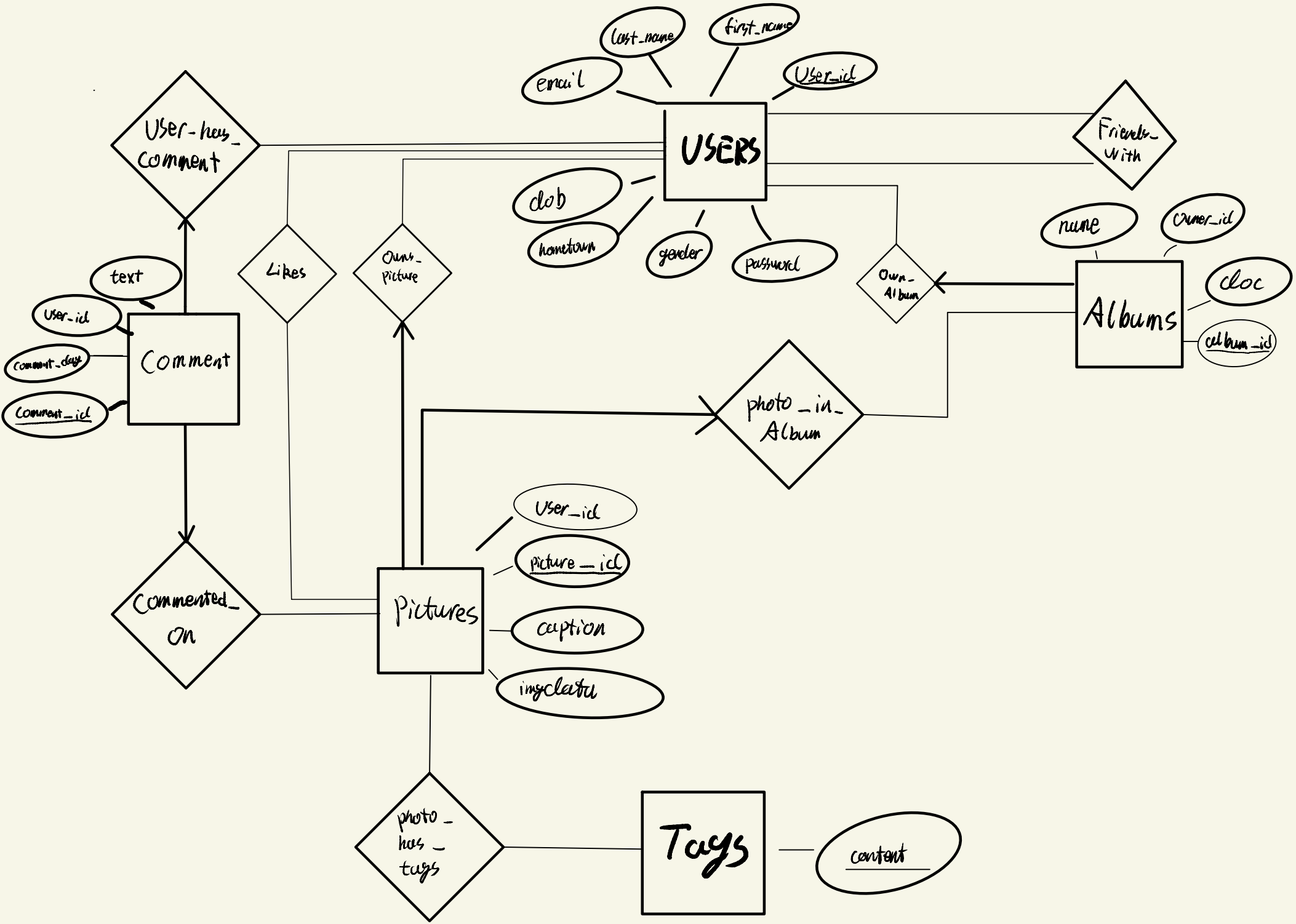
last_name
first_name
email
User_id
USERS
User-has-Comment
Friends-With
dob
hometown
gender
password
Likes
Own-Picture
Own-Album
name
Owner_id
Albums
clac
album_id
text
User_id
Comment_day
Comment_id
Comment
photo_in_Album
User_id
picture_id
caption
imgdata
Commented-On
Pictures
photo-has-tags
Tags
content

Assumptions:

Each photo will and only will in 1 album.

Each comment will and only will be on 1 picture.

User can creat empty Albums.

Each Album will and only will have 1 owner(user)

Ecach tag is unique and can be identified by tag itself

Each piture will and only will have 1 owner(user)

User email must not be null

User password must not be null

Picture imgdata must not be null

```
CREATE DATABASE IF NOT EXISTS photoshare;
USE photoshare;
DROP TABLE IF EXISTS photo_has_tags CASCADE;
DROP TABLE IF EXISTS Commented_on CASCADE;
DROP TABLE IF EXISTS User_has_comment CASCADE;
DROP TABLE IF EXISTS Likes CASCADE;
DROP TABLE IF EXISTS Photo_in_album CASCADE;
DROP TABLE IF EXISTS Owns_album CASCADE;
DROP TABLE IF EXISTS Pictures CASCADE;
DROP TABLE IF EXISTS Albums CASCADE;
DROP TABLE IF EXISTS Friends_with CASCADE;
DROP TABLE IF EXISTS Comment CASCADE;
DROP TABLE IF EXISTS Users CASCADE;
DROP TABLE IF EXISTS Tags CASCADE;
DROP TABLE IF EXISTS Owns_picture CASCADE;


CREATE TABLE Users (
    user_id int4  AUTO_INCREMENT,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    email varchar(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    gender VARCHAR(255),
    homeTown VARCHAR(255),
    dob DATETIME,
  CONSTRAINT users_pk PRIMARY KEY (user_id)
);

CREATE TABLE Friends_with
(
    user1_id int4,
    user2_id int4,
  CONSTRAINT friend_pk PRIMARY KEY (user1_id,user2_id),
  FOREIGN KEY (user1_id) REFERENCES Users(user_id),
  FOREIGN KEY (user2_id) REFERENCES Users(user_id)
);

CREATE TABLE Pictures
(
    picture_id int4  AUTO_INCREMENT,
    user_id int4 NOT NULL,
    imgdata longblob NOT NULL,
    caption VARCHAR(255),
    INDEX upid_idx (user_id),
  CONSTRAINT pictures_pk PRIMARY KEY (picture_id),
  CONSTRAINT pictures_fk FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

CREATE TABLE Owns_picture
(
    picture_id int4 UNIQUE NOT NULL,
    user_id int4,
    CONSTRAINT owns_picture_pk PRIMARY KEY(picture_id,user_id),
    FOREIGN KEY(user_id) REFERENCES Users(user_id),
    FOREIGN KEY(picture_id) REFERENCES Pictures(picture_id)
```

```
);


CREATE TABLE Albums
(
    album_id int4 AUTO_INCREMENT,
    name VARCHAR(255),
    owner_id int4 NOT NULL,
    doc DATETIME,
  CONSTRAINT album_pk PRIMARY KEY (album_id),
  FOREIGN KEY (owner_id) REFERENCES Users(user_id)
);

CREATE TABLE Owns_album
(
    user_id int4,
    album_id int4 UNIQUE NOT NULL,
  CONSTRAINT album_pk PRIMARY KEY (album_id),
  FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

CREATE TABLE Photo_in_album
(
    picture_id int4 UNIQUE NOT NULL,
    album_id int4,
  CONSTRAINT photo_album_pk PRIMARY KEY(picture_id,album_id),
  FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id),
  FOREIGN KEY (album_id) REFERENCES Albums(album_id)
);

CREATE TABLE Likes
(
    user_id int4,
    picture_id int4,
  CONSTRAINT likes_pk PRIMARY KEY (user_id,picture_id),
  FOREIGN KEY (user_id) REFERENCES Users(user_id),
  FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id)
);

CREATE TABLE Comment
(
    comment_id int4  AUTO_INCREMENT,
    comment_day DATETIME,
    text VARCHAR(255),
    user_id int4 NOT NULL,
  CONSTRAINT comments_pk PRIMARY KEY (comment_id),
  FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

CREATE TABLE User_has_comment
(
    comment_id int4 UNIQUE NOT NULL,
    user_id int4,
  CONSTRAINT user_comment_pk PRIMARY KEY(comment_id,user_id),
  FOREIGN KEY (comment_id) REFERENCES Comment(comment_id),
  FOREIGN KEY (user_id) REFERENCES Users(user_id)
```

```
);

CREATE TABLE Commented_on
(
    comment_id int4 UNIQUE NOT NULL,
    picture_id int4,
  CONSTRAINT comment_on_pk PRIMARY KEY(comment_id,picture_id),
  FOREIGN KEY (comment_id) REFERENCES Comment(comment_id),
  FOREIGN KEY (picture_id)REFERENCES Pictures(picture_id)
);

CREATE TABLE Tags
(
   content VARCHAR(255) UNIQUE,
  CONSTRAINT tags_pk PRIMARY KEY (content)
);

CREATE TABLE photo_has_tags
(
    content VARCHAR(255),
    picture_id int4,
  CONSTRAINT photo_has_tags_pk PRIMARY KEY (content,picture_id),
  FOREIGN KEY (content) REFERENCES Tags(content),
  FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id)
);

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