

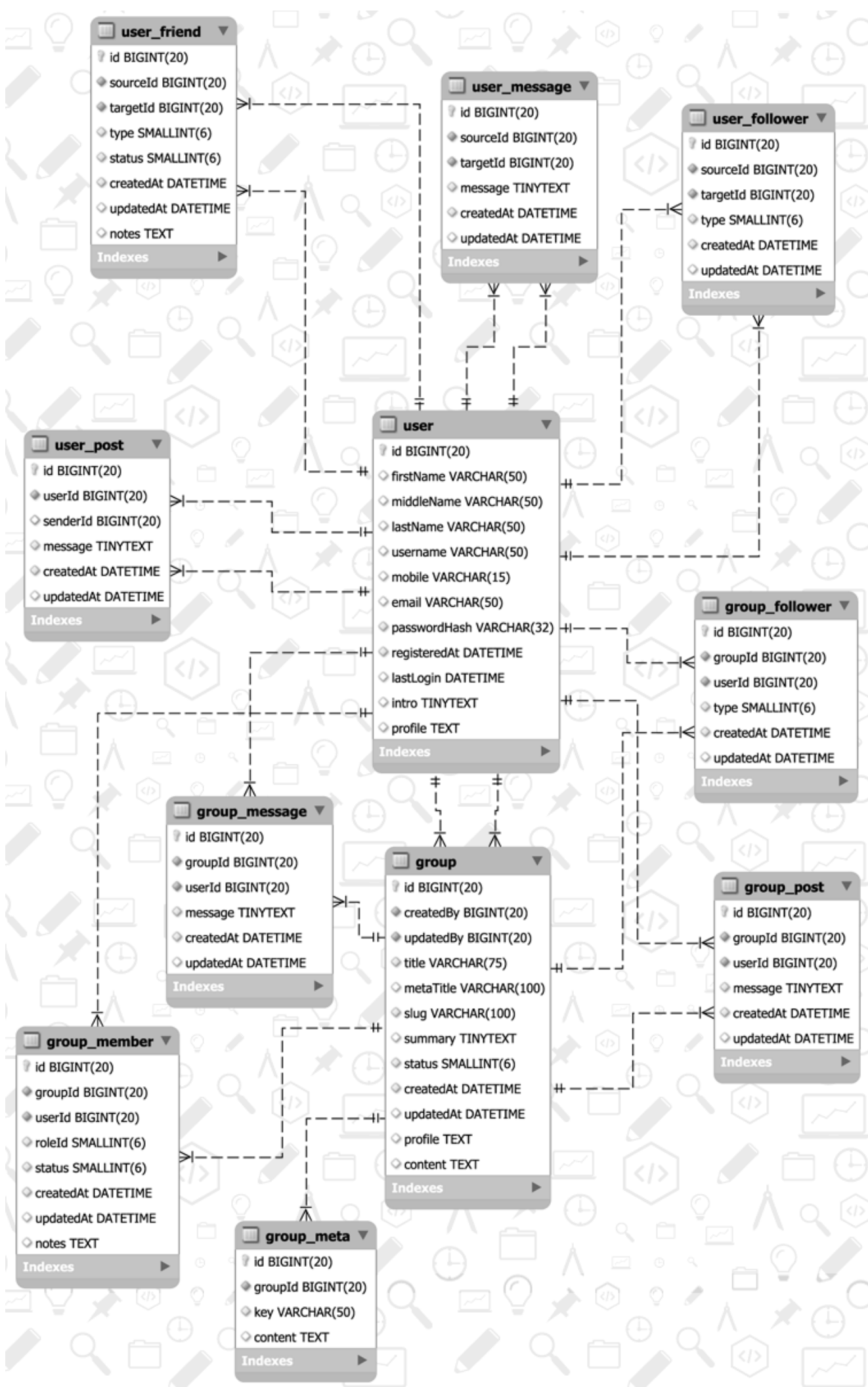
DBMS Mini Project

Topic: Database For Social Network System

Team :

Aaditya P	RA1911027010003
Harishankar P V	RA1911027010022

1. The Entity Relationship Diagram



User Table

The **User Table** is used to store user information. Below mentioned is the description of all the columns of the User Table.

Id	The unique id to identify the user.
First Name	The first name of the user.
Middle Name	The middle name of the user.
Last Name	The last name of the user.
Mobile	The mobile number of the user. It can be used for login and registration purposes.
Email	The email of the user. It can be used for login and registration purposes.
Password Hash	The password hash generated by the appropriate algorithm. We must avoid storing plain or encrypted passwords.
Registered At	This column can be used to calculate the life of the user with the application.
Last Login	It can be used to identify the last login of the user.
Intro	The brief introduction of the User.
Profile	User details.

User Friend Table

The **User Friend Table** to store the user's friends. The friend status can be used to track the friendship status and type can be used to specify the type of friendship. Below mentioned is the description of all the columns of the User Friend Table.

Id	The unique id to identify the friendship.
Source Id	The user id to identify the user who initiated the friendship.
Target Id	The user id of the friend.
Type	The type to classify friends. It can be School, College, or Acquaintance.
Status	The status can be New, Rejected, or Active.

Created At	It stores the date and time at which the friend request was initiated.
Updated At	It stores the date and time at which the friend request was updated.
Notes	It stores the notes specific to the friendship.

User Follower Table

The **User Follower Table** to store the user followers. The follower type can be used to specify the type of follower among Like, Dislike, or Follow. Below mentioned is the description of all the columns of the User Follower Table.

Id	The unique id to identify the follower.
Source Id	The user id to identify the follower user.
Target Id	The user id to identify the following user.
Type	The type to classify followers. It can be Like, Dislike, or Follow.
Created At	It stores the date and time at which the follower was created.
Updated At	It stores the date and time at which the follower was updated.

User Message Table

The **User Message Table** to store the user chat messages. Below mentioned is the description of all the columns of the User Message Table.

Id	The unique id to identify the message.
Source Id	The user id to identify the sender.
Target Id	The user id to identify the receiver.
Message	The message body.
Created At	It stores the date and time at which the message was created.
Updated At	It stores the date and time at which the message was updated.

User Post Table

The **User Post Table** to store the user posts. The sender might be required to allow the other users with appropriate permissions to post on the user wall. Below mentioned is the description of all the columns of the User Post Table.

Id	The unique id to identify the post.
User Id	The user id to identify the corresponding user.
Sender Id	The sender id to identify the corresponding sender.
Message	The message body.
Created At	It stores the date and time at which the post was created.
Updated At	It stores the date and time at which the post was updated.

Group Table

The **Group Table** to store the group data. Below mentioned is the description of all the columns of the Group Table.

Id	The unique id to identify the group.
Created By	The user id to identify the user who registered the group.
Updated By	The user id to identify the user who updated the group.
Title	The group title.
Meta Title	The meta title to be used for browser title and SEO purposes.
Slug	The slug to form the unique URL.
Summary	The summary to mention the key highlights.
Status	The status of the group can be New, Approved, Active, or Blocked.
Created At	It stores the date and time at which the group is created.
Updated At	It stores the date and time at which the group is updated.
Profile	The column used to store the profile details of the group.
Content	The column used to store the additional details of the group.

Group Meta

The Group Meta Table can be used to store additional information about groups including the group banner URL etc. Below mentioned is the description of all the columns of the Group Meta Table.

Id	The unique id to identify the group meta.
Group Id	The group id to identify the parent group.
Key	The key identifying the meta.
Content	The column used to store the group's metadata.

Group Member Table

The **Group Member Table** to store the group members. The member status can be used to track the membership status and member role can be used to identify the member privileges. Below mentioned is the description of all the columns of the Group Member Table.

Id	The unique id to identify the membership.
Group Id	The group id to identify the corresponding group.
User Id	The user id to identify the corresponding user.
Role Id	The role to check user privileges.
Status	The status can be New, Rejected, Active, or Blocked.
Created At	It stores the date and time at which the member request was initiated.
Updated At	It stores the date and time at which the member was updated.
Notes	It stores the notes specific to the membership.

Group Follower Table

The **Group Follower Table** to store the group followers. The follower type can be Like, Dislike, or Follow. Below mentioned is the description of all the columns of the Group Follower Table.

Id	The unique id to identify the follower.
Group Id	The group id to identify the corresponding group.
User Id	The user id to identify the corresponding user.

Type	The follower type can be Like, Dislike, or Follow.
Created At	It stores the date and time at which the follower was created.
Updated At	It stores the date and time at which the follower was updated.

Group Message Table

The **Group Message Table** to store the group chat messages. Below mentioned is the description of all the columns of the Group Message Table.

Id	The unique id to identify the message.
Group Id	The group id to identify the corresponding group.
User Id	The user id to identify the corresponding user.
Message	The message body.
Created At	It stores the date and time at which the message was created.
Updated At	It stores the date and time at which the message was updated.

Group Post Table

The **Group Post Table** to store the group posts. The members having the appropriate role can post to the group. Below mentioned is the description of all the columns of the Group Post Table.

Id	The unique id to identify the post.
Group Id	The group id to identify the corresponding group.
User Id	The user id to identify the corresponding user.
Message	The message body.
Created At	It stores the date and time at which the post was created.
Updated At	It stores the date and time at which the post was updated.

MySQL Commands

```
sudo mysql -u harryneo -p

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> CREATE SCHEMA `sns` DEFAULT CHARACTER SET utf8mb4 COLLATE u
tf8mb4_unicode_ci;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE TABLE `sns`.`user` (
  -> `id` BIGINT NOT NULL AUTO_INCREMENT,
  -> `firstName` VARCHAR(50) NULL DEFAULT NULL,
  -> `middleName` VARCHAR(50) NULL DEFAULT NULL,
  -> `lastName` VARCHAR(50) NULL DEFAULT NULL,
  -> `username` VARCHAR(50) NULL DEFAULT NULL,
  -> `mobile` VARCHAR(15) NULL,
  -> `email` VARCHAR(50) NULL,
  -> `passwordHash` VARCHAR(32) NOT NULL,
  -> `registeredAt` DATETIME NOT NULL,
  -> `lastLogin` DATETIME NULL DEFAULT NULL,
  -> `intro` TINYTEXT NULL DEFAULT NULL,
  -> `profile` TEXT NULL DEFAULT NULL,
  -> PRIMARY KEY (`id`),
  -> UNIQUE INDEX `uq_username` (`username` ASC),
  -> UNIQUE INDEX `uq_mobile` (`mobile` ASC),
  -> UNIQUE INDEX `uq_email` (`email` ASC));
Query OK, 0 rows affected (0.05 sec)

mysql>
```

```
sudo mysql -u harryneo -p

mysql> CREATE TABLE `sns`.`user_friend` (
  -> `id` BIGINT NOT NULL AUTO_INCREMENT,
  -> `sourceId` BIGINT NOT NULL,
  -> `targetId` BIGINT NOT NULL,
  -> `type` SMALLINT NOT NULL DEFAULT 0,
  -> `status` SMALLINT NOT NULL DEFAULT 0,
  -> `createdAt` DATETIME NOT NULL,
  -> `updatedAt` DATETIME NULL DEFAULT NULL,
  -> `notes` TEXT NULL DEFAULT NULL,
  -> PRIMARY KEY (`id`),
  -> INDEX `idx_friend_source` (`sourceId` ASC),
  -> CONSTRAINT `fk_friend_source`
  -> FOREIGN KEY (`sourceId`)
  -> REFERENCES `sns`.`user` (`id`)
  -> ON DELETE NO ACTION
  -> ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.05 sec)

mysql>
mysql> ALTER TABLE `sns`.`user_friend`
  -> ADD INDEX `idx_friend_target` (`targetId` ASC);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE `sns`.`user_friend`
  -> ADD CONSTRAINT `fk_friend_target`
  -> FOREIGN KEY (`targetId`)
  -> REFERENCES `sns`.`user` (`id`)
  -> ON DELETE NO ACTION
  -> ON UPDATE NO ACTION;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```



```
mysql> CREATE TABLE `sns`.`user_follower` (  
-> `id` BIGINT NOT NULL AUTO_INCREMENT,  
-> `sourceId` BIGINT NOT NULL,  
-> `targetId` BIGINT NOT NULL,  
-> `type` SMALLINT NOT NULL DEFAULT 0,  
-> `createdAt` DATETIME NOT NULL,  
-> `updatedAt` DATETIME NULL DEFAULT NULL,  
-> PRIMARY KEY (`id`),  
-> INDEX `idx_ufollower_source` (`sourceId` ASC),  
-> CONSTRAINT `fk_ufollower_source`  
-> FOREIGN KEY (`sourceId`)  
-> REFERENCES `sns`.`user` (`id`)  
-> ON DELETE NO ACTION  
-> ON UPDATE NO ACTION);  
Query OK, 0 rows affected (0.04 sec)  
  
mysql>  
mysql> ALTER TABLE `sns`.`user_follower`  
-> ADD INDEX `idx_ufollower_target` (`targetId` ASC);  
Query OK, 0 rows affected (0.02 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE `sns`.`user_follower`  
-> ADD CONSTRAINT `fk_ufollower_target`  
-> FOREIGN KEY (`targetId`)  
-> REFERENCES `sns`.`user` (`id`)  
-> ON DELETE NO ACTION  
-> ON UPDATE NO ACTION;  
Query OK, 0 rows affected (0.08 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql>  
mysql>
```

```
mysql> CREATE TABLE `sns`.`user_message` (  
-> `id` BIGINT NOT NULL AUTO_INCREMENT,  
-> `sourceId` BIGINT NOT NULL,  
-> `targetId` BIGINT NOT NULL,  
-> `message` TINYTEXT NULL DEFAULT NULL,  
-> `createdAt` DATETIME NOT NULL,  
-> `updatedAt` DATETIME NULL DEFAULT NULL,  
-> PRIMARY KEY (`id`),  
-> INDEX `idx_umessage_source` (`sourceId` ASC),  
-> CONSTRAINT `fk_umessage_source`  
-> FOREIGN KEY (`sourceId`)  
-> REFERENCES `sns`.`user` (`id`)  
-> ON DELETE NO ACTION  
-> ON UPDATE NO ACTION);  
Query OK, 0 rows affected (0.03 sec)  
  
mysql>  
mysql> ALTER TABLE `sns`.`user_message`  
-> ADD INDEX `idx_umessage_target` (`targetId` ASC);  
Query OK, 0 rows affected (0.02 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE `sns`.`user_message`  
-> ADD CONSTRAINT `fk_umessage_target`  
-> FOREIGN KEY (`targetId`)  
-> REFERENCES `sns`.`user` (`id`)  
-> ON DELETE NO ACTION  
-> ON UPDATE NO ACTION;  
Query OK, 0 rows affected (0.07 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql>
```

```

mysql> CREATE TABLE `sns`.`user_post` (
  -> `id` BIGINT NOT NULL AUTO_INCREMENT,
  -> `userId` BIGINT NOT NULL,
  -> `senderId` BIGINT NOT NULL,
  -> `message` TINYTEXT NULL DEFAULT NULL,
  -> `createdAt` DATETIME NOT NULL,
  -> `updatedAt` DATETIME NULL DEFAULT NULL,
  -> PRIMARY KEY (`id`),
  -> INDEX `idx_upost_user` (`userId` ASC),
  -> CONSTRAINT `fk_upost_user`
  -> FOREIGN KEY (`userId`)
  -> REFERENCES `sns`.`user` (`id`)
  -> ON DELETE NO ACTION
  -> ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql> ALTER TABLE `sns`.`user_post`
  -> ADD INDEX `idx_upost_sender` (`senderId` ASC);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE `sns`.`user_post`
  -> ADD CONSTRAINT `fk_upost_sender`
  -> FOREIGN KEY (`senderId`)
  -> REFERENCES `sns`.`user` (`id`)
  -> ON DELETE NO ACTION
  -> ON UPDATE NO ACTION;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>

```

```

mysql> CREATE TABLE `sns`.`group` (
  -> `id` BIGINT NOT NULL AUTO_INCREMENT,
  -> `createdBy` BIGINT NOT NULL,
  -> `updatedBy` BIGINT NOT NULL,
  -> `title` VARCHAR(75) NOT NULL,
  -> `metaTitle` VARCHAR(100) NULL,
  -> `slug` VARCHAR(100) NOT NULL,
  -> `summary` TINYTEXT NULL,
  -> `status` SMALLINT NOT NULL DEFAULT 0,
  -> `createdAt` DATETIME NOT NULL,
  -> `updatedAt` DATETIME NULL DEFAULT NULL,
  -> `profile` TEXT NULL DEFAULT NULL,
  -> `content` TEXT NULL DEFAULT NULL,
  -> PRIMARY KEY (`id`),
  -> UNIQUE INDEX `uq_slug` (`slug` ASC),
  -> INDEX `idx_group_creator` (`createdBy` ASC),
  -> CONSTRAINT `fk_group_creator`
  -> FOREIGN KEY (`createdBy`)
  -> REFERENCES `sns`.`user` (`id`)
  -> ON DELETE NO ACTION
  -> ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.05 sec)

mysql>
mysql> ALTER TABLE `sns`.`group`
  -> ADD INDEX `idx_group_modifier` (`updatedBy` ASC);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE `sns`.`group`
  -> ADD CONSTRAINT `fk_group_modifier`
  -> FOREIGN KEY (`updatedBy`)
  -> REFERENCES `sns`.`user` (`id`)

```

```
mysql> ALTER TABLE `sns`.`group`
-> ADD INDEX `idx_group_modifier` (`updatedAt` ASC);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE `sns`.`group`
-> ADD CONSTRAINT `fk_group_modifier`
-> FOREIGN KEY (`updatedAt`)
-> REFERENCES `sns`.`user` (`id`)
-> ON DELETE NO ACTION
-> ON UPDATE NO ACTION;
Query OK, 0 rows affected (0.11 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE TABLE `sns`.`group_meta` (
-> `id` BIGINT NOT NULL AUTO_INCREMENT,
-> `groupId` BIGINT NOT NULL,
-> `key` VARCHAR(50) NOT NULL,
-> `content` TEXT NULL DEFAULT NULL,
-> PRIMARY KEY (`id`),
-> INDEX `idx_meta_group` (`groupId` ASC),
-> UNIQUE INDEX `uq_meta_group` (`groupId` ASC, `key` ASC),
-> CONSTRAINT `fk_meta_group`
-> FOREIGN KEY (`groupId`)
-> REFERENCES `sns`.`group` (`id`)
-> ON DELETE NO ACTION
-> ON UPDATE NO ACTION)
-> ENGINE = InnoDB;
Query OK, 0 rows affected (0.04 sec)

mysql>
```

```
mysql> CREATE TABLE `sns`.`group_member` (
-> `id` BIGINT NOT NULL AUTO_INCREMENT,
-> `groupId` BIGINT NOT NULL,
-> `userId` BIGINT NOT NULL,
-> `type` SMALLINT NOT NULL DEFAULT 0,
-> `status` SMALLINT NOT NULL DEFAULT 0,
-> `createdAt` DATETIME NOT NULL,
-> `updatedAt` DATETIME NULL DEFAULT NULL,
-> `notes` TEXT NULL DEFAULT NULL,
-> PRIMARY KEY (`id`),
-> INDEX `idx_member_group` (`groupId` ASC),
-> CONSTRAINT `fk_member_group`
-> FOREIGN KEY (`groupId`)
-> REFERENCES `sns`.`group` (`id`)
-> ON DELETE NO ACTION
-> ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql> ALTER TABLE `sns`.`group_member`
-> ADD INDEX `idx_member_user` (`userId` ASC);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE `sns`.`group_member`
-> ADD CONSTRAINT `fk_member_user`
-> FOREIGN KEY (`userId`)
-> REFERENCES `sns`.`user` (`id`)
-> ON DELETE NO ACTION
-> ON UPDATE NO ACTION;
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> CREATE TABLE `sns`.`group_follower` (  
-> `id` BIGINT NOT NULL AUTO_INCREMENT,  
-> `groupId` BIGINT NOT NULL,  
-> `userId` BIGINT NOT NULL,  
-> `type` SMALLINT NOT NULL DEFAULT 0,  
-> `createdAt` DATETIME NOT NULL,  
-> `updatedAt` DATETIME NULL DEFAULT NULL,  
-> PRIMARY KEY (`id`),  
-> INDEX `idx_gfollower_group` (`groupId` ASC),  
-> CONSTRAINT `fk_gfollower_group`  
-> FOREIGN KEY (`groupId`)  
-> REFERENCES `sns`.`group` (`id`)  
-> ON DELETE NO ACTION  
-> ON UPDATE NO ACTION);  
Query OK, 0 rows affected (0.04 sec)  
  
mysql>  
mysql> ALTER TABLE `sns`.`group_follower`  
-> ADD INDEX `idx_gfollower_user` (`userId` ASC);  
Query OK, 0 rows affected (0.02 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE `sns`.`group_follower`  
-> ADD CONSTRAINT `fk_gfollower_user`  
-> FOREIGN KEY (`userId`)  
-> REFERENCES `sns`.`user` (`id`)  
-> ON DELETE NO ACTION  
-> ON UPDATE NO ACTION;  
Query OK, 0 rows affected (0.07 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql>  
mysql> ALTER TABLE `sns`.`group_follower` ADD UNIQUE `uq_friend`(`
```

```
mysql> CREATE TABLE `sns`.`group_message` (  
-> `id` BIGINT NOT NULL AUTO_INCREMENT,  
-> `groupId` BIGINT NOT NULL,  
-> `userId` BIGINT NOT NULL,  
-> `message` TINYTEXT NULL DEFAULT NULL,  
-> `createdAt` DATETIME NOT NULL,  
-> `updatedAt` DATETIME NULL DEFAULT NULL,  
-> PRIMARY KEY (`id`),  
-> INDEX `idx_gmessage_group` (`groupId` ASC),  
-> CONSTRAINT `fk_gmessage_group`  
-> FOREIGN KEY (`groupId`)  
-> REFERENCES `sns`.`group` (`id`)  
-> ON DELETE NO ACTION  
-> ON UPDATE NO ACTION);  
Query OK, 0 rows affected (0.03 sec)  
  
mysql>  
mysql> ALTER TABLE `sns`.`group_message`  
-> ADD INDEX `idx_gmessage_user` (`userId` ASC);  
Query OK, 0 rows affected (0.02 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE `sns`.`group_message`  
-> ADD CONSTRAINT `fk_gmessage_user`  
-> FOREIGN KEY (`userId`)  
-> REFERENCES `sns`.`user` (`id`)  
-> ON DELETE NO ACTION  
-> ON UPDATE NO ACTION;  
Query OK, 0 rows affected (0.07 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql>
```

```
sudo mysql -u harryneo -p

mysql> CREATE TABLE `sns`.`group_post` (
  -> `id` BIGINT NOT NULL AUTO_INCREMENT,
  -> `groupId` BIGINT NOT NULL,
  -> `userId` BIGINT NOT NULL,
  -> `message` TINYTEXT NULL DEFAULT NULL,
  -> `createdAt` DATETIME NOT NULL,
  -> `updatedAt` DATETIME NULL DEFAULT NULL,
  -> PRIMARY KEY (`id`),
  -> INDEX `idx_gpost_group` (`groupId` ASC),
  -> CONSTRAINT `fk_gpost_group`
  -> FOREIGN KEY (`groupId`)
  -> REFERENCES `sns`.`group` (`id`)
  -> ON DELETE NO ACTION
  -> ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> ALTER TABLE `sns`.`group_post`
  -> ADD INDEX `idx_gpost_user` (`userId` ASC);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE `sns`.`group_post`
  -> ADD CONSTRAINT `fk_gpost_user`
  -> FOREIGN KEY (`userId`)
  -> REFERENCES `sns`.`user` (`id`)
  -> ON DELETE NO ACTION
  -> ON UPDATE NO ACTION;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> |
```

MySQL Workbench Screenshots of Database

