```
1
     CREATE SCHEMA SocialDB DEFAULT CHARACTER SET 'UTF8MB4';
 2
     -- UTF8MBR IS USE FOR ASCII (American Standard Code for Information Interchange)
 3
     USE SocialDB
 4
     CREATE TABLE UserProfile (
 5
     UserProfileID INTEGER PRIMARY KEY AUTO_INCREMENT,
 6
     UserName VARCHAR(20) UNIQUE NOT NULL,
 7
     Password VARCHAR(20) NOT NULL,
 8
     Email
             VARCHAR(40) UNIQUE NOT NULL,
 9
     FirstName VARCHAR(20) NOT NULL,
10
     MiddleName VARCHAR(20),
11
     LastName VARCHAR(20) NOT NULL,
12
     CreatedDTTM DATETIME NOT NULL DEFAULT NOW(),
13
     UpdatedDTTM DATETIME,
14
     IsDeleted BIT NOT NULL DEFAULT 0
15
     );
16
17
     INSERT INTO UserProfile(UserName, Password, Email, FirstName, LastName)
18
     VALUES('a','a','a@test.com','a','a');
19
20
     INSERT INTO UserProfile(UserName, Password, Email, FirstName, LastName)
21
     VALUES('b','b','b@test.com','b','b');
22
23
     INSERT INTO UserProfile(UserName, Password, Email, FirstName, LastName)
24
     VALUES('c','c','c@test.com','c','c');
25
26
     INSERT INTO UserProfile(UserName, Password, Email, FirstName, LastName)
27
     VALUES('d','d','d@test.com','d','d');
28
29
     SELECT * FROM UserProfile;
```

31 -- UserProfileExt Table <--1-1--> UserProfile

30

```
32
      -- tinytext = 255, text = 64kb, mediumtext = 16mb, longtext = 4gb (memory size for each category)
33
34
      CREATE TABLE UserProfileExt (
35
      UserProfileExtID INTEGER PRIMARY KEY AUTO_INCREMENT,
36
      UserProfileID INTEGER UNIQUE NOT NULL,
      ProfileImage VARCHAR(100),
37
38
      Phone
                  VARCHAR(12),
39
      Website
                   VARCHAR(100),
40
      HeadLine
                   VARCHAR(256),
41
      Country
                  VARCHAR(50),
      Summary
42
                    text,
43
      CONSTRAINT fk_userprofileid FOREIGN KEY(UserProfileID) REFERENCES UserProfile(UserProfileID)
44
      );
45
46
      INSERT INTO UserProfileExt(UserProfileID, ProfileImage, Phone)
47
      VALUES(1, '/storage/1/image.png', '1234');
48
49
      SELECT * FROM UserProfileExt;
50
51
      -- To Displey user info along with user profile extentions.
52
53
      SELECT * FROM UserProfile AS u
54
      LEFT OUTER JOIN UserProfileExt AS upe ON (u.UserProfileID = upe.UserProfileID);
55
56
      -- User Connections Table
57
      -- Many users can get connected with many users.
58
      -- Many users can follow many users.
59
60
      CREATE TABLE UserConnections(
61
      UserOne
                 INTEGER NOT NULL,
62
      UserTwo
                 INTEGER NOT NULL,
```

```
63
      ISConnection BIT NOT NULL,
64
      ISFollower BIT NOT NULL,
65
      Connected DTTM DATETIME NOT NULL DEFAULT NOW(),
66
      CONSTRAINT fk_userone_userprofid FOREIGN KEY(UserOne) REFERENCES UserProfile(UserProfileID),
67
      constraint fk_usertwo_userprofid foreign key(USerTwo) references UserPRofile(UserProfileID)
68
      );
69
70
      -- Insert values: a connected with b
71
      -- (a, b) and (b, a)
72
73
      INSERT INTO UserConnections
74
      VALUES(1, 2, 1, 0, NOW());
75
      INSERT INTO UserConnections
76
      VALUES(2, 1, 1, 0, NOW());
77
78
      -- (a, c) and (c, a)
79
80
      insert into UserConnections values(1, 3, 1, 0, now());
81
      insert into UserConnections values(3, 1, 1, 0, now());
82
83
      -- (a, d) d follows a
84
      insert into UserConnections values(1, 4, 0, 1, now());
85
86
      SELECT * FROM UserConnections;
87
88
      -- Create Table User Posts
89
      -- One user can post many posts.
90
91
      create table Post (
92
      PostID
                integer primary key auto_increment,
93
      Title
              varchar(250) not null,
```

```
94
       Content text not null,
 95
       PostedBy integer not null,
 96
       PostedDTTM datetime not null default now(),
 97
       constraint fk_postedby_userid foreign key(PostedBy) references UserProfile(UserProfileID)
 98
       );
 99
100
       insert into post(Title, Content, PostedBy)
101
       values('SamplePost', 'Sample Post Content', 1);
102
103
       select * from post;
104
105
       -- Post likes Table
106
107
       create table PostLike (
108
       PostLikeID integer primary key auto_increment,
109
       PostID
                  integer not null,
                  integer not null,
110
       LikedBy
111
       ActionDTTM datetime not null default now(),
112
       constraint fk_postid foreign key(PostID) references Post(PostID),
113
       constraint fk_likedby foreign key(LikedBy) references UserProfile(UserProfileID),
114
       constraint unq_postid_likeby unique(PostID, LikedBy)
115
       );
116
117
       insert into PostLike(PostID, LikedBy)
118
       values(1, 2);
119
120
       insert into PostLike(PostID, LikedBy)
121
       values(1, 3);
122
123
       select * from postlike;
```

124

```
125
       -- Post Comment Table
126
127
       create table PostComment (
128
       PostCommentID
                           integer primary key auto_increment,
129
       PostID
                     integer not null,
130
       CommentForCommentID integer,
131
       CommentText
                          text not null,
132
       CommentedBy
                          integer not null,
133
       constraint fk_postid_postcomment foreign key(PostID) references Post(PostID),
134
       constraint fk_commentedby foreign key(CommentedBy) references UserProfile(UserProfileID),
135
       constraint fk_comment_for_comment foreign key(CommentForCommentID) references
       PostComment(PostCommentID)
136
137
       );
138
139
       insert into PostComment(PostID, CommentText, CommentedBy)
140
       values(1, 'good post', 2);
141
142
       insert into PostComment(PostID, CommentForCommentID, CommentText, CommentedBy)
143
       values(1, 1, 'well said', 3);
144
145
       select * from PostComment;
146
147
       -- retrieve the posts posted by 'a'
148
       select * from Post
149
       where PostedBy = (select UserProfileID from UserProfile where username = 'a');
150
       -- retrieve the no of likes for postID - 1
151
152
       select count(*) as 'Likes for post 1' from PostLike
153
       where PostID = 1;
154
155
```

```
156
       -- Problem 1) Write a query to get connection of a (solved using subqueries)
157
       select UserName as connections_with_a from UserProfile
158
       where UserProfileID in (
159
                       select UserTwo from UserConnections
160
            where UserOne = ( select UserProfileID from UserProfile where UserName = 'a')
161
            and ISConnection = 1
162
       );
163
164
       -- get connection of a (solved using join)
165
       select UserName as connections_with_a from UserProfile as u
166
       inner join UserConnections as uc
167
       on (u.UserProfileID = uc.UserTwo
168
               and UserOne = (select UserPRofileID from UserProfile where UserName = 'a')
169
               and ISConnection = 1);
170
171
       -- Problem 2): Write a query to retreive all the followers of 'a'
172
                Note:- followers includes connections + only followers.
173
       select UserName from UserProfile as u
174
       inner join UserConnections as uc
175
       on (u.UserProfileID = uc.UserTwo
176
       and UserOne = (select UserProfileID from UserProfile where UserName = 'a')
177
       and (ISConnection = 1 or ISFollower = 1));
178
179
       -- Problem 3) Display the total likes for the posts posted by 'a'
180
       select count(*) as 'Likes for post by a' from PostLike
181
       where PostID = (select UserProfileID from UserProfile where username = 'a');
182
183
       -- Problem 4) Display the PostID and Likes count for the posts posted by 'a'
184
       select PostID, count(*) as 'Likes for post by a' from PostLike
185
       where PostID = (select UserProfileID from UserProfile where username = 'a');
186
```

```
187
       -- Problem 6) Display the name of the user with maximum post likes.
188
       -- (give name for derived table using AS)
189
       select UserName from UserProfile as u
190
       inner join Post as p
191
       on (u.UserProfileID = p.PostedBy
192
       and PostID = (select PostID from
193
       (select PostID, max(mycount) from
194
       (select PostID, count(PostID) as mycount from PostLike) as pi_cnt ) as pi_max));
195
196
197
       -- Problem 7) Display the users with maximum posts.
198
       select UserName from UserProfile as u
199
       inner join Post as p
200
       on (u.UserProfileID = p.PostedBy
201
       and PostedBy = (select PostedBy from
```

(select Postedby, count(PostedBy) as mycount from Post) as po_cnt) as po_max));

202

203

(select PostedBy, max(mycount) from