```
1:
2: -- Project Part 2
3: -- Group Members: Dov Salomon, Danny Lin, Shlomo Oved
4:
5: -- Part A
6:
7: -- Tables:
8: -- person(uname, password, fname, lname)
9:           -----
10: -- content(cid, uname, date, file_path, name, is_pub)
11:            ---
12: -- comment(uname, cid, timestamp, text)
13:            -----  ---  ---------
14: -- tag(tagger, taggee, cid, timestamp, status)
15:        ------  ------  ---
16: -- friendgroup(owner, gname, description)
17:                 -----  -----
18: -- member(owner, gname, member)
19:           -----  -----  ------
20: -- share(cid, owner, gname)
21:          ---  -----  -----
22:
23: create table if not exists person (
24:         uname varchar(64) not null,
25:         password varchar(255) not null,
26:         fname varchar(60),
27:         lname varchar(60),
28:         primary key(uname)
29: );
30:
31: create table if not exists content (
32:         cid int(10) unsigned not null auto_increment,
33:         uname varchar(64) not null,
34:         date timestamp default current_timestamp,
35:         file_path varchar(255) default null,
36:         name varchar(255) not null,
37:         is_pub boolean default false,
38:         primary key(cid),
39:         foreign key(uname) references person(uname)
40: );
41:
42: create table if not exists comment (
43:         uname varchar(64) not null,
44:         cid int(10) unsigned not null,
45:         timestamp timestamp not null default current_timestamp,
46:         text longtext default null,
47:         primary key(uname, cid, timestamp),
48:         foreign key(uname) references person(uname),
49:         foreign key(cid) references content(cid)
50: );
51:
52: create table if not exists tag (
53:         tagger varchar(64) not null,
54:         taggee varchar(64) not null,
55:         cid int(10) unsigned not null,
56:         timestamp timestamp default current_timestamp,
57:         status varchar(64),
58:         primary key(tagger, taggee, cid),
59:         foreign key(tagger) references person(uname),
60:         foreign key(tagger) references person(uname),
61:         foreign key(cid) references content(cid)
62: );
63:
64: create table if not exists friendgroup (
65:         owner varchar(64) not null,
66:         gname varchar(64) not null,
67:         description longtext default null,
```

```
 68:         primary key(owner, gname),
 69:         foreign key(owner) references person(uname)
 70: );
 71:
 72: create table if not exists member (
 73:         owner varchar(64) not null,
 74:         gname varchar(64) not null,
 75:         member varchar(64) not null,
 76:         primary key(owner, gname, member),
 77:         foreign key(owner, gname) references friendgroup(owner, gname),
 78:         foreign key(member) references person(uname)
 79:         -- no need for FK on owner, its already contrained by friendgroup FK
 80: );
 81:
 82: create table if not exists share (
 83:         cid int(10) unsigned not null,
 84:         owner varchar(64) not null,
 85:         gname varchar(64) not null,
 86:         primary key(cid, owner, gname),
 87:         foreign key(cid) references content(cid),
 88:         foreign key(owner, gname) references friendgroup(owner, gname)
 89: );
 90:
 91: --Part B
 92:
 93: insert into person(uname, password, fname, lname) values
 94: ('AA', md5('AA'), 'Ann', 'Anderson'),
 95: ('BB', md5('BB'), 'Bob', 'Baker'),
 96: ('CC', md5('CC'), 'Cathy', 'Chang'),
 97: ('DD', md5('DD'), 'David', 'Davidson'),
 98: ('EE', md5('EE'), 'Ellen', 'Ellenberg'),
 99: ('FF', md5('FF'), 'Fred', 'Fox'),
100: ('GG', md5('GG'), 'Gina', 'Gupta'),
101: ('HH', md5('HH'), 'Helen', 'Harper');
102:
103: insert  into friendgroup(owner, gname) values
104: ('AA', 'family'),
105: ('BB', 'family'),
106: ('AA', 'besties');
107:
108: insert into member(owner, gname, member) values
109: ('AA', 'family', 'AA'),
110: ('AA', 'family', 'CC'),
111: ('AA', 'family', 'DD'),
112: ('AA', 'family', 'EE'),
113: ('BB', 'family', 'BB'),
114: ('BB', 'family', 'FF'),
115: ('BB', 'family', 'EE'),
116: ('AA', 'besties', 'AA'),
117: ('AA', 'besties', 'GG'),
118: ('AA', 'besties', 'HH');
119:
120: insert into content(cid, uname, name, is_pub) values
121: (1, 'AA', 'Whiskers', false),
122: (2, 'AA', 'My birthday party', false),
123: (3, 'BB', 'Rover', false);
124:
125: insert into share(cid, owner, gname) values
126: (1, 'AA', 'family'),
127: (2, 'AA', 'besties'),
128: (3, 'BB', 'family');
129:
130: -- Part C
131:
132: -- Query to find all content shared with David
133:
134: select distinct name
```

```
135: from content natural join share natural join member
136: where member = 'DD';
137:
138: -- distinct is required, since David may be in more than one group
139: -- that a content item is shared with.
```