NIU CONNECT

Photo and Video Sharing App

MEMBERS:

Covenant Falana [Z1836845]
Gouri Patil [Z1931431]
Hita Kadam [Z1908140]
Hithesh Saikam [Z1937821]

Brief Summary:

NIU Connect developed in 2018 is a social media platform that was created to engage people with each other and help them stay updated about ongoing events and activities happening around them. The app was created to allow users to share their content in the form of pictures, videos, music Etc. Users can also use Hashtags and Location along with their content. Other users can engage with content available on the app by viewing it, liking it and commenting on it.

Users can sign in and create their profiles by providing basic information such as Name, DOB, Email, Bio etc. They can also set a profile picture to their profile. The app will authenticate each user by the username and password provided at the time of Sign In. Every user will have unique Usernames to differentiate among profiles. Users will also have a unique profile page, where their pictures, videos, and content are shared with their followers. This page allows for the content shared by the users to be viewed, liked, or commented on by their followers.

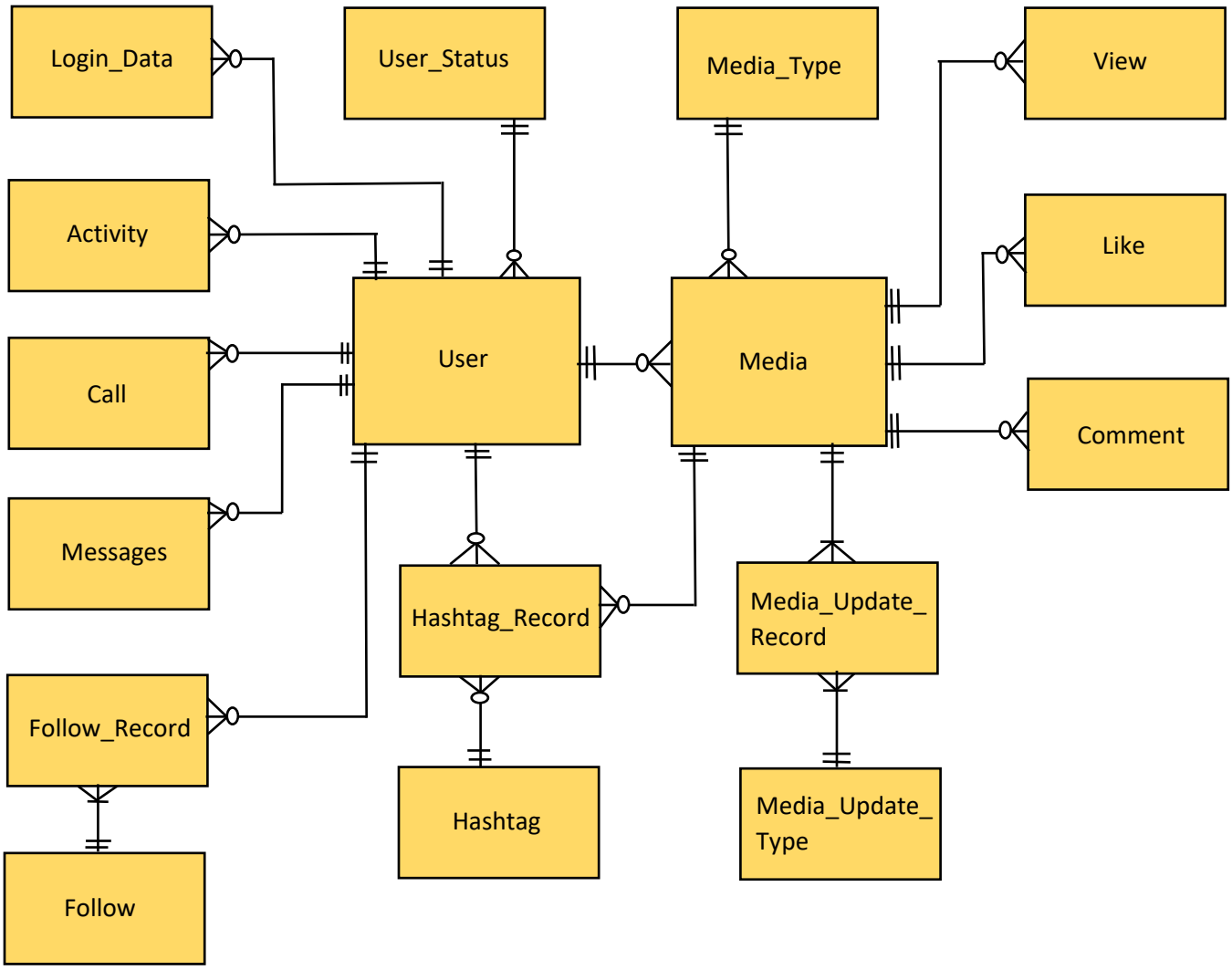
It primarily focuses on video and photography sharing. It allows users to share high quality photography and videos. Users also have the ability to send a private message to their followed profiles and have a chat. Along with messages, users can also video call or audio call another user.

NIU Connect will allow users to follow other users. Following other profiles lets users to view and engage with the content posted by those profiles. As stated in the above paragraph, engagement can be through likes, views and or comments. Each post by a user appears on their followers' feeds and can also be viewed by the public when tagged using Hashtags or Locations.

NIU Connect allows users to edit and upload photos and short videos through a mobile app. Users can add a caption to each of their posts and use hashtags and location-based geotags to index these posts and make them searchable by other users within the app.

During this assignment, we will take a bird's eye view of NIU Connect database. As it flows from the User (User Status/Login/Activity) to their Followers, the type of Posts or Content type they share on their profile, and finally likes and comments.

ERD Diagram:


Login_Data
User_Status
Media_Type
View
Activity
Like
User
Media
Call
Comment
Messages
Hashtag_Record
Media_Update_
Record
Follow_Record
Hashtag
Media_Update_
Type
Follow

Tables in SQL Server Management Studio:

Table Name	Primary_Key	Foreign_Keys
Activity	Activity_ID	UID
Call	Call_ID	Caller, Receiver
Comment	Comment_ID	UID, Media_ID
Follow	F_ID	Follower_ID, Followee_ID
Follow_Record	Follow_Record_ID	UID, F_ID
Hashtag	Hashtag_ID	
Hashtag_Record	Hashtag_Record_ID	UID, Media_ID, Hashtag_ID
Like	Like_ID	UID, Media_ID
Login_Data	Login_ID	UID
Media	Media_ID	Type_ID, UID, Last_Update_ID
Media_Type	Type_ID	
Media_Update_Record	Media_Update_ID	Media_ID, Update_Type_ID
Media_Update_Type	Update_Type_ID	
Messages	Message_ID	Sender, Receipient
User	UID	User_Status
User_Status	Status_ID	
Views	View_ID	UID, Media_ID

Queries:

1. Display user details of users who are active in the system

```
select Fname, Lname, Email, Phone, username
from  [dbo].[User] u inner join [dbo].[User_Status] us
on u.User_status = us.Status_ID
where us.Status_Type like '%Active%'
```

2. Display user details and the number of people they are following in descending order

```
select u.fname as UserName, COUNT(f.FID) as TotalFollowing
from follow f inner join [dbo].[User] u
on f.followerID=u.uid
where status=1
group by u.fname
order by totalfollowing desc
```

3. Displaying the user details of user having highest number of followers

```
select top 1 u.fname as UserName, (count(f.FID)) as HighestFollowers
from follow f inner join [dbo].[User] u
on f.followeeID=u.uid
where status=1
group by u.fname
order by HighestFollowers desc
```

4. Display user details and count of each type of media uploaded by that user.

```
select u.UID, u.fname, u.lname, mt.type, count(m.media_id) as TotalUploads
from [dbo].[User] u, media_type mt, media m
where u.uid = m.uid and m.type_id= mt.type_id
group by u.UID, u.fname, u.lname, mt.type
order by totaluploads desc
```

5. Display UID of users who unfollowed another user and also the user whom they unfollowed

```
select followerID as UID , followeeID as unfollowed
from follow
where FID in (select FID from follow_record where status=0)
```

6. Display all usernames who are active for more than 30 minutes

```
Select u.username, DATEDIFF(MINUTE, a.activity_starttime, a.activity_endtime) as duration
from activity a, [dbo].[User] u
where u.UID= a.UID and DATEDIFF(MINUTE, a.activity_starttime, a.activity_endtime)> 30
order by duration desc
```

7. Display user's media type with highest likes

```
select u.UID, mt.Type, COUNT(l.Like_Id) as highestlikes
from [dbo].[User] u, media_type mt, media m, likes l
where u.UID=m.UID and m.media_Id=l.media_Id and m.type_Id=mt.type_Id
group by u.UID, mt.Type
order by highestlikes desc
```

8. Display username and media type which received highest number of comments

```
select u.username, mt.Type, COUNT(c.comment_ID) as highestcomments
from [dbo].[User] u, media_type mt, media m, comments c
where u.UID=m.UID and m.media_Id=c.media_Id and m.type_Id=mt.type_Id
group by u.username, mt.Type
order by highestcomments desc
```

9. Disliked media

```
select u.UID, m.media
from [dbo].[User] u, media m, likes l
where u.UID=m.UID and m.media_Id=l.media_Id and l.status=0
group by u.UID, m.media
```

10. Display the reels and their viewcount having atleast two views

```
select m.Media as ReelName,  COUNT(*) as NumberofViews
from Views v inner join Media m
on v.Media_ID=m.Media_Id
where m.Type_Id = 3 and m.Status=1
group by m.Media
having COUNT(*) > 2
```

11. Display medianame and musicfilename used for reels and videos

```
select media, Music_Filename
from Media
where Type_Id in (select Type_Id from Media_Type where Type in ('video', 'reel')) and status =1
```

12. Display the media which are deleted or archived

```
select m.Media_ID, mut.Update_Type
from Media m, Media_Update_Type mut, Media_Update_Record mr
where m.Type_Id=mut.Update_Type_ID and m.Media_Id=mr.Media_ID and mut.Update_Type
in ('Deleted', 'Archived')
```

13. Display the call duration for each call in desc order

```
select Call_ID, u.Fname, c.Type, DATEDIFF(MINUTE,start_time, end_time) as
duration_minutes
from calls c inner join [dbo].[User] u
on c.caller = u.UID
order by duration_minutes desc
```

14. Display the count of audio and video calls made by users

```
select u.UID, c.Type, count(c.Call_ID) as TotalCalls, count(m.Message_ID) as TotalMessages
from Calls c, [dbo].[Users] u, Messages m
where c.Caller=u.UID and m.Sender = u.UID
group by u.UID, c.Type
```

15. Displaying the user name, the hash value for the media

```
select m.Media, u.Fname, h.value as Hash_Value
from Media m, [dbo].[User] u, Hashtag_Record r, Hashtag h
where r.Media_ID= m.media_ID and r.UID= u.UID and h.Hashtag_ID = r.Hashtag_ID
```

16. Display the number of messages sent by user in desc order

```
select Sender, COUNT(*) as numberofmessages
from Messages m, [dbo].[User] u
where m.Sender = u.UID
group by Sender
order by numberofmessages desc
```

17. Display the user name, media and caption given at the time of creation by the user.

```
select u.fname, m.media, m.Caption
from [dbo].[User] u, media m, Media_Update_Record mr
where u.UID=m.UID and  mr.Media_ID = m.Media_Id and mr.Update_Type_ID = 1
```

18. Display the username and all devices from which the user logged in to the app.

```
select u.Username, ld.Device
from Login_Data ld,  [dbo].[User] u
where ld.UID = u.UID and ld.Status =1
order by u.username
```

19. Display user name of the user who commented on a post along with the comment.

```
select u.username, m.media, c.text
from [dbo].[Users] u, Media m, Comments c
where u.UID = c.UID and m.Media_Id = c.Media_ID
```

20. Display count of number of uploads of each type with the highest first.

```
select mt.type, COUNT(*) as uploadsofeachtype
from Media m, Media_Type mt
where m.Type_Id = mt.Type_Id
group by mt.Type_Id , mt.Type
order by uploadsofeachtype desc
```