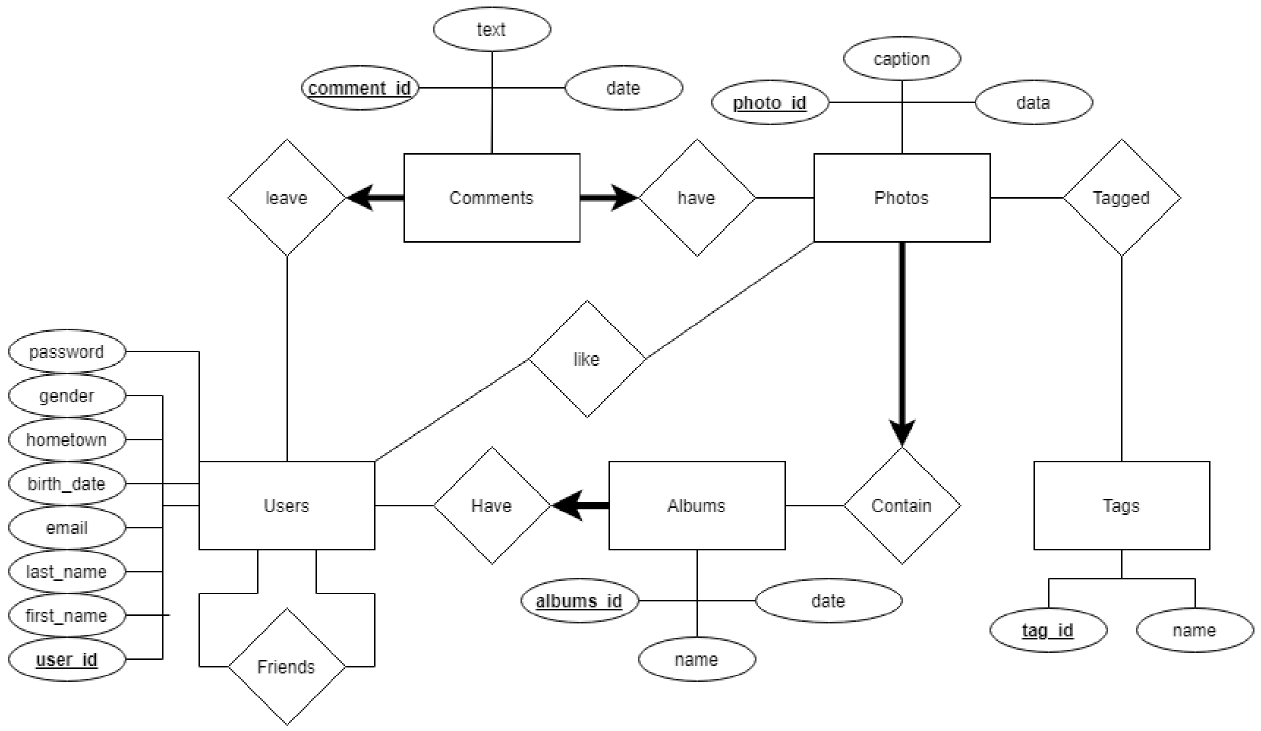
Albert Kulikowski
Piotr Nojszewski

ER Diagram:



SQL Schema:

```
CREATE TABLE Users(
 user_id INTEGER NOT NULL AUTO_INCREMENT,
 first_name VARCHAR(100),
 last_name VARCHAR(100),
 email VARCHAR(100) UNIQUE,
 birth_date DATE,
 hometown VARCHAR(100),
 gender VARCHAR(100),
 password VARCHAR(100) NOT NULL,
 PRIMARY KEY (user_id)
 );

 CREATE TABLE Friends(
 user_id1 INTEGER,
 user_id2 INTEGER,
 PRIMARY KEY (user_id1, user_id2),
 FOREIGN KEY (user_id1)
 REFERENCES Users(user_id)
 ON DELETE CASCADE,
 FOREIGN KEY (user_id2)
 REFERENCES Users(user_id)
 ON DELETE CASCADE,
 CONSTRAINT not_self
 CHECK (user_id1 <> user_id2),
CONSTRAINT Unique_Pair1 UNIQUE (user_id1, user_id2),
CONSTRAINT Unique_Pair2 UNIQUE (user_id2, user_id1)
);

CREATE TABLE Albums(
 albums_id INTEGER NOT NULL AUTO_INCREMENT,
 name VARCHAR(100),
 date DATE,
 user_id INTEGER NOT NULL,
 PRIMARY KEY (albums_id),
 FOREIGN KEY (user_id)
 REFERENCES Users(user_id)
 ON DELETE CASCADE
);

CREATE TABLE Tags(
 tag_id INTEGER NOT NULL AUTO_INCREMENT,
 name VARCHAR(100),
 PRIMARY KEY (tag_id),
 CONSTRAINT check_lowercase
 CHECK (LOWER(name) = name)
);

CREATE TABLE Photos(
 photo_id INTEGER NOT NULL AUTO_INCREMENT,
 caption VARCHAR(100),
 data LONGBLOB,
 albums_id INTEGER NOT NULL,
 user_id INTEGER NOT NULL,
 PRIMARY KEY (photo_id),
 FOREIGN KEY (albums_id)
 REFERENCES Albums (albums_id)
 ON DELETE CASCADE,
 FOREIGN KEY (user_id)
 REFERENCES Users (user_id)
);

CREATE TABLE Tagged(
 photo_id INTEGER,
 tag_id INTEGER,
 PRIMARY KEY (photo_id, tag_id),
 FOREIGN KEY(photo_id)
 REFERENCES Photos (photo_id)
 ON DELETE CASCADE,
 FOREIGN KEY(tag_id)
 REFERENCES Tags (tag_id)
);

CREATE TABLE Comments(
 comment_id INTEGER NOT NULL AUTO_INCREMENT,
 user_id INTEGER NOT NULL,
 photo_id INTEGER NOT NULL,
 text VARCHAR (100) NOT NULL,
 date DATE,
 PRIMARY KEY (comment_id),
 FOREIGN KEY (user_id)
 REFERENCES Users (user_id),
 FOREIGN KEY (photo_id)
 REFERENCES Photos (photo_id)
 ON DELETE CASCADE,
CONSTRAINT not_own_comment
CHECK (user_id <> Photos.user_id)
);

CREATE TABLE Likes(
 photo_id INTEGER,
 user_id INTEGER,
 PRIMARY KEY (photo_id,user_id),
 FOREIGN KEY (photo_id)
 REFERENCES Photos (photo_id)
 ON DELETE CASCADE,
 FOREIGN KEY (user_id)
 REFERENCES Users (user_id)
 ON DELETE CASCADE
);
```

ASSUMPTIONS

1. Every table that is not a many to many relationship will have AUTO_INCREMENT as their primary key. These keys are: user_id, albums_id, tag_id, photo_id, and comment_id.
    a. This will automatically increment the id values of these tables
2. Every child table will have an ON DELETE CASCADE statement. These tables are: Friends, Albums, Photos, Tagged, Comments, and Likes.
    a. This will automatically delete attributes of a child entity if a parent entity is deleted.
3. The email attribute in the Users table needs to be UNIQUE. This should prevent people from creating an account using an email that already exists.
4. The password attribute in the Users table needs to be NOT NULL so that each account has a definite account associated with it.
5. The text attribute in the Comments table needs to be NOT NULL so that people cannot leave a blank comment, or a comment with no text in it.
6. The "not_self" constraint in the Friends table of the SQL schema prevents a user from adding themselves as a friend.
7. The "unique_pair" constraint in the Friends table of the SQL schema keeps each pair of friends unique in the database, so that each instance of a friend is stored only once.
8. The "not_own_comment" constraint in the Comments table of the SQL schema prevents a user from commenting on their own photo.
9. The "check_lowercase" constraint in the Tags table of the SQL schema prevents a user from entering a tag that is not all lowercase letters.
10. We have also assumed that a user must first create an album and then upload images to a specific album.