

Final Report

Relational schema

Users(user_id, First_name, Last_name, email, Hometown, Gender, Date_of_birth, password)

Friends(Friends_id, **user_id**, Friends_email)

Albums(Album_id, Album_name, Date_of_creation, **user_id**)

Pictures(picture_id, imgdata, caption, **Album_id**, **user_id**)

Tags(tag_name)

Tags_and_pics(**tag_name**, **picture_id**)

Comments(Comment_id, text, date, **picture_id**, **user_id**)

Likes(**user_id**, **picture_id**)

SQL statements

```
CREATE DATABASE IF NOT EXISTS photoshare;
```

```
USE photoshare;
```

```
DROP TABLE IF EXISTS Comments CASCADE;
DROP TABLE IF EXISTS Likes CASCADE;
DROP TABLE IF EXISTS Tags_and_pics CASCADE;
DROP TABLE IF EXISTS Pictures CASCADE;
DROP TABLE IF EXISTS Tags CASCADE;
DROP TABLE IF EXISTS Albums CASCADE;
DROP TABLE IF EXISTS Friends CASCADE;
DROP TABLE IF EXISTS Users CASCADE;
```

```
CREATE TABLE Users (
    user_id int4 AUTO_INCREMENT,
    First_name VARCHAR(30) NOT NULL,
    Last_name VARCHAR(30) NOT NULL,
    email varchar(255) UNIQUE,
    password varchar(255),
    Hometown VARCHAR(100),
    Gender VARCHAR(10) NOT NULL,
    Date_of_birth DATE NOT NULL,
    constraint chk1 check (Gender = 'F' or Gender = 'M' or Gender='other'),
    CONSTRAINT users_pk PRIMARY KEY (user_id)
);
```

```
CREATE TABLE Friends(  
    Friends_id INTEGER,  
    Friends_email varchar(225) UNIQUE,  
    user_id INTEGER,  
    FOREIGN KEY (user_id) REFERENCES Users(user_id),  
    PRIMARY KEY(user_id, Friends_id)  
    );
```

```
CREATE TABLE Albums(  
    Album_id int4 AUTO_INCREMENT,  
    Album_name VARCHAR(30) NOT NULL,  
    Date_of_creation DATE NOT NULL,  
    user_id INTEGER,  
    FOREIGN KEY (user_id) REFERENCES Users(user_id),  
    CONSTRAINT Album_id PRIMARY KEY (Album_id)  
  
    );
```

```
CREATE TABLE Tags(  
    tag_name VARCHAR(20),  
    CONSTRAINT tags_pk PRIMARY KEY (tag_name)  
    );
```

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

```
CREATE TABLE Tags_and_pics(  
    tag_name VARCHAR(20),  
    picture_id int4 AUTO_INCREMENT,  
    FOREIGN KEY (tag_name) REFERENCES Tags(tag_name),
```

```
FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id)
);
```

```
CREATE TABLE Comments(
  Comment_id int4 AUTO_INCREMENT,
  text VARCHAR(1000) NOT NULL,
  date DATE,
  picture_id int4,
  user_id int4,
  FOREIGN KEY(picture_id) REFERENCES Pictures(picture_id),
  FOREIGN KEY(user_id) REFERENCES Users(user_id),
  CONSTRAINT comment_pk PRIMARY KEY (Comment_id)
);
```

```
CREATE TABLE Likes(
  user_id int4,
  picture_id INTEGER,
  FOREIGN KEY (User_id) REFERENCES Users(User_id),
  FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id)
);
```

```
INSERT INTO Users(email, password, First_name, Last_name, Date_of_birth, Hometown,
Gender) VALUES ("anon@anon",
                  "anon123",
                  "anon", "anon",
                  "1900 - 01 - 01 ",
                  "anon", "F");
```

Assumption:

1. A photo must be contained within an album.
2. Each tag has its own tag_id.
3. An album must be created by a user.
4. A comment must be associated with a user.
5. A comment must be associated with a photo.
6. Gender must be female or male or other.
7. Friend relationship is associated with both user1 and user2.
8. A like must be associated with a user and a photo.
9. Assume we have an Anonymous user which stores all anonymous operations like comments at the beginning by initiating and inserting an entry of the anonymous user to the database User table.
10. When a user adds a friend, the user will be asked to enter their friends' email address and add them to their friend list. The friendship will also be shown on the friend's list.
11. Assume a user creates a tag separately. Then add the tag to a photo during the photo uploading page.
12. User_id, album_id, comment_id and picture_id are all auto-increment.

Integrity Constraint is listed in the SQL statements above.