William Garnes Ebuka Okpala Team 13

Database Management Systems Technical Report

System Design: For our database based on the requirements we create seven different entities that we wanted our database to be built around. These entities were users, media, playlists, groups, conversations, comments, and messages. Although those were the main entities that we worked with, we also has many other tables in our system to represent the different relationships between the different entities. We made sure that data was not being repeated throughout the system which is why we had so many tables in order to eliminate any redundancy that might exist throughout the system.

ER Diagram: We have placed an image of our ER diagram at the end of this report. It shows our chosen entities as well as the relationships that exist between each one of those entities.

Database Schema: Below are the SQL commands that we used to create our tables. At the bottom of this report there is also a picture of the schema and what the database design looks like which we got from phpmyadmin.

```
create table users
(
  user_id INTEGER NOT NULL AUTO_INCREMENT,
  first_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  email VARCHAR(100) NOT NULL,
  user_name VARCHAR(100) NOT NULL,
  user_password VARCHAR(100) NOT NULL,
  display_name VARCHAR(100) NOT NULL,
  PRIMARY KEY ( user_id )
) ENGINE=INNODB;

create table contacts
(
  contact_1 INTEGER NOT NULL REFERENCES users(user_id),
  contact_2 INTEGER NOT NULL REFERENCES users(user_id),
  category INT NOT NULL,
  FOREIGN KEY (contact_1) REFERENCES users(user_id),
  FOREIGN KEY (contact_2) REFERENCES users(user_id),
  primary key(contact_1, contact_2)
) ENGINE=INNODB;

create table media
(
  media_id INTEGER NOT NULL AUTO_INCREMENT,
  uploader INTEGER NOT NULL REFERENCES users(user_id),
  title VARCHAR(100) NOT NULL,
  media_description VARCHAR(255) NOT NULL,
  media_type VARCHAR(10) NOT NULL,
  extension VARCHAR(10) NOT NULL,
  share_type TINYINT NOT NULL,
  discussion_type TINYINT NOT NULL,
  rating_type TINYINT NOT NULL,
  view_count INTEGER NOT NULL,
  file_path VARCHAR(2048) NOT NULL,
  file_size INTEGER NOT NULL,
  upload_time DATETIME,
  FOREIGN KEY (uploader) REFERENCES users(user_id),
  PRIMARY KEY ( media_id )
) ENGINE=INNODB;

create table media_categories
(
  media_id INTEGER NOT NULL REFERENCES media(media_id),
  category TINYINT NOT NULL,
  FOREIGN KEY (media_id) REFERENCES media(media_id) ON DELETE CASCADE
) ENGINE=INNODB;

create table media_keywords
(
  media_id INTEGER NOT NULL REFERENCES media(media_id),
  keyword VARCHAR(100) NOT NULL,
  FOREIGN KEY (media_id) REFERENCES media(media_id) ON DELETE CASCADE
) ENGINE=INNODB;

create table media_blocks
(
  media_id INTEGER NOT NULL REFERENCES media(media_id),
  user_id INTEGER NOT NULL REFERENCES users(user_id),
  FOREIGN KEY (media_id) REFERENCES media(media_id),
  FOREIGN KEY (user_id) REFERENCES users(user_id)
) ENGINE=INNODB;

create table media_ratings
(
  media_id INTEGER NOT NULL REFERENCES media(media_id),
  user_id INTEGER NOT NULL REFERENCES users(user_id),
  rating TINYINT NOT NULL,
  FOREIGN KEY (media_id) REFERENCES media(media_id) ON DELETE CASCADE,
  FOREIGN KEY (user_id) REFERENCES users(user_id)
) ENGINE=INNODB;

create table subscriptions
(
  contact_1 INTEGER NOT NULL REFERENCES users(user_id),
  contact_2 INTEGER NOT NULL REFERENCES users(user_id),
  FOREIGN KEY (contact_1) REFERENCES users(user_id),
  FOREIGN KEY (contact_2) REFERENCES users(user_id),
  primary key(contact_1, contact_2)
) ENGINE=INNODB;

create table conversations
(
  conv_id INTEGER NOT NULL AUTO_INCREMENT,
  conv_name VARCHAR(100) NOT NULL,
  conv_type TINYINT NOT NULL,
  primary key(conv_id)
) ENGINE=INNODB;

create table comments
(
  user_id INTEGER NOT NULL REFERENCES users(user_id),
  conv_id INTEGER NOT NULL REFERENCES conversations(conv_id),
  write_time DATETIME NOT NULL,
  comment VARCHAR(255) NOT NULL,
  FOREIGN KEY (user_id) REFERENCES users(user_id),
  FOREIGN KEY (conv_id) REFERENCES conversations(conv_id),
  primary key(user_id, write_time)
) ENGINE=INNODB;

create table groups
(
  group_id INTEGER NOT NULL AUTO_INCREMENT,
  group_name VARCHAR(100) NOT NULL,
  primary key(group_id)
) ENGINE=INNODB;

create table group_conversations
(
  conv_id INTEGER NOT NULL REFERENCES conversations(conv_id),
  group_id INTEGER NOT NULL REFERENCES groups(group_id),
  FOREIGN KEY (conv_id) REFERENCES conversations(conv_id),
  FOREIGN KEY (group_id) REFERENCES groups(group_id),
  primary key(conv_id, group_id)
) ENGINE=INNODB;

create table comment_section
(
  media_id INTEGER NOT NULL REFERENCES media(media_id),
  conv_id INTEGER NOT NULL REFERENCES conversations(conv_id),
  FOREIGN KEY (media_id) REFERENCES media(media_id),
  FOREIGN KEY (conv_id) REFERENCES conversations(conv_id),
  primary key (media_id, conv_id)
) ENGINE=INNODB;

create table group_members
(
  group_id INTEGER NOT NULL REFERENCES groups(group_id),
  user_id INTEGER NOT NULL REFERENCES users(user_id),
  FOREIGN KEY (group_id) REFERENCES groups(group_id),
  FOREIGN KEY (user_id) REFERENCES users(user_id),
  primary key(group_id,user_id)
) ENGINE=INNODB;

create table messages
(
  contact_1 INTEGER NOT NULL REFERENCES users(user_id),
  contact_2 INTEGER NOT NULL REFERENCES users(user_id),
  conv_id INTEGER NOT NULL REFERENCES conversations(conv_id),
  FOREIGN KEY (contact_1) REFERENCES users(user_id),
  FOREIGN KEY (contact_2) REFERENCES users(user_id),
  FOREIGN KEY (conv_id) REFERENCES conversations(conv_id),
  primary key (contact_1, contact_2)
) ENGINE=INNODB;

create table playlists
(
  playlist_id INTEGER NOT NULL AUTO_INCREMENT,
  user_id INTEGER NOT NULL REFERENCES users(user_id),
  playlist_name VARCHAR(100) NOT NULL,
  FOREIGN KEY (user_id) REFERENCES users(user_id),
  primary key (playlist_id)
) ENGINE=INNODB;
```

```
create table playlist_media
(
  playlist_id INTEGER NOT NULL REFERENCES playlists(playlist_id),
  media_id INTEGER NOT NULL REFERENCES media(media_id),
  FOREIGN KEY (playlist_id) REFERENCES playlists(playlist_id) ON DELETE CASCADE,
  FOREIGN KEY (media_id) REFERENCES media(media_id) ON DELETE CASCADE,
  primary key (playlist_id, media_id)
) ENGINE=INNODB;
```

Function Design: For this project all of our database queries that we performed, we did in a file within our project called database_queries.php. These queries were the basis for the functionality of of system. Please take a look at that file to see the different queries that we wrote for our project.

Implementation details: To implement our system we used the web design tools that we were taught in class including HTML, Javascript, CSS, PHP. In order to to build certain parts of the website we often referred to w3schools.com to learn/review different tools as well as to look at different examples on how to create different parts of a website. We also used some other commonly used third party tool web development tools like JQuery and Bootstrap to assist in the design of our website. JQuery was especially helpful because it allowed us to perform different posts without having to navigate to a new page. Of course we also used MySQL with was taught to us in class. To store our progress as well as in order to share code and work independently we used buffet in order to hold our repsosity.

Test cases/results: In order to test our system we would write and perform queries into phpmyadmin before we placed them into our system. From doing this we were able to make sure the queries were valid before we attempted to use them. This also allowed us to to trial and error on different queries especially when we started doing multiple joins on tables in order to get data that was spread across more than one table. From doing this we were able to spend less time debugging errors that were related to SQL queries.

We also would test our system as we developed each feature. As we developed a feature we would test small parts of them to make sure that we were not making a lot a changes and then coming up with bugs that were too hard to fix. After a feature was done we would test it with different different accounts and by navigating from different pages, as well as testing features while logged in and logged out. By performing iterative tests on small parts we were able to prevents ourselves from being hung up for too long one one particular problem.

Another thing that we did in relation to iterative tests is we would test previously finished features when new ones were completed. The reason for this is because sometimes when we completed a new feature we sometimes introduced new bugs to features that we had already completed. From doing this we were able to find flaws with the design of our system and we also had more or an idea where the bug could have been introduced because it would have occured when then last feature was implemented.

User manual:

Registration: Click on sign in and then click the create account button. The will lead you to the create account page where you can register your information

Sign In: Click on the sign in button and then enter the information that you used to sign in. You can sign in with either your username or password.

Profile Update: On either the left hand collapsible menu or the icon with your initial at the top right click on the settings button. From that point there will setting were you can change your name, your email, your username, your password and your display name

Contacts: Go to the initial at the top right hand side of the screen and click on contacts. Contacts will be split into three different groups (contacts, friends, and blocked). People who are friends will also show up in the contacts column. People can be switched from one group to another at any time. However, if you are blocked by someone else you can add them to your block list as well but you can not add them to your friends or contact group.

Uploading Media: While logged in click on the upload button to upload information. Right now once you set the information of a piece of media you cannot change it.

Media Blocking: Navigate to your channel by clicking on the initial button on the top right corner. From there you can see your media. Click on the manage access button on one the the videos and the search for a person. After that you can block them.

Media Unblocking: Go to your setting page and then click on view blocked users. From there you can unblock people from different media.

Media Browsing: You can browse media on your own channel or other peoples channels. Media can then be filtered or sorted based on different categories or different information about the media.

Playlists: To add things to a playlist, go to their page where you can view them and there will be a save button. This will allow you to add things to your playlists as well as create new playlists if you need to.

Messaging: Go to your contacts page and click on send message for friends or normal contacts. You can also click on the speech bubble in the top right corner to go directly to messages for specific users.

Media Commenting: You can make comments on media if the settings are enabled for it. To make a comment click on the text box and then create a message. In order to see a your new message you will need to refresh the page.

Media Rating: You can rate media by clicking on one of the five stars that appears underneath the media.

Groups: Go to the initial at the top right hand side of the screen and click on my groups. From there you can see the groups that you are apart of as well as make new groups. When you click on a group you can see all of the discussions that are involved with that particular group. From there you can add new people to a group or create a new discussion within that group. The only way a be apart of a group is to make one or to be added to one by someone who is already a member of the group. When you click on go to discussion for a particular group, you will be able to view the discussion as well as add your own comments.

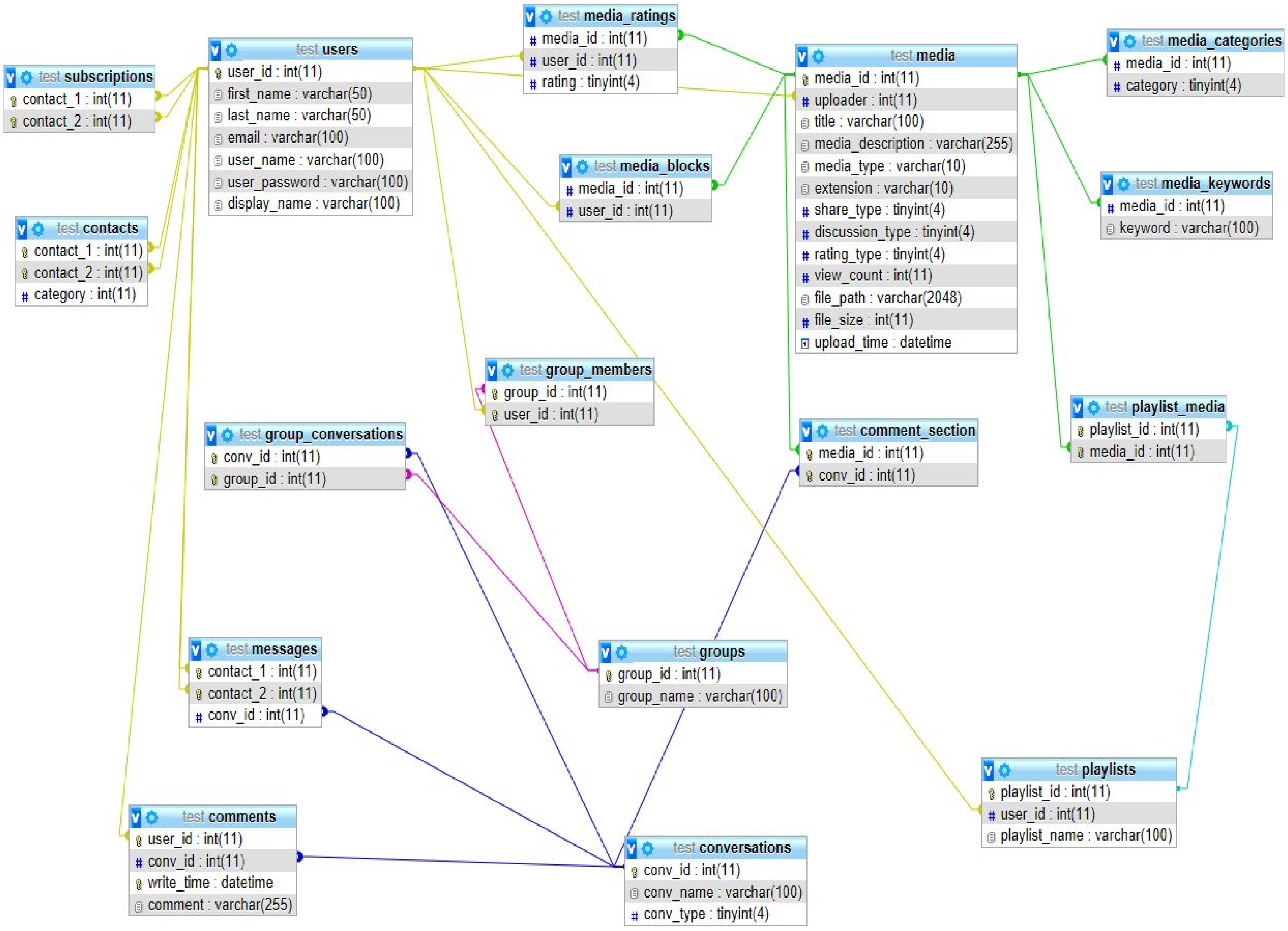
Searching: You can search media by using the search bar at the top. Results that come up will be results that have the same keywords as the information that you entered. One you search for something you can then filter out the results by using the filter bar. You can filter things by their category as well as order the results by different features of the media.

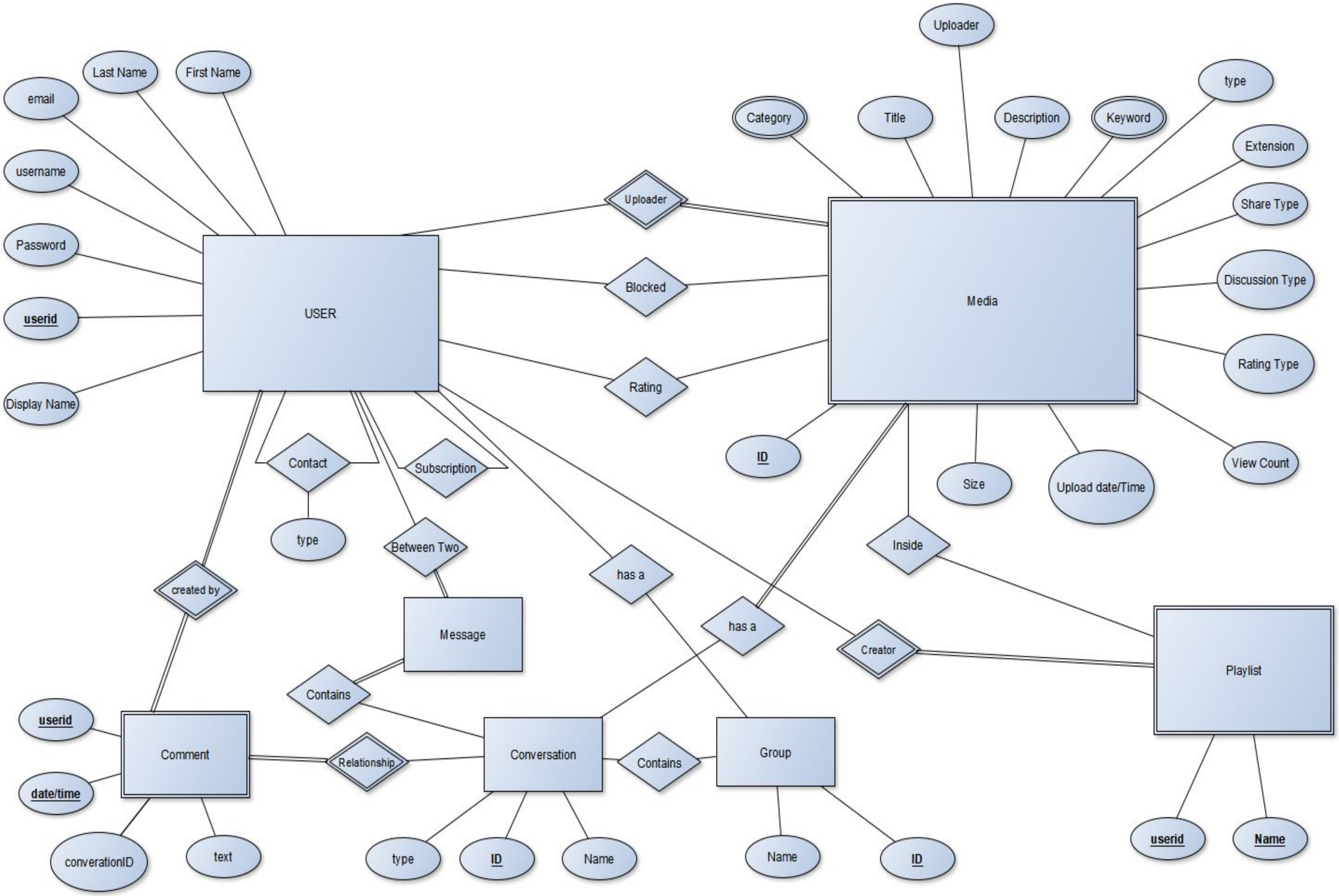
Word Cloud: When you click on the word cloud you will see the different keywords that are used in the system. If you click one of the keywords it will perform a search for you for that particular word.

Viewing/Downloading Media: You can view media when you click on that media from places like a channel, the home page, searching, etc. Once you view the media you can download it by clicking on the download button. Below that comments for a video you will be able to see other recommended media. Media is recommended if it has the same keywords and/or categories as the media that you are currently viewing. The media recommendations will max out at 10 medium.

Other Information:

Display Name: What you can other people see when you make comments. This will also be the name of your channel.


test subscriptions
contact_1 : int(11)
contact_2 : int(11)
test users
user_id : int(11)
first_name : varchar(50)
last_name : varchar(50)
email : varchar(100)
user_name : varchar(100)
user_password : varchar(100)
display_name : varchar(100)
test media_ratings
media_id : int(11)
user_id : int(11)
rating : tinyint(4)
test media
media_id : int(11)
uploader : int(11)
title : varchar(100)
media_description : varchar(255)
media_type : varchar(10)
extension : varchar(10)
share_type : tinyint(4)
discussion_type : tinyint(4)
rating_type : tinyint(4)
view_count : int(11)
file_path : varchar(2048)
file_size : int(11)
upload_time : datetime
test media_categories
media_id : int(11)
category : tinyint(4)
test contacts
contact_1 : int(11)
contact_2 : int(11)
category : int(11)
test media_blocks
media_id : int(11)
user_id : int(11)
test media_keywords
media_id : int(11)
keyword : varchar(100)
test group_members
group_id : int(11)
user_id : int(11)
test playlist_media
playlist_id : int(11)
media_id : int(11)
test group_conversations
conv_id : int(11)
group_id : int(11)
test comment_section
media_id : int(11)
conv_id : int(11)
test messages
contact_1 : int(11)
contact_2 : int(11)
conv_id : int(11)
test groups
group_id : int(11)
group_name : varchar(100)
test playlists
playlist_id : int(11)
user_id : int(11)
playlist_name : varchar(100)
test comments
user_id : int(11)
conv_id : int(11)
write_time : datetime
comment : varchar(255)
test conversations
conv_id : int(11)
conv_name : varchar(100)
conv_type : tinyint(4)



email
Last Name
First Name
username
Password
userid
Display Name
USER
Uploader
Blocked
Rating
Contact
type
Subscription
Between Two
Message
created by
Contains
has a
has a
Creator
Inside
Media
Uploader
Category
Title
Description
Keyword
type
Extension
Share Type
Discussion Type
Rating Type
View Count
ID
Size
Upload date/Time
Playlist
userid
Name
userid
Comment
date/time
conveationID
text
Relationship
Conversation
type
ID
Name
Contains
Group
Name
ID