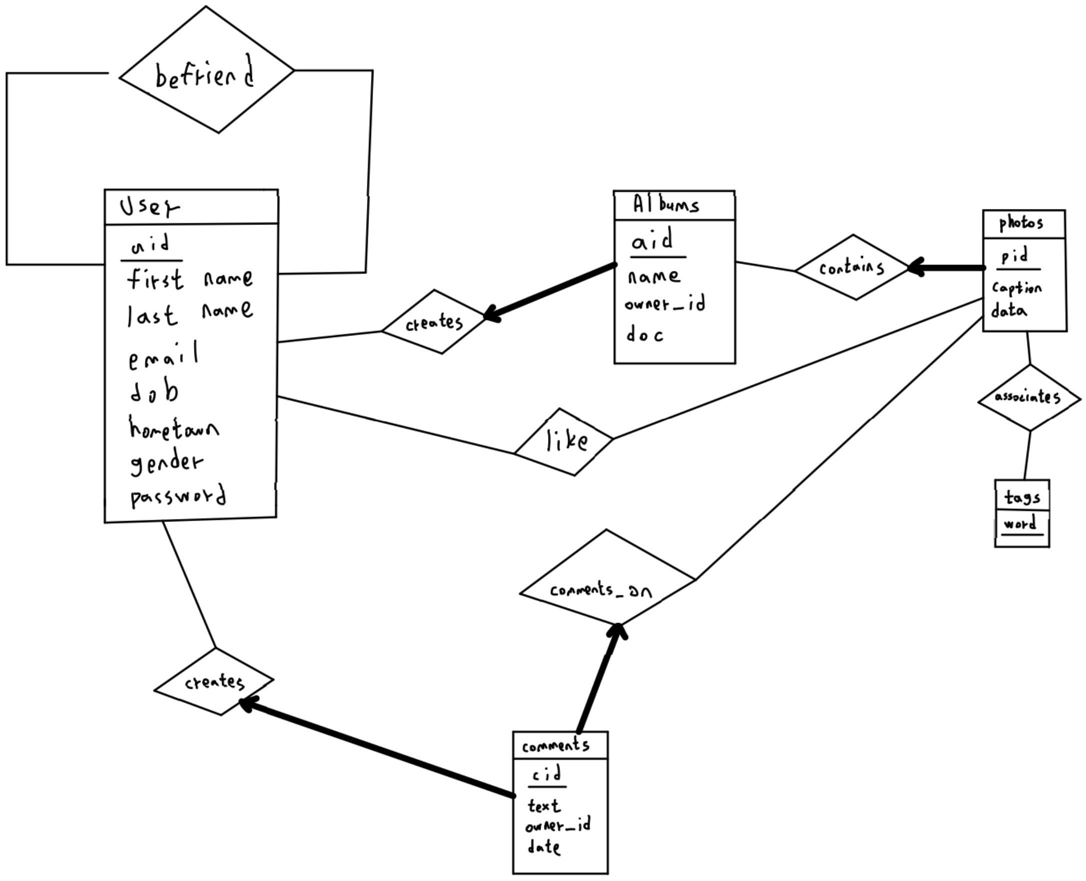

Additional Assumptions:

1. “Friend” functions more like “follow” — it does not need permission for user A to add user B as friend.
2. Each album must have exactly one owner(creator).
3. Each photo must belong to exactly one album (A user has to create an album before that user can upload a photo).
4. Each comment (identified by cid) can only be created by one user, to comment on one photo (but different comments may have the same content).

```
CREATE DATABASE IF NOT exists PA1;
use PA1;

CREATE TABLE Users ( -- capitalized entitys for notations
    uid INT AUTO_INCREMENT,
    first_name VARCHAR(20) NOT NULL, -- user must have a name
    last_name VARCHAR(20) NOT NULL,
    email VARCHAR(30) NOT NULL, -- user need to register with a email
    job VARCHAR(10),
    hometown VARCHAR(20),
    gender VARCHAR(20),
    user_password VARCHAR(20) NOT NULL,
    PRIMARY KEY (uid)
);

CREATE TABLE be_friend(
    uid_from INT,
    uid_to INT,
    PRIMARY KEY (uid_from, uid_to),
    FOREIGN KEY (uid_to) REFERENCES Users(uid),
    FOREIGN KEY (uid_from) REFERENCES Users(uid)
);
 -- ALTER TABLE be_friend ADD INDEX(uid1);
 -- ALTER TABLE be_friend CHANGE uid1 uid1 INT AUTO_INCREMENT;

CREATE TABLE Albums(
    aid INT PRIMARY KEY AUTO_INCREMENT,
    album_name VARCHAR(20),
    owner_id INT,
    date_creation date
);

CREATE TABLE create_album(  -- for each albums, it should be created by
only one user.
    uid INT NOT NULL,  -- total participation constraint that for each
albums, it need has at least one user as owner.
    aid INT PRIMARY KEY,-- aid is unique identify this relationship due to
key constraint that for each album, it should have one user at most
    FOREIGN KEY (aid) references Albums(aid),
    FOREIGN KEY (uid) references Users(uid)
);

CREATE TABLE Photos(
    pid INT PRIMARY KEY AUTO_INCREMENT,
    data_photo LONGBLOB, -- store data in binary
    caption VARCHAR(255)
);

CREATE TABLE Tags(
    word PRIMARY KEY VARCHAR(25)
);

CREATE TABLE associate(
    pid INT,
    word VARCHAR(25),
    PRIMARY KEY (pid, word),
    FOREIGN KEY (pid) REFERENCES Photos(pid),
    FOREIGN KEY (word) REFERENCES Tags(word)
);

CREATE TABLE album_contain_photo(  -- for each photos, it should be
contained in only one album
    aid INT NOT NULL,-- total participation constraint that for each photo
it should be contained at least one album
    pid INT PRIMARY KEY, -- key constraint that each photo can have at
most one album
    FOREIGN KEY (pid) REFERENCES Photos(pid),
    FOREIGN KEY (aid) REFERENCES  Albums(aid)
);

CREATE TABLE user_like_photo(
    uid INT,
    pid INT,
    PRIMARY KEY (uid, pid),
    FOREIGN KEY (uid) REFERENCES Users(uid),
    FOREIGN KEY (pid) REFERENCES Photos(pid)
);

CREATE TABLE Comments(
    cid INT PRIMARY KEY AUTO_INCREMENT,
    content VARCHAR(255),
    owner_id INT NOT NULL,
    date_comment date
);

CREATE TABLE comments_comment_on_photo( -- each comments should related to
only one photo
    pid INT NOT NULL, -- participation constraint
    cid INT PRIMARY KEY, -- KEY constraint so cid is unique to identify
this relationship
    FOREIGN KEY (pid) REFERENCES photos(pid),
    FOREIGN KEY (cid) REFERENCES Comments(cid)
);

CREATE TABLE user_create_comment( -- comments for each should be created
by only one user
    uid INT NOT NULL,
    cid INT PRIMARY KEY, -- key contraints that for each comments, it can
only be created by onw user.
    FOREIGN KEY (uid) REFERENCES Users(uid),
    FOREIGN KEY (cid) REFERENCES Comments(cid)
);
```