

```

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

```

```

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

```

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