

```
DROP DATABASE photoshare;
CREATE DATABASE IF NOT EXISTS photoshare;
USE photoshare;
DROP TABLE IF EXISTS Pictures CASCADE;
DROP TABLE IF EXISTS Users CASCADE;
```

```
CREATE TABLE Users (
    user_id int4  AUTO_INCREMENT,
    email varchar(255) UNIQUE,
    password varchar(255),
    fName varchar(255),
    lName varchar(255),
    DOB DATE,
    hometown varchar(255),
    gender varchar(255),
    CONSTRAINT users_pk PRIMARY KEY (user_id)
);
```

```
CREATE TABLE Albums(
    albumID          INTEGER AUTO_INCREMENT,
    albumName        CHAR(20) NOT NULL,
    albumDate        DATE,
    albumOwnedBy     int4,
    PRIMARY KEY (albumID),
    FOREIGN KEY (albumOwnedBy)
REFERENCES Users (user_id)
);
```

```
CREATE TABLE Pictures
(
    picture_id int4  AUTO_INCREMENT,
    user_id int4,
    imgdata longblob ,
    caption VARCHAR(255),
    belongs INTEGER NOT NULL,
    INDEX upid_idx (user_id),
    CONSTRAINT pictures_pk PRIMARY KEY (picture_id)
);
```

```
CREATE TABLE FriendsWith (
    userID1          INTEGER,
```

```
        userID2      INTEGER,  
        PRIMARY KEY (userID1, userID2),  
        FOREIGN KEY (userID1) REFERENCES Users (user_id),  
        FOREIGN KEY (userID2) REFERENCES Users (user_id),  
        CONSTRAINT selfFriend CHECK (userID1 != userID2)  
);
```

```
CREATE TABLE Likes(  
    userID      INTEGER,  
    photoID     INTEGER,  
    PRIMARY KEY (userID, photoID),  
    FOREIGN KEY (userID)  
REFERENCES Users (user_id),  
    FOREIGN KEY (photoID)  
REFERENCES Pictures (picture_id)  
);
```

```
CREATE TABLE Tags(  
    tagDescription CHAR(20),  
    PRIMARY KEY (tagDescription)  
);
```

```
CREATE TABLE taggedWith(  
    photoID INTEGER,  
    tagDescription CHAR(20),  
    PRIMARY KEY (photoID, tagDescription),  
    FOREIGN KEY (photoID)  
        REFERENCES Pictures(picture_id),  
    FOREIGN KEY (tagDescription)  
        REFERENCES Tags(tagDescription)  
);
```

```
CREATE TABLE Comments(  
    commentID      INTEGER AUTO_INCREMENT,  
    commentText     CHAR(100) NOT NULL,  
    commentDate     DATE,  
    commentOwnedBy  INTEGER,  
    commentedUnder  INTEGER NOT NULL,  
    PRIMARY KEY (commentID),  
    FOREIGN KEY (commentOwnedBy)  
REFERENCES Users (user_id),
```

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

```
INSERT INTO Users (email, password) VALUES ('test@bu.edu', 'test');
INSERT INTO Users (email, password) VALUES ('test1@bu.edu', 'test');
```