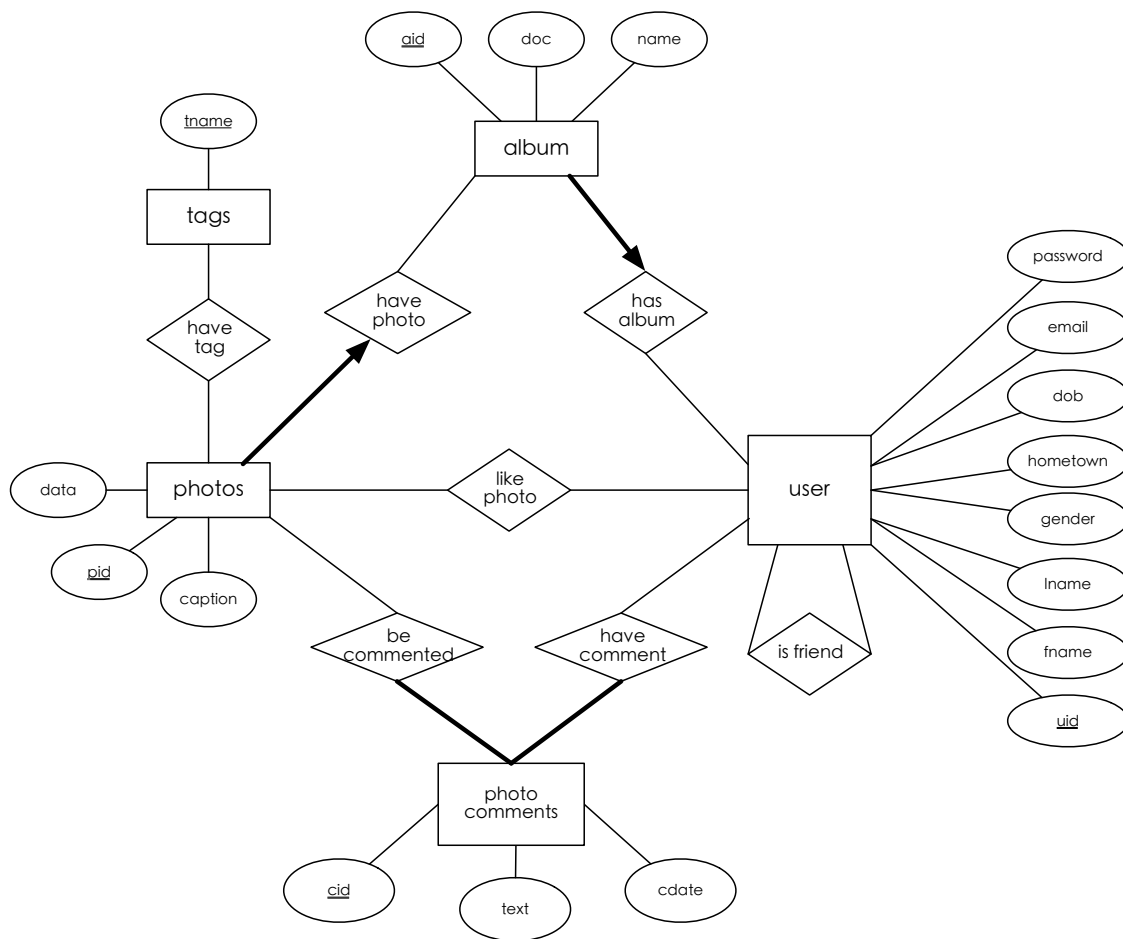


CS 660: Project Assignment #1

Bowen Zhang (U881677028)
Yiwen Gu (U58170649)

1 The ER-Diagram



2 The Schema

```
CREATE DATABASE photoshare;
USE photoshare;

CREATE TABLE Users(
    uid    INTEGER NOT NULL AUTOINCREMENT,
    fname  VARCHAR(20) NOT NULL,
    lname  VARCHAR(20) NOT NULL,
    email  VARCHAR(50) UNIQUE,
    dob    DATE,
    hometown VARCHAR(20) ,
    gender  CHAR(1) ,
    password VARCHAR(20) NOT NULL,
    PRIMARY KEY (uid)
);

CREATE TABLE Albums(
    aid    INTEGER NOT NULL AUTOINCREMENT,
    aname  VARCHAR(20) NOT NULL,
    doc    TIMESTAMP DEFAULT current_timestamp ,
    uid    INTEGER NOT NULL,
    PRIMARY KEY (aid) ,
    FOREIGN KEY (uid) REFERENCES Users(uid)
        ON DELETE CASCADE
);
/* Assumption:
   each album has to belong to one user
   includes album entity and 'own' relationship
*/

CREATE TABLE Photos(
    pid    INTEGER NOT NULL AUTOINCREMENT,
    caption VARCHAR(200) ,
    path   VARCHAR(200) NOT NULL,
    aid    INTEGER NOT NULL,
    PRIMARY KEY (pid) ,
    FOREIGN KEY (aid) REFERENCES Albums(aid)
        ON DELETE CASCADE
);
/* Assumption:
```

each photo has to belong to one album,
 a photo copied from one album to another will have its
 own (different) pid. */

```
CREATE TABLE Tags(
    pid INTEGER NOT NULL,
    tname VARCHAR(50) NOT NULL,
    PRIMARY KEY (pid, tname),
    FOREIGN KEY (pid) REFERENCES Photos(pid) ON DELETE
        CASCADE
);
```

```
CREATE TABLE isFriend(
    uid INTEGER NOT NULL,
    fuid INTEGER NOT NULL,
    PRIMARY KEY (uid, fuid),
    FOREIGN KEY (uid) REFERENCES Users(uid)
        ON DELETE CASCADE,
    FOREIGN KEY (fuid) REFERENCES Users(uid)
        ON DELETE CASCADE,
    CHECK (uid <> fuid)
);
/* Assumption: a user cannot be friend to him/herself;
   (A,B) and (B,A) are different in this table */
```

```
CREATE TABLE Comments(
    cid INTEGER NOT NULL AUTOINCREMENT,
    cdate TIMESTAMP DEFAULT current_timestamp,
    text VARCHAR(255),
    pid INTEGER NOT NULL,
    uid INTEGER NOT NULL,
    PRIMARY KEY (cid),
    FOREIGN KEY (pid) REFERENCES Photos(pid)
        ON DELETE CASCADE,
    FOREIGN KEY (uid) REFERENCES Users(uid)
);
```

```
CREATE TABLE likePhoto(
    uid INTEGER,
    pid INTEGER,
    PRIMARY KEY (uid, pid),
```

```

FOREIGN KEY (uid) REFERENCES Users(uid)
ON DELETE CASCADE
FOREIGN KEY (pid) REFERENCES Photos(pid)
ON DELETE CASCADE
);

CREATE VIEW contribution AS AS
SELECT up,fname,cc,cp,cc+cp FROM Users ,
      (SELECT uid AS up, IFNULL(COUNT(pid),0) AS cp FROM
        (SELECT u.uid,p.pid FROM photos p
          RIGHT JOIN albums a ON a.aid = p.aid
          RIGHT JOIN users u ON u.uid=a.uid) b
      GROUP BY uid) a1,
      (SELECT uid AS uc, IFNULL(cid),0) AS cc FROM
        (SELECT u.uid,c.cid FROM comments c
          RIGHT JOIN users u ON c.uid = u.uid) b
      GROUP BY uid) b1
WHERE uc=up AND uc<>-1 AND uc=uid
ORDER BY cc+cp DESC;
/* for viewing the users' contribution*/

CREATE VIEW tagcount AS
SELECT tname, COUNT(tname) AS ct
FROM Tags
GROUP BY tname
ORDER BY ct DESC;
/*for viewing the photos that tagged by a single tag*/

INSERT INTO Users(uid,fname,lname,password)
VALUES ('-1','guest','guest','password');
/* for unregistered users */

```

3 The Constrains

The entity set **Album** has a key constraint in the relationship set **hasAlbum**.
The entity set **Photos** has a key constraint in the relationship set **havePhoto**.

4 Achieved Function

All the pages:

- 1.Access to view the Popular page.
- 2.Access to go back Home page.

When logged out

- 1.Access to register.
- 2.Access to Login.

When logged in

- 1.Access to Logout.
- 2.Access to go to user's main page.

Home page:

Allowed to search photo by tag/comment or search user by name(first or last). If you are logged in, you are allowed to choose to Search your own photos using the same two methods.

User's main page:

- 1.Same searching functions as the Home page.
- 2.Shows the owner's Albums and you can access by click.
- 3.Shows the owner's Friends and you can access by click.

If you are the Owner

- 1.Access to delete the Album.
- 2.Show the users you may like.(recommending your friends' friends)
- 3.Show the photos you may like.(recommending by tags you own for your photos)

If you are not the Owner

Access to add/cancel the owner as your friend

Album page:

Show the photos in the Album.

If your are the Owner

Access to upload photos to the Album or delete the photos.

Photo page:

1.Show the photo, the owner of the photo, the album the photo belongs to and it's caption.

2.Show how many people like the photo.

3.Show the tags of the photo. When tags are clicked, show the search result of photos with the tag you clicked

4.Show others' Comments and access to leave comment.

If logged in

- 1.Access to like the photo, and cancel your like after a second click.
- 2.Access to delete the comment you leave.

If it's your photo

1. Access to add tags.
2. You are not permitted to leave comments but you can delete the comments of your photo.

Search result page:

Show the search result.

Popular page:

1. Show the active Users with the number of their comments, photos and contributions.
2. Show the most popular tags.