MySQL Script:

- *Creating Table,*
- *Inserting sample values*
- *Retrieve Table Content*

```
1  CREATE SCHEMA SocialDB DEFAULT CHARACTER SET 'UTF8MB4';
```

UTF8MBR IS USE FOR ASCII (American Standard Code for Information Interchange)

```
2  USE SocialDB;
```

Creating UserProfile Table for storing user data.

```
3  CREATE TABLE UserProfile (
4  UserProfileID INTEGER PRIMARY KEY AUTO_INCREMENT,
5  UserName      VARCHAR(20) UNIQUE NOT NULL,
6  Password      VARCHAR(20) NOT NULL,
7  Email         VARCHAR(40) UNIQUE NOT NULL,
8  FirstName     VARCHAR(20) NOT NULL,
9  MiddleName    VARCHAR(20),
10 LastName      VARCHAR(20) NOT NULL,
11 CreatedDTTM   DATETIME NOT NULL DEFAULT NOW(),
12 UpdatedDTTM   DATETIME,
13 IsDeleted     BIT NOT NULL DEFAULT 0
14 );
```

Inserting sample user data in table.

```
15 INSERT INTO UserProfile(UserName, Password, Email, FirstName, LastName)
16 VALUES('a','a','a@test.com','a','a');
17 INSERT INTO UserProfile(UserName, Password, Email, FirstName, LastName)
18 VALUES('b','b','b@test.com','b','b');
19 INSERT INTO UserProfile(UserName, Password, Email, FirstName, LastName)
20 VALUES('c','c','c@test.com','c','c');
```

```
21  INSERT INTO UserProfile(UserName, Password, Email, FirstName, LastName)
22  VALUES('d','d','d@test.com','d','d');

23  SELECT * FROM UserProfile;
```

Creating UserProfile Extension Table for storing user's personal data.

-- UserProfileExt Table <--1-1--> UserProfile. Use constraint.

-- tinytext = 255, text = 64kb, mediumtext = 16mb, longtext = 4gb (memory size for each category)

```
24  CREATE TABLE UserProfileExt (
25  UserProfileExtID  INTEGER PRIMARY KEY AUTO_INCREMENT,
26  UserProfileID     INTEGER UNIQUE NOT NULL,
27  ProfileImage      VARCHAR(100),
28  Phone             VARCHAR(12),
29  Website           VARCHAR(100),
30  HeadLine          VARCHAR(256),
31  Country           VARCHAR(50),
32  Summary           text,
33  CONSTRAINT fk_userprofileid FOREIGN KEY(UserProfileID) REFERENCES UserProfile(UserProfileID)
34  );

35  INSERT INTO UserProfileExt(UserProfileID, ProfileImage, Phone)
36  VALUES(1, '/storage/1/image.png', '1234');

37  SELECT * FROM UserProfileExt;
```

To Display user info along with user profile extensions.

```
38  SELECT * FROM UserProfile AS u
39  LEFT OUTER JOIN UserProfileExt AS upe ON (u.UserProfileID = upe.UserProfileID);
```

```
    User Connections Table.
    -- Many users can get connected with many users.
    -- Many users can follow many users.
40  CREATE TABLE UserConnections(
41  UserOne      INTEGER NOT NULL,
42  UserTwo      INTEGER NOT NULL,
43  ISConnection  BIT NOT NULL,
44  ISFollower    BIT NOT NULL,
45  ConnectedDTTM DATETIME NOT NULL DEFAULT NOW(),
46  CONSTRAINT fk_userone_userprofid FOREIGN KEY(UserOne) REFERENCES UserProfile(UserProfileID),
47  constraint fk_usertwo_userprofid foreign key(USerTwo) references UserPRofile(UserProfileID)
48  );

    Insert values: a connected with b
    -- (a, b) and (b, a)
49  INSERT INTO UserConnections
50  VALUES(1, 2, 1, 0, NOW());
51  INSERT INTO UserConnections
52  VALUES(2, 1, 1, 0, NOW());

53  -- (a, c) and (c, a)
54  insert into UserConnections values(1, 3, 1, 0, now());
55  insert into UserConnections values(3, 1, 1, 0, now());

56   -- (a, d) d follows a
57  insert into UserConnections values(1, 4, 0, 1, now());

58  SELECT * FROM UserConnections;
```

```
    Create Table User Posts
    -- One user can post many posts.
59  create table Post (
60  PostID      integer primary key auto_increment,
61  Title       varchar(250) not null,
62  Content     text not null,
63  PostedBy    integer not null,
64  PostedDTTM  datetime not null default now(),
65  constraint fk_postedby_userid foreign key(PostedBy) references UserProfile(UserProfileID)
66  );

67  insert into post(Title, Content, PostedBy)
68  values('SamplePost', 'Sample Post Content', 1);

69  select * from post;

    Post likes Table
70  create table PostLike (
71  PostLikeID    integer primary key auto_increment,
72  PostID        integer not null,
73  LikedBy       integer not null,
74  ActionDTTM    datetime not null default now(),
75  constraint fk_postid foreign key(PostID) references Post(PostID),
76  constraint fk_likedby foreign key(LikedBy) references UserProfile(UserProfileID),
77  constraint unq_postid_likeby unique(PostID, LikedBy)
78  );
79  insert into PostLike(PostID, LikedBy)
80  values(1 , 2);
81  insert into PostLike(PostID, LikedBy)
82  values(1, 3);
83  select * from postlike;
```

Post Comment Table

```
84  create table PostComment (
85  PostCommentID        integer primary key auto_increment,
86  PostID               integer not null,
87  CommentForCommentID  integer,
88  CommentText          text not null,
89  CommentedBy          integer not null,
90  constraint fk_postid_postcomment foreign key(PostID) references Post(PostID),
91  constraint fk_commentedby foreign key(CommentedBy) references UserProfile(UserProfileID),
92  constraint fk_comment_for_comment foreign key(CommentForCommentID) references
93  PostComment(PostCommentID)
94  );


95  insert into PostComment(PostID, CommentText, CommentedBy)
96  values(1, 'good post', 2);
97  insert into PostComment(PostID, CommentForCommentID, CommentText, CommentedBy)
98  values(1, 1, 'well said', 3);


99  select * from PostComment;
```

Retrieve the posts posted by 'a'

```
100 select * from Post
101 where PostedBy = (select UserProfileID from UserProfile where username = 'a');
```

Retrieve the no of likes for postID - 1

```
102 select count(*) as 'Likes for post 1' from PostLike
103 where PostID = 1;
```