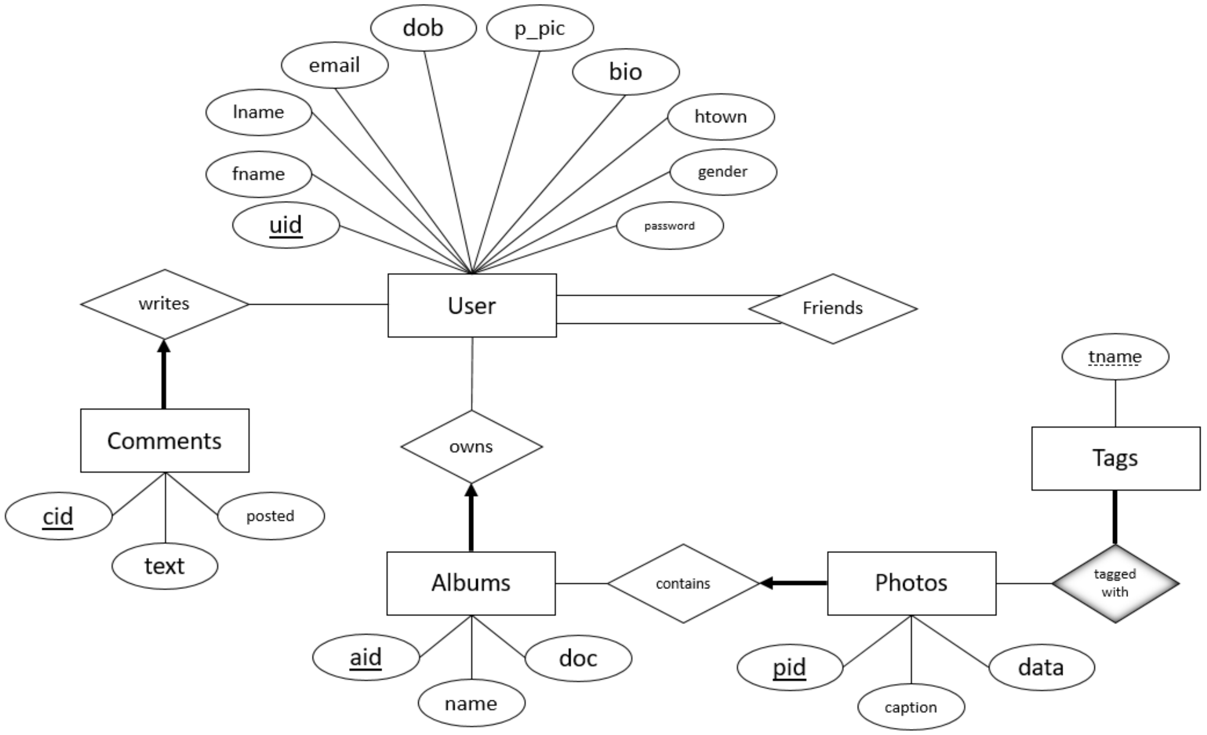
Programming Assignment 1: Photoshare

Jean P. Vazquez

U61571136



I. Data

a. The above ER-Diagram describes the entities that will make up the tables of the database and the relationships between them. However, in order to actually program this website, it is necessary to then convert this to DDL as such:

```
CREATE TABLE users (

        uid int primary key,

        fname char(35) not null,

        lname char(35) not null,

        email char(255) unique not null,

        password char(20) not null

        username char(35),

        dob date not null,
        p_pic longblob,
        bio text,
        hometown char(35),
        gender char(10),
        contribution int
) ;

CREATE TABLE friends (
        uid int primary key,
        fid int not null,
        foreign key (uid) references users (uid) on delete cascade,
        foreign key (fid) references users (uid) on delete cascade
);

CREATE TABLE comments (
        cid int primary key,
        uid int not null,
        pid int not null
        text text,
        posted datetime not null,
        foreign key owner (uid) references users (uid) on delete cascade
);

CREATE TABLE albums (
        aid int primary key,
        uid int not null,
        name char(30) not null,
        doc datetime not null,
        foreign key owner (uid) references users (uid) on delete cascade
);

CREATE TABLE photos (
        pid int primary key,
        aid int not null,
        caption text,
        likes int,
        data longblob not null,
        foreign key album (aid) references albums (aid) on delete cascade
);

CREATE TABLE tags (
        pid int primary key,
        tname char(20) not null,
        foreign key (pid) references photos (pid) on delete cascade
);
```

II. Constraints

a. On delete cascades added to every object besides user in order to ensure that if a user deletes their account, all information related to them is also deleted. This includes being deleted off of all their friend's friends list, all albums are deleted (which in turn causes all pictures to be deleted along with the tables holding the picture tags), and all their comments.

b. For the table containing the tags, pid was made the primary key both because not tags can be unique and it eases retrieval of picture tags later on.

c. Several assumptions were made with respect to not null constraints on attributes since many of these entities wouldn't make much sense to exist without these attributes. For example, all albums should have a name and a user should be able to see when it was created so the creation date should also not be null.

d. Unique constraint added to email in user entity in addition to not null in order to ensure that there exists no other user with the same email.

e. While not listed in the tables as it has to be performed outside, I also plan on adding a trigger to the comments table which checks if the picture a user is commenting on was posted by them, which is why I stored pid as a foreign key in the comment table.

III. Application Description (Functionality)

a. Users

i. In order to register, a user must provide their first and last name, an email address not associated with another user, their date of birth, and a password for their account. All other fields are not required, including the username field which will default to first and last name if left blank. Their username will be the name that will be displayed on a user's personal page.

ii. Once registered, a user will be able to search for other users by entering that other user's username and add them as friends with no restrictions. In other words, when a user selects to add another user as a friend, that user will instantly be added with no verification.

iii. Users will have access to a list containing all their friends via their bio page.

iv. Users will have a value associated with them (name contribution) which reflects how much they interact with the website (through uploads) and with other users (commenting and liking). The 10 users with the highest values will have their most popular pictures or their profiles featured on the front page of the website.

b. Albums and Photos

i. All photos and albums are publicly viewable to both registered and unregistered users of the site.

ii. Once registered, users can post pictures and create albums, the restrictions of which are listed above. Only the creator of an album can edit/delete the album and if a non-empty album is deleted, then all the pictures within will be deleted as well.

iii. Via their bio, registered users are able to search through their published images and albums.

iv. Photos can also be "liked" by registered users which will be displayed alongside the image and a list of all users who have liked the image.

c. Tags

i. Via their bio, users are able to search through their published images via the tags posted on those images.

ii. Tags can be freely added or removed from images provided that the user adding/removing these tags is the one who posted the image.

iii. If a tag is entered into the main search bar, all images containing said tag will be returned to the user. This functionality is available to both registered and non-registered users.

iv. All tags will be implemented as hyperlinks which, when pressed, will redirect users to the same web page they would have received if they had entered that tag into the main search bar

v. There will be a function on the main web page that gives users the option to view the most popular tags which will, after aggregating over the tags table, return the top 10 most used tags within the past week.

d. Comments

i. Both registered and unregistered users can leave comments on any image provided that they did not publish the image themselves (in the case of registered users). If a registered user leaves a comment, this will count towards their contribution score mentioned above.

e. Recommendations

i. If a registered user who is logged in visits the main web page, they will be see a timeline of the most popular pictures which contain any/all of the user's own 5 most used tags. The order in which images are displayed is determined by how many of the user's most used tags it contains, followed by the date when the image was uploaded.

ii. When posting an image, a registered user will be recommended tags for the image based on the tags they themselves have put on them. Using the user's entered tags, a search through all images containing those tags will be run and a list containing different tags most commonly posted alongside those provided by the user.