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 |

1. **broadcasts**

|  |  |
| --- | --- |
| broadcastid | integer |
| postdesc | varchar |
| posttime | timestamp |
| userid | int |
| title | varchar |

1. **comments**

|  |  |
| --- | --- |
| comment\_id | int |
| user\_id | int |
| date\_time | varchar |
| post\_id | int |
| comment\_string | varchar |
| module\_type | int |

1. **dashboard**

|  |  |
| --- | --- |
| entry\_id | int |
| entry\_desc | varchar |
| entry\_type | int |
| post\_id | int |
| activity\_desc | varchar |
| create\_date\_time | varchar |
| user\_id | int |

1. **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 |

1. **moduleenum**

|  |  |
| --- | --- |
| module\_enum\_value | int |
| Module\_enum\_name | varchar |

1. **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 |

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.

## **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 ;

## **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 ;

## **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 ;

## **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 ;

## **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 ;

## **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 ;

## **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 column. Different module types are:

***EVENTS***(1),

***BROADCAST***(2),

***DISCUSSION***(3);

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