Lucy Kisova
Valentina Haddad

CS 460: Photoshare report

Relational Schema:

Users(user_id, gender, email, password, dob, hometown, fname, name)

Albums (album_id, Name, date_of_creation, user_id)

Photos (photo_id, user_id, caption, imgdata, album_id)

Comments (comment_id, text, date, user_id, photo_id,)

Likes (user_id, photo_id)

Tags (tag_id, name)

Tagged (photo_id, tag_id);

Friends (UID1,UID2)

Assumptions:

Users:
The user's email must be valid

Albums:

- Created before any photo can be uploaded
- Can have multiple
- Can have the same names
- Can be empty
- Can be seen by an anonymous user

Photos:

- Must be put inside an album to be uploaded
- Can be seen by an anonymous user

Comments;

- Cannot comment on own posts
- Can not view all comments
- Must know what a comment is to see what user commented on it

Likes:

- User can like their own pictures
- Can like other users' pictures

Tags:

- Can be multiple words long
- Can only tag photos, cannot tag albums
- Only a User can tag their own photo

Friends:

- You cannot be friends with yourself
- You can friend someone but they will not automatically be your friend, they have to add u as well.

SCHEMA:

```
CREATE DATABASE IF NOT EXISTS photoshare;
USE photoshare;
DROP TABLE IF EXISTS Photos CASCADE;
DROP TABLE IF EXISTS Users CASCADE;
DROP TABLE IF EXISTS Albums CASCADE;
DROP TABLE IF EXISTS Comments CASCADE;
DROP TABLE IF EXISTS Likes CASCADE;
DROP TABLE IF EXISTS Tags CASCADE;
DROP TABLE IF EXISTS Tagged CASCADE;
DROP TABLE IF EXISTS Friendship CASCADE;


CREATE TABLE Users (
user_id INT NOT NULL AUTO_INCREMENT,
gender VARCHAR(6),
email VARCHAR(255) NOT NULL UNIQUE,
password VARCHAR(40) NOT NULL,
dob DATE NOT NULL,
hometown VARCHAR(40),
fname VARCHAR(40) NOT NULL,
lname VARCHAR(40) NOT NULL, PRIMARY KEY (user_id)
);

CREATE TABLE Albums (
  album_id INT AUTO_INCREMENT,
  Name VARCHAR(40) NOT NULL,
  date_of_creation DATETIME DEFAULT CURRENT_TIMESTAMP,
  user_id INT NOT NULL,
  PRIMARY KEY (album_id),
  FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE
);

CREATE TABLE Photos (
  photo_id INT AUTO_INCREMENT,
  user_id INT,
  caption VARCHAR(200),
  imgdata LONGBLOB,
  album_id INT NOT NULL,
  PRIMARY KEY (photo_id),
  FOREIGN KEY (user_id) REFERENCES Users(user_id),
  FOREIGN KEY (album_id) REFERENCES Albums(album_id) ON DELETE CASCADE
);

CREATE TABLE Comments (
  comment_id INT NOT NULL AUTO_INCREMENT,
  text TEXT NOT NULL,
  date DATETIME DEFAULT CURRENT_TIMESTAMP,
  user_id INT NOT NULL,
  photo_id INT NOT NULL,
  PRIMARY KEY (comment_id),
  FOREIGN KEY (user_id) REFERENCES Users (user_id) ON DELETE CASCADE,
  FOREIGN KEY (photo_id) REFERENCES Photos (photo_id) ON DELETE CASCADE
);

CREATE TABLE Likes (
  user_id INT NOT NULL,
  photo_id INT NOT NULL,
  PRIMARY KEY (photo_id, user_id),
  FOREIGN KEY (user_id) REFERENCES Users (user_id) ON DELETE CASCADE,
  FOREIGN KEY (photo_id) REFERENCES Photos (photo_id) ON DELETE CASCADE
);

CREATE TABLE Tags (
  tag_id INT AUTO_INCREMENT,
  name VARCHAR(100),
  PRIMARY KEY (tag_id)
);

CREATE TABLE Tagged (
  photo_id INTEGER,
  tag_id INTEGER,
  PRIMARY KEY (photo_id, tag_id),
  FOREIGN KEY(photo_id) REFERENCES Photos(photo_id),
  FOREIGN KEY(tag_id) REFERENCES Tags (tag_id)
);

CREATE TABLE Friends (
  UID1 INT NOT NULL,
  UID2 INT NOT NULL,
  CHECK (UID1 <> UID2),
  PRIMARY KEY(UID1, UID2),
  FOREIGN KEY (UID1) REFERENCES Users (user_id) ON DELETE CASCADE,
  FOREIGN KEY (UID2) REFERENCES Users (user_id) ON DELETE CASCADE
);



INSERT INTO Users (email, password, gender, dob, hometown, fname, lname) VALUES
('test@bu.edu', 'test', 'male', '2002-01-24', 'Boston', 'Test', 'User');
```

Limitations:

Users/friends:

- We do not have an option to view all users on the site before you add them as a friend. You have to know the email of the person you want to add as a friend
- Future updates could include a search function for users so you can add them as a friend without knowing their whole email, or searching by first/last name
- No option to unfollow/block users

Comments:

- Cannot view all comments for each photo
- A comments button can be added so that all comments from all users could be viewed under a photo, sorted chronologically, with an option to add a comment in a thread
- Must know what it commented in order to go to users profile, can only search based on what you know had been commented

Albums/like:

- Anonymous users cannot create albums, like photos, or comment on other people’s photos