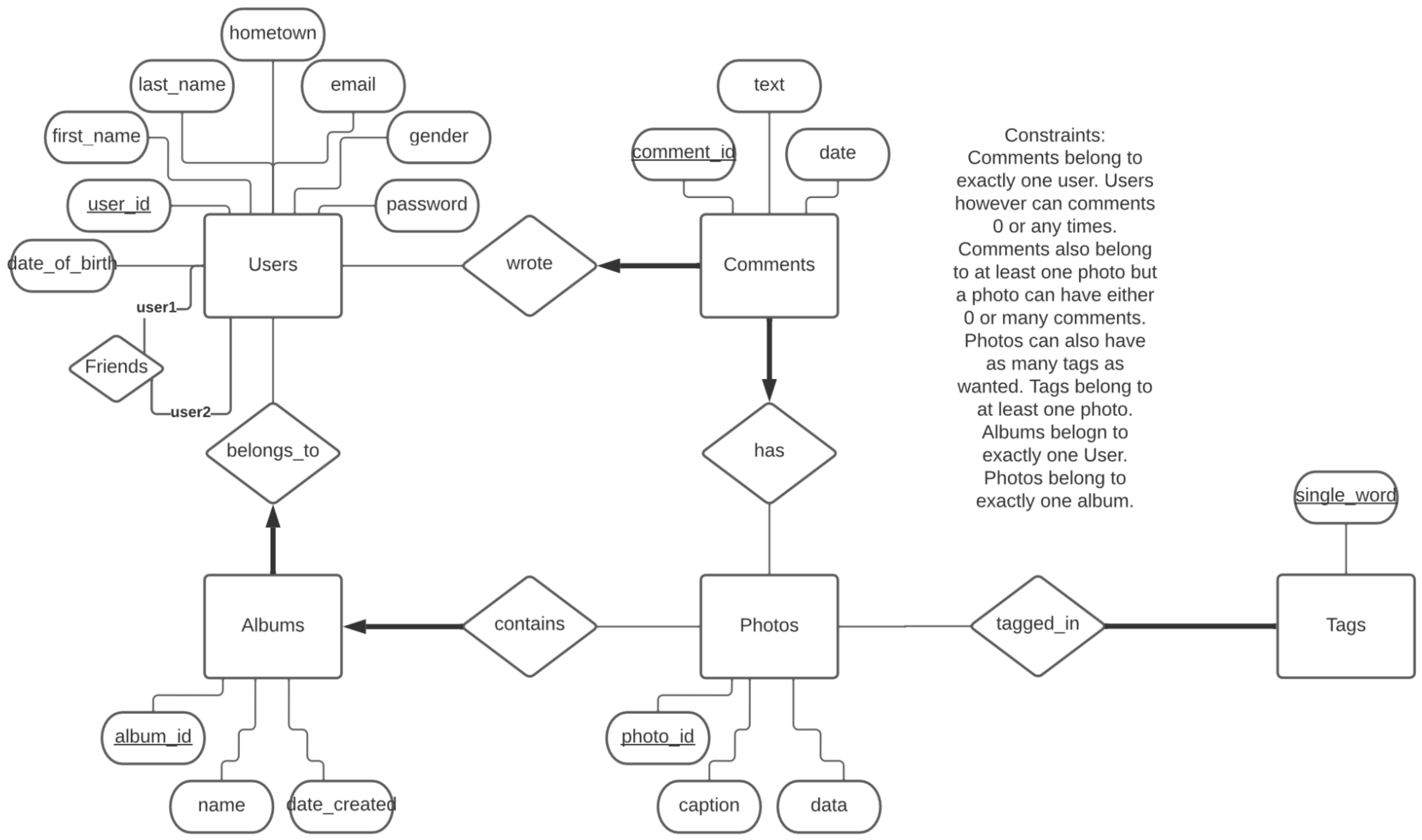

Constraints:
Comments belong to exactly one user. Users however can comments 0 or any times. Comments also belong to at least one photo but a photo can have either 0 or many comments. Photos can also have as many tags as wanted. Tags belong to at least one photo. Albums belogn to exactly one User. Photos belong to exactly one album.

**code below

```
CREATE TABLE Users(
        user_id CHAR(20),
        first_name CHAR(10),
        last_name CHAR(15),
        email CHAR(20),
        date_of_birth DATE,
        hometown CHAR(10),
        gender CHAR(10),
        password CHAR(15),
        PRIMARY KEY(user_id)

CREATE TABLE Albums(
        album_id CHAR(20)
        date_created CHAR(20)
        name CHAR(20)
        PRIMARY KEY(album_id)

CREATE TABLE contains(
        photo_id CHAR (20),
        album_id CHAR (20),
        PRIMARY KEY (photo_id, album_id)
        FOREIGN KEY (photo_id) REFERENCES Photos(photo_id),
        FOREIGN KEY (album_id) REFERENCES Albums(album_id)
                ON DELETE CASCADE);

CREATE TABLE belong_to(
        album_id CHAR (20),
        user_ID CHAR(20),
        PRIMARY KEY (album_id),
        FOREIGN KEY (user_id) REFERENCES Users(user_id),
        FOREIGN KEY (album_id) REFERENCES Albums(album_id));

CREATE TABLE Photos(
        photo_id CHAR(20)
        data BLOB
        caption CHAR(60)
        PRIMARY KEY(photo_id)

CREATE TABLE Tags(
        single_word CHAR(100)
        PRIMARY KEY(single_word));

CREATE TABLE Comments(
        comment_id CHAR(20)
        text CHAR(100)
        date CHAR(20)
        PRIMARY KEY(comment_id)

CREATE TABLE wrote(
        user_id CHAR(20),
        comment_id CHAR(20),
        FOREIGN KEY (user_id) REFERENCES Users(user_id),
        FOREIGN KEY (commented_id) REFERENCES Comments(commented_id));

CREATE TABLE tagged_in(
        single_word CHAR(100),
        photo_id CHAR(20),
        PRIMARY KEY (word, photo_id),
        FOREIGN KEY (word) REFERENCES Tags(word),
        FOREIGN KEY (photo_id) REFERENCES Photos(photo_id)
                ON DELETE SET NULL);

CREATE TABLE has(
        comment_id CHAR(20),
        photo_id CHAR(20),
        PRIMARY KEY (comment_id),
        FOREIGN KEY (photo_id) REFERENCES Photos(photo_id),
        FOREIGN KEY (commented_id) REFERENCES Comments (commented_id)
                ON DELETE SET NULL);

CREATE TABLE friends(
        user1_id CHAR(11),
        user2_id CHAR(11),
        PRIMARY KEY (user1_id, user2_id),
        FOREIGN KEY (user1_id) REFERENCES Users(user_id),
        FOREIGN KEY (user2_id) REFERENCES Users(user_id));
```

RELATIONAL SCHEMA:
Users(user_id: char, first_name: char, last_name: char, password: char, date_of_birth: date, gender: char, hometown: char, email: char)
Albums(album_id: char, name: chat, date: date)
Comments (comment_id: char, date: date, text: char)
Photos(photo_id: char, caption: , data: blob)
Tags(word: char)
friends _with(User1_id: char, User2_id: char)
Owns(user_id: char, album_id: char)
contains(photo_id: char, album_id: char)
wrote(comment_id: char, user_id: char)

has(comment_id: char, photo_id: char)
tagged_in(word: char, photo_id: char)

```
use tangram;

CREATE TABLE Users(
	user_id CHAR(20),
	first_name CHAR(10),
	last_name CHAR(15),
	email CHAR(20),
        date_of_birth DATE,
        hometown CHAR(10),
        gender CHAR(10),
        password CHAR(15),
        PRIMARY KEY(user_id));

CREATE TABLE Tags(
        single_word CHAR(100),
        PRIMARY KEY(single_word),
        FOREIGN KEY(photo_id) REFERENCES Tags(photo_id)
                ON DELETE CASCADE);

CREATE TABLE Photos(
        photo_id CHAR(20),
        data BINARY(20),
        caption CHAR(60),
        PRIMARY KEY(photo_id)
        FOREIGN KEY(album_id) REFERENCES Albums(album_id));

CREATE TABLE Albums(
        album_id CHAR(20),
        date_created CHAR(20),
        name CHAR(20),
        PRIMARY KEY(album_id),
        FOREIGN KEY(photo_id) REFERENCES Photos(photo_id)
                ON DELETE CASCADE);

CREATE TABLE Comments(
        comment_id CHAR(20),
        text CHAR(100),
        date CHAR(20),
        PRIMARY KEY(comment_id),
        FOREIGN KEY(user_id) REFERENCES Users(user_id),
        FOREIGN KEY(photo_id) REFERENCES Photos(photo_id)
                ON DELETE SET NULL);
```