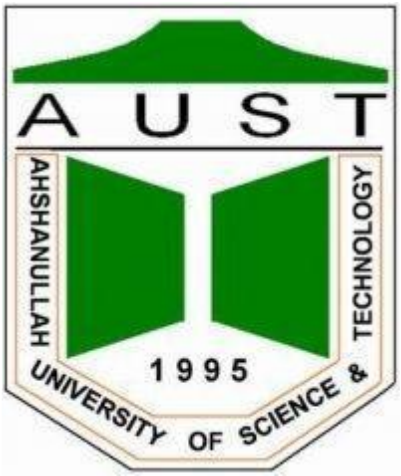


Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Project Name : Database System for a General Blog Site.

Course No : CSE3104
Course Title : Database Lab

Date of Submission: 30.09.2018
Submitted To : Mr. Nazmus Sakib
Mr. Mir Tafseer Nayeem

Submitted By-
Name: Tahmeed Asaad Rahman
Id : 16.01.04.039
Name : MD. Faisal Ahmed
Id : 16.01.04.046
Section : A_2

Project objective : In our project, we have designed a database system for any general dynamic blog site. Every user will have an account and by using it he can do all the work(create post, like, comment, reply etc.) allowed by the administrator of the blog site.

All of the user's full name, password, email, post and other information will be stored in the database. Also various information about posts, posts categories, comments, likes, replies will be stored in the database.

Features of the Project: After analyzing the necessary features, we made an ERD for the project. Then the relational model was created. There are total nine entity in the database system.

For different search operation or to view, update, delete, retrieve data from the database ,there are overall 70 queries in the project.

Types of users : There are three types of users. Firstly some queries is for the admin who handles users information, another one is for the admin who handles post related information and the last one is the admin who handles comments and replies related information. All the necessary queries for these users are written.

Feature grouping according to the users :

1. **User Related Admin**: This user can view all the users details in the database, update them and if needed he can also delete user id and other information.

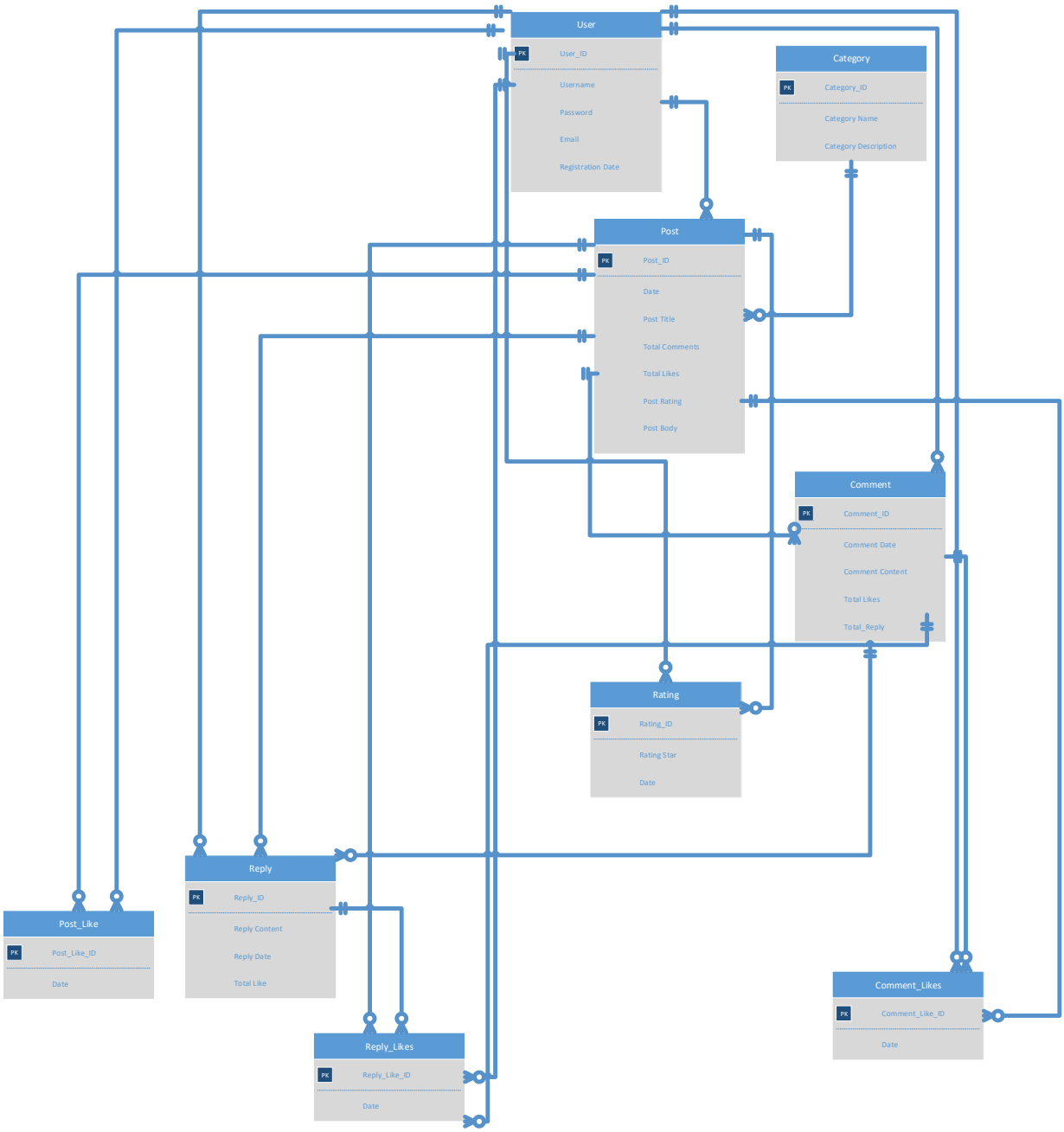
2. **Post Related Admin**: This user can view all posts, category, rating, likes on a post. How much likes/ comments the nth most liked/commented post has can be found easily. He can edit the post, also how many comments, rating each post has can be easily found. There is also a feature to know which post has the maximum activity. All the categories, and highest rated posts are present in which category has can be found.

3. **Comment and Reply Related Admin**: This user can find the post title, post body, comment of that post with the reply. All replies, comments each post has can be viewed. All the replies after a specific date can be easily found.

Names of the entity with primary key :

Entity Name	Primary Key
User	User_ID
Category	Category_ID
Post	Post_ID
Comment	Comment_ID
Rating	Rating_ID
Reply	Reply_ID
Comment_Likes	Comment_Likes_ID
Post_Like	Post_Like_ID
Reply_Like	Reply_Like_ID

ER Diagram :

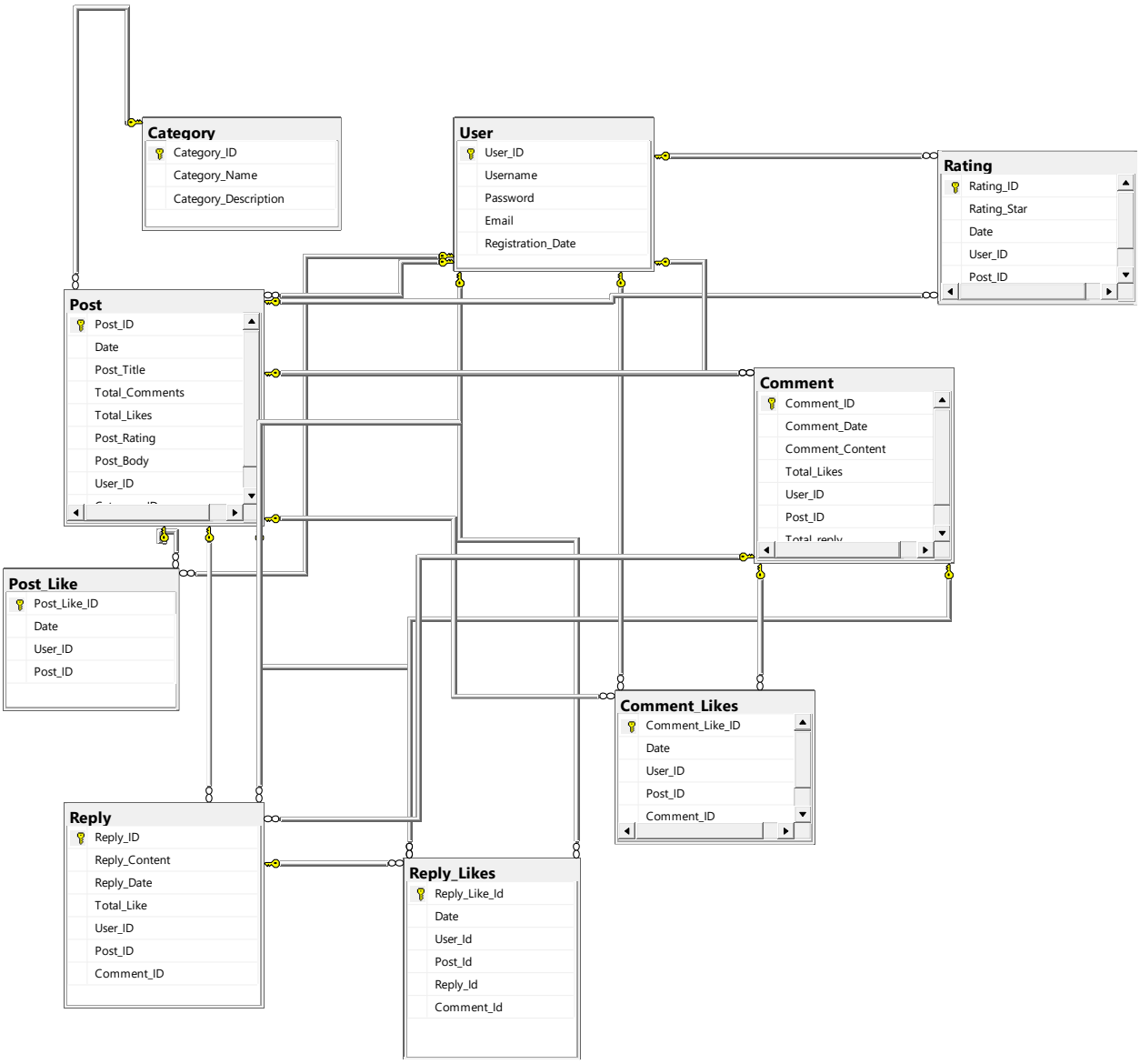

User
PK
User_ID
Username
Password
Email
Registration Date
Category
PK
Category_ID
Category Name
Category Description
Post
PK
Post_ID
Date
Post Title
Total Comments
Total Likes
Post Rating
Post Body
Comment
PK
Comment_ID
Comment Date
Comment Content
Total Likes
Total_Reply
Rating
PK
Rating_ID
Rating Star
Date
Reply
PK
Reply_ID
Reply Content
Reply Date
Total Like
Post_Like
PK
Post_Like_ID
Date
Reply_Likes
PK
Reply_Like_ID
Date
Comment_Likes
PK
Comment_Like_ID
Date

Relational Model:

```
CREATE TABLE [User]
(
User_ID int IDENTITY(1,1) PRIMARY KEY,
Username varchar(50) NOT NULL,
[Password] varchar(30) NOT NULL,
Email varchar(40) NOT NULL,
Registration_Date date NOT NULL
)

CREATE TABLE Category
(
Category_ID int IDENTITY(1,1) PRIMARY KEY,
Category_Name varchar(100) NOT NULL,
Category_Description varchar(100) NOT NULL
)

CREATE TABLE Post
(
Post_ID int IDENTITY(1,1) PRIMARY KEY,
[Date] date NOT NULL,
Post_Title varchar(150) NOT NULL,
Total_Comments int NOT NULL,
Total_Likes int NOT NULL,
Post_Rating int NOT NULL,
Post_Body varchar(2000) NOT NULL,
User_ID int not null foreign key references [User](User_ID),
Category_ID int not null foreign key references Category(Category_ID)
)

CREATE TABLE Comment
(
Comment_ID int IDENTITY(1,1) PRIMARY KEY,
Comment_Date date NOT NULL,
Comment_Content varchar(200) NOT NULL,
Total_Likes int NOT NULL,
User_ID int not null foreign key references [User](User_ID),
Post_ID int not null foreign key references Post(Post_ID),
Total_reply  int not null

)
```

```
CREATE TABLE Rating
(
Rating_ID int IDENTITY(1,1) PRIMARY KEY,
Rating_Star INT NOT NULL,
[Date] date NOT NULL,
User_ID int not null foreign key references [User](User_ID),
Post_ID int not null foreign key references Post(Post_ID)
)

CREATE TABLE Reply
(
Reply_ID int IDENTITY(1,1) PRIMARY KEY,
Reply_Content varchar(200) NOT NULL,
Reply_Date date NOT NULL,
Total_Like int NOT NULL,
User_ID int not null foreign key references [User](User_ID),
Post_ID int not null foreign key references Post(Post_ID),
Comment_ID int not null foreign key references Comment(Comment_ID)
)

CREATE TABLE Comment_Likes
(
Comment_Like_ID int IDENTITY(1,1) PRIMARY KEY,
[Date] date NOT NULL,
User_ID int not null foreign key references [User](User_ID),
Post_ID int not null foreign key references Post(Post_ID),
Comment_ID int not null foreign key references Comment(Comment_ID)
)

CREATE TABLE Post_Like
(
Post_Like_ID int IDENTITY(1,1) PRIMARY KEY,
[Date] date NOT NULL,
User_ID int not null foreign key references [User](User_ID),
Post_ID int not null foreign key references Post(Post_ID)
)

CREATE TABLE Reply_Likes
(
Reply_Like_Id int IDENTITY(1,1) PRIMARY KEY,
[Date] date NOT NULL,
User_Id int not null foreign key references [User](User_Id),
Post_Id int not null foreign key references Post(Post_Id),
Reply_Id int not null foreign key references Reply(Reply_Id),
Comment_Id int not null foreign key references Comment(Comment_Id)
)
```

Database Diagram :



SQL queries grouped under different types of users:

• **Post Related Admin**:

-->1.View the Post table

```
Select *from Post
```

-->2.View the Rating table

```
Select *from Rating
```

-->3.View the Post_Like table

```
Select *from Post_Like
```

-->4.View the Category table

```
Select *from Category
```

-->5.Edit the Title of a specific post

```
Update Post Set Post_Title = 'Christiano Ronaldo' Where Post_ID = '1'
```

-->6.View the total number of comments each post has

```
Select Comment.Post_ID, COUNT(Comment.Comment_ID)
From Comment
Group By Post_ID
```

-->7.View the total number of likes the 3rd most liked post has

```
Select Top 1 Total_Likes From
```

(Select Distinct Top 3 Total_Likes From Post Order By Total_Likes Desc)

Result

Order By Total_Likes

-->8.View the total number of comments the 2nd most commented post has

Select Top 1 Total_Comments From

(Select Distinct Top 2 Total_Comments From Post Order By Total_Comments Desc)

Result

Order By Total_Comments

-->9.View all the posts which has one or more comments on a specific date such as '2018-11-08'

Select * From Post

Where Post_ID IN

(Select Post_ID From Comment Where Comment_Date = '2018-11-08')

-->10.View all the posts which has at least one comment, also arrange them in increasing order of comment_date

Select * From Post

Full Join Comment

On Post.Post_ID = Comment.Post_ID

Order By Comment_Date

-->11.View all the posts who has the word 'Ronaldo' present in their title or in the post body.

```
Select Post_Title, Post_Body

From Post

Where Post_Body Like '%Ronaldo%' Or Post_Title Like '%Ronaldo%'
```

-->12.View the Posts in the decreasing order of their total comments

```
Select * From Post Order By Total_Comments Desc
```

-->13.View the Posts in the increasing order of their total likes

```
Select * From Post Order By Total_Likes
```

-->14.View the Posts in the increasing order of their total rating

```
Select * From Post Order By Post_Rating
```

-->15.View the post which has maximum activity means post which has total likes+comment+reply+...etc more than other posts

```
Select Post_Title, Post_Body, Post.Total_Likes, Total_Comments,
(Post.Total_Likes+Total_Comments+Reply_ID+Reply.Total_Like+Comment.Comment_ID+Comment_Like_ID) As 'Total Activity'

From Post

Join Reply

On Post.Post_ID = Reply.Reply_ID

Join Comment_Likes

On Post.Post_ID = Comment_Likes.Post_ID

Join Comment
```

On Post.Post_ID = Comment.Post_ID

Order By
(Post.Total_Likes+Total_Comments+Reply_ID+Reply.Total_Like+Comment.Comment_ID+Comment_Like_ID) DESC

-->16.To select all the post with titles and body which recieved a comment on 28th,29th,30th september,2018

```
Select Post_Id,post_title,post_body from post where post_id IN
    (select post_id from Comment where Comment_Date IN('2018-9-28','2018-9-29','2018-9-30'))
```

-->17.To select all the post with titles and body which recieved a comment from 2016 to 2018

```
Select Post_Id,post_title,post_body from post where post_id IN (select post_id from Comment where Comment_Date Between '2016-1-1' AND '2018-12-31')
```

-->18.To find average rating of posts of User-Faisal

```
Select AVG(Rating_Star) from Rating where Post_ID IN(Select Post_Id from Post where User_ID IN(Select USER_ID from [User] where USERNAME='Faisal'))
```

-->19.To find max rating recieved by User-Mrgrath

```
Select MAx(Rating_Star) from Rating where Post_ID IN(Select Post_Id from Post where User_ID IN(Select USER_ID from [User] where USERNAME='Mcgrath'))
```

-->20.To find minimum rating recieved by User-Mahmud

```
Select Min(Rating_Star) from Rating where Post_ID IN(Select Post_Id from Post where User_ID IN(Select USER_ID from [User] where USERNAME='Mahmud'))
```

-->21.To truncate all the post_rating to its integer value

Select ROUND(Post_rating,00) from Post

-->22.To show the category which contains most post

Select *from Category where Category_ID IN (Select Top 1 Category_ID from Post group by Category_ID Order By COUNT(Post_ID) DESC)

-->23.To decrease a post's rating by half which has received the lowest likes in 2016

Update Post set Post_Rating=Post_Rating-0.5 where Post_ID IN(Select TOP 1 Post_id from Post where Date>='2016-1-1' and Date<='2016-12-31' Order By Total_Likes)

-->24.To delete a post which has received the lowest rating in 2016

Delete from Post where Post_Id IN

(Select TOP 1 Post_id from Post where Date>='2016-1-1' and Date<='2016-12-31' Order By Post_Rating)

-->25.To show Username,email,all that user,s Post title and body and rating of user-Mcgrath

Select USERNAME,email,Post_Title,Post_Body,Post_Rating from [User]

Inner Join Post ON [User].User_ID=Post.User_ID where USERNAME='Mcgrath'

-->26.To find the category of the Post which have the highest rating among the post's having atleast 4.5 rating

```
Select Top 1
Post_Title,Post_Body,Post_rating,Category_name,Category_description

from Post Right Join Category on Post.Category_ID=Category.Category_ID

where Post_Rating>=4.5 Order by Post_Rating Desc
```

-->27.To find the category of the Post which have the highest rating among the post's posted after August 2018

```
Select Top 1
Post_Title,Post_Body,Post_rating,Category_name,Category_description,Date

from Post Left Join Category on Post.Category_ID=Category.Category_ID

where Date >='2018-9-1' Order by Post_Rating  DESC
```

-->28.To find the Posts which recieved a rating between 4.00 to 4.5 OR 4.5 to 5

```
Select Post_Title from Post where Post_Rating>='4.00'  and Post_Rating<=4.5

UNION

Select Post_Title from Post where Post_Rating>'4.50'
```

-->29.To find the users who recieved a rating between 4.00 to 4.5 and from 4.5 to 5

```
Select User_ID from Post where Post_Rating>='4.00'  and Post_Rating<=4.5

Intersect

Select User_ID from Post where Post_Rating>'4.50'
```

-->30.To find first 20 alphabets of all Post's title

```
Select SUBSTRING(Post_Title,1,20)  from Post
```

-->31.To cast all the ratings as integer

Select CAST(Post_Rating as integer) from Post

-->32.To convert all the ratings as integer

Select Convert(Integer,Post_Rating) from Post

-->33.To show all categories with their description as full infromation

Select Category_Name+ ' : '+ Category_Description

as full_information from Category

-->34.all the replies and comments except user_id 1

Select Comment.User_id, Reply_ID, Reply_Content, Reply_Date, Comment_Content, Comment_Date

From Reply

Inner Join Comment

On Reply.User_ID = Comment.User_ID

Where Reply.User_ID != '1'

Union All

Select Comment.User_id,Reply_ID, Reply_Content, Reply_Date, Comment_Content, Comment_Date

From Reply

Inner Join Comment

On Reply.User_ID = Comment.User_ID

Where Reply.User_ID != '1'

-->35.To find the post title, post body, comment of that post with the reply

--> which was posted on 28 September 2018 along with the Post's creator

```
select username,
Post.Post_Title,Post.Post_Body,Comment_Content,Reply_Content,Reply_Date
from Post,Comment,Reply,[User]

where Reply_Date='2018-09-28' and Reply.Comment_ID=Comment.Comment_ID

and Comment.Post_ID=Post.Post_ID

and Post.User_ID=[User].User_ID
```

• User Related Admin:

-->1.View the User table

```
Select *from [user]
```

-->2.Update Email of a specific user

```
Update [User] Set Email = 'faisal1111@gmail.com' Where Username = 'Faisal'
```

-->3.Delete all the information of an user from User table

```
Delete From [User] Where Username = 'Tahmeed'
```

-->4.View all the information of an user via his Username or Email

```
Select * From [User] Where Username = 'Faisal' OR Email = 'tahmedad@gaf.com'
```

-->5.View all the information of the user whose Username is 'Faisal' as well as Email is 'faisal1111@gmail.com'

```
Select * From [User] Where Username = 'Faisal' And Email = 'faisal1111@gmail.com'
```

-->6.View all the users information whose Registration_Date is not null

```
Select * From [User] Where Registration_Date IS NOT NULL
```

-->7.View the total number of users registered on each day

```
Select Count(User_ID), Registration_Date
From [User]
Group by Registration_Date
```

-->8.View how many usres has registered on October 2010

```
Select Count(User_ID)
From [User]
Where DATEPART(month, Registration_Date) = 10 And DATEPART(year, Registration_Date) = 2010
```

-->9.View all the users who joined on a particular date

```
Select * From [User] Where Registration_Date = '2010-10-08'
```

-->10.To add a new attribute named Date of birth in the user table

```
Alter table [User]
Add date_of_birth date
```

-->11.To change datatype of date_of_birth to datetime

```
Alter table [User]
Alter column date_of_birth datetime
```

-->12. To drop attribute named Date of birth in the user table

```
Alter table [User]
Drop column date_of_birth
```

-->13.To select all users who have registered after 2001 or who have commmented atleast three times

```
Select *from [User] where Registration_Date>'2001/12/31' OR   User_id IN
        (Select USER_ID from Comment group by User_ID having
COUNT(Comment_ID)>=3)
```

-->14. To select all users who have registered after 2001 AND who have liked any post atleast two times

```
Select *from [User] where Registration_Date>'2001/12/31' AND   User_id IN(
Select USER_ID from Post_Like group by User_ID  having
COUNT(Post_Like_ID)>=2)
```

--15>.To find average rating of posts of User-Faisal

```
Select AVG(Rating_Star) from Rating where Post_ID IN(Select Post_Id from Post
where User_ID IN(Select USER_ID from [User] where USERNAME='Faisal'))
```

--16.>To find max rating recieved by User-Mrgrath

```
Select MAx(Rating_Star) from Rating where Post_ID IN(Select Post_Id from Post
where User_ID IN(Select USER_ID from [User] where USERNAME='Mcgrath'))
```

--17.>To find minimum rating recieved by User-Mahmud

```
Select Min(Rating_Star) from Rating where Post_ID IN(Select Post_Id from Post
where User_ID IN(Select USER_ID from [User] where USERNAME='Mahmud'))
```

--18.>To find total likes given by User-Faisal combining all posts

```
Select SUM(Total_Likes) from Post where User_ID IN(Select USER_ID from [User]
where USERNAME='Faisal')
```

--19.>To select all usernames converted to uppercase

```
Select UPPER(USERNAME) from [User]
```

--20.>>To select all usernames converted to lowercase

```
Select LOWER(USERNAME) from [User]
```

--21.>To select the user having longest username

```
Select *from [User] where LEN(Username) IN (  Select MAX(Len(USERNAME)) from [User] )
```

-->22.To show Username,email,all that user,s Post title and body and rating of user-Mcgrath

```
Select USERNAME,email,Post_Title,Post_Body,Post_Rating from [User]
Inner Join Post ON [User].User_ID=Post.User_ID where USERNAME='Mcgrath'
```

-->23.To select the Users who have Not registered in 2001

```
Select *from [User]
EXCEPT
Select *from [User] where Registration_Date like '2001%'
```

-->24.Select the 3 leftmost letters of each username in ascending order

```
Select LEFT(Username,3) from [User] order by USERNAME
```

-->25.Select registration month and date of each user in ascending order

```
Select RIGHT(Registration_Date,5) from [User] order by Registration_Date
```

-->26.To reverse all the user's username

```
Select Reverse (Username) from [USer]
```

-->27.all the replies and comments except user_id 1

```
Select Comment.User_id,Reply_ID, Reply_Content, Reply_Date,
Comment_Content, Comment_Date

From Reply

Inner Join Comment

On Reply.User_ID = Comment.User_ID

Where Reply.User_ID != '1'

Union All

Select Comment.User_id,Reply_ID, Reply_Content, Reply_Date,
Comment_Content, Comment_Date

From Reply

Inner Join Comment

On Reply.User_ID = Comment.User_ID

Where Reply.User_ID != '1'
```

-->28.To find the date of previous year of each user's registration date

```
Select DATEADD(YEAR,-1,Registration_Date) from [user]
```

-->29.To find difference of days of joining of user_id 1 and 3

```
Select DATEDIFF(DAY,(Select Registration_Date from [user] where
User_ID=3),(Select Registration_Date from [user] where User_ID=1))
```

-->30.To find the months of all user's registration date

Select DATEPART(MONTH,Registration_Date) from [user]

-->31.To find the days of all user's registration date

Select DAY(Registration_Date) from [user]

-->32.To find the post title,post body,comment of that post with the reply

--> which was posted on 28th september 2018 along with the Post's creator

select username,
Post.Post_Title,Post.Post_Body,Comment_Content,Reply_Content,Reply_Date
from Post,Comment,Reply,[User]

where Reply_Date='2018-09-28' and Reply.Comment_ID=Comment.Comment_ID

and Comment.Post_ID=Post.Post_ID

and Post.User_ID=[User].User_ID

• **Comment and Reply Related Admin**:

-->1.View the Comment table

Select *from Comment

-->2.View the Reply table

Select *From Reply

-->3.View the Comment_Likes table

Select *from Comment_Likes

-->4.View the Reply_Likes table

```
Select *from Reply_Likes
```

-->5.View the total number of comments each post has

```
Select Comment.Post_ID, COUNT(Comment.Comment_ID)
From Comment
Group By Post_ID
```

-->6.View all the posts which has one or more comments on a specific date such as '2018-11-08'

```
Select * From Post
Where Post_ID IN
(Select Post_ID From Comment Where Comment_Date = '2018-11-08')
```

-->7.View all the posts which has at least one comment, also arrange them in increasing order of comment_date

```
Select * From Post
Full Join Comment
On Post.Post_ID = Comment.Post_ID
Order By Comment_Date
```

-->8.View the post which has maximum activity means post which has total likes+comment+reply+...etc more than other posts

```
Select Post_Title, Post_Body, Post.Total_Likes, Total_Comments,
(Post.Total_Likes+Total_Comments+Reply_ID+Reply.Total_Like+Comment.Comment_ID+Comment_Like_ID) As 'Total Activity'

From Post

Join Reply

On Post.Post_ID = Reply.Reply_ID

Join Comment_Likes

On Post.Post_ID = Comment_Likes.Post_ID

Join Comment

On Post.Post_ID = Comment.Post_ID

Order By
(Post.Total_Likes+Total_Comments+Reply_ID+Reply.Total_Like+Comment.Comment_ID+Comment_Like_ID) DESC
```

-->9.To select all the post with titles and body which recieved a comment on 28th,29th,30th september,2018

```
Select      Post_Id,post_title,post_body from post where post_id IN
      (select post_id from Comment where Comment_Date  IN('2018-9-28','2018-9-29','2018-9-30'))
```

-->10.To select all the post with titles and body which recieved a comment from 2016 to 2018

```
Select Post_Id,post_title,post_body from post where post_id IN   (select post_id from Comment where Comment_Date  Between '2016-1-1' AND '2018-12-31')
```

-->11.To select all users who have registered after 2001 or who have commmented atleast three times

```
Select *from [User] where Registration_Date>'2001/12/31' OR   User_id IN
        (Select USER_ID from Comment group by User_ID having
COUNT(Comment_ID)>=3)
```

-->12.To erase the records of reply_likes keeping its structure

```
Delete from Reply_Likes
```

-->13.To erase the records of reply_likes keeping its structure and to re_initialize the table

```
Truncate Table Reply_likes
```

-->14.To erase the table reply_likes from database destroying its strcuture

```
Drop Table Reply_likes
```

-->15. To find all the replies after september 2018

```
Select          Reply_Content from Reply where Reply_Date >ALL(Select Reply_date
from Reply where Reply_date<='2018-09-30' )
```

-->16.To find date of any like received by a reply after august 2018

```
select reply_like_id, date from Reply_likes where date>ANY      (select DATE from
Reply_Likes where date like '2018-09%')
```

-->17.all the replies and comments except user_id 1

```
Select Comment.User_id,Reply_ID, Reply_Content, Reply_Date,
Comment_Content, Comment_Date

From Reply
```

```
Inner Join Comment

On Reply.User_ID = Comment.User_ID

Where Reply.User_ID != '1'

Union All

Select Comment.User_id,Reply_ID, Reply_Content, Reply_Date,
Comment_Content, Comment_Date

From Reply

Inner Join Comment

On Reply.User_ID = Comment.User_ID

Where Reply.User_ID != '1'
```

-->18.To find the post title,post body,comment of that post with the reply

--> which was posted on 28th september 2018 along with the Post's creator

```
select username,
Post.Post_Title,Post.Post_Body,Comment_Content,Reply_Content,Reply_Date
from Post,Comment,Reply,[User]

where Reply_Date='2018-09-28' and Reply.Comment_ID=Comment.Comment_ID

and Comment.Post_ID=Post.Post_ID

and Post.User_ID=[User].User_ID
```

Project Limitations:

The database may have a few limitations. All the tables may not be normalized upto BCNF. So it might not be fully consistent.

Conclusion & Future work:

This project can be useful for a blog site. All it's information can be access, retrieve easily. The future plan is to make it more normalized, efficient also implementing various new queries depending on the tasks. Also more additional features can be there.