



primary key: uid, pid, aid, word, cid.

weak entities: photos.

```
CREATE TABLE USERS(  
    uid INT NOT NULL,  
    PRIMARY KEY(uid));
```

```
CREATE TABLE Registered_Users(  
    uid INT NOT NULL,  
    first_name VARCHAR(255) NOT NULL,  
    last_name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    DOB DATE NOT NULL,  
    hometown VARCHAR(255),  
    gender CHAR(20),  
    pswd VARCHAR(255) NOT NULL,  
    contribution INT,  
    PRIMARY KEY (uid,email),  
    FOREIGN KEY (uid) REFERENCES USERS(uid),  
    CHECK (( SELECT COUNT(email)  
            FROM Registered_Users  
            GROUP BY uid) < 2));
```

```
CREATE TABLE Anonymous_Users (  
    uid INT NOT NULL,  
    PRIMARY KEY(uid),  
    FOREIGN KEY(uid) REFERENCES USERS(uid));
```

```
CREATE TABLE Friends (  
    f_name VARCHAR(255) NOT NULL,  
    uid INT NOT NULL,  
    PRIMARY KEY (uid,f_name));
```

```
CREATE TABLE Add_f (  
    f_name VARCHAR(255) NOT NULL,  
    uid INT NOT NULL,  
    PRIMARY KEY (uid,f_name),  
    FOREIGN KEY (uid) REFERENCES Registered_Users(uid),  
    FOREIGN KEY (f_name) REFERENCES Friends(f_name));
```

```
CREATE TABLE Albums (  
    aid INT NOT NULL,  
    a_name VARCHAR(255),  
    uid INT NOT NULL,  
    creation_date DATE,  
    PRIMARY KEY (aid));
```

```
CREATE TABLE create_album(  
    aid INT NOT NULL,  
    uid INT NOT NULL,  
    PRIMARY KEY (aid,uid),  
    FOREIGN KEY (aid) REFERENCES Albums(aid),  
    FOREIGN KEY (uid) REFERENCES Registered_Users(uid));
```

```
CREATE TABLE Photos(  
    pid INT NOT NULL,  
    caption VARCHAR(255),  
    p_data BINARY,  
    PRIMARY KEY (pid));
```

```
CREATE TABLE belong_to_album (  
    pid INT,  
    caption VARCHAR(255),  
    p_data BINARY,  
    aid INT NOT NULL,  
    PRIMARY KEY (aid, pid),  
    FOREIGN KEY (aid) REFERENCES Albums(aid)  
    ON DELETE CASCADE);
```

```
CREATE TABLE Tags(  
    word VARCHAR(255),  
    PRIMARY KEY (word));
```

```
CREATE TABLE Being_Tagged (  
    word VARCHAR(255),  
    pid INT,  
    PRIMARY KEY (pid,word),  
    FOREIGN KEY (pid) REFERENCES Photos(pid),  
    FOREIGN KEY (word) REFERENCES Tags(word));
```

```
CREATE TABLE Comments(  
    cid INT NOT NULL,  
    C_text VARCHAR(255),  
    User_id INT,  
    Date_created DATE,  
    PRIMARY KEY (cid));
```

```
CREATE TABLE has(  
    uid INT,  
    cid INT,  
    PRIMARY KEY (uid, cid),
```

```
FOREIGN KEY (uid) REFERENCES Users(uid),
FOREIGN KEY (cid) REFERENCES Comments(cid));
```

```
CREATE TRIGGER contribution_Calculation
AFTER INSERT
ON Comments
FOR EACH ROW
SET Registered_Users.Contribution = 0 AND Registered_Users.Contribution =
Registered_Users.Contribution +
(SELECT COUNT(cid)
FROM Comments
INNER JOIN Registered_Users
ON Comments.User_id =
Registered_Users.uid
GROUP BY (uid));
```

```
CREATE TRIGGER contribution_Calculation
AFTER DELETE
ON Comments
FOR EACH ROW
SET Registered_Users.Contribution = Registered_Users.Contribution -
(SELECT COUNT(cid)
FROM Comments
INNER JOIN Registered_Users
ON Comments.User_id =
Registered_Users.uid
GROUP BY (uid));
```

```
CREATE TRIGGER contribution_Calculation
AFTER INSERT
ON Photos
FOR EACH ROW
SET Registered_Users.Contribution = Registered_Users.Contribution +
(SELECT COUNT(pid)
FROM ((Photos
INNER JOIN belong_to_album
ON Photos.pid = belong_to_album.pid)
INNER JOIN Albums
ON belong_to_album.aid = Albums.aid)
INNER JOIN Registered_Users
ON Registered_Users.uid = Albums.uid
GROUP BY (uid));
```

```
CREATE TRIGGER contribution_Calculation
```

```
AFTER DELETE
ON Photos
FOR EACH ROW
SET Registered_Users.Contribution = Registered_Users.Contribution -
    (SELECT COUNT(pid)
     FROM ((Photos
           INNER JOIN belong_to_album
             ON Photos.pid = belong_to_album.pid)
          INNER JOIN Albums
            ON belong_to_album.aid = Albums.aid)
          INNER JOIN Registered_Users
            ON Registered_Users.uid = Albums.uid
     GROUP BY (uid));
```