

```

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

```

```

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

```

```
135: ('AA', 2),
136: ('BB', 3);
137:
138: insert into share(cid, owner, gname) values
139: (1, 'AA', 'family'),
140: (2, 'AA', 'besties'),
141: (3, 'BB', 'family');
142:
143: -- Part C
144:
145: -- Query to find all content shared with David
146:
147: select distinct name
148: from content natural join share natural join member
149: where member = 'DD';
150:
151: -- distinct is required, since David may be in more than one group
152: -- that a content item is shared with.
153:
154: -- To find all that content that David has permission to view:
155:
156: select distinct name
157: from content natural join share natural join member
158: where member = 'DD' or owner = 'DD' or is_pub;
```