

Project Part #1

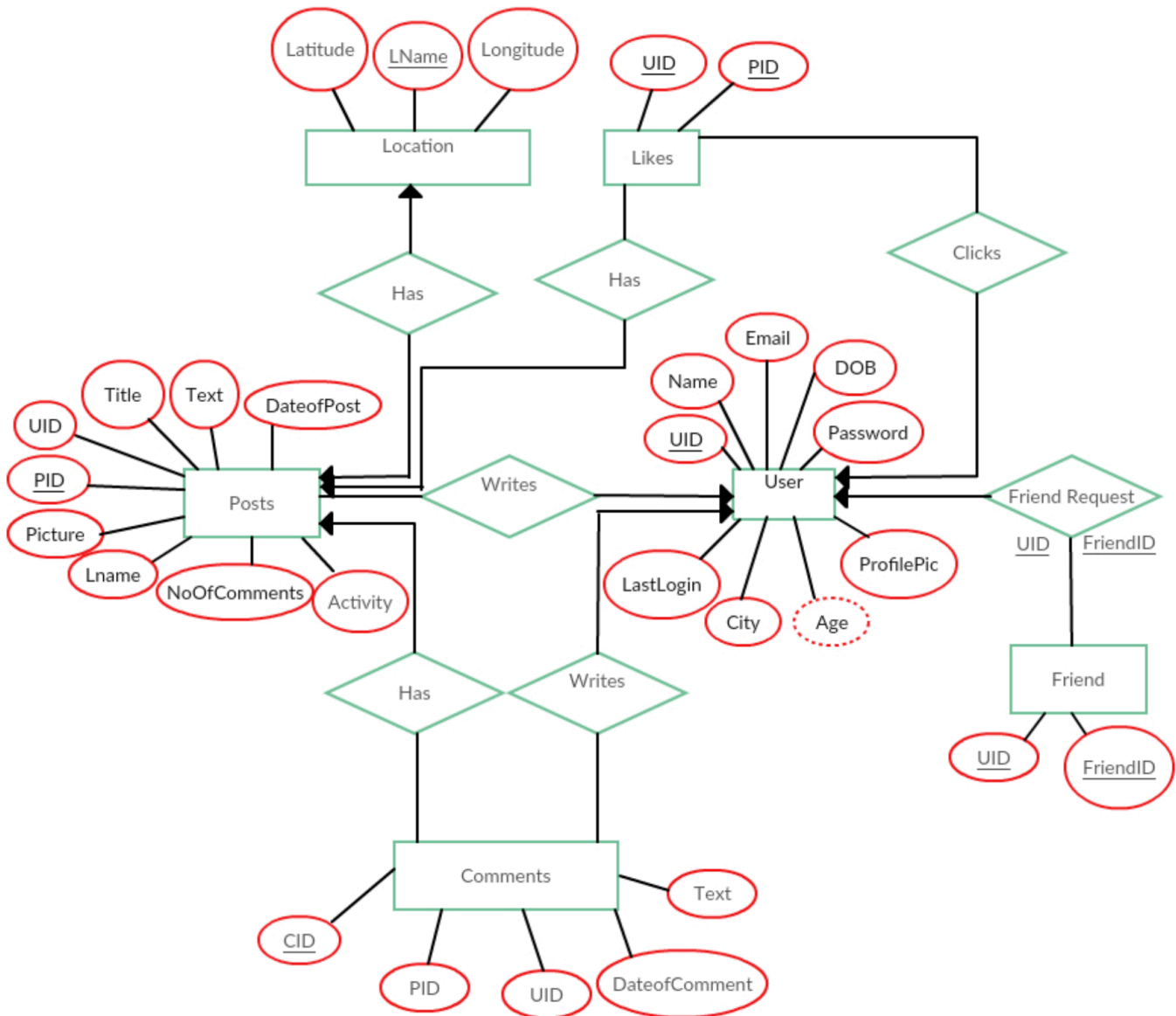
Submitted By: Geethaanjali Karupanna Murugesan (gkm253) and Vaishali Pari (vp1096)

Outdoor Enthusiasts Social Network

The outdoor enthusiast social network site has a database with six entities- User, Posts, Friends, Friend Requests, Comments, Likes and Location. The idea is that Users interested in outdoor activities such as Kayaking, Boating, Hiking, Skiing etc sign up in the site and share their experiences. Users sign up with their email id and edit details about themselves such as Date of Birth, City they live in etc and upload a profile picture for them as well. They can post about their experiences by uploading pictures and writing text descriptions along with a title. The location of the activity and the activity name are tagged in the post. Users can add friends by sending out friend requests. The friend request is automatically updates as a friend when a user accepts a friend request. Posts posted by friends are visible on the user's feed ordered by the date of the post. Users can also choose to comment on their friend's posts. Posts display the picture, text description given by the user, the Date of Post, the activity and location tags along with the User's Name and Profile picture who made the post. While comments, display the text comment written by the user's friends along with the name of the friend who wrote the comment and the date when the comment was written on the post. Users could also choose to like posts posted by their friends.

The following is the ER diagram for the described scenario. It indicates all attributes of the six entities and the relationship along with the kind of relationship the entities hold with each other. The ER diagram has been drawn using www.creately.com

ER Diagram – Outdoor Enthusiast Social Network



Relational Schema

The following is the relational schema for the described scenario. The attributes that are underlined are the primary keys of the relation.

User (UID, Name, Email, Password, DOB, Age, City, LastLogin, ProfilePic)

Posts (PID, UID, Title, Texts, DateofPost, Picture, NoOfComments, LName, Activity)

Friends (UID, FriendID)

Friend_Request (UID, FriendID)

Comments (CID, PID, UID, Texts, DateOfComment)

Location (LName, Latitude, Longitude)

Likes (UID, PID)

1. **User** relation contains information about the user such as Name, Email ID, Date of Birth, City and Profile Picture. UID, that is, User ID is the primary key of this relation.
2. **Posts** has PID, that is, Post ID as its primary key. A post can have a title, a text description along with a picture. The post also has information about the activity described in the post and its location. Posts has UID and LName as foreign keys.
3. **Friends** has both User ID and the Friend's ID as primary key so as to uniquely identify each record in the relation.
4. **Friend_Request** is similar to the Friends relation. A record is present in this table from the time the user has given a friend request until the friend accepts the request. Once the request is accepted, the record is deleted in this relation and updated in the Friends relation since they have become friends.
5. **Comments** has CID, that is, Comment ID as its primary key. Comments has information about the text that was commented, by which user and when the comment was made. Comment has PID and UID as foreign keys to identify on which post the comment has been made and by which user.
6. **Location** relation has information about different locations. LName or the location name is the primary key. It also has the latitude and longitude of the location.
7. **Likes** relation contains information about which user likes which posts and thus has attributes UID and PID.

DDL Queries

Creation of Database

```
CREATE SCHEMA `Outdoor_enthusiasts_social_network` ;
```

Creation of User Table

```
CREATE TABLE `Outdoor_enthusiasts_social_network`.`User` (  
  `UID` VARCHAR(20) NOT NULL,  
  `Name` VARCHAR(30) NOT NULL,  
  `Email` VARCHAR(70) NOT NULL UNIQUE,  
  `Password` VARCHAR(32) NOT NULL,  
  `DOB` DATETIME,  
  `Age` INT,  
  `City` VARCHAR(30),  
  `LastLogin` DATETIME,  
  `ProfilePic` BLOB,  
  PRIMARY KEY (`UID`));
```

Creation of Friends Table

```
CREATE TABLE `Outdoor_enthusiasts_social_network`.`Friends` (  
  `FriendID` VARCHAR(20) NOT NULL,  
  `UID` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`Friend_id`, `UID`),  
  INDEX `UID_idx` (`UID` ASC),  
  CONSTRAINT `UID`  
  FOREIGN KEY (`UID`) REFERENCES `Outdoor_enthusiasts_social_network`.`User` (`UID`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE);
```

Creation of Friend_Requests Table

```
CREATE TABLE `Outdoor_enthusiasts_social_network`.`Friend_request` (  
  `FriendID` INT NOT NULL,  
  `UID` INT NOT NULL,  
  PRIMARY KEY (`FriendID`, `UID`),
```

```
INDEX `UID_idx` (`UID` ASC),  
CONSTRAINT `UID`  
FOREIGN KEY (`UID`) REFERENCES `Outdoor_enthusiasts_social_network`.`User` (`UID`)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

Creation of Location Table

```
CREATE TABLE `Outdoor_enthusiasts_social_network`.`Location` (  
  `LName` VARCHAR(30) NOT NULL,  
  `Latitude` FLOAT NOT NULL,  
  `Longitude` FLOAT NOT NULL,  
  PRIMARY KEY (`LName`));
```

Creation of Posts Table

```
CREATE TABLE `Outdoor_enthusiasts_social_network`.`Posts` (  
  `PID` VARCHAR(20) NOT NULL,  
  `UID` VARCHAR(20) NOT NULL,  
  `Title` VARCHAR(30) NOT NULL,  
  Texts TEXT,  
  DateOfPost DATETIME,  
  Picture LONGBLOB,  
  NoOfComments int,  
  LName VARCHAR(30),  
  Activity VARCHAR(30)  
  PRIMARY KEY (PID),  
  FOREIGN KEY (`UID`) REFERENCES `Outdoor_enthusiasts_social_network`.`User` (`UID`)  
  FOREIGN KEY (`LName`) REFERENCES `Outdoor_enthusiasts_social_network`.`Location`  
  (`LName`));
```

Creation of Comments Table

```
CREATE TABLE `Outdoor_enthusiasts_social_network`.`Comments` (  
  `CID` VARCHAR(20) NOT NULL,  
  `PID` VARCHAR(20) NOT NULL,  
  `UID` VARCHAR(20) NOT NULL,  
  DateOfComment DATETIME,
```

Texts TEXT,
PRIMARY KEY (CID));

Creation of Likes Table

```
CREATE TABLE `Outdoor_enthusiasts_social_network`.`Likes` (  
  'UID' VARCHAR(20) NOT NULL,  
  'PID' VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`UID`, `PID`)  
  FOREIGN KEY (`UID`) REFERENCES `Outdoor_enthusiasts_social_network`.`User` (`UID`),  
  FOREIGN KEY (`LName`) REFERENCES  
  `Outdoor_enthusiasts_social_network`.`Posts`(`PID`));
```

Table with Sample Data

1. User

SELECT * FROM User

UID	Name	Email	Password	DOB	Age	City	LastLogin	ProfilePic
1	Monica	monica@mail.com	monica123	1991-01-01	26	Boston	2017-03-30 16:31:24	[BLOB - 23.5 KiB]
2	Rachael	rachael@mail.com	rachael123	1994-07-07	22	New York	2017-03-30 16:34:59	[BLOB - 64 KiB]
3	Phoebe	phoebe@mail.com	phoebe123	1988-06-05	28	New York	2017-03-30 17:02:29	[BLOB - 64 KiB]
4	Chandler	chandler@mail.com	chandler123	1994-03-01	23	Chicago	2017-03-30 17:04:35	[BLOB - 59.3 KiB]
5	Ross	ross@mail.com	ross123	1990-01-02	27	Phoenix	2017-03-30 17:08:14	[BLOB - 14 KiB]
6	Joey	joey@mail.com	joey23	1994-06-06	22	New York	2017-03-30 17:11:01	[BLOB - 19.9 KiB]

2. Location

SELECT * FROM Location

LName	Latitude	Longitude
Charles River	42.193	-71.5119
Colorado Springs	38.8339	104.821
Grand Canyon	36.1128	113.996
Hudson River	40.788	-73.9836
Lake Pleasant	33.8887	112.28
North Woods	38.7042	90.2834
Sedona	34.8697	111.761

3. Posts

SELECT * FROM Posts

PID	UID	Title	Text	DateOfPost	Picture	NoOfComments	LName	Activity
1	1	Most favorite hike	I recently hiked the grand canyon and you guys mus...	2017-03-30 18:04:24	[BLOB - 239 KiB]	2	Grand Canyon	Hiking
10	5	GC Hike	GC has some of the best trails to hike!	2017-03-30 19:17:57	[BLOB - 239 KiB]	0	Grand Canyon	Hiking
2	2	Kayaking at Charles	Kayaking at the Charles river is such a wonderful ...	2017-03-30 18:05:34	[BLOB - 4.5 KiB]	3	Charles River	Kayaking
3	2	Trip to Sedona	Sedona is a beautiful place in Arizona meant for h...	2017-03-30 18:04:42	[BLOB - 4.9 KiB]	0	Sedona	Hiking
4	5	Best Kayaking Trip	I went on a trip to Boston last summer and had the...	2017-03-30 18:06:57	[BLOB - 4.5 KiB]	1	Charles River	Kayaking
5	4	Hudson and me	Hudson river is one of the best places to go on a ...	2017-03-30 18:09:26	[BLOB - 4.8 KiB]	0	Hudson River	Boating
6	4	Boating - Charles	Had a good time boating at Charles!	2017-03-30 19:11:00	[BLOB - 4.5 KiB]	0	Charles River	Boating
7	1	Kayak at Hudson	One of the best things to do in the East Coast dur...	2017-03-30 19:13:47	[BLOB - 4.8 KiB]	0	Hudson River	Kayaking
8	1	The North Woods Hike	Hiking through the North Woods was fun!	2017-03-30 19:15:09	[BLOB - 4.8 KiB]	0	North Woods	Hiking
9	6	Boating - Hudson	Had a lovely time boating across the Hudson River	2017-03-30 19:16:35	[BLOB - 4.8 KiB]	0	Hudson River	Boating

4. Friend_Request

SELECT * FROM Friend_Request

UID	FriendID
3	4
6	1

5. Friends

SELECT * FROM Friends

UID	FriendID
1	2
1	3
1	4
1	5
2	1
2	3
2	5
3	1
3	2
4	1
4	5
4	6
5	1
5	2
5	4
5	6
6	4
6	5

6. Comments

SELECT * FROM Comments

CID	PID	UID	DateOfComment	Texts
1	1	3	2017-03-31 17:46:29	That sounds cool.
2	1	1	2017-03-30 19:32:06	You should try it.
3	2	1	2017-03-30 19:32:46	The picture is beautiful
4	2	3	2017-03-30 19:33:03	I second that!
5	2	2	2017-03-30 19:33:23	Thanks guys.
6	4	4	2017-03-30 19:34:03	Looks lovely. I should try this.

7. Likes

SELECT * FROM Likes

UID	PID
1	2
5	9
6	4

Sample queries(DML)

For the sample queries, we assume that User with UID = 1, that is, Monica is logged in.

1. List Friends

```
SELECT A.Name, A.Email, A.DOB, A.Age, A.City, A.ProfilePic
FROM User A, User B, Friends
WHERE A.UID = Friends.FID and B.UID = Friends.UID and B.UID = 1
```

Name	Email	DOB	Age	City	ProfilePic
Rachael	rachael@mail.com	1994-07-07	22	New York	[BLOB - 64 KiB]
Phoebe	phoebe@mail.com	1988-06-05	28	New York	[BLOB - 64 KiB]
Chandler	chandler@mail.com	1994-03-01	23	Chicago	[BLOB - 59.3 KiB]
Ross	ross@mail.com	1990-01-02	27	Phoenix	[BLOB - 14 KiB]

2. List Posts by Friends

```
SELECT A.Name as Friend_Name, Post.Title, Post.Text, Post.Picture, Post.LName as
Location, Post.Activity, Post.NumberOfComments, Post.DateOfPost
FROM User A, User B, Friends, Post
WHERE A.UID = Friends.FID and B.UID = Friends.UID and B.UID = 1 and Post.UID =
A.UID
ORDER BY Post.DateOfPost
```

Friend_Name	Title	Text	Picture	Location	Activity	NumberOfComments	DateOfPost
Rachael	Trip to Sedona	Sedona is a beautiful place in Arizona meant for h...	[BLOB - 4.9 KiB]	Sedona	Hiking	0	2017-03-30 18:04:42
Rachael	Kayaking at Charles	Kayaking at the Charles river is such a wonderful ...	[BLOB - 4.5 KiB]	Charles River	Kayaking	3	2017-03-30 18:05:34
Ross	Best Kayaking Trip	I went on a trip to Boston last summer and had the...	[BLOB - 4.5 KiB]	Charles River	Kayaking	1	2017-03-30 18:06:57
Chandler	Hudson and me	Hudson river is one of the best places to go on a ...	[BLOB - 4.8 KiB]	Hudson River	Boating	0	2017-03-30 18:09:26
Chandler	Boating - Charles	Had a good time boating at Charles!	[BLOB - 4.5 KiB]	Charles River	Boating	0	2017-03-30 19:11:00
Ross	GC Hike	GC has some of the best trails to hike!	[BLOB - 239 KiB]	Grand Canyon	Hiking	0	2017-03-30 19:17:56

3. Updating email ID

```
UPDATE User
SET Email = 'mons@mail.com'
WHERE User.UID = 1
```

4. Updating Age given DOB

```
UPDATE User
SET Age = ((DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(`DOB`, '%Y')) -
(DATE_FORMAT(NOW(), '00-%m-%d') < DATE_FORMAT(`DOB`, '00-%m-%d')))
```

5. View Friends who have given Friend Requests

```
Select A.Name, A.ProfilePic FROM User A, User B, Friend_Request
WHERE Friend_Request.UID = A.UID and Friend_Request.FID = B.UID and B.UID = 1
```

Name	ProfilePic
Joey	[BLOB - 19.9 KiB]

6. List all locations tagged by Friends

```
SELECT B.Name as Friend, Location.LName as LocationTagged
FROM User A, User B, Friends, Location, Post
WHERE A.UID = Friends.UID and Friends.FriendID = B.UID and A.UID = 1 and B.UID =
Post.UID and Post.LName = Location.LName
ORDER BY Post.DateOfPost DESC
```

Friend	LocationTagged
Rachael	Sedona
Rachael	Charles River
Ross	Charles River
Chandler	Hudson River
Chandler	Charles River
Ross	Grand Canyon

7. List Mutual Friends between Monica and Joey

```
SELECT User.Name as MutualFriends
FROM Friends f1 INNER JOIN Friends f2 ON f1.FriendID = f2.FriendID INNER JOIN
User ON f1.FriendID = User.UID
WHERE f1.UID = 1 AND f2.UID = 6
```

MutualFriends

Chandler

Ross

8. List Friends of Friends

```
SELECT DISTINCT User.Name
FROM Friends f1 INNER JOIN Friends f2 ON f1.FriendID = f2.UID INNER JOIN User
ON f2.FriendID = User.UID WHERE f1.UID = 1 and f1.UID != f2.FriendID
```

FOF

Phoebe

Ross

Rachael

Joey

Chandler

9. List posts that have tagged Hudson River from Friends

```
SELECT User.Name, User.ProfilePic, Post.Title, Post.Texts, Post.Picture,
Post.DateOfPost
FROM Post, Friends A, Friends B, User
WHERE A.UID= 1 AND B.FriendID = 1 AND User.UID != 1 AND B.UID = User.UID
AND A.FriendID = B.UID AND User.UID = Post.UID AND Post.LName = "Hudson
River"
ORDER BY Post.DateOfPost DESC
```

Name	ProfilePic	Title	Text	Picture	DateOfPost
Chandler	[BLOB - 59.3 KiB]	Hudson and me	Hudson river is one of the best places to go on a ...	[BLOB - 4.8 KiB]	2017-03-30 18:09:26

10. List posts that have tagged Hudson River from every user

```
SELECT User.Name, User.ProfilePic, Post.Title, Post.Text, Post.Picture,
Post.DateOfPost
FROM Post, User
WHERE User.UID != 1 AND Post.LName = "Hudson River" and Post.UID = User.UID
ORDER BY Post.DateOfPost DESC
```

Name	ProfilePic	Title	Text	Picture	DateOfPost
Joey	[BLOB - 19.9 KiB]	Boating - Hudson	Had a lovely time boating across the Hudson River	[BLOB - 4.8 KiB]	2017-03-30 19:16:35
Chandler	[BLOB - 59.3 KiB]	Hudson and me	Hudson river is one of the best places to go on a ...	[BLOB - 4.8 KiB]	2017-03-30 18:09:26

11. List posts about Hiking from Friends

```
SELECT User.Name, User.ProfilePic, Post.Title, Post.Text, Post.Picture,
Post.DateOfPost
FROM Post, Friends A, Friends B, User
WHERE A.UID = 1 and B.FriendID = 1 and User.UID != 1 and B.UID = User.UID and
A.FriendID = B.UID and User.UID = Post.UID and Post.Activity = "Hiking"
ORDER BY Post.DateOfPost DESC
```

Name	ProfilePic	Title	Text	Picture	DateOfPost
Ross	[BLOB - 14 KiB]	GC Hike	GC has some of the best trails to hike!	[BLOB - 239 KiB]	2017-03-30 19:17:57
Rachael	[BLOB - 64 KiB]	Trip to Sedona	Sedona is a beautiful place in Arizona meant for h...	[BLOB - 4.9 KiB]	2017-03-30 18:04:42

12. List posts about Boating from everyone

```
SELECT User.Name, User.ProfilePic, Post.Title, Post.Text, Post.Picture,
Post.DateOfPost
FROM Post, User
WHERE User.UID != 1 and Post.activity = "Boating" and Post.UID = User.UID
ORDER BY Post.DateOfPost DESC
```

Name	ProfilePic	Title	Text	Picture	DateOfPost
Joey	[BLOB - 19.9 KiB]	Boating - Hudson	Had a lovely time boating across the Hudson River	[BLOB - 4.8 KiB]	2017-03-30 19:16:35
Chandler	[BLOB - 59.3 KiB]	Boating - Charles	Had a good time boating at Charles!	[BLOB - 4.5 KiB]	2017-03-30 19:11:00
Chandler	[BLOB - 59.3 KiB]	Hudson and me	Hudson river is one of the best places to go on a ...	[BLOB - 4.8 KiB]	2017-03-30 18:09:26

13. View Comments on a Post

```
SELECT Comments.Texts, Comments.DateOfComment
FROM Comments, Post
WHERE Comments.PID = Post.PID AND Comments.pid = 2
```

Texts	DateOfComment
The picture is beautiful	2017-03-30 19:32:46
I second that!	2017-03-30 19:33:03
Thanks guys.	2017-03-30 19:33:23

14. List Users who like a post

```
SELECT User.Name as PeopleWhoLikedThePost, Post.Title as PostTitle
FROM User, Post, Likes
WHERE Likes.PID = Post.PID and Likes.UID = User.UID and Post.PID = 2
```

PeopleWhoLikedThePost	PostTitle
Monica	Kayaking at Charles

15. List Posts with keyword Kayaking

```
SELECT User.Name, User.ProfilePic, Post.Title, Post.Texts, Post.Picture,
Post.DateOfPost
FROM User INNER JOIN Post on Post.UID = User.UID
WHERE User.UID != 1 and Post.Texts like "%kayak%"
ORDER BY Post.DateOfPost DESC
```

Name	ProfilePic	Title	Texts	Picture	DateOfPost
Ross	[BLOB - 14 KiB]	Best Kayaking Trip	I went on a trip to Boston last summer and had the...	[BLOB - 4.5 KiB]	2017-03-30 18:06:5
Rachael	[BLOB - 64 KiB]	Kayaking at Charles	Kayaking at the Charles river is such a wonderful ...	[BLOB - 4.5 KiB]	2017-03-30 18:05:3

16. List of post made by everyone in the last week

```

SELECT User.Name, User.ProfilePic, Post.Title, Post.Text, Post.Picture,
Post.DateOfPost
FROM Post, User
WHERE User.UID != 1 and Post.UID = User.UID and (Post.DateOfPost <= curdate()
and Post.DateOfPost >= curdate()-7)
ORDER BY Post.DateOfPost DESC

```

Stored Procedures

1. Inserting values into User Table

```

CREATE PROCEDURE `USER_INSERT`(`UID` VARCHAR(20), `Name` VARCHAR(30),
`Email` VARCHAR(70), `Password` VARCHAR(32), `DOB` DATETIME, `Age` INT, `City`
VARCHAR(20))
Insert Into User Values (@UID, @Name, @Email, @Password, @DOB, @Age,
@City)

```

2. Writing a new post

```

CREATE PROCEDURE `POST_INSERT`(`PID` VARCHAR(20), `UID` VARCHAR(20), `Title`
VARCHAR(30), Texts TEXT, DateOfPost DATETIME, Picture LONGBLOB,
NoOfComments int, LName VARCHAR(30), Activity VARCHAR(30))
Insert Into Post Values (@PID, @UID, @Title, @Texts, @DateOfPost, @Picture,
@NoOfComments, @LName, @Activity)

```

3. Liking a Post

```

CREATE PROCEDURE `ADD_LIKE`(`UID` VARCHAR(20), `PID` VARCHAR(20))
Insert Into Likes Values (@UID, @PID)

```

4. Commenting on a Post

```
CREATE PROCEDURE 'ADD_COMMENT' ('CID' VARCHAR(20), 'PID' VARCHAR(20),  
'UID' VARCHAR(20), 'DateOfComment' DATETIME, 'Texts' Text)  
INSERT Into Comments Values(@CID, @PID, @UID, @DateOfComment, @Texts)
```

5. On Accepting Friend Requests

```
DELETE FROM Friend_Request  
WHERE FID.Friend_Request = 'AcceptedID' and UID.Friend_Request =  
'RequestedID'
```

```
INSERT INTO Friends (UID, FID) VALUES ('RequestedID' , 'AcceptedID');  
INSERT INTO Friends (UID, FID) VALUES ('AcceptedID' , 'RequestedID');
```

6. Increasing number of comments

```
DECLARE @count int  
SET @count = 0  
UPDATE Posts  
SET @count = NoOfComments = @count + 1
```