');

call insert\_employee(378092902,'pns234@nyu.edu');

call insert\_employee(407840577,'akk198@nyu.edu');

call insert\_employee(940237837,'ryk123@nyu.edu');

call insert\_employee(379486342,'gpn967@nyu.edu');



*Insert into signin*

call insert\_signin\_details(742347663,'password');

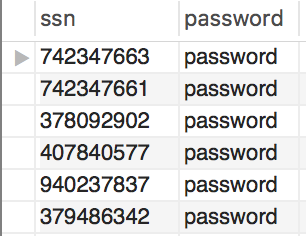
call insert\_signin\_details(742347661,'password');

call insert\_signin\_details(378092902,'password');

call insert\_signin\_details(407840577,'password');

call insert\_signin\_details(940237837,'password');

call insert\_signin\_details(379486342,'password');



*Insert into registered\_employee*

call insert\_registered\_employee('jj2196',742347663,'Joby','Joy','New York','Senior Engineer','Java Php','Dancing','Karl','A','jj2196@nyu.edu',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','Male','S');

call insert\_registered\_employee('gss383',742347661,'Gauri','Sarode','New York','Senior Engineer','Python Php','Movies','Karl','A','gss383@nyu.edu',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','Female','P');

call insert\_registered\_employee('pns234',378092902,'Pranav','Reddy','Bayridge','Data Engineer','Java React','Mingling','Gauri','B','pns23@nyu.edu',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','Male','S');

call insert\_registered\_employee('akk198',407840577,'Arun','Kondani','Manhattan','Website developer','Python React','Cribbing','Gauri','B','akk198@nyu.edu',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','Male','P');

call insert\_registered\_employee('ryk123',940237837,'Rahul','Keswani','Long Island','Data Scientist','React Php','Coding','Pranav','C','ryk123@nyu.edu',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','Male','S');

call insert\_registered\_employee('gpn967',379486342,'Geetesh','Nikhade','Boston','Full stack developer','Python Java','Drinking','Pranav','C','gpn967@nyu.edu',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','Male','P');



*Insert into post*

call insert\_post('post\_1','2010-07-22 22:30:12','First post','Day one it is!',' ');

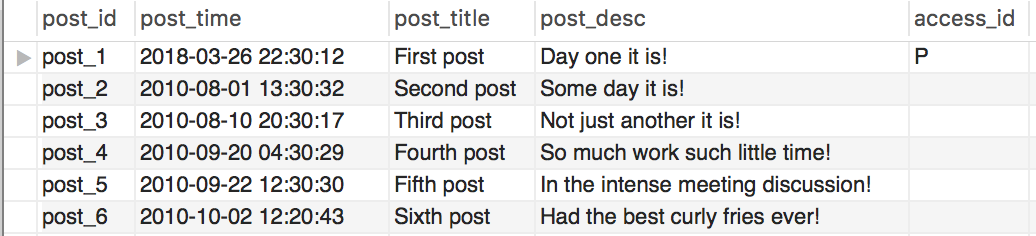
call insert\_post('post\_2','2010-08-01 13:30:32','Second post','Some day it is!',' ');

call insert\_post('post\_3','2010-08-10 20:30:17','Third post','Not just another it is!',' ');

call insert\_post('post\_4','2010-09-20 04:30:29','Fourth post','So much work such little time!',' ');

call insert\_post('post\_5','2010-09-22 12:30:30','Fifth post','In the intense meeting discussion!',' ');

call insert\_post('post\_6','2010-10-02 12:20:43','Sixth post','Had the best curly fries ever!',' ');



*Insert into wall*

call add\_to\_wall('post\_1','jj2196');

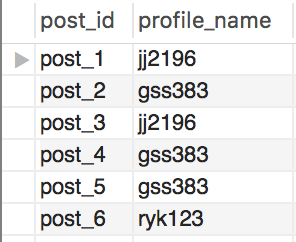
call add\_to\_wall('post\_2','gss383');

call add\_to\_wall('post\_3','jj2196');

call add\_to\_wall('post\_4','gss383');

call add\_to\_wall('post\_5','gss383');

call add\_to\_wall('post\_6','ryk123');



*Insert into relationship*

call insert\_relationship('jj2196','gss383','T','2010-07-22 22:30:12','Accepted');

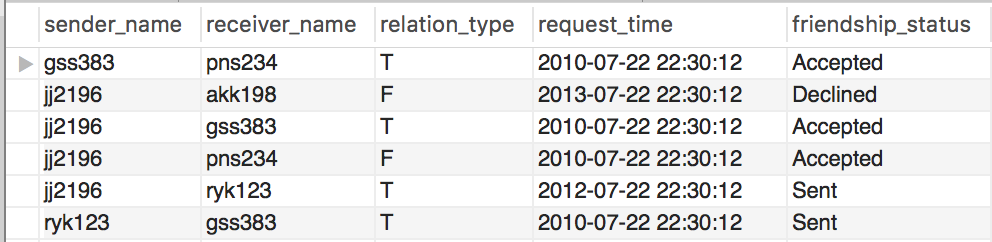
call insert\_relationship('jj2196','ryk123','T','2012-07-22 22:30:12','Sent');

call insert\_relationship('jj2196','akk198','F','2013-07-22 22:30:12','Declined');

call insert\_relationship('jj2196','pns234','F','2010-07-22 22:30:12','Accepted');

call insert\_relationship('gss383','pns234','T','2010-07-22 22:30:12','Accepted');

call insert\_relationship('ryk123','gss383','T','2010-07-22 22:30:12','Sent');



*Insert into multimedia\_content*

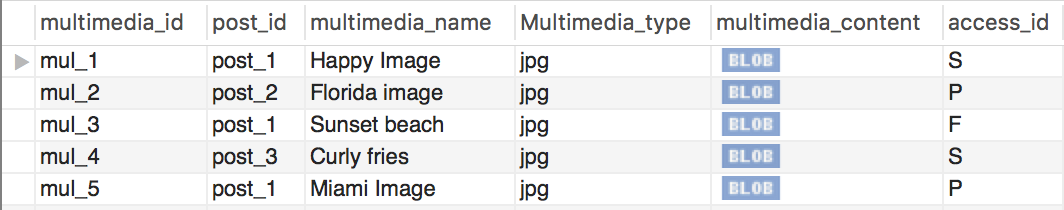
call insert\_multimedia('mul\_1','post\_1','Miami Image','jpg',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','S');

call insert\_multimedia('mul\_2','post\_2','Florida image','jpg',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','P');

call insert\_multimedia('mul\_3','post\_1','Sunset beach','jpg',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','F');

call insert\_multimedia('mul\_4','post\_3','Curly fries','jpg',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','S');

call insert\_multimedia('mul\_5','post\_1','Miami Image','jpg',x'C9CBBBCCCEB9C8CABCCCCEB9C9CBBB','P');



*Insert into comment*

call insert\_comments('cmnt\_1','post\_3','So cool!','2010-07-22 22:30:12','gss383','P');

call insert\_comments('cmnt\_2','post\_4','Let me join in','2010-09-21 04:30:29','ryk123','F');

call insert\_comments('cmnt\_3','post\_5','Where is this!','2010-09-22 15:30:30','ryk123','F');

call insert\_comments('cmnt\_4','post\_5','How does this work?','2010-09-23 12:30:302','gpn967','P');

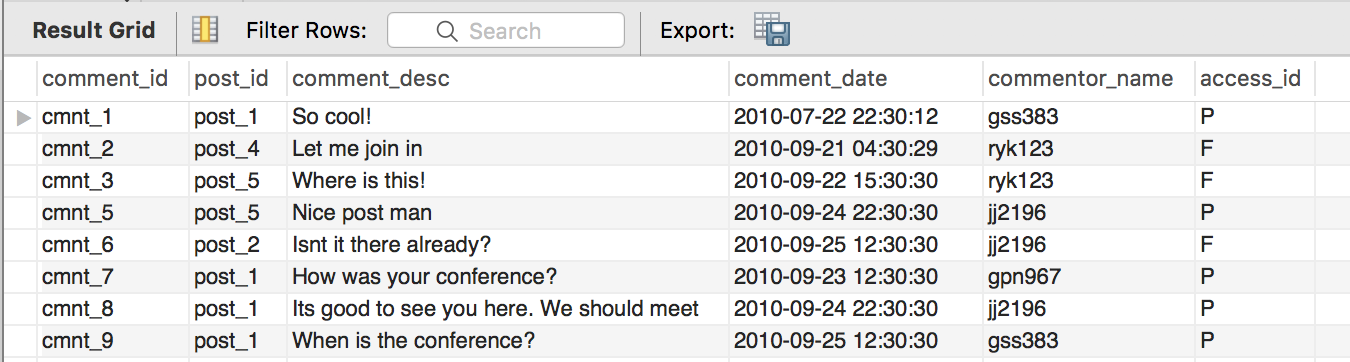
call insert\_comments('cmnt\_5','post\_5','Nice post man','2010-09-24 22:30:30','jj2196','P');

call insert\_comments('cmnt\_6','post\_2','Isnt it there already?','2010-09-25 12:30:30','jj2196','F');

call insert\_comments('cmnt\_7','post\_1','How was your conference?','2010-09-23 12:30:30','gpn967','P');

call insert\_comments('cmnt\_8','post\_1','Its good to see you here. We should meet','2010-09-24 22:30:30','jj2196','P');

call insert\_comments('cmnt\_9','post\_1','When is the conference?','2010-09-25 12:30:30','gss383','P');



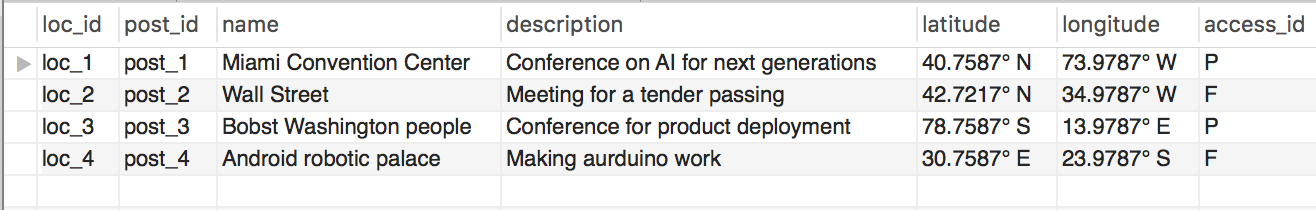
*Insert into location*

call insert\_locations('loc\_1','post\_1','Miami Convention Center','Conference on AI for next generations','40.7587° N', '73.9787° W','P');

call insert\_locations('loc\_2','post\_2','Wall Street','Meeting for a tender passing','42.7217° N', '34.9787° W','F');

call insert\_locations('loc\_3','post\_3','Bobst Washington people','Conference for product deployment','78.7587° S', '13.9787° E','P');

call insert\_locations('loc\_4','post\_4','Android robotic palace','Making aurduino work','30.7587° E', '23.9787° S','F');



*Insert into likes*

call insert\_likes('like\_1','post\_1','mul\_1','cmnt\_1',NULL,'gss383','2010-07-22 22:30:12');

call insert\_likes('like\_2','post\_2','mul\_1','cmnt\_1',NULL,'akk198','2010-08-12 16:45:45');

call insert\_likes('like\_3','post\_1','mul\_1','cmnt\_1',NULL,'akk198','2010-09-14 18:20:23');

call insert\_likes('like\_4','post\_2','mul\_1','cmnt\_1',NULL,'pns234','2010-09-22 19:35:19');

call insert\_likes('like\_5','post\_2','mul\_2','cmnt\_1',NULL,'pns234','2010-09-22 19:35:19');

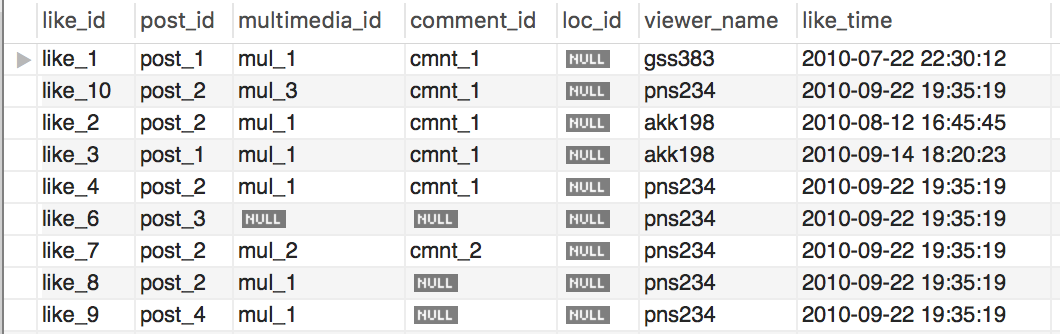
call insert\_likes('like\_6','post\_3',NULL,NULL,NULL,'pns234','2010-09-22 19:35:19');

call insert\_likes('like\_7','post\_2','mul\_2','cmnt\_2',NULL,'pns234','2010-09-22 19:35:19');

call insert\_likes('like\_8','post\_2','mul\_1',NULL,NULL,'pns234','2010-09-22 19:35:19');

call insert\_likes('like\_9','post\_4','mul\_1',NULL,NULL,'pns234','2010-09-22 19:35:19');

call insert\_likes('like\_10','post\_2','mul\_3','cmnt\_1',NULL,'pns234','2010-09-22 19:35:19');



**Views:**

**#Display Posts by all the users**

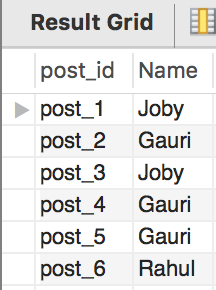
CREATE OR REPLACE VIEW All\_Posts\_view AS

SELECT post\_id, first\_name as Name

FROM wall as w natural join registered\_employee as re

where w.profile\_name=re.profile\_name;

select \* from All\_Posts\_view;

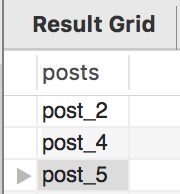


*Posts by a user:*

select post\_id AS posts

from All\_Posts\_view

where Name='Gauri';



**#Display Multimedia Contents of a specific location**

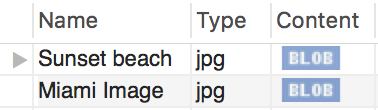
CREATE OR REPLACE VIEW Multimedia\_list\_view AS

select m.multimedia\_name as Name,m.Multimedia\_type as Type,m.multimedia\_content as Content

from multimedia\_content as m, location as l

where l.post\_id=m.post\_id and l.name like '%Miami%' and (m.access\_id='P' or m.access\_id='F');

select \* from Multimedia\_list\_view;



**#Display friends list of any User.**

Create OR Replace VIEW friendlist\_view AS

select first\_name as Friend,friendship\_status as Status, request\_time as Time

from relationship as r,registered\_employee as e

where r.receiver\_name=e.profile\_name and r.sender\_name = 'jj2196';

**Queries:**

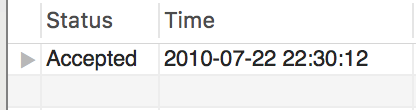
These queries were created to check for different features of the Mingle application:

*Check for connections between two users*

select friendship\_status as Status, request\_time as Time

from relationship as r

where r.sender\_name = 'jj2196' AND r.receiver\_name='gss383';

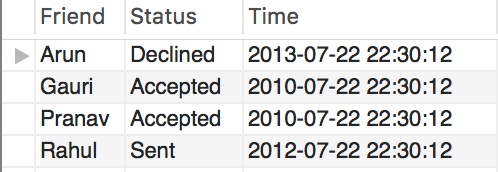


*Display friends list of any User.*

select first\_name as Friend,friendship\_status as Status, request\_time as Time

from relationship as r,registered\_employee as e

where r.receiver\_name=e.profile\_name and r.sender\_name = 'jj2196';



*Display friends list by connection request status.*

select \*

from friendlist\_view

where Status='Sent';



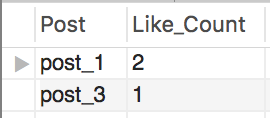
*Count Total likes by post for specific user*

select l.post\_id as Post,count(l.post\_id) as Like\_Count

from likes as l natural join wall as w

where w.profile\_name='jj2196'

group by l.post\_id;



*Filter post by date*

select post\_title,post\_desc

from post

where post\_time between date\_sub(now(), interval 10 year )

and date(now()) ;

