图示

描述已自动生成Additional Assumptions:

1. “Friend” functions more like “follow” — it does not need permission for user A to add user B as friend.
2. Each album must have exactly one owner(creator).
3. Each photo must belong to exactly one album (A user has to create an album before that user can upload a photo).
4. Each comment (identified by comment\_id) can only be created by one user, to comment on one photo (but different comments may have the same content).
5. User uses email to register, so email must be unique.
6. Users cannot befriend with themselves.
7. Users cannot comment on their own photos.

In the SQL below, some “exactly one” restriction is translated by using attributes. For example, since one album must have exactly one user, we add a NOT NULL attribute named user\_id. Similar for photos and comments.

CREATE DATABASE IF NOT exists PA1;

use PA1;

DROP TABLE IF EXISTS user\_create\_comment CASCADE;

DROP TABLE IF EXISTS user\_like\_Photo CASCADE;

DROP TABLE IF EXISTS be\_friend CASCADE;

DROP TABLE IF EXISTS associate CASCADE;

DROP TABLE IF EXISTS Tags CASCADE;

DROP TABLE IF EXISTS Comments CASCADE;

DROP TABLE IF EXISTS Photos CASCADE;

DROP TABLE IF EXISTS Albums CASCADE;

DROP TABLE IF EXISTS Users CASCADE;

CREATE TABLE Users ( -- capitalized entitys for notations

    user\_id INT4 AUTO\_INCREMENT,

    first\_name VARCHAR(20),

    last\_name VARCHAR(20),

    email VARCHAR(30) UNIQUE,

    job VARCHAR(255),

    hometown VARCHAR(20),

    gender VARCHAR(20),

    password VARCHAR(255),

    CONSTRAINT users\_pk PRIMARY KEY (user\_id)

);

CREATE TABLE be\_friend(

    user\_id\_from INT4,

    user\_id\_to INT4,

    PRIMARY KEY (user\_id\_from, user\_id\_to),

    FOREIGN KEY (user\_id\_to) REFERENCES Users(user\_id) ON DELETE CASCADE,

    FOREIGN KEY (user\_id\_from) REFERENCES Users(user\_id) ON DELETE CASCADE,

    CONSTRAINT diff\_user

        CHECK ((user\_id\_from) <> (user\_id\_to))

);

 -- ALTER TABLE be\_friend ADD INDEX(user\_id1);

 -- ALTER TABLE be\_friend CHANGE user\_id1 user\_id1 INT4 AUTO\_INCREMENT;

CREATE TABLE Albums(

    album\_id INT4 PRIMARY KEY AUTO\_INCREMENT,

    album\_name VARCHAR(255),

    user\_id INT4 NOT NULL,

    date\_created date,

    FOREIGN KEY (user\_id) REFERENCES Users(user\_id) ON DELETE CASCADE

);

CREATE TABLE Photos(

  photo\_id INT4 AUTO\_INCREMENT,

  user\_id INT4 NOT NULL,

  album\_id INT4 NOT NULL,

  imgdata LONGBLOB, -- store data in binary

  caption VARCHAR(255),

  INDEX uphoto\_id\_idx (user\_id),

  CONSTRAINT photos\_pk PRIMARY KEY (photo\_id),

  FOREIGN KEY (user\_id) REFERENCES Users(user\_id) ON DELETE CASCADE,

  FOREIGN KEY (album\_id) REFERENCES Albums(album\_id) ON DELETE CASCADE

);

CREATE TABLE Tags(

    word VARCHAR(25) PRIMARY KEY

);

CREATE TABLE associate(

    photo\_id INT4,

    word VARCHAR(25),

    PRIMARY KEY (photo\_id, word),

    FOREIGN KEY (photo\_id) REFERENCES Photos(photo\_id) ON DELETE CASCADE,

    FOREIGN KEY (word) REFERENCES Tags(word)

);

CREATE TABLE user\_like\_Photo(

    user\_id INT4,

    photo\_id INT4,

    PRIMARY KEY (user\_id, photo\_id),

    FOREIGN KEY (user\_id) REFERENCES Users(user\_id) ON DELETE CASCADE,

    FOREIGN KEY (photo\_id) REFERENCES Photos(photo\_id) ON DELETE CASCADE

);

CREATE TABLE Comments(

    comment\_id INT4 PRIMARY KEY AUTO\_INCREMENT,

    user\_id INT4 NOT NULL,

    photo\_id INT4 NOT NULL,

    content VARCHAR(255),

    date\_comment date,

    FOREIGN KEY (user\_id) REFERENCES Users(user\_id) ON DELETE CASCADE,

    FOREIGN KEY (photo\_id) REFERENCES Photos(photo\_id) ON DELETE CASCADE

);