```
-- Project Part 2
-- Theodore Kim
-- JinZhao Su
-- PART A: Table Creation Queries
CREATE TABLE Person (
     email varchar(80) PRIMARY KEY,
     password char(64) NOT NULL,
     first name varchar(50),
     last name varchar(50)
);
CREATE TABLE FriendGroup (
     owner email varchar(80),
     fg name varchar(80),
     description varchar(250),
     PRIMARY KEY (owner_email, fg_name),
     FOREIGN KEY (owner_email) REFERENCES Person(email)
);
CREATE TABLE Belongs (
     member email varchar(80),
     fg name varchar(80),
     owner email varchar(80),
     PRIMARY KEY (owner_email, fg_name, member_email),
     FOREIGN KEY (owner_email, fg_name)
           REFERENCES FriendGroup(owner email, fg name),
     FOREIGN KEY (member_email) REFERENCES Person(email)
);
CREATE TABLE ContentItem (
     item id int AUTO INCREMENT PRIMARY KEY,
     post_time timestamp default CURRENT_TIMESTAMP,
     file path varchar(200) NOT NULL,
     item_name varchar(80),
     is pub boolean default false
);
CREATE TABLE Tag (
     tagger email varchar(80),
     taggee_email varchar(80),
     item_id int,
     tag_time timestamp default CURRENT_TIMESTAMP,
     status int,
```

```
PRIMARY KEY (tagger email, taggee email, item id),
      FOREIGN KEY (item_id) REFERENCES ContentItem(item_id),
      FOREIGN KEY (tagger email, taggee email)
            REFERENCES Person(email, email)
);
CREATE TABLE Rate (
      item id int,
      rater_email varchar(80),
      rate time timestamp default CURRENT TIMESTAMP,
      emoji char(1),
      PRIMARY KEY (item id, rater email),
      FOREIGN KEY (item_id) REFERENCES ContentItem(item_id),
      FOREIGN KEY (rater email) REFERENCES Person(email)
);
CREATE TABLE Posted (
      item id int,
      poster email varchar(80),
      PRIMARY KEY (item_id, poster_email),
      FOREIGN KEY (item_id) REFERENCES ContentItem(item_id),
      FOREIGN KEY (poster_email) REFERENCES Person(email)
);
CREATE TABLE Shares (
      item_id int,
      owner_email varchar(80),
      fg_name varchar(80),
      PRIMARY KEY (item id, owner email, fg name),
      FOREIGN KEY (item_id) REFERENCES ContentItem(item_id),
      FOREIGN KEY (owner_email, fg_name)
            REFERENCES FriendGroup(owner email, fg name)
);
```

```
-- PART B: Insert Oueries
INSERT INTO Person VALUES ("AA@nyu.edu", SHA2("AA", 256), "Ann", "Anderson"),
                          ("BB@nyu.edu", SHA2("BB", 256), "Bob", "Baker"),
                          ("CC@nyu.edu", SHA2("CC", 256), "Cathy", "Cheng"),
                          ("DD@nyu.edu", SHA2("DD", 256), "David", "Davidson"),
                          ("EE@nyu.edu", SHA2("EE", 256), "Ellen", "Ellenberg"),
                           ("FF@nyu.edu", SHA2("FF", 256), "Fred", "Fox"),
                          ("GG@nyu.edu", SHA2("GG", 256), "Gina", "Gupta"),
                          ("HH@nyu.edu", SHA2("HH", 256), "Hellen", "Harper")
INSERT INTO FriendGroup VALUES ("AA@nyu.edu", "family", NULL),
                                ("BB@nyu.edu", "family", NULL),
                                ("AA@nyu.edu", "roommates", NULL)
("DD@nyu.edu", "family", "AA@nyu.edu"),
                            ("EE@nyu.edu", "family", "AA@nyu.edu"), ("BB@nyu.edu", "family", "BB@nyu.edu"),
                            ("FF@nyu.edu", "family", "BB@nyu.edu"), ("EE@nyu.edu", "family", "BB@nyu.edu"),
                            ("AA@nyu.edu", "roommates", "AA@nyu.edu"),
                            ("GG@nyu.edu", "roommates", "AA@nyu.edu"),
                            ("HH@nyu.edu", "roommates", "AA@nyu.edu")
INSERT INTO ContentItem (file_path, item_name)
VALUES ("/pic/of/cat.jpg", "Whiskers"),
       ("/pic/of/food.jpg", "leftovers in fridge"),
       ("/pic/of/rover.jpg", "Rover");
INSERT INTO Posted VALUES (1, "AA@nyu.edu"),
                          (2, "AA@nyu.edu"),
                          (3, "BB@nyu.edu");
INSERT INTO Shares VALUE (1, "AA@nyu.edu", "family"),
                         (2, "AA@nyu.edu", "roommates"),
                         (3, "BB@nyu.edu", "family")
```

-- PART C: Query to get the ID and name of each content item that is shared with David

SELECT DISTINCT c.item_id, c.item_name FROM ContentItem AS c
NATURAL JOIN Shares
NATURAL JOIN FriendGroup
NATURAL JOIN Belongs AS b
WHERE b.member_email = "DD@nyu.edu" OR c.is_pub = true