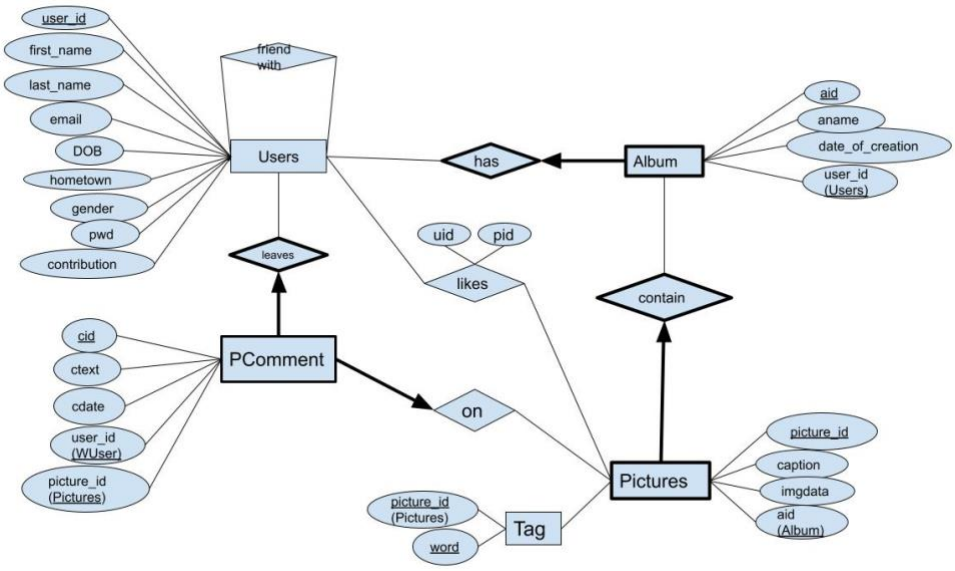
CS460 Project Report

Ruihang Liu, Shuo Han

Final ER Diagram:



Final schema of the database:

```
CREATE DATABASE IF NOT EXISTS photoshare;
USE photoshare;
DROP TABLE IF EXISTS Pictures CASCADE;
DROP TABLE IF EXISTS Users CASCADE;
DROP TABLE IF EXISTS Album CASCADE;
DROP TABLE IF EXISTS PComment CASCADE;
DROP TABLE IF EXISTS friendWith CASCADE;
DROP TABLE IF EXISTS tag CASCADE;
DROP TABLE IF EXISTS likes CASCADE;

CREATE TABLE Users (
    user_id         int4 PRIMARY KEY AUTO_INCREMENT,
    first_name      CHAR(10),
    last_name       CHAR(10),
    email           varchar(255) NOT NULL,
    DOB             DATE,
    hometown        CHAR(10),
    gender          CHAR(10),
    password        varchar(255),
    contribution    INT DEFAULT 0,
    CONSTRAINT sameEmail UNIQUE (email)
);

CREATE TABLE friendWith (
    user1           INT REFERENCES Users(user_id) ON DELETE CASCADE,
    user2           INT REFERENCES Users(user_id) ON DELETE CASCADE
);
```

```sql
CREATE TABLE PComment (
    cid INT4 PRIMARY KEY AUTO_INCREMENT,
    ctext CHAR(200),
    cdate DATE,
    user_id INT4,
    picture_id INT4,
    FOREIGN KEY (user_id)
        REFERENCES Users (user_id)
        ON DELETE CASCADE,
    FOREIGN KEY (picture_id)
        REFERENCES Pictures (picture_id)
        ON DELETE CASCADE
);
```

```sql
CREATE TABLE Pictures (
    picture_id      int4  AUTO_INCREMENT,
    caption         VARCHAR(255),
    imgdata         longblob,
    aid             int4,
    FOREIGN KEY (aid) REFERENCES Album (aid) ON DELETE CASCADE,
    PRIMARY KEY(picture_id)
);
```

```sql
CREATE TABLE Album (
    aid             int4 PRIMARY KEY AUTO_INCREMENT,
    aname           VARCHAR(50) UNIQUE,
    user_id         int4 REFERENCES Users on DELETE CASCADE,
    date_of_creation DATE
);
```

```
CREATE TABLE tag (
    word            VARCHAR(20),
    picture_id      int4 REFERENCES Pictures ON DELETE CASCADE,
    PRIMARY KEY(word)
);

CREATE TABLE likes (
    user_id         INT REFERENCES Users ON DELETE CASCADE,
    picture_id      int4 REFERENCES Pictures ON DELETE CASCADE
);

CREATE TRIGGER after_photo AFTER INSERT ON Pictures
    FOR EACH ROW
        UPDATE Users
        SET contribution = contribution + 1
        WHERE Users.user_id = (SELECT * FROM (SELECT U.user_id
            FROM Users AS U, Album AS A, Pictures AS P
            WHERE P.picture_id = NEW.picture_id AND P.aid = A.aid AND A.user_id = U.user_id) AS X);

CREATE TRIGGER after_comment AFTER INSERT ON PComment
    FOR EACH ROW
        UPDATE Users
        SET contribution = contribution + 1
        WHERE Users.user_id = NEW.user_id;

CREATE TRIGGER after_photo_d BEFORE DELETE ON Pictures
    FOR EACH ROW
        UPDATE Users
        SET contribution = contribution - 1
        WHERE Users.user_id = (SELECT * FROM (SELECT U.user_id
            FROM Users AS U, Album AS A, Pictures AS P
            WHERE P.picture_id = OLD.picture_id AND P.aid = A.aid AND A.user_id = U.user_id) AS X);

CREATE TRIGGER after_album_d BEFORE DELETE ON Album
    FOR EACH ROW
        UPDATE Users
        SET contribution = contribution - (SELECT COUNT(*)
                FROM Pictures AS P
                WHERE P.aid = OLD.aid)
        WHERE Users.user_id = (SELECT * FROM (SELECT U.user_id
            FROM Users AS U, Album AS A
            WHERE OLD.aid = A.aid AND A.user_id = U.user_id) AS X);

CREATE TRIGGER after_comment_d BEFORE DELETE ON PComment
    FOR EACH ROW
        UPDATE Users
        SET contribution = contribution - 1
        WHERE Users.user_id = OLD.user_id;

INSERT INTO Users (email, password) VALUES ('test@bu.edu', 'test');
INSERT INTO Users (email, password) VALUES ('test1@bu.edu', 'test');
```

Brief description of the system:

In this system, we have constructed the database to track the pictures, users, albums, comments, friends, tags, and likes in the photoshare website. They are linked together to get various data for every specific user to provide service for every user.

Users are the key part for the whole website. Once a user is deleted, we will delete all dependent data, user information, friend, comment, album, and likes. Also, when the data from album were deleted, the related data, pictures, comments, and tags, are all deleted accordingly.

Also, we have introduced some triggers, after_photo, after_comment, after_photo_d, and after_comment_d, in our schema to calculate the contribution of every user. When a user adds a photo and comment, the contribution of these users will be increased accordingly. When photo, album and comment get deleted, the contribution of these users will be decreased then.
(PS. We have also added a delete comment function in our website as a bonus)