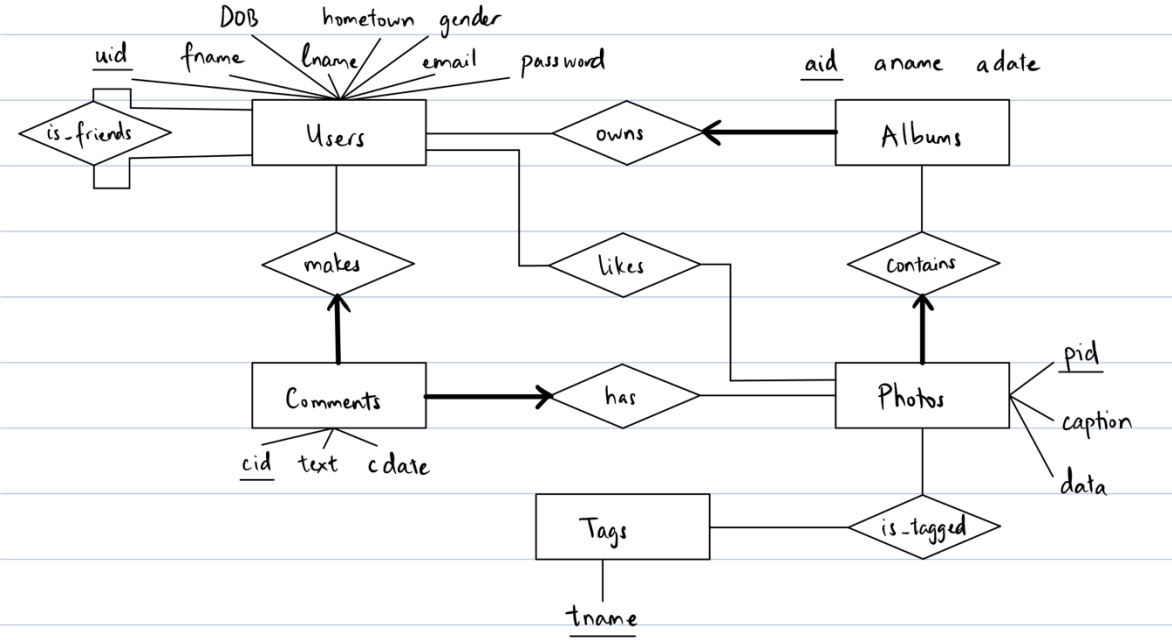
CS460 PA1 – Database Design
Viet Pham: U29541063
Duong Nguyen: U32906640

1.E/R Model:



2. Translation to relational schema

```
CREATE TABLE Users (
        uid int,
        fname varchar(255),
        lname varchar(255),
        dob datetime,
        hometown varchar(255),
        email varchar(255) UNIQUE,
        gender ENUM('Male', 'Female','Other'),
        password varchar(255),
        PRIMARY KEY (uid)
);
```

- This handles the constraint where emails have to be unique, using UNIQUE
- Unregistered users are also given an unique uid, but all other attributes are NULL

```
CREATE TABLE Friendship (
        uid int,
        fid int,
        FOREIGN KEY (uid) REFERENCES Users(uid) ON DELETE CASCADE,
        FOREIGN KEY (fid) REFERENCES Users(uid) ON DELETE CASCADE,
        PRIMARY KEY (uid, fid),
        CHECK (uid != fid)
);
```

- This handles the constraint where users cannot be friends with themselves, using CHECK.
- Also handles friendship deletion when user or friend is removed

```
CREATE TABLE Albums (
        aid int,
        uid int NOT NULL,
        aname varchar(255) NOT NULL,
        adate datetime NOT NULL,
        PRIMARY KEY (aid),
        FOREIGN KEY (uid) REFERENCES Users(uid) ON DELETE CASCADE
);
```

- This handles the constraint where albums must be owned by exactly 1 user, using NOT NULL
- Also handles album deletion when its owner is removed, using ON DELETE CASCADE

```
CREATE TABLE Photos (
        pid int,
        aid int NOT NULL,
        caption varchar(255),
        data varchar(255) NOT NULL,
        PRIMARY KEY (pid),
        FOREIGN KEY (aid) REFERENCES Albums(aid) ON DELETE CASCADE
);
```

- This handles the constraint where photos must belong to exactly 1 album, using NOT NULL
- Also handles photo deletion when its album is removed, using ON DELETE CASCADE

```
CREATE TABLE Tags (
        tname varchar(255),
        pid int,
        PRIMARY KEY (tname, pid),
        FOREIGN KEY (pid) REFERENCES Photos(pid) ON DELETE CASCADE
);
```

- This handles the removal of a tag-photo relationship when a photo is deleted

```
CREATE TABLE Comments (
        cid int,
        content varchar(255) NOT NULL,
        cdate datetime NOT NULL,
        uid int NOT NULL,
        pid int NOT NULL,
        PRIMARY KEY (cid),
        FOREIGN KEY (uid) REFERENCES Users(uid) ON DELETE CASCADE,
        FOREIGN KEY (pid) REFERENCES Photos(pid) ON DELETE CASCADE
);
```

- This handles the constraint where comments must belong to exactly 1 photo and made by exactly 1 user, using NOT NULL
- Also handles comment deletion when its owner or the photo is deleted, using ON DELETE CASCADE
- This does not handle the constraint where users cannot comment on their own photos, which will be handled on the application level using Python

```
CREATE TABLE Likes (
        uid int,
        pid int,
        PRIMARY KEY (uid, pid),
        FOREIGN KEY (uid) REFERENCES Users(uid) ON DELETE CASCADE,
        FOREIGN KEY (pid) REFERENCES Photos(pid) ON DELETE CASCADE
);
```

- This handles the constraint where user can give at most 1 like for each photo
- Also handles like removal when its owner or the photo is deleted, using ON DELETE CASCADE