Database configuration

Database is hosted at: **proj-514-02.cs.iastate.edu**
UserName: **coms514user**
Password: **password**
Port: **3306**
Database/Schema name: **socialDb**

TABLES:

1. **Discussion**
   Columns:

discussionID	int
userID	int
adminID	int
adminMessage	varchar
message	varchar
time	varchar
admintime	varchar

2. **broadcasts**

broadcastid	integer
postdesc	varchar
posttime	timestamp
userid	int
title	varchar

3. **comments**

comment_id	int
user_id	int
date_time	varchar
post_id	int
comment_string	varchar
module_type	int

4. **dashboard**

entry_id	int
entry_desc	varchar
entry_type	int
post_id	int

5. **events**

event_id	int
event_desc	varchar
created_date_time	varchar
user_id	int
resources_needed	varchar
place	varchar
event_date_time	timestamp
is_archived	varchar
is_resources_satisfied	varchar
notify_sms_sent	varchar
time_to_display	timestamp

6. **moduleenum**

module_enum_value	int
Module_enum_name	varchar

7. **users**

userid	int
usertype	varchar
dob	date
gender	varchar
phone	varchar
email	varchar
address	varchar
emergencycontact_name	varchar
emergencycontact_phone	int
name	varchar
password	varchar

Note: Primary keys in related tables are highlighted by color.

PROCEDURES

All these procedures are related to dashboard and is executed when you get data from the database.

A. getHomeInfo()

```
DELIMITER ;;
CREATE DEFINER=`coms514user`@`%` PROCEDURE `getHomeInfo`()
BEGIN

SET SQL_SAFE_UPDATES = 0;
drop table if exists homeinfo;
drop table if exists homeinfotemp;

create table homeinfotemp(
```

```
entry_id int,
entry_desc varchar(1000),
entry_type int,
post_id int,
activity_desc varchar(1000),
create_date_time varchar(100),
user_id Int
);
create table homeinfo(
entry_id int,
entry_desc varchar(1000),
entry_type int,
post_id int,
activity_desc varchar(1000),
create_date_time varchar(100),
user_id int
);

insert into homeinfotemp select a.entry_id, a.entry_desc, a.entry_type,
a.post_id, b.event_desc, b.created_date_time,b.user_id from dashboard a,
events b where a.post_id = b.event_id and a.entry_type = 1;
insert into homeinfotemp select a.entry_id, a.entry_desc, a.entry_type,
a.post_id, b.postdesc, b.posttime,b.userid from dashboard a, broadcasts b
where a.post_id = b.broadcastid and a.entry_type = 2;

insert into homeinfo select * from homeinfotemp order by create_date_time
desc;
select * from homeinfo;

drop table homeinfo;
drop table homeinfotemp;
SET SQL_SAFE_UPDATES = 1;
END ;;
DELIMITER ;
```

B. getInfoByDateBroadcast()

```
DELIMITER ;;
CREATE DEFINER=`coms514user`@`%` PROCEDURE `getInfoByDateBroadcast`()
BEGIN

SET SQL_SAFE_UPDATES = 0;
drop table if exists homeinfo;
drop table if exists homeinfotemp;

create table homeinfotemp(
entry_id int,
entry_desc varchar(1000),
entry_type int,
post_id int,
activity_desc varchar(1000),
create_date_time varchar(100),
user_id Int
);
```

```
create table homeinfo(
entry_id int,
entry_desc varchar(1000),
entry_type int,
post_id int,
activity_desc varchar(1000),
create_date_time varchar(100),
user_id int
);

insert into homeinfotemp select a.entry_id, a.entry_desc, a.entry_type,
a.post_id, b.postdesc, b.posttime,b.userid from dashboard a, broadcasts b
where a.post_id = b.broadcastid and a.entry_type = 2;

insert into homeinfo select * from homeinfotemp order by create_date_time
desc;
select * from homeinfo order by create_date_time desc;

drop table homeinfo;
drop table homeinfotemp;
SET SQL_SAFE_UPDATES = 1;
END ;;
DELIMITER ;
```

C. getInfoByDateEvent()

```
DELIMITER ;;
CREATE DEFINER=`coms514user`@`%` PROCEDURE `getInfoByDateEvent`()
BEGIN

SET SQL_SAFE_UPDATES = 0;
drop table if exists homeinfo;
drop table if exists homeinfotemp;

create table homeinfotemp(
entry_id int,
entry_desc varchar(1000),
entry_type int,
post_id int,
activity_desc varchar(1000),
create_date_time varchar(100),
user_id Int
);
create table homeinfo(
entry_id int,
entry_desc varchar(1000),
entry_type int,
post_id int,
activity_desc varchar(1000),
create_date_time varchar(100),
user_id int
);
```

```
insert into homeinfotemp select a.entry_id, a.entry_desc, a.entry_type,
a.post_id, b.event_desc, b.created_date_time,b.user_id from dashboard a,
events b where a.post_id = b.event_id and a.entry_type = 1;
insert into homeinfo select * from homeinfotemp order by create_date_time
desc;
select * from homeinfo order by create_date_time desc;

drop table homeinfo;
drop table homeinfotemp;
SET SQL_SAFE_UPDATES = 1;
END ;;
DELIMITER ;
```

D. Trigger for updating dashboard by using information from events

```
delimiter $
create trigger update_dashboard_by_event
after insert on events for each row
begin
insert into dashboard(entry_desc,entry_type,post_id)
values('event',1,new.event_id);
end$
delimiter ;
```

E. Trigger for deleting dashboard by using information from events

```
delimiter $
create trigger delete_dashboard_by_event
before delete on events for each row
begin
declare entryid Integer;
select a.entry_id into @entryid from dashboard a where a.post_id =
old.event_id and a.entry_type = 1 limit 1;
delete from dashboard where entry_id = @entryid;
end$
delimiter ;
```

F. Trigger for updating dashboard by using information from broadcast

```
delimiter $
create trigger update_dashboard_by_broadcast
after insert on broadcasts for each row
begin
insert into dashboard(entry_desc,entry_type,post_id)
values('broadcast',2,new.broadcastid);
end$
delimiter ;
```

G. Trigger for deleting dashboard by using information from broadcast

```
delimiter $
create trigger delete_dashboard_by_broadcast
```

```
before delete on broadcasts for each row
begin
declare entryid Integer;
select a.entry_id into @entryid from dashboard a, broadcasts b where a.post_id
= old.broadcastid and a.entry_type = 2 limit 1;
delete from dashboard where entry_id = @entryid;
end$
delimiter ;
```

Important points:

Comments is a table in which we put comments from all modules and the modules are distinguished by the ***module_type*** column. Different module types are:

```
EVENTS(1),
BROADCAST(2),
DISCUSSION(3);
```

When we insert a comment from Event module, we put **1** in this column and also insert the event_id in the ***post_id*** column so as to find the event which this comment relates to.