CS460
Final Report of PhotoShare

Hanzi Yu & Mengshu Zhang

Final schema:

```
CREATE DATABASE photoshare;
USE photoshare;

CREATE TABLE Users (
    user_id int4  AUTO_INCREMENT,
    first_name VARCHAR(255) NOT NULL,
            last_name VARCHAR(255) NOT NULL,
    email varchar(255) NOT NULL,
    date_of_birth DATE NOT NULL,
            gender VARCHAR(255),
    hometown VARCHAR(100),
            password VARCHAR (255) NOT NULL,
    UNIQUE (email),
    CONSTRAINT users_pk PRIMARY KEY (user_id)
);


CREATE TABLE Friends (
              user_id1 INT4 NOT NULL,
              user_id2 INT4 NOT NULL,
  CONSTRAINT users_fk FOREIGN KEY (user_id1) REFERENCES Users(user_id),
  CONSTRAINT friends_fk FOREIGN KEY (user_id2) REFERENCES Users(user_id)
);

CREATE TABLE Albums (
            album_id INT4 AUTO_INCREMENT,
            name VARCHAR(255),
            date DATE,
    user_id INT4 NOT NULL,
  CONSTRAINT albums_pk PRIMARY KEY (album_id),
  CONSTRAINT albums_fk FOREIGN KEY (user_id) REFERENCES Users(user_id) ON UPDATE
CASCADE ON DELETE CASCADE
);

CREATE TABLE Photos (
    photo_id INT4 AUTO_INCREMENT,
    caption VARCHAR(255),
            data LONGBLOB,
    user_id INT4 NOT NULL,
    album_id INT4 NOT NULL,
  CONSTRAINT photos_pk PRIMARY KEY (photo_id),
  CONSTRAINT photos_fk FOREIGN KEY (user_id) REFERENCES Users(user_id),
  CONSTRAINT photos_fk2 FOREIGN KEY (album_id) REFERENCES Albums(album_id) ON
UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Tagged (
          word VARCHAR(50),
          photo_id INT4 NOT NULL,
   CONSTRAINT tagged_fk FOREIGN KEY (photo_id) REFERENCES Photos(photo_id)
);

CREATE TABLE Comments (
          comment_id INT4 AUTO_INCREMENT,
   text VARCHAR(300),
          user_id INT4 NOT NULL,
          date DATE,
          photo_id INT4 NOT NULL,
  CONSTRAINT comments_pk PRIMARY KEY (comment_id),
  CONSTRAINT comments_fk FOREIGN KEY (user_id) REFERENCES Users(user_id),
  CONSTRAINT comments_fk2 FOREIGN KEY (photo_id) REFERENCES Photos(photo_id)
);

CREATE TABLE Likes (
  user_id INT4,
  photo_id INT4,
  CONSTRAINT likes_fk FOREIGN KEY (user_id) REFERENCES Users(user_id),
  CONSTRAINT likes_fk2 FOREIGN KEY (photo_id) REFERENCES Photos(photo_id)
);


INSERT INTO Users (user_id, first_name, last_name, password, email) VALUES (-1, 'Guest', '',
'','anonymous@anonymous');
```

Additional assumptions:

The anonymous user has user_id=-1
Anonymous user cannot Like.

For each user, he/she cannot have two albums which have same names

Like and Tags, Comments and Friends cannot be deleted.

We are adding friend in the both ends: A added B and then in B’s friend list, it can also see A.

Our system return top10 most popular tags

Our system return friends recommendation based on the how many times each recommended friend appears in the lists of friends of friends in a descending order.

The Limitations of our system:

For each user, he/she cannot have two albums which have same names
Like and Tags, Comments and Friends cannot be deleted.
We cannot upload a photo with a new album name.