Alex O’Connor

(Updated Schema)

Relational Schema

users(uid, first_name, last_name, email, dob, hometown, gender, password)
pictures(picture_id, imgdata, caption)
albums(album_id, name, uid, date_started)
stored_in(photo_id, album_id)
friends_with(uid, friend_uid)
tags(tag_id, tag)
tagged_picture(picture_id, tag_id)
comments(comment_id, uid, date, words)
commented_on(picture_id, comment_id)
liked_picture(user_id, picture_id)

Assumptions:

Users

- Each user has a valid and unique email address

Friends

- If a user1 is friends with user2, user2 does not have to be friends with user1 (no confirmation)

Albums

- An album can be empty after it is created
- An album can only belong to one user

Photos

- Assume that a tag/s are added to the picture before being uploaded to an album
- Assume that an album is present before uploading a photo
- A photo can only belong to one album

Tags

- Assume that a tag/s are added to a photo before it is uploaded to an album
- Assume that a tag cannot be added in a comment

Comments

- Assume that a comment is added to a photo only after it is uploaded.
- A comment can only be added to one picture

Likes

- A user can like his own picture

SQL Statements

```
CREATE TABLE Users (
   user_id int4  AUTO_INCREMENT NOT NULL,
   email varchar(255) UNIQUE,
   password varchar(255),
   first_name varchar(255),
   last_name varchar(255),
   dob date,
   gender varchar(255),
   hometown varchar(255),
   CONSTRAINT users_pk PRIMARY KEY (user_id)
);

CREATE TABLE Pictures
(
  picture_id int4  AUTO_INCREMENT NOT NULL,
  user_id int4,
  FOREIGN KEY (user_id) REFERENCES Users(user_id),
  imgdata longblob ,
  caption varchar(255),
  INDEX upid_idx (user_id),
  CONSTRAINT pictures_pk PRIMARY KEY (picture_id)
);


CREATE TABLE Albums
(
          album_id int4 AUTO_INCREMENT NOT NULL,
          name varchar(255),
          user_id int4,
          FOREIGN KEY (user_id) REFERENCES Users(user_id),
          date_started date,
          CONSTRAINT albums_pk PRIMARY KEY (album_id)
);

CREATE TABLE Stored_In
(
          picture_id int4,
          FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id),
        album_id int4,
        FOREIGN KEY (album_id) REFERENCES Albums(album_id)
);




CREATE TABLE Friends_with
(       user_id int4,
        FOREIGN KEY (user_id) REFERENCES Users(user_id),
        friend_uid int4,
        FOREIGN KEY (friend_uid) REFERENCES Users(user_id)
);


CREATE TABLE Tags
(
        tag_id int4 AUTO_INCREMENT NOT NULL,
        tag varchar(255),
        CONSTRAINT tag_pk PRIMARY KEY (tag_id)
);

CREATE TABLE Tagged_Picture
(
        picture_id int4 ,
        FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id),
        tag_id int4,
        FOREIGN KEY (tag_id) REFERENCES Tags(tag_id)
);

CREATE TABLE Comments
(
        comment_id int4 AUTO_INCREMENT NOT NULL,
        user_id int4,
        FOREIGN KEY (user_id) REFERENCES Users(user_id),
        date_written date,
        words varchar(255),
        CONSTRAINT comments_pk PRIMARY KEY (comment_id)
);

CREATE TABLE Commented_on
(
        comment_id int4,
        FOREIGN KEY (comment_id) REFERENCES Comments(comment_id),
        picture_id int4,
        FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id)
```

```
);



CREATE TABLE Liked_pictures
(
	user_id int4,
	FOREIGN KEY (user_id) REFERENCES Users(user_id),
	picture_id int4,
	FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id)
);
```