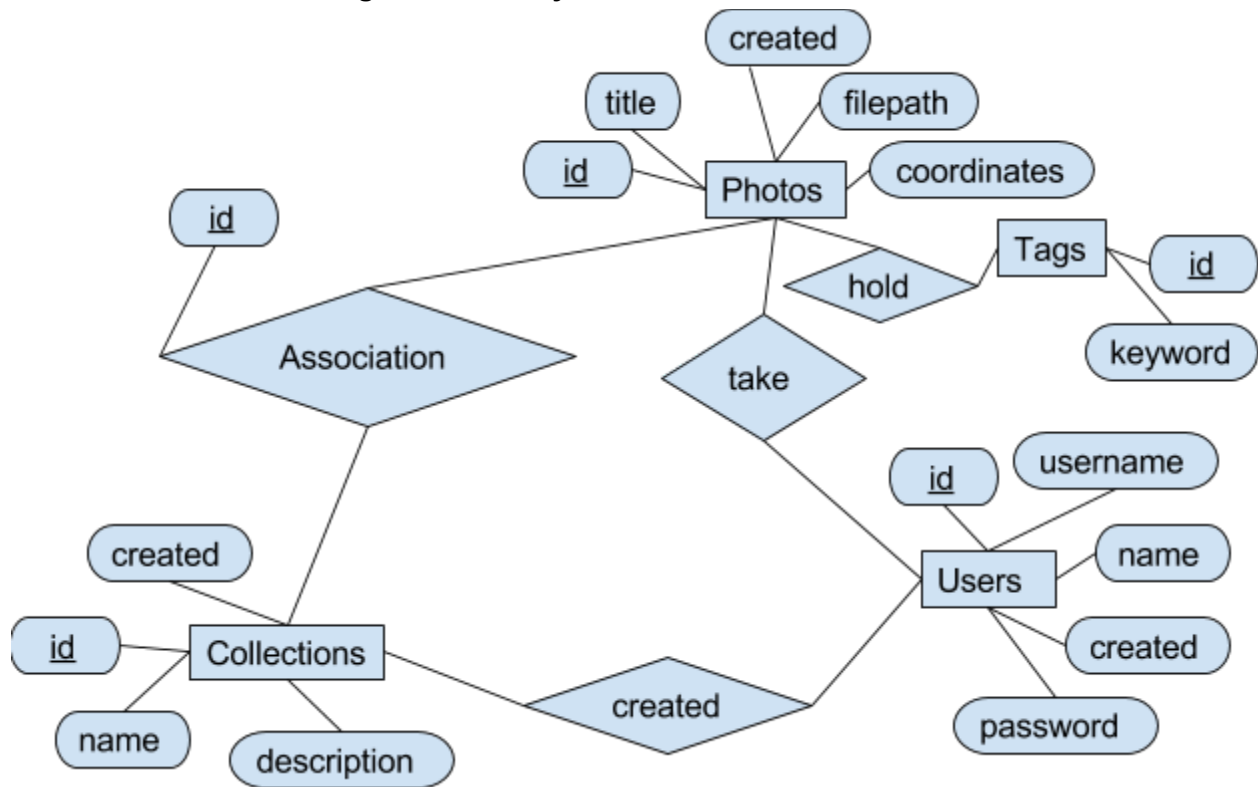**1.   Create the E/R Diagram for the system**.

**2.   Write the SQL statements to CREATE each of the tables in the given schema.
Write them in the order that they will be executed in a script without failing.**

CREATE TABLE Users (
        id INT AUTO_INCREMENT PRIMARY KEY,
        username VARCHAR(20),
        name VARCHAR(20),
        password VARCHAR(16)
        created DATETIME DEFAULT CURRENT_TIMESTAMP,
);
CREATE TABLE Photos (
        id INT AUTO_INCREMENT PRIMARY KEY,
        title VARCHAR(20),
        created DATETIME DEFAULT CURRENT_TIMESTAMP,
        filepath VARCHAR(255),
        coordinates VARCHAR(255),
        use_id int,
        FOREIGN KEY (use_id) REFERENCES Users(id) ON DELETE CASCADE

```
);
CREATE TABLE Tags (
        id INT AUTO_INCREMENT PRIMARY KEY,
        Keyword VARCHAR(20),
        photo_id int,
        FOREIGN KEY (photo_id) REFERENCES Photos(id) ON DELETE CASCADE
);
CREATE TABLE Collections (
        id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(20),
        created DATETIME DEFAULT CURRENT_TIMESTAMP,
        description VARCHAR(255),
        creator_id int,
        FOREIGN KEY (creator_id) REFERENCES Users(id) ON DELETE CASCADE

);
CREATE TABLE Associations (
        id INT AUTO_INCREMENT PRIMARY KEY,
        collection_id int,
        FOREIGN KEY (photo_id) REFERENCES Photos(id) ON DELETE CASCADE,
        FOREIGN KEY (collection_id) REFERENCES Collections(id) ON DELETE CASCADE
);
```

3.   Why do you think we use a filepath to point to the photo instead of putting the entire image into the database?

Because a size of an image can be very large, for example, if the image is the RAW Image Format, it can be at least 20M. That is to say, if store many of these images into SQL database, the database will be occupied as a large space only use to store images. However, if we use a filepath, it can save many spaces for storing filepath, and we can store many images into database without large spaces.

4.   Your photo sharing application is humming along swimmingly and your client has hired you on full-time. Your first task as the new "Database Administrator" is to create a user account for yourself called ps_admin. This is so that you aren't always using the root user.
Create a user called ps_admin and give them SELECT, INSERT, and UPDATE permissions on the tables: 'associations', 'collections', 'photos', 'tags', and 'users' in the system.
ALSO, give ps_admin the ability to grant these permissions to other users.
Write the SQL commands to create the user, set their password, and grant the desired permissions.

```
INSERT INTO users(username, name, password) VALUES ('ps_admin', 'RUI', 123456);
```

GRANT SELECT,INSERT,UPDATE ON Associations TO ps_admin WITH GRANT OPTION;
GRANT SELECT,INSERT,UPDATE ON Collections TO ps_admin WITH GRANT OPTION;
GRANT SELECT,INSERT,UPDATE ON Photos TO ps_admin WITH GRANT OPTION;
GRANT SELECT,INSERT,UPDATE ON Tags TO ps_admin WITH GRANT OPTION;
GRANT SELECT,INSERT,UPDATE ON Users TO ps_admin WITH GRANT OPTION;

5.   Now that you have an admin user for your DBMS, you need to create a user for the system to use so that is it not always using the root user (this is dangerous). This user will be called ps_system. You specifically never want anything deleted from your system so you decide to give ps_system only the ability to SELECT, INSERT, and UPDATE the tables: associations, collections, photos, tags, and user only. Make sure that ps_system cannot grant others these privileges.
Create the user ps_system, set their password, and give ps_system the SELECT, INSERT, and UPDATE privileges.

        INSERT INTO users(username, name, password) VALUES ('ps_system', 'JING', 12345);
        GRANT SELECT,INSERT,UPDATE ON Associations TO ps_system;
        GRANT SELECT,INSERT,UPDATE ON Collections TO ps_system;
        GRANT SELECT,INSERT,UPDATE ON Photos TO ps_system;
        GRANT SELECT,INSERT,UPDATE ON Tags TO ps_system;
        GRANT SELECT,INSERT,UPDATE ON Users TO ps_system;

6.   Let's say that each table in your photo-system database is a file saved to disk. Assuming that your system has no caches or intermediate storage, which of the following queries would execute faster and why?
Query 1: "Get all of the titles of the photos in the collection named 'photography'."
Query 2: "Get all of the titles of the photos in the collection with id 1"

        The Query 2 will execute faster. Because the file in the collection only named integer, which is a primary key. Thus, the name may be used repeatedly but the id is different.

7.   Read the following Stack Overflow question and top answer:
http://stackoverflow.com/questions/1108/how-does-database-indexing-work

        Answer the following questions:

        (a) In your own words, what is indexing (do NOT just copy and paste from the post).
        Indexing is a proach of sorting many records on diverse fields.

(b) What is one benefit and drawback of of indexing?

Benefit: The index structure is easy to use, traverse, and sorted. For example, we can use binary search to locate content we need quickly.

Drawback: The index structure needs more space stored on the disk.

(c) Given table users, write the SQL to create an index called unames on the username column.

CREATE INDEX unames ON Users (username);