राष्ट्रीय प्रौद्योगिकी संस्थान वारंगल
National Institute of Technology Warangal | Department of Computer Science and Engineering

SOCIAL MEDIA DATABASE MANAGEMENT

A detailed Database Model of a Social Media Platform implemented in the Structured Query Language (SQL).

CHINMAY JOSHI
197222 - SECTION B

DEEKSHITA TIRUMALA
197125 - SECTION A

NANDEPU INDIRA
197155 - SECTION A

Acknowledgement

Inspiration and motivation play a key role in the success of any venture. Successful completion of any project requires proper guidance and help. We express our gratitude to the following people who have made this possible.

- We thank DBMS professors, **Dr. T Ramakrishnudu** and **Dr. RBV Subramanyam** for giving us the opportunity to delve deep into database concepts. Their lessons have been really helpful in learning the fundamental DBMS concepts like relations, E-R diagrams, Structured Query Language, functional dependencies, normalization, indexing, etc. in detail.
- We extend our thanks to the whole CSE Department for being so helpful and available whenever we required any kind of guidance. The DBMS Lab faculty has also trained and made us practice the Structured Query Language with the help of the periodic assignments on Google Classroom.
- The students of CSE batch (II-Year) have been very interactive and healthy class discussions are always a boon for students.
- We thank the NITW management, Dean - Academic and Director - NITW, for giving us this opportunity to discover this subject in the course, and enter new avenues in the field of database management.

We also thank our parents, elders and well-wishers for being there with us and giving us all kinds of technical and moral support.

Regards,

[illegible]
Chinmay Joshi
197222 (Section B)

T. Deekshita
Deekshita Tirumala
197125 (Section A)

N. [illegible]
Nandepu Indira
197155 (Section A)

Table of Contents

Problem Statement

In this project, we have designed a social media management system to let users connect with each other. A social media model involves the management of big data, regarding the user profiles, the communication among them, broadcast posting and other features such as recommendation of users and posts. In our project we have implemented these concepts using SQL-based constructs. The important functionalities have been implemented using PL/SQL procedures. The various functional dependencies of the relations have been analyzed and the tables have been normalized for optimum performance.

The database contains important information about the users and other users will be able to make friends online and also chat with each other. In this model we are implementing a messaging system that can handle both private messaging as well as group messaging. Users can post whatever they want to share, which can be accessed publicly. The database management system inherits properties of both private messaging apps like WhatsApp as well as public broadcast media such as Instagram, Facebook, etc.

Users can like and comment on each other's posts and create an interactive environment where they can make many friends. The users receive customized recommended content as per their most frequently liked and commented topics. The users can post stories too, which are basically short duration posts. Additionally, there is an option to mention their friends in stories and post hashtags which their posts fall under. The users can make a very detailed profile and their connections are also classified based on the relationship they hold with others - such as work, alumni, family, friend, partner, etc. Users with mutual friends will be suggested to each other.

There are 13 relations in total, in this social media database model. The brief explanation of these relations, their structure as well as data are given below.

1. Users

The users relation is the collection of all the profile data of the users of this social media model. This profile data includes name, date of birth, user ID, password, account type, etc. This is somewhat like a reference relation to many other relations in this model.

Attribute	Data Type	Constraints
UserID	NUMBER	PRIMARY KEY
UserName	VARCHAR(30)	
FirstName	VARCHAR(20)	
LastName	VARCHAR(20)	
EmailID	VARCHAR(30)	
DOB	DATE	
DOJ	DATE	
Bio	VARCHAR(100)	
Password		
AccountType	VARCHAR(30)	CHECK(AccountType IN ('Business','Personal'))
LastSeen	TIMESTAMP	
StoryID	NUMBER	UNIQUE NOT NULL

UserID	UserName	FirstName	LastName	EmailID	DOB	DOJ
1201	rosie123	Rosie	Thomas	rosie123@gmail.com	12-Jun-94	15-Feb-09
1202	jaydie156	Judy	Rhodes	judyiscute@hotmail.com	13-Sep-96	18-Jul-11
1203	averyfatso	Aviral	Singh	realaviral@aol.com	11-May-98	11-Oct-09
1204	sweety	Sweety	Sharma	sweetgal@gmail.com	29-Dec-95	6-Jan-11
1205	honeyhan	Haniya	Ahmed	honeyhan100@gmail.com	21-Mar-97	4-Apr-12
1206	martha_xoxo	Martha	Jacob	marthaxoxo@aol.com	8-Jun-98	9-Aug-08
1207	averitas	Avery	Luther	averitas@outlook.com	2-Apr-96	1-Sep-08
1208	raziaaaa	Razia	Sultana	raziaaa01@gmail.com	5-Aug-99	18-Feb-12
1209	suzieeeee1234	Suzie	Everton	suzfuz@aol.com	13-May-96	12-Jul-12
1210	beingron	Ron	Peter	thatguyron@hotmail.com	6-Mar-98	17-Sep-09
1211	harry3101	Harmanpreet	Kaur	thepunjabikudi@aol.com	31-Jan-96	14-Dec-08
1212	shelly_13	Shaurya	Trivedi	kingshaurya@outlook.com	13-Jan-97	26-Feb-09

Bio	Password	AccountType	LastSeen	StoryID
Classy, bossy and sassy :D	thorosie987	Personal	26-MAR-21 08.51.16.78 AM	50209
Living like a princess!	jd*$1$5$6	Personal	30-MAR-21 07.48.16.38 AM	50210
Defining life in a word: Food	awaryfat171	Personal	30-MAR-21 11.23.18.61 AM	50211
Keepin it real since 95	99s$wee$ty$s99	Business	28-MAR-21 11.41.43.76 AM	50212
Pizza lover	##**ahhhhmed**##	Personal	01-APR-21 05.21.08.64 AM	50213
Vibing on tea	*m@onnneymar11tha	Business	01-APR-21 06.31.16.28 AM	50214
Fashion designing is fun!	*awaryloother1&	Personal	28-MAR-21 10.57.34.66 AM	50215
Loving basketball	rsauzlitaana	Personal	24-MAR-21 09.22.43.54 AM	50216
To the reader, mgbu with more work	seuvzeriteon	Personal	29-MAR-21 06.50.11.38 AM	50217
Swimming is my passion	ronipter4738	Business	30-MAR-21 01.15.37.67 AM	50218
Brown kudi :)	ruakhar9292	Personal	30-MAR-21 02.21.11.59 AM	50219
Fitness addict	*34shoreyeah43*	Personal	29-MAR-21 02.41.12.43 AM	50220

2. Friends

The Friends relation is the collection of all the friendships between User X and User Y of this social media model. This is best implemented by using two ID attributes - one for source and the other for destination. Each record in this relation is unique i.e. if 12aa - 12bb are in one record, then 12bb - 12aa will not be again inserted because both of them represent the same pair.

Attribute	Data Type	Constraints
SourceUserID	NUMBER	PRIMARY KEY, FOREIGN KEY
DestUserID	NUMBER	PRIMARY KEY, FOREIGN KEY
DOC	DATE	
Relationship	VARCHAR(30)	CHECK (Relationship IN ('Family','Educational','Colleague'))

SourceUserID	DestUserID	DOC	Relationship
1201	1202	27-Oct-19	Colleague
1201	1203	4-Oct-19	Colleague
1201	1204	18-Dec-18	Family
1201	1205	17-Dec-18	Colleague
1201	1206	18-Nov-18	Educational
1202	1209	4-Oct-18	Family
1202	1210	5-Dec-16	Educational
1202	1211	2-Nov-17	Colleague
1202	1212	12-Nov-17	Family
1203	1204	1-Sep-18	Educational
1203	1205	15-Nov-16	Colleague
1203	1210	3-May-17	Educational
1204	1207	5-Jun-16	Family
1204	1208	9-Jul-18	Educational
1204	1209	6-Apr-15	Educational
1204	1212	3-Mar-14	Family
1205	1206	10-Jan-16	Educational
1205	1207	18-Jun-16	Colleague
1205	1212	16-Sep-18	Colleague
1206	1207	26-Nov-14	Educational
1206	1208	9-May-16	Educational
1207	1208	8-Jul-14	Family
1208	1209	17-Nov-16	Educational
1209	1210	10-Jun-14	Educational
1210	1212	4-Feb-17	Colleague
1211	1212	8-Oct-19	Family

3. Stories

The Stories relation shows all the stories shared by users and each story has a story ID which will be displayed in this relation. Stories are basically short-duration (24-hr) status updates by users to share some interesting stuff whenever they come across it. We can also see how many views the story has received and what time the story was shared.

Attribute	Data Type	Constraints
StoryID	NUMBER	PRIMARY KEY, FOREIGN KEY
Views	NUMBER	
StoryTime	TIMESTAMP	

StoryID	Views	StoryTime
50209	23	30-MAR-21 04.43.09.24 AM
50210	43	30-MAR-21 11.01.55.55 AM
50211	68	30-MAR-21 02.49.41.76 AM
50212	12	30-MAR-21 09.06.14.54 AM
50213	54	30-MAR-21 10.35.06.32 AM
50214	26	30-MAR-21 07.41.05.24 AM
50215	76	30-MAR-21 08.17.09.11 AM
50216	33	30-MAR-21 11.33.16.08 AM
50217	39	30-MAR-21 07.58.08.88 AM
50218	49	30-MAR-21 11.14.49.43 AM
50219	31	30-MAR-21 08.27.05.29 AM
50220	61	30-MAR-21 04.55.08.19 AM

4. Mentions

The Mentions relation displays all the mentions done by people on the social media model. Mentions are actually usernames of other people, who are mentioned on the stories of users. So it can be used to tag friends and family when a memorable moment is to be shared. The Mentions relations stores the mention ID and also the ID of the story in which the mentioning has occurred.

Attribute	Data Type	Constraints
StoryID	NUMBER	PRIMARY KEY, FOREIGN KEY
MentionID	NUMBER	PRIMARY KEY, FOREIGN KEY

StoryID	MentionID
50211	1201
50212	1206
50214	1212
50218	1204
50219	1203

5. DirectMessage

The DirectMessage relation contains all the messages that are communicated from any User A to any User B of the Social Media model. There are two ID attributes referenced from the Users table to represent the source and destination of the message. The ID values along with timestamp of the message uniquely identify each record of the relation. The message contents are represented by its type and data. If the message is of file type, we store its extension in type and path to the file in data.

Attribute	Data Type	Constraints
SourceUserID	NUMBER	PRIMARY KEY, FOREIGN KEY
DestUserID	NUMBER	PRIMARY KEY, FOREIGN KEY
MsgTime	TIMESTAMP	PRIMARY KEY
MsgData	VARCHAR(100)	
MsgType	VARCHAR(30)	

SourceUserID	DestUserID	MsgTime	MsgData	MsgType
1201	1202	12-MAR-12 04.29.18.73 AM	HEY	text
1201	1202	13-MAR-12 04.29.18.73 AM	See you at 5.00 pm	text
1202	1201	14-OCT-13 05.19.48.73 AM	https://newspaper.html	.html
1201	1203	15-SEP-15 06.29.38.73 AM	How much is the fee?	text
1204	1205	16-DEC-14 07.29.28.73 AM	/files/info.png	.png
1205	1204	17-MAR-17 08.24.18.73 AM	/folder/assignment.pdf	.pdf
1205	1202	18-JAN-18 09.29.08.73 AM	Happy Birthday!	text
1204	1203	19-MAR-12 04.39.58.73 AM	/files/song.mp3	.mp3
1207	1212	13-FEB-18 06.27.38.73 AM	Goodbye	text
1209	1210	16-AUG-19 07.09.15.73 AM	/files/stars.png	.png

6. Groups

The Groups relation is a collection of all the existing groups in the Social Media model. Each group is uniquely identified by its GroupID and has properties of GroupName, its description and time of its creation. Each group also has a composite attribute for its group members which is represented by a separate relation. Groups is also referenced by GroupMessage relation representing the messages of the group chat.

Attribute	Data Type	Constraints
GroupID	NUMBER	PRIMARY KEY
GroupName	VARCHAR(30)	
Description	VARCHAR(100)	
CreationTime	TIMESTAMP	

GroupID	GroupName	Description	CreationTime
78545	College Friends	To share memories and casual chat	26-JUN-12 09.39.16.78 AM
78546	Carpool for Work	Driver no.: 1234 Pickup time: 8.00 am	16-SEP-12 12.19.17.98 PM
78547	Presentation Team	Presentation on traffic data. Deadline tomorrow	07-OCT-14 03.23.08.65 AM
78548	Family Group	Peace, love and happiness	29-MAR-12 07.29.14.72 AM
78549	Painting Club	Paint away the world!	12-FEB-12 04.39.18.73 AM

7. GroupMembers

GroupMembers relation stores data of the members who are part of any groups of the social media. Each member is uniquely identified by their UserID and the name of the group that they belong to. In addition each user can have their own username within a group and can be either a member or admin of the group. Their date of joining the group is also recorded. A user can be part of multiple groups.

Attribute	Data Type	Constraints
UserID	NUMBER	PRIMARY KEY, FOREIGN KEY
GroupName	VARCHAR(30)	PRIMARY KEY
UserName	VARCHAR(30)	
Privilege	VARCHAR(30)	CHECK (Privilege IN ('admin','member'))
DOJ	DATE	

UserID	GroupName	UserName	Privilege	DOJ
1201	College Friends	rosie	member	12-Apr-14
1202	College Friends	lastjedi	admin	13-Apr-15
1203	College Friends	avery	member	14-Oct-19
1204	College Friends	sweety	member	15-Apr-13
1205	Painting Club	han	admin	16-Nov-15
1206	Painting Club	martha	member	1-Apr-12
1203	Presentation Team	avery	admin	17-Dec-17
1208	Presentation Team	rosie	member	2-Apr-16
1209	Presentation Team	suzie	member	17-Mar-18
1210	Carpool for Work	ron	admin	3-Aug-13
1211	Carpool for Work	harry	member	11-May-14
1212	Family Group	shelly	member	10-Oct-17

8. GroupMessage

The GroupMessage relation shows which user has sent what message, and in which group.

Attribute	Data Type	Constraints
GroupID	NUMBER	PRIMARY KEY, FOREIGN KEY
UserID	NUMBER	PRIMARY KEY, FOREIGN KEY
MsgData	VARCHAR(100)	
MsgType	VARCHAR(30)	
MsgTime	TIMESTAMP	PRIMARY KEY

GroupID	UserID	MsgData	MsgType	MsgTime
78545	1201	Hello	text	12-FEB-12 04.39.18.73 AM
78546	1205	/files/beach.jpeg	.jpeg	22-SEP-17 04.38.18.73 AM
78546	1206	This is the plan	text	02-FEB-12 04.37.18.73AM
78547	1203	Here is the file	text	17-FEB-13 04.39.18.73 AM
78547	1208	Agenda/speech.mp3	.mp3	12-DEC-12 04.59.18.73 AM
78547	1209	Thank you	text	18-FEB-18 04.37.18.73 AM
78549	1210	Send the details	text	12-MAR-12 04.29.18.73 AM
78549	1211	folder/rules.txt	.txt	13-FEB-14 04.49.18.73 AM

9. TopicTag

The TopicTag relation consists of all the hashtags trending in the social media model

Attribute	Data Type	Constraints
TagID	NUMBER	PRIMARY KEY
TagName	VARCHAR(30)	

TagID	TagName
7653	CaptureIt
7654	Trendz
7655	Vibing
7656	Inspire
7657	Art_fo_life
7658	foodiez
7659	OceanWaves
7660	chill
7661	EDMLove
7662	HipHop
7663	Assignment
7664	YOLO
7665	Engineering
7666	Cheems
7667	meme

10. Posts

The Posts relation stores all the posts ,being posted by the users. It gives complete data of a post like time of post ,like and comment count too. This is best implemented by using two ID attributes - one for the user posting and the other for the post .

Attribute	Data Type	Constraints
PostID	NUMBER	PRIMARY KEY
UserID	NUMBER	FOREIGN KEY
Captions	VARCHAR(100)	
PostData	VARCHAR(100)	
PostTagID	NUMBER	FOREIGN KEY
PostTime	TIMESTAMP	
LikesNo	NUMBER	
CommentsNo	NUMBER	

PostID	UserID	Captions	PostData	PostTagID	PostTime	LikesNo	CommentsNo
293434	1206	Sea's swayin'...	/files/videos/20190127_042010.mp4	7659	27-JAN-19 04.20.10.01	1	3
293435	1204	I'm a Bachelors of Memology(B.Meme) graduate.	/files/images/20200122_071011.jpg	7667	22-JAN-20 07.10.10.51	1	1
293436	1209	To live is to Inspire	/files/mpeg-4/20190205_014517.m4a	7656	05-FEB-19 01.45.16.31	0	3
293437	1211	Dance to death...	/files/videos/20170407_062539.mp4	7662	07-APR-17 06.25.38.58	1	1
293438	1208	Expressin heart ...with art!	/files/images/20180311_071517.jpeg	7657	11-MAR-18 07.15.17.05	3	2
293439	1203	Plate was Full...now am Filled...Plate is empty	/files/images/20170812_054511.jpg	7658	12-AUG-17 05.45.10.49	2	1
293440	1208	Blur's beauty in times...focus ain't mandated	/files/images/20200628_080253.jpg	7653	28-JUN-20 08.02.52.13	2	0
293441	1211	You Only Live Once...Damn!	/files/videos/20171106_013112.mp4	7664	06-NOV-17 01.31.11.01	2	1
293442	1204	I type m instead n...am i the Cheems??	/files/videos/20190423_091623.mp4	7666	23-APR-19 09.16.23.02	3	1
293443	1202	Century!!...	.files/images/20181009_060916.jpeg	7654	09-OCT-18 06.09.16.01	1	3
293444	1205	Assignments->to do list...Procatination->In progress list	/files/images/20200709_064018.jpg	7663	09-JUL-20 06.40.17.41	0	3
293445	1212	had enough of life...*Slurps*	/files/images/20210224_031238.jpg	7658	24-FEB-21 03.12.38.07	2	2
293446	1210	rock it YOLO	/files/mpeg-4/20161229_080027.m4a	7664	29-DEC-16 08.00.27.15	2	1
293447	1201	Million days with terrace air	/files/images/20190605_060353.jpg	7660	05-JUN-19 06.03.53.21	1	0
293448	1202	keep inspiring...	/files/videos/20190922_014546.mp4	7656	22-SEP-19 01.45.46.18	1	0
293449	1206	Smile in pain...	/files/images/20181020_094612.jpg	7665	20-OCT-18 09.46.11.41	1	0
293450	1212	Damn....just EDM's are Enough	/files/music/20201127_111158.mp3	7661	27-NOV-20 11.11.58.01	4	1
293451	1207	Trop tard, Trop tard...	/files/videos/201903_021446.mp4	7662	25-MAR-17 02.14.46.32	0	1
293452	1205	Wise men say...only Fools rush in	/files/music/20180814_120022.mp3	7655	14-AUG-18 12.00.22.09	0	0

11. Likes

The Likes relation records the likes on all the posts by users of this social media model. This is best implemented by using two ID attributes - one for the post and the other for the user who liked the post. Each record in this relation is unique not just by candidate keys but the timestamp of like time too.

Attribute	Data Type	Constraints
PostID	NUMBER	PRIMARY KEY, FOREIGN KEY
UserID	NUMBER	PRIMARY KEY, FOREIGN KEY
LikeTime	TIMESTAMP	

PostID	UserID	LikeTime
293441	1208	25-FEB-18 11.25.44.12 AM
293438	1205	07-APR-18 11.45.46.34 AM
293450	1203	07-NOV-21 12.47.52.52 PM
293450	1201	15-AUG-21 10.26.24.24 AM
293440	1208	08-JAN-21 06.42.15.16 AM
293450	1210	02-AUG-21 02.53.24.26 AM
293450	1207	25-FEB-21 05.57.24.43 AM
293439	1201	01-APR-18 11.08.36.51 AM
293440	1203	05-JAN-21 06.49.27.15 AM
293438	1204	09-NOV-18 08.55.52.00 AM
293446	1201	19-SEP-17 07.23.11.46 AM
293439	1211	11-JUL-18 03.57.43.45 AM
293442	1208	23-MAR-20 10.38.54.13 AM
293446	1210	04-AUG-17 02.56.32.16 AM
293434	1208	20-MAY-19 12.14.57.43 PM
293441	1204	11-APR-18 12.52.05.41 PM
293437	1211	20-JUN-17 12.57.59.15 PM
293442	1205	23-SEP-19 05.01.00.35 AM
293449	1210	20-MAR-19 03.55.43.16 AM
293448	1206	10-JUN-20 11.19.27.53 AM
293447	1207	26-SEP-19 11.07.05.59 AM
293445	1203	09-FEB-22 04.57.08.43 AM
293445	1209	20-JUN-21 03.09.43.16 AM
293442	1207	03-JAN-20 09.14.12.13 AM
293438	1211	24-JUN-18 11.42.56.12 AM
293443	1208	11-FEB-19 09.39.52.14 AM
293435	1207	10-OCT-20 06.50.53.41 AM

12. Comments

The Comments relation stores comments from users on all the posts from the users of this social media model. This is best implemented by using two ID attributes - one for the post and the other for the user who commented on the post. Each record in this relation is unique not just by candidate key pair but the timestamp of comment time too.

Attribute	Data Type	Constraints
PostID	NUMBER	PRIMARY KEY, FOREIGN KEY
UserID	NUMBER	PRIMARY KEY, FOREIGN KEY
CmtData	VARCHAR(100)	
CmtTime	TIMESTAMP	

PostID	UserID	CmtData	CmtTime
293444	1201	Go strt ur Assigns now...	15-JUL-20 02.44.30.43 AM
293441	1209	tht's the coolest thing i've heard today	28-DEC-17 05.04.16.16 AM
293451	1203	litt.....what's the song btw??	10-MAY-17 12.02.11.23 PM
293445	1206	our way ...the foodiez way!!	28-FEB-21 10.07.27.00 AM
293434	1211	the sunset + oceans..the timing	17-FEB-19 06.44.30.09 AM
293434	1203	Beauty...thts how we define it!	03-FEB-19 12.02.33.08 PM
293436	1202	Made my day <3	28-MAR-19 11.23.58.01 AM
293443	1203	in petrol pump tho	05-NOV-18 09.49.38.59 AM
293435	1202	damn...keep going...	25-JAN-20 08.38.09.22 AM
293439	1203	lurin us ehhh??	14-SEP-17 05.36.38.18 AM
293450	1205	ikr...damn clarity X your name..	25-JAN-21 08.04.02.57 AM
293443	1208	true..not in stadium tho	13-OCT-18 06.29.56.48 AM
293442	1207	RIP cheems...	11-MAY-19 12.09.41.14 PM
293446	1201	wohww..the perfect song with perfect caption!!	02-MAR-17 02.47.15.51 AM
293438	1205	seriously!!?	13-MAY-18 12.17.15.52 PM
293437	1206	u've got tht grace...thts litt	26-APR-17 05.32.51.26 AM
293445	1201	uhmm uhmm... *even longer slurp*	24-FEB-21 05.28.47.53 AM
293434	1201	aye...swayin'<3	05-MAR-19 04.24.52.51 AM
293444	1208	bro!..i topped the procastination test	19-JUL-20 05.20.46.45 AM
293436	1209	thts great!	06-FEB-19 11.34.22.01 AM
293438	1210	killerrrr!!	20-APR-18 04.53.05.05 AM
293443	1202	aye..	22-OCT-18 12.42.52.50 PM
293436	1206	okkeiii..	15-MAR-19 10.49.18.03 AM
293444	1209	smile in pain...	17-JUL-20 09.57.58.06 AM

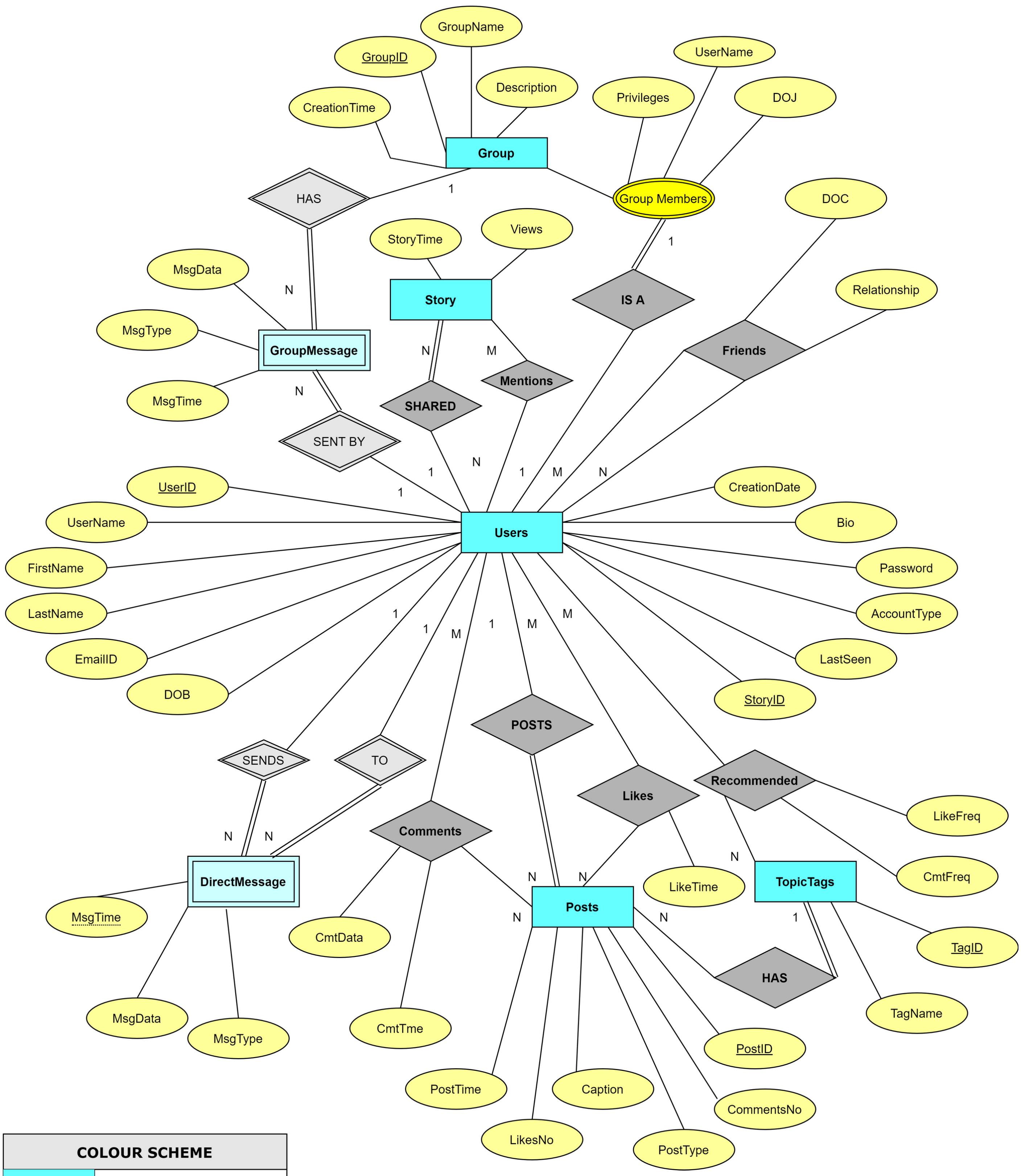
13. Recommended

The Recommended relation stores recommendations based on topic tags frequently liked or commented by a user. This is best implemented by taking top 3 tags that we get from the various tables and a PL/SQL stored procedure.

Attribute	Data Type	Constraints
TagID	NUMBER	PRIMARY KEY, FOREIGN KEY
UserID	NUMBER	PRIMARY KEY, FOREIGN KEY
LikeFreq	NUMBER	
CmtFreq	NUMBER	

TagID	UserID	LikeFreq	CmtFreq
7654	1201	1	1
7653	1201	1	0
7666	1201	1	0
7654	1202	1	1
7653	1202	1	0
7666	1202	1	0
7654	1203	1	1
7653	1203	1	0
7666	1203	1	0
7654	1204	1	1
7653	1204	1	0
7666	1204	1	0
7654	1205	1	1
7653	1205	1	0
7666	1205	1	0
7654	1206	1	1
7653	1206	1	0
7666	1206	1	0
7654	1207	1	1
7653	1207	1	0
7666	1207	1	0
7654	1208	1	1
7653	1208	1	0
7666	1208	1	0
7654	1209	1	1
7653	1209	1	0
7666	1209	1	0
7654	1210	1	1
7653	1210	1	0
7666	1210	1	0
7654	1211	1	1
7653	1211	1	0
7666	1211	1	0

Entity-Relationship (E-R) Diagram



COLOUR SCHEME	
🟦	STRONG ENTITY
🟦	WEAK ENTITY
🟨	ATTRIBUTE
⬜	WEAK ENTITY RELATIONSHIP
⬛	RELATION
🟨	MULTIVALUED ATTRIBUTE

Description of the E-R Diagram

The Social Media Database system broadly consists of 13 relations in total, out of which five are Strong entities, two are Weak entities, one is a multivalued attribute expressed as an entity and remaining five are relationships.

Strong Entities: Group, Story, Posts, TopicTag & Users
Weak Entities: DirectMessage & GroupMessage
Relationship Entities: Likes, Comments, Recommended, Mentions & Friends
Multivalued Attribute: GroupMembers

Assumptions, Cardinality and Participation in Relationships:

Users-POSTS-Posts: A given user can post multiple posts, however a given post can only belong to a single user. Therefore, the cardinality is 1:N. Every user is not required to have posted something, however every post must belong to some user. Therefore, there is partial participation of Users and total participation of posts.

Users-COMMENTS-Posts: A given user can comment on multiple posts, and a given post can receive comments from multiple users. Therefore, the cardinality is M:N. Every user is not required to have commented something, and every post need not receive a comment from a user. Therefore, there is partial participation on both sides.

Users-LIKES-Posts: A given user can like multiple posts, and a given post can receive likes from multiple users. Therefore, the cardinality is M:N. Every user is not required to have liked something, and every post need not receive a like from a user. Therefore, there is partial participation on both sides.

Users-SENDS-DirectMessage: A given user can send multiple messages, however a given message can only belong to a single user. Therefore, the cardinality is 1:N. Every user is not required to have messaged something, however every message must belong to some user. Therefore, there is partial participation of Users and total participation of DirectMessage.

DirectMessage-TO-Users: A given user can receive multiple messages, however a given message can only be sent to a single user. Therefore, the cardinality is N:1. Every user is not required to have received a message, however every message must be sent to some user. Therefore, there is partial participation of Users and total participation of DirectMessage.

Posts-HAS-TopicTags: A given post can have a single topictag, however a given topictag can belong to multiple posts. Therefore, the cardinality is N:1. Every post is required to have a topic, however every topictag need not have a post on it. Therefore, there is partial participation of TopicTags and total participation of Posts.

GroupMessage-SENT BY-Users: A given user can post multiple messages on a group, however a given message can only belong to a single user. Therefore, the cardinality is N:1. Every user is not required to have messaged something, however every message must belong to some user. Therefore, there is partial participation of Users and total participation of GroupMessage.

Group-HAS-GroupMessage: A given group can have multiple messages, however a given message can only belong to a single group. Therefore, the cardinality is 1:N. Every group is not required to have some message, however every message must belong to some group. Therefore, there is partial participation of group and total participation of GroupMessage.

Users-SHARED-Story: A given user can post multiple stories, however a given story can only belong to a single user. Therefore, the cardinality is 1:N. Every user is not required to have posted some story, however every story must belong to some user. Therefore, there is partial participation of Users and total participation of Story.

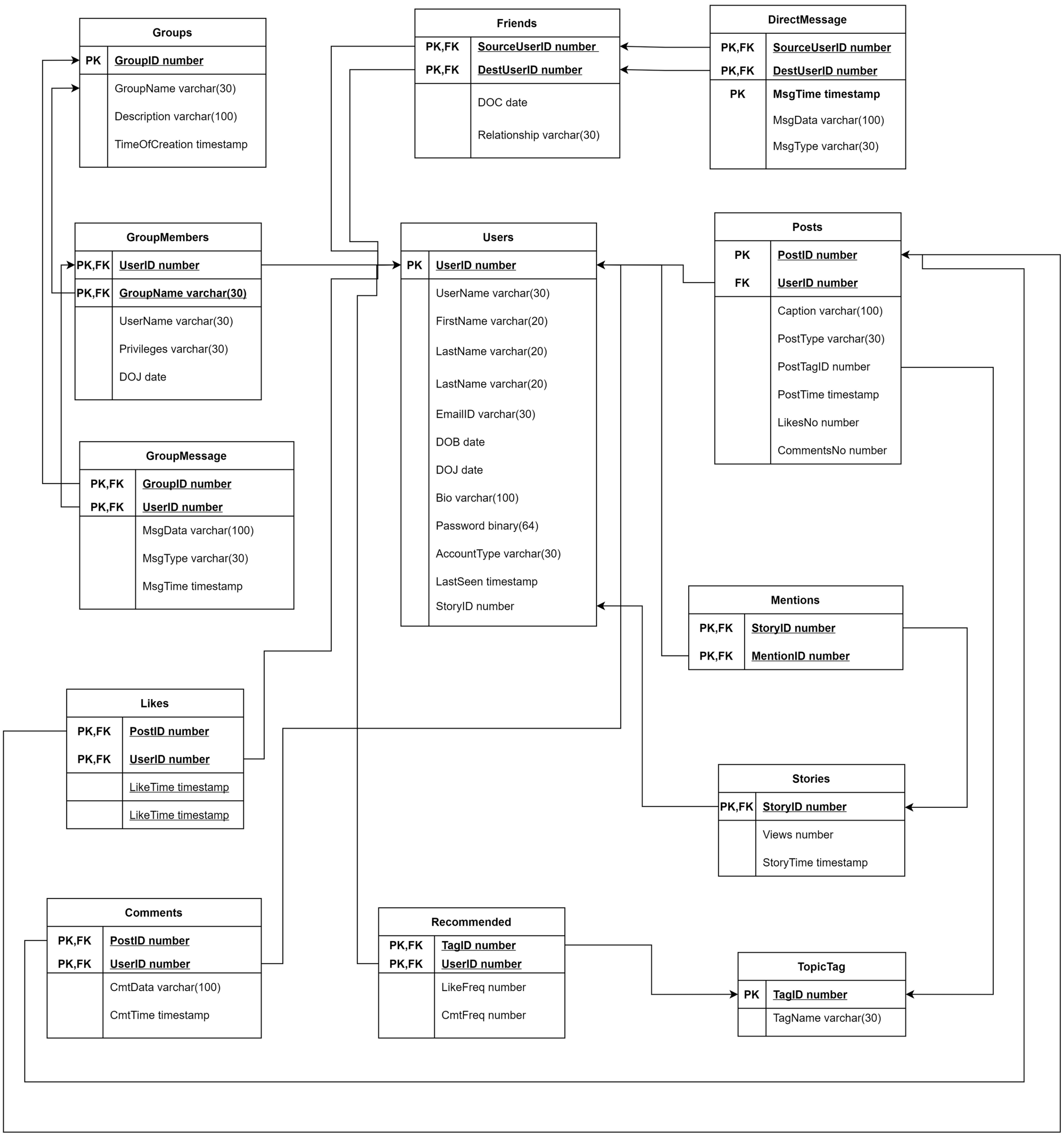
Story-Mentions-Users: A given user can be mentioned on multiple stories, and a given story can mention multiple users. Therefore, the cardinality is M:N. Every user is not required to have been mentioned, and every story need not mention some user. Therefore, there is partial participation on both sides.

Users-Friends-Users: A given user can have multiple friends, and a given user can be friend of multiple users. Therefore, the cardinality is M:N. Every user is not required to have a friend or be some user's friend. Therefore, there is partial participation on both sides.

GroupMembers-IS A-Users: A given groupmember corresponds to a single user and vice versa. Therefore, the cardinality is 1:1. A given groupmember is part of the users entity, however not every user has to be member of some group. Therefore, there is partial participation of Users and total participation of GroupMember.

Users-Recommended-TopicTag: A given user can be recommended multiple topics, and a given topic can be recommended to multiple users. Therefore, the cardinality is M:N. Every user is not required to be recommended something, and every topic need not be recommended to some user. Therefore, there is partial participation on both sides.

Relational Model



1. Creation of Relations

```
CREATE TABLE Users(
	UserID NUMBER PRIMARY KEY,
	UserName VARCHAR(30),
	FirstName VARCHAR(20),
	LastName VARCHAR(20),
	EmailID VARCHAR(30),
	DOB DATE,
	DOJ DATE,
	Bio VARCHAR(100),
	Password VARCHAR(50),
	AccountType VARCHAR(30),
	LastSeen TIMESTAMP,
	StoryID NUMBER UNIQUE NOT NULL,
	CONSTRAINT account_type CHECK(AccountType in ('Business','Personal'))
);

CREATE TABLE Friends(
	SourceUserID NUMBER,
	DestUserID NUMBER,
	DOC DATE,
	Relationship VARCHAR(30),
	PRIMARY KEY(SourceUserID,DestUserID),
	FOREIGN KEY(SourceUserID) REFERENCES Users(UserID),
	FOREIGN KEY(DestUserID) REFERENCES Users(UserID),
	CONSTRAINT friend_rel_type
	CHECK ( Relationship in ('Family','Educational','Colleague'))
);

CREATE TABLE Stories(
	StoryID NUMBER PRIMARY KEY,
	Views NUMBER,
	StoryType TIMESTAMP,
	FOREIGN KEY(StoryID) REFERENCES Users(StoryID)
);

CREATE TABLE Mentions(
	StoryID NUMBER,
	MentionID NUMBER,
	PRIMARY KEY(StoryID,MentionID),
	FOREIGN KEY(StoryID) REFERENCES Stories(StoryID),
	FOREIGN KEY(MentionID) REFERENCES Users(UserID)
);
```

```
CREATE TABLE DirectMessage(
	SourceUserID NUMBER,
	DestUserID NUMBER,
	MsgTime TIMESTAMP,
	MsgData VARCHAR(100),
	MsgType VARCHAR(30),
	PRIMARY KEY(SourceUserID,DestUserID,MsgTime),
	FOREIGN KEY(SourceUserID) REFERENCES USERS(UserID),
	FOREIGN KEY(DestUserID) REFERENCES USERS(UserID)
);

CREATE TABLE Groups(
	GroupID NUMBER PRIMARY KEY,
	GroupName VARCHAR(30),
	Description VARCHAR(100),
	CreationTime TIMESTAMP
);

CREATE TABLE GroupMessage(
	GroupID NUMBER,
	UserID NUMBER,
	MsgData VARCHAR(100),
	MsgType VARCHAR(30),
	MsgTime TIMESTAMP,
	PRIMARY KEY (GroupID,UserID,MsgTime),
	FOREIGN KEY (GroupID) REFERENCES GROUPS(GroupID),
	FOREIGN KEY (UserID) REFERENCES USERS(UserID)
);

CREATE TABLE GroupMembers(
	UserID NUMBER,
	GroupName VARCHAR(30),
	UserName VARCHAR(30),
	Privilege VARCHAR(30),
	DOJ DATE,
	PRIMARY KEY(GroupName,UserID),
	FOREIGN KEY(UserID) REFERENCES USERS(UserID),
	CHECK ( Privilege in ('admin','member'))
);

CREATE TABLE TopicTag(
	TagID NUMBER PRIMARY KEY,
	TagName VARCHAR(30)
);
```

```
CREATE TABLE Posts(
    PostID NUMBER PRIMARY KEY,
    UserID NUMBER,
    Captions VARCHAR(100),
    PostData VARCHAR(100),
    PostTagID NUMBER,
    PostTime TIMESTAMP,
    LikesNo NUMBER,
    CommentsNo NUMBER,
    FOREIGN KEY (UserId) REFERENCES Users(UserID),
    FOREIGN KEY (PostTagID) REFERENCES TopicTag(TagID)
);

CREATE TABLE Likes(
    PostID NUMBER,
    UserID NUMBER,
    LikeTime TIMESTAMP,
    PRIMARY KEY (PostID,UserID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    FOREIGN KEY (PostID) REFERENCES Posts(PostID)
);

CREATE TABLE Comments(
    PostID NUMBER,
    UserID NUMBER,
    CmtData VARCHAR(100),
    CmtTime TIMESTAMP,
    PRIMARY KEY (PostID,UserID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    FOREIGN KEY (PostID) REFERENCES Posts(PostID)
);

CREATE TABLE Recommended(
    TagID NUMBER,
    UserId NUMBER,
    LikeFreq NUMBER,
    CmtFreq NUMBER,
    PRIMARY KEY (TagID,UserID),
    FOREIGN KEY (TagID) REFERENCES TopicTag(TagID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);
```

2. Insertion of Values

```
INSERT INTO Users VALUES
(1201,'rosie123','Rosie','Thomas','rosie123@gmail.com','12-JUN-1994','15-FEB-2009',
'Classy, bossy and sassy :D','thorosie987','Personal','26-MAR-21 08:51:16.78',50209);
INSERT INTO Users VALUES
(1202,'jaydie156','Judy','Rhodes','judyiscute@hotmail.com','13-SEP-1996','18-JUL-2011',
'Living like a princess!','jd*$1$5$6','Personal','30-MAR-21 07:48:16.38',50210);
INSERT INTO Users VALUES
(1203,'averyfatso','Aviral','Singh','realaviral@aol.com','11-MAY-1198','11-OCT-2009',
'Defining life in a word: Food','awaryfat171','Personal','30-MAR-21 11:23:18.61',50211);
INSERT INTO Users VALUES
(1204,'sweety','Sweety','Sharma','sweetgal@gmail.com','29-DEC-1995','06-JAN-2011',
'Keepin it real since 95','99s$wee$ty$s99','Business','28-MAR-21 11:41:43.76',50212);
INSERT INTO Users VALUES
(1205,'honeyhan','Haniya','Ahmed','honeyhan100@gmail.com','21-MAR-1997','04-APR-2012',
'Pizza lover','##**ahhhhmed**##','Personal','01-APR-21 05:21:08.64',50213);
INSERT INTO Users VALUES
(1206,'martha_xoxo','Martha','Jacob','marthaxoxo@aol.com','08-JUN-1998','09-AUG-2008',
'Vibing on tea','*m@onnneymar11tha','Business','01-APR-21 06:31:16.28',50214);
INSERT INTO Users VALUES
(1207,'averitas','Avery','Luther','averitas@outlook.com','02-APR-1996','01-SEP-2008',
'Fashion designing is fun!','*awaryloother1&','Personal','28-MAR-21 10:57:34.66',50215);
INSERT INTO Users VALUES
(1208,'raziaaaa','Razia','Sultana','raziaaa01@gmail.com','05-AUG-1999','18-FEB-2012',
'Loving basketball','rsauzlitaana','Personal','24-MAR-21 09:22:43.54',50216);
INSERT INTO Users VALUES
(1209,'suzieeeee1234','Suzie','Everton','suzfuz@aol.com','13-MAY-1996','12-JUL-2012',
'To the reader, mgbu with more work','seuvzeriteon','Personal','29-MAR-21 06:50:11.38',50217);
INSERT INTO Users VALUES
(1210,'beingron','Ron','Peter','thatguyron@hotmail.com','06-MAR-1998','17-SEP-2009',
'Swimming is my passion','ronipter4738','Business','30-MAR-21 01:15:37.67',50218);
INSERT INTO Users VALUES
(1211,'harry3101','Harmanpreet','Kaur','thepunjabikudi@aol.com','31-JAN-1996','14-DEC-2008',
'Brown kudi :)','ruakhar9292','Personal','30-MAR-21 02:21:11.59',50219);
INSERT INTO Users VALUES
(1212,'shelly_13','Shaurya','Trivedi','kingshaurya@outlook.com','13-JAN-1997','26-FEB-2009',
'Fitness addict','*34shoreyeah43*','Personal','29-MAR-21 02:41:12.43',50220);
```

```
INSERT INTO Friends VALUES(1202,1209,'04-OCT-2018','Family');
INSERT INTO Friends VALUES(1202,1210,'05-DEC-2016','Educational');
INSERT INTO Friends VALUES(1202,1211,'02-NOV-2017','Colleague');
INSERT INTO Friends VALUES(1202,1212,'12-NOV-2017','Family');
INSERT INTO Friends VALUES(1203,1204,'01-SEP-2018','Educational');
INSERT INTO Friends VALUES(1203,1205,'15-NOV-2016','Colleague');
INSERT INTO Friends VALUES(1203,1210,'03-MAY-2017','Educational');
INSERT INTO Friends VALUES(1204,1207,'05-JUN-2016','Family');
INSERT INTO Friends VALUES(1204,1208,'09-JUL-2018','Educational');
INSERT INTO Friends VALUES(1204,1209,'06-APR-2015','Educational');
INSERT INTO Friends VALUES(1204,1212,'03-MAR-2014','Family');
INSERT INTO Friends VALUES(1205,1206,'10-JAN-2016','Educational');
INSERT INTO Friends VALUES(1205,1207,'18-JUN-2016','Colleague');
INSERT INTO Friends VALUES(1205,1212,'16-SEP-2018','Colleague');
INSERT INTO Friends VALUES(1206,1207,'26-NOV-2014','Educational');
INSERT INTO Friends VALUES(1206,1208,'09-MAY-2016','Educational');
INSERT INTO Friends VALUES(1207,1208,'08-JUL-2014','Family');
INSERT INTO Friends VALUES(1208,1209,'17-NOV-2016','Educational');
INSERT INTO Friends VALUES(1209,1210,'10-JUN-2014','Educational');
INSERT INTO Friends VALUES(1210,1212,'04-FEB-2017','Colleague');
INSERT INTO Friends VALUES(1211,1212,'08-OCT-2019','Family');


INSERT INTO Stories VALUES(50209,23,'30-MAR-21 04:43:09.24');
INSERT INTO Stories VALUES(50210,43,'30-MAR-21 11:01:55.55');
INSERT INTO Stories VALUES(50211,68,'30-MAR-21 02:49:41.76');
INSERT INTO Stories VALUES(50212,12,'30-MAR-21 09:06:14.54');
INSERT INTO Stories VALUES(50213,54,'30-MAR-21 10:35:06.32');
INSERT INTO Stories VALUES(50214,26,'30-MAR-21 07:41:05.24');
INSERT INTO Stories VALUES(50215,76,'30-MAR-21 08:17:09.11');
INSERT INTO Stories VALUES(50216,33,'30-MAR-21 11:33:16.08');
INSERT INTO Stories VALUES(50217,39,'30-MAR-21 07:58:08.88');
INSERT INTO Stories VALUES(50218,49,'30-MAR-21 11:14:49.43');
INSERT INTO Stories VALUES(50219,31,'30-MAR-21 08:27:05.29');
INSERT INTO Stories VALUES(50220,61,'30-MAR-21 04:55:08.19');


INSERT INTO Mentions VALUES(50211,1201);
INSERT INTO Mentions VALUES(50212,1206);
INSERT INTO Mentions VALUES(50214,1212);
INSERT INTO Mentions VALUES(50218,1204);
INSERT INTO Mentions VALUES(50219,1203);
INSERT INTO DirectMessage VALUES
(1201,1202,'12-MAR-12 04:29:18.73','HEY','text');
INSERT INTO DirectMessage VALUES
(1201,1202,'13-MAR-12 04:29:18.73','See you at 5.00 pm','text');
INSERT INTO DirectMessage VALUES
(1202,1201,'14-OCT-13 05:19:48.73','https://newspaper.html','.html');
INSERT INTO DirectMessage VALUES
(1201,1203,'15-SEP-15 06:29:38.73','How much is the fee?','text');
INSERT INTO DirectMessage VALUES
(1204,1205,'16-DEC-14 07:29:28.73','/files/info.png','.png');
INSERT INTO DirectMessage VALUES
(1205,1204,'17-MAR-17 08:24:18.73','/folder/assignment.pdf','.pdf');
INSERT INTO DirectMessage VALUES
(1205,1202,'18-JAN-18 09:29:08.73','Happy Birthday!','text');
INSERT INTO DirectMessage VALUES
(1204,1203,'19-MAR-12 04:39:58.73','/files/song.mp3','.mp3');
INSERT INTO DirectMessage VALUES
(1207,1212,'13-FEB-18 06:27:38.73','Goodbye','text');
INSERT INTO DirectMessage VALUES
(1209,1210,'16-AUG-19 07:09:15.73','/files/stars.png','.png');

INSERT INTO Groups VALUES
(78545,'College Friends','To share memories and casual chat','26-JUN-12 09:39:16.78');
INSERT INTO Groups VALUES
(78546,'Carpool for Work','Driver no.: 1234 Pickup time: 8.00 am','16-SEP-182 12:19:17.98');
INSERT INTO Groups VALUES
(78547,'Presentation Team','Presentation on traffic data. Deadline tomorrow','07-OCT-14 03:23:08.65');
INSERT INTO Groups VALUES
(78548,'Family Group','Peace, love and happiness','29-MAR-12 07:29:14.72');
INSERT INTO Groups VALUES
(78549,'Painting Club','Paint away the world!','12-FEB-12 04:39:18.73');

INSERT INTO GroupMembers VALUES(1201,'College Friends','rosie','member','12-APR-14');
INSERT INTO GroupMembers VALUES(1202,'College Friends','lastjedi','admin','13-APR-15');
INSERT INTO GroupMembers VALUES(1203,'College Friends','avery','member','14-OCT-19');
INSERT INTO GroupMembers VALUES(1204,'College Friends','sweety','member','15-APR-13');
INSERT INTO GroupMembers VALUES(1205,'Painting Club','han','admin','16-NOV-15');
INSERT INTO GroupMembers VALUES(1206,'Painting Club','martha','member','01-APR-12');
INSERT INTO GroupMembers VALUES(1203,'Presentation Team','avery','admin','17-DEC-17');
INSERT INTO GroupMembers VALUES(1208,'Presentation Team','rosie','member','02-APR-16');
INSERT INTO GroupMembers VALUES(1209,'Presentation Team','suzie','member','17-MAR-18');
INSERT INTO GroupMembers VALUES(1210,'Carpool for Work','ron','admin','03-AUG-13');
INSERT INTO GroupMembers VALUES(1211,'Carpool for Work','harry','member','11-MAY-14');
INSERT INTO GroupMembers VALUES(1212,'Family Group','shelly','member','10-OCT-17');
INSERT INTO GroupMessage VALUES
(78545,1201,'Hello','text','12-FEB-12 04:39:18.73');
INSERT INTO GroupMessage VALUES
(78545,1201,'Hello','text','12-NOV-12 04:39:18.73');
INSERT INTO GroupMessage VALUES
(78545,1202,'/files/flower.png','.png','12-OCT-125 04:30:18.73');
INSERT INTO GroupMessage VALUES
(78545,1201,'Hi','text','16-FEB-12 04:36:18.73');
INSERT INTO GroupMessage VALUES
(78546,1205,'/files/beach.jpeg','.jpeg','22-SEP-17 04:38:18.73');
INSERT INTO GroupMessage VALUES
(78546,1206,'This is the plan','text','02-FEB-12 04:37:18.73');
INSERT INTO GroupMessage VALUES
(78547,1203,'Here is the file','text','17-FEB-13 04:39:18.73');
INSERT INTO GroupMessage VALUES
(78547,1208,'Agenda/speech.mp3','.mp3','12-DEC-12 04:59:18.73');
INSERT INTO GroupMessage VALUES
(78547,1209,'Thank you','text','18-FEB-18 04:37:18.73');
INSERT INTO GroupMessage VALUES
(78549,1210,'Send the details','text','12-MAR-12 04:29:18.73');
INSERT INTO GroupMessage VALUES
(78549,1211,'folder/rules.txt','.txt','13-FEB-14 04:49:18.73');

INSERT INTO TopicTag VALUES (7653,'CaptureIt');
INSERT INTO TopicTag VALUES (7654,'Trendz');
INSERT INTO TopicTag VALUES (7655,'Vibing');
INSERT INTO TopicTag VALUES (7656,'Inspire');
INSERT INTO TopicTag VALUES (7657,'Art_fo_life');
INSERT INTO TopicTag VALUES (7658,'foodiez');
INSERT INTO TopicTag VALUES (7659,'OceanWaves');
INSERT INTO TopicTag VALUES (7660,'chill');
INSERT INTO TopicTag VALUES (7661,'EDMLove');
INSERT INTO TopicTag VALUES (7662,'HipHop');
INSERT INTO TopicTag VALUES (7663,'Assignment');
INSERT INTO TopicTag VALUES (7664,'YOLO');
INSERT INTO TopicTag VALUES (7665,'Engineering');
INSERT INTO TopicTag VALUES (7666,'Cheems');
INSERT INTO TopicTag VALUES (7667,'meme');

INSERT INTO Posts VALUES
(293434, 1206, 'Sea''s swayin''... ', '/files/videos/20190127_042010.mp4', 7659, '27-JAN-2019
04:20:10.01','','');
```

```
INSERT INTO Posts VALUES
(293435, 1204, 'I''m a Bachelors of Memology(B.Meme) graduate.',
'/files/images/20200122_071011.jpg', 7667, '22-JAN-2020 07:10:10.51','','');
INSERT INTO Posts VALUES
(293436, 1209, 'To live is to Inspire', '/files/mpeg-4/20190205_014517.m4a',
7656, '05-FEB-2019 01:45:16.31','','');
INSERT INTO Posts VALUES
(293437, 1211, 'Dance to death...', '/files/videos/20170407_062539.mp4',
7662, '07-APR-2017 06:25:38.58','','');
INSERT INTO Posts VALUES
(293438, 1208, 'Expressin heart ...with art!', '/files/images/20180311_071517.jpeg',
7657, '11-MAR-2018 07:15:17.05','','');
INSERT INTO Posts VALUES
(293439, 1203, 'Plate was Full...now am Filled...Plate is empty', '/files/images/20170812_054511.jpg',
7658, '12-AUG-2017 05:45:10.49','','');
INSERT INTO Posts VALUES
(293440, 1208, 'Blur''s beauty in times...focus ain''t mandated', '/files/images/20200628_080253.jpg',
7653, '28-JUN-2020 08:02:52.13','','');
INSERT INTO Posts VALUES
(293441, 1211, 'You Only Live Once...Damn!', '/files/videos/20171106_013112.mp4',
7664, '06-NOV-2017 01:31:11.01','','');
INSERT INTO Posts VALUES
(293442, 1204, 'I type m instead n...am i the Cheems??', '/files/videos/20190423_091623.mp4',
7666, '23-APR-2019 09:16:23.02','','');
INSERT INTO Posts VALUES
(293443, 1202, 'Century!!...', '.files/images/20181009_060916.jpeg',
7654, '09-OCT-2018 06:09:16.01','','');
INSERT INTO Posts VALUES
(293444, 1205, 'Assignments->to do list...Procatination->In progress list ',
'/files/images/20200709_064018.jpg', 7663, '09-JUL-2020 06:40:17.41','','');
INSERT INTO Posts VALUES
(293445, 1212, 'had enough of life...*Slurps*', '/files/images/20210224_031238.jpg',
7658, '24-FEB-2021 03:12:38.07','','');
INSERT INTO Posts VALUES
(293446, 1210, 'rock it YOLO', '/files/mpeg-4/20161229_080027.m4a',
7664, '29-DEC-2016 08:00:27.15','','');
INSERT INTO Posts VALUES
(293447, 1201, 'Million days with terrace air', '/files/images/20190605_060353.jpg',
7660, '05-JUN-2019 06:03:53.21','','');
INSERT INTO Posts VALUES
(293448, 1202, 'keep inspiring...', '/files/videos/20190922_014546.mp4',
7656, '22-SEP-2019 01:45:46.18','','');
```

```
INSERT INTO Posts VALUES
(293449, 1206, 'Smile in pain...', '/files/images/20181020_094612.jpg',
7665, '20-OCT-2018 09:46:11.41','','');
INSERT INTO Posts VALUES
(293450, 1212, 'Damn....just EDM''s are Enough', '/files/music/20201127_111158.mp3',
7661, '27-NOV-2020 11:11:58.01','','');
INSERT INTO Posts VALUES
(293451, 1207, 'Trop tard, Trop tard...', '/files/videos/201903_021446.mp4',
7662, '25-MAR-2017 02:14:46.32','','');
INSERT INTO Posts VALUES
(293452, 1205, 'Wise men say...only Fools rush in', '/files/music/20180814_120022.mp3',
7655, '14-AUG-2018 12:00:22.09','','');


INSERT INTO Likes VALUES ( 293441, 1208, '25-02-18 11:25:44.12');
INSERT INTO Likes VALUES ( 293438, 1205, '07-04-18 11:45:46.34');
INSERT INTO Likes VALUES ( 293450, 1203, '07-11-21 12:47:52.52');
INSERT INTO Likes VALUES ( 293450, 1201, '15-08-21 10:26:24.24');
INSERT INTO Likes VALUES ( 293440, 1208, '08-01-21 06:42:15.16');
INSERT INTO Likes VALUES ( 293450, 1210, '02-08-21 02:53:24.26');
INSERT INTO Likes VALUES ( 293450, 1207, '25-02-21 05:57:24.43');
INSERT INTO Likes VALUES ( 293439, 1201, '01-04-18 11:08:36.51');
INSERT INTO Likes VALUES ( 293440, 1203, '05-01-21 06:49:27.15');
INSERT INTO Likes VALUES ( 293438, 1204, '09-11-18 08:55:52.00');
INSERT INTO Likes VALUES ( 293446, 1201, '19-09-17 07:23:11.46');
INSERT INTO Likes VALUES ( 293439, 1211, '11-07-18 03:57:43.45');
INSERT INTO Likes VALUES ( 293442, 1208, '23-03-20 10:38:54.13');
INSERT INTO Likes VALUES ( 293446, 1210, '04-08-17 02:56:32.16');
INSERT INTO Likes VALUES ( 293434, 1208, '20-05-19 12:14:57.43');
INSERT INTO Likes VALUES ( 293441, 1204, '11-04-18 12:52:05.41');
INSERT INTO Likes VALUES ( 293437, 1211, '20-06-17 12:57:59.15');
INSERT INTO Likes VALUES ( 293442, 1205, '23-09-19 05:01:00.35');
INSERT INTO Likes VALUES ( 293449, 1210, '20-03-19 03:55:43.16');
INSERT INTO Likes VALUES ( 293448, 1206, '10-06-20 11:19:27.53');
INSERT INTO Likes VALUES ( 293447, 1207, '26-09-19 11:07:05.59');
INSERT INTO Likes VALUES ( 293445, 1203, '09-02-22 04:57:08.43');
INSERT INTO Likes VALUES ( 293445, 1209, '20-06-21 03:09:43.16');
INSERT INTO Likes VALUES ( 293442, 1207, '03-01-20 09:14:12.13');
INSERT INTO Likes VALUES ( 293438, 1211, '24-06-18 11:42:56.12');
INSERT INTO Likes VALUES ( 293443, 1208, '11-02-19 09:39:52.14');
INSERT INTO Likes VALUES ( 293435, 1207, '10-10-20 06:50:53.41');
INSERT INTO Comments VALUES
( 293444, 1201, 'Go strt ur Assigns now...','15-07-20 02:44:30.43');
INSERT INTO Comments VALUES
( 293441, 1209, 'tht''s the coolest thing i''ve heard today ','28-12-17 05:04:16.16');
INSERT INTO Comments VALUES
( 293451, 1203, 'litt.....what''s the song btw??','10-05-17 12:02:11.23');
INSERT INTO Comments VALUES
( 293445, 1206, 'our way ...the foodiez way!!','28-02-21 10:07:27.00');
INSERT INTO Comments VALUES
( 293434, 1211, 'the sunset + oceans..the timing','17-02-19 06:44:30.09');
INSERT INTO Comments VALUES
( 293434, 1203, 'Beauty...thts how we define it!','03-02-19 12:02:33.08');
INSERT INTO Comments VALUES
( 293436, 1202, 'Made my day <3','28-03-19 11:23:58.01');
INSERT INTO Comments VALUES
( 293443, 1203, 'in petrol pump tho','05-11-18 09:49:38.59');
INSERT INTO Comments VALUES
( 293435, 1202, 'damn...keep going...','25-01-20 08:38:09.22');
INSERT INTO Comments VALUES
( 293439, 1203, 'lurin us ehhh??','14-09-17 05:36:38.18');
INSERT INTO Comments VALUES
( 293450, 1205, 'ikr...damn clarity X your name..','25-01-21 08:04:02.57');
INSERT INTO Comments VALUES
( 293443, 1208, 'true..not in stadium tho','13-10-18 06:29:56.48');
INSERT INTO Comments VALUES
( 293442, 1207, 'RIP cheems...','11-05-19 12:09:41.14');
INSERT INTO Comments VALUES
( 293446, 1201, 'wohww..the perfect song with perfect caption!!','02-03-17 02:47:15.51');
INSERT INTO Comments VALUES
( 293438, 1205, 'seriously!!?','13-05-18 12:17:15.52');
INSERT INTO Comments VALUES
( 293437, 1206, 'u''ve got tht grace...thts litt','26-04-17 05:32:51.26');
INSERT INTO Comments VALUES
( 293445, 1201, 'uhmm uhmm... *even longer slurp*','24-02-21 05:28:47.53');
INSERT INTO Comments VALUES
( 293434, 1201, 'aye...swayin''<3','05-03-19 04:24:52.51');
INSERT INTO Comments VALUES
( 293444, 1208, 'bro!..i topped the procastination test','19-07-20 05:20:46.45');
INSERT INTO Comments VALUES
( 293436, 1209, 'thts great!','06-02-19 11:34:22.01');
```

```
INSERT INTO Comments VALUES
( 293438, 1210, 'killerrrr!!','20-04-18 04:53:05.05');
INSERT INTO Comments VALUES
( 293443, 1202, 'aye..','22-10-18 12:42:52.50');
INSERT INTO Comments VALUES
( 293436, 1206, 'okkeiii..','15-03-19 10:49:18.03');
INSERT INTO Comments VALUES
( 293444, 1209, 'smile in pain...','17-07-20 09:57:58.06');
```

3. PL/SQL Procedures

3.1. likesCount() Function

The likesCount() function takes PostID from Posts table and UserID from Users table as parameters, and whenever a user clicks the like button on a post, this function is executed, which increments the number of likes on that post by one. It also adds the like details (for eg. LikeTime) to the Likes table.

```
CREATE OR REPLACE PROCEDURE
likesCount(post_id IN Posts.PostID%TYPE, user_id IN Users.UserID%TYPE) AS
BEGIN
    INSERT INTO Likes VALUES (post_id,user_id,TO_CHAR(SYSDATE,'dd-mon-yyyy hh24:mi:ss'));
    UPDATE Posts
    SET LikesNo = LikesNo + 1
    WHERE PostID = post_id;
END;
```

3.2. commentCount() Function

The commentCount() function takes PostID from Posts table and UserID from Users table as parameters, and whenever a user comments on a post, the function is executed to increment number of comments on the post. It also adds the details (for eg. CmtData, CmtTime) to the Comments table.

```
CREATE OR REPLACE PROCEDURE
commentCount(post_id IN Posts.PostID%TYPE, User_id IN Users.UserID%TYPE, commentData
IN Comments.CmtData%TYPE) AS
BEGIN
    INSERT INTO Comments VALUES(post_id,user_id,commentData,TO_CHAR(SYSDATE,'dd-
    mon-yyyy hh24:mi:ss'));
    UPDATE Posts
    SET CommentsNo = CommentsNo + 1
    WHERE PostID = post_id;
END;
```

3.3. userSearch() Function

The userSearch() function takes UserName from Users table and a reference cursor as parameters to search if there is any user in the system, with the given UserName.

```
CREATE OR REPLACE PROCEDURE
userSearch(user_name IN Users.UserName%TYPE, user_refCursor IN OUT SYS_REFCURSOR) AS
BEGIN
    OPEN user_refCursor FOR
    SELECT UserName FROM Users WHERE UserName = user_name;
END;
```

3.4. encryptPassword() Function

The encryptPassword() function takes Password from Users table as a parameter, and it encrypts the password by converting it to SHA1 hash code. This is done using the built-in function HASH() which belongs to the DBMS_CRYPTO package, the permission for which, should be granted first by SYS user.

```
CREATE OR REPLACE PROCEDURE
encryptPassword(password IN Users.Password%TYPE) AS
BEGIN
    UPDATE Users SET Password = DBMS_CRYPTO.HASH(RAWTOHEX(password),3);
END;
```

3.5. Notify Mentioned Users

This procedure helps to notify user A (receiver) when another user B (sender) mentions the user A in a particular story. This is achieved by inputting the StoryID and MentionID in the beginning and then producing a message output.

```
SET SERVEROUTPUT ON
DECLARE
    sender Users.UserName%TYPE;
    receiver Users.UserName%TYPE;
    storyNo Mentions.StoryID%TYPE := &storyNo;
    mentionNo Mentions.MentionID%TYPE := &mentionNo;
BEGIN
    SELECT UserName INTO sender FROM Users
    WHERE Users.StoryID = storyNo;
    SELECT UserName INTO receiver FROM Users
    WHERE Users.UserID = MentionNo;
    DBMS_OUTPUT.PUT_LINE('To @'||receiver||' send notification: '||sender||' has mentioned
    you in their story!');
END;
```

3.6. recommendFriend() Function

The recommendFriend() function takes SourceUserID from Friends table as a parameter to represent a user. It creates two cursors - C1 for list of that user's friends, C2 for list of users that are not friends with any of the users in C1. This function recommends friends to the given SourceUserID.

```
CREATE OR REPLACE PROCEDURE
recommendFriend(srcid IN Friends.SourceUserID%TYPE) AS
    CURSOR C1 IS
        SELECT DISTINCT SourceUserID FROM Friends WHERE SourceUserID NOT IN (SELECT
        SourceUserID FROM Friends WHERE DestUserID = srcid) AND SourceUserID NOT IN
        (SELECT DestUserID FROM friends WHERE SourceUserID = srcid);
    CURSOR C2 IS
        SELECT DISTINCT DestUserID FROM Friends WHERE DestUserID NOT IN (SELECT
        DestUserID FROM Friends WHERE SourceUserID = srcid) AND DestUserID NOT IN (SELECT
        SourceUserID FROM Friends WHERE DestUserID = srcid);
    count1 NUMBER;
    count2 NUMBER;
BEGIN
    FOR notf IN C1 LOOP
        SELECT COUNT(DISTINCT SourceUserID) INTO count1 FROM Friends
        WHERE SourceUserID = notf.SourceUserID AND DestUserID IN ((SELECT DestUserID
        FROM Friends WHERE SourceUserID = srcid) UNION (SELECT SourceUserID FROM Friends
        WHERE DestUserID = srcid));
         IF count1 > 0 AND notf.sourceuserid!=srcid THEN
            DBMS_OUTPUT.PUT_LINE('Recommended: ' || notf.SourceUserID);
         END IF;
    END LOOP;
    FOR notf IN C2 LOOP
        SELECT COUNT(DISTINCT DestUserID) INTO count2 FROM Friends WHERE DestUserID =
        notf.DestUserID AND SourceUserID IN ((SELECT DestUserID FROM Friends WHERE
        SourceUserID = srcid) UNION (SELECT SourceUserID FROM Friends WHERE DestUserID =
        srcid));
        IF count2 > 0 AND notf.DestUserID != srcid THEN
            DBMS_OUTPUT.PUT_LINE('Recommended: ' || notf.DestUserID);
        END IF;
    END LOOP;
END;
SET SERVEROUTPUT ON;
BEGIN
    recommendFriend(1210);
END;
```

3.7. recommendTopic() Function

The recommendTopic() function takes UserID from Users table as a parameter. We calculate the interactions of the given user with any post of a given Topic in terms of their Likes and Comment frequencies on that particular Topics. The top three interacted topics are shortlisted for each user and those details are inserted into the Recommended relation.

```
CREATE OR REPLACE PROCEDURE
recommendTopic(u IN Users.UserID%TYPE) AS
    CURSOR C IS
        SELECT * FROM(SELECT nvl(tab1.PostTagID,tab2.PostTagID) AS
        TagID(nvl(tab1.likecount,0) + nvl(tab2.commentcount,0)) AS
        allcount, nvl(tab1.likecount,0) AS lcount, nvl(tab2.commentcount,0)
        AS ccount FROM (SELECT Posts.PostTagID, COUNT(Likes.UserID) AS
        likecount FROM Posts, Likes WHERE Posts.PostID = Likes.PostID
        AND Likes.UserID = 1208 GROUP BY Posts.PostTagID) tab1
        FULL OUTER JOIN
        (SELECT Posts.PostTagID, COUNT(Comments.UserID) AS commentcount FROM
        Posts, Comments WHERE Posts.PostID = Comments.PostID AND Comments.UserID = 1208
        GROUP BY Posts.PostTagID) tab2 ON tab1.PostTagID = tab2.PostTagID
        ORDER BY allcount DESC) WHERE ROWNUM <= 3;
BEGIN
    FOR rec IN C LOOP
        INSERT INTO Recommended VALUES(rec.TagID,u,rec.lcount,rec.ccount);
    END LOOP;
END;

PROCEDURE 2:
CREATE PROCEDURE addrec AS
    CURSOR C IS
        SELECT DISTINCT UserID FROM Likes
        UNION
        SELECT DISTINCT UserID FROM Comments ORDER BY UserID;
BEGIN
    FOR u IN C LOOP
        recommendTopic(u.UserID);
    END LOOP;
END;

EXEC addrec;
```

Functional Dependency (FD) is a constraint that determines the relation of one attribute to another attribute in a database. It is denoted by an arrow "→". The functional dependency of X on Y is represented by X → Y.

1. Users

Users is an independent entity, it holds 2 candidate keys namely :

UserID → { UserName, FirstName, LastName, EmailID, DOB, DOJ, Bio, Password, AccountType, LastSeen, StoryID }
EmailID → {UserID, UserName, FirstName, LastName, DOB, DOJ, Bio, Password, AccountType, LastSeen, StoryID}

Out of which UserID is chosen to be Primary key over others as the ID can be auto incremented for every new insertion in the users relation and so uniquely allotted to every User in this database. And the closure of UserID gives us the entire relation.
(UserID)+ → R

2. Friends

Friends a Dependent entity entirely dependent on Users relation for its existence. It has a single candidate key .

{SourceUserID, DestUserID} → {DOC, Relationship}

The SourceUserID, DestUserID pair can uniquely identify the tuple holding the connection between two users and is the only possible candidate key in Friends relation. Therefore chosen to be the composite primary key. If We find the closure of {SourceUserID, DestUserID} we get the entire relation.
({SourceUserID, DestUserID})+ → R.

3. Stories

A Dependent entity whose existence is possible only with Users relation. It too has only a single candidate key namely :

StoryID → {Views, StoryTime}

As every user's allotted a StoryID alongside the UserID uniquely. In addition to that, restricting the story time to 24 hours and the number of stories by a user per day to one, every record in stories relation can be uniquely identified by just StoryID making it the primary key.
(StoryID)+ → R.

4. Mentions

It’s dependent on both Stories and Users relations. It has the whole attribute set as one and only candidate key set.

{StoryID, MentionID} → { StoryID, MentionID}

A single story can have multiple mentions, so the key pair makes it possible to have each record uniquely and hence is the composite primary key , whose closure gives the whole relation because of the trivial dependency.

(StoryID, MentionID)+ → R

5. DirectMessage

It’s a dependent relation depending solely on users table and it holds 3 candidate keys

{SourceUserID, DestUserID, MsgTime} → {SourceUserID, DestUserID, MsgData, MsgType, MsgTime }

We choose {SourceUserID, DestUserID, MsgTime} ,even when MsgTime can most probably differentiate every single record from the other with the precision of time in timestamp given to 6 milliseconds in DirectMessage table.

It is because of the reliability in unique identification when we include both SourceUserID and DestUserID with MsgTime and for easier search of texts with userID's .

({SourceUserID, DestUserID, MsgTime})+ → R.

6. Groups

This is one of the 3 independent entities in database.It has 2 candidate keys

GroupID → {GroupID, GroupName, Description, CreationTime}

{GroupName, CreationTime} → {GroupID, GroupName, Description, CreationTime}

The groupID can solely identify a record in the groups table and can be easily allotted by auto increment to every record entered in groups. So, it is chosen over other candidate keys.

(GroupID)+ → R.

7 . GroupMembers

GroupMembers is a Characteristic Entity i.e., A Multi valued Attribute. And has a single candidate key given by

{GroupName, UserID} → {GroupName, UserID, UserName, Privilege, DOJ}

A user can be part of many groups.So, all the records corresponding to the same user in the GroupMembers table will have a different groupID associated with it and so can uniquely identify records in it's closure.

({GroupName, UserID})+ → R.

8. GroupMessage

GroupMessage is a dependent attribute ,depending on two of the independent entities Groups and Groupmembers .It holds a candidate key

{GroupID, UserID, MsgTime} → {GroupID, UserID, MsgData, MsgType, MsgTime }

Multiple messages can be sent by a user in a group, so MsgTime differentiates records from each other. Making {GroupID, UserID, MsgTime} the primary key of the Relation.

({GroupID, UserID})+ → R

10. Posts

It's a dependent entity with its own primary key . Holding two candidate keys namely :

PostID → {PostID, UserId, Caption, PostType, PostTagID, PostTime, LikesNo, CommentsNo }
{UserID,PostTime} → {PostID, UserId, Caption, PostType, PostTagID, PostTime, LikesNo, CommentsNo }

Out of which postID is taken as the Primary key than the other, because of the auto Incremented postIDs and good organised content in the post table. Moreover the closure of postID gives the whole relation.

(PostID)+ → R.

9. TopicTag

Topic Tag's the last independent entity in the database. It has a single candidate key

TagID → {TagID, TagName}

It is auto incremented for each distinct tagname. And it's closure gives us the whole relation

(TagID)+ → R

11. Likes

Entirely dependent on Posts table and Users table.Considering the single primary key

{PostID,UserID} → {PostID, UserID, LikeTime}

Each post is liked by multiple users and so it is best recorded by making the postID, UserID pair as primary key i.e., unique.It's closure gives us the entire relation.

({PostID,UserID})+ → R.

12. Comments

Entirely dependent on Posts table and Users table.Considering the single primary key

{PostID,UserID} → {PostID, UserID, CmtData, CmtTime}

Each post is commented by multiple users and so it is best recorded by making the postID, UserID pair as primary key i.e., unique.It's closure gives us the entire relation.

({PostID,UserID})+ → R.

13. Recommended

This is too a dependent entity having a candidate key

{TagID,UserID} → {TagID, UserID, LikeFreq, CmtFreq}

The top 3 distinct TagIDs being liked and commented by a user are recorded in this table making the {TagID,UserID} pair distinct in every record .We get the other attributes from this pair of values if specified and can be clearly depicted as

({TagID,UserID})+ → R.

Normalization

Normalization is a database design technique that reduces data redundancy. We have restricted the discussion upto four normal forms: 1NF (First Normal Form), 2NF (Second Normal Form), 3NF (Third Normal Form) and BCNF (Boyce-Codd Normal Form). The relations have been splitted so as to meet the first four normal forms.

1. Users

Candidate keys :
UserID
EmailID

- All of the Attributes of the Users relation are single valued and so it's in 1NF.
- No Partial dependencies are possible in Users relation with a single determining attribute UserID keeping it in 2NF.
- Not a single attribute of users table transitively depends on the candidate key UserID, making it positive in 3NF.
- All determinants in functional dependencies are super keys, hence the table is in BCNF.

2. Friends

Candidate keys:
{SourceUserID, DestUserID}

- All of the Attributes of the Friends relation are single valued and so it's in 1NF.
- No Partial dependencies can be spotted as both SourceUserID,DestUser together identify a row uniquely keeping it in 2NF.
- Every attribute of the table Directly depends on the determining attributes pair , so it attained 3NF.
- The determinant here is a super key.So, the table is in BCNF.

3. Stories

Candidate keys :
{StoryID}

- All of the Attributes of the Stories relation are single valued and so it's in 1NF.
- No Partial dependencies are possible in this relation with a single determining attribute StoryID keeping it in 2NF.
- Not a single attribute of users table transitively depends on the candidate key StoryID, satisfying 3NF.
- All determinants in functional dependencies are super keys, hence the table is in BCNF.

4. Mentions

Candidate keys:

{StoryID, MentionID}

- All of the Attributes of the Mentions relation are single valued and so it’s in 1NF.
- No Partial dependencies can be spotted as both StoryID, MentionID together identify a row uniquely keeping it in 2NF.
- Every attribute of the table Directly depends on the determining attributes pair, making it positive in 3NF.
- The determinant here is a super key. So, the table is in BCNF.

5. DirectMessage

Candidate keys:

{SourceUserID, DestUserID, MsgTime}

- All of the Attributes of the DirectMessage relation are single valued and so it’s in 1NF.
- No Partial dependencies can be spotted as all SourceUserID, DestUserID, MsgTime together identify a row uniquely keeping it in 2NF.
- Every attribute of the table Directly depends on the determining attributes , attaining 3NF
- The determinant here is a super key. So, the table is in BCNF.

6. Groups

Candidate keys :

{GroupID}

{GroupName, CreationTime}

- All of the Attributes of the Groups relation are single valued and Group members is implemented as a so it’s in 1NF.
- No Partial dependencies are possible in this relation with a single determining attribute GroupID keeping it in 2NF.
- Not a single attribute of groups table transitively depends on the candidate key GroupID, making it positive in 3NF.
- All determinants in functional dependencies are super keys, hence the table is in BCNF.

7. GroupMembers

Candidate keys:

{GroupName, UserID}

- All of the Attributes of the GroupMembers relation are single valued and so it's in 1NF.
- No Partial dependencies can be spotted as both GroupName, UserID together identify a row uniquely UserID keeping it in 2NF.
- Every attribute of the table Directly depends on the determining attributes , making it positive in 3NF.
- The determinant here is a super key. So, the table is in BCNF.

8. GroupMessage

Candidate keys:

{GroupID, UserID, MsgTime}

- All of the Attributes of the GroupMessages relation are single valued and so it's in 1NF.
- No Partial dependencies can be spotted as all GroupID, UserID, MsgTime together identify a row uniquely keeping it in 2NF.
- Every attribute of the table Directly depends on the determining attributes , making it positive in 3NF.
- The determinant here is a super key. So, the table is in BCNF.

9. Posts

Candidate keys :

{PostID}

{UserID,PostTime}

- All of the Attributes of the Posts relation are single valued so it's in 1NF.
- No Partial dependencies are possible in this relation with a single determining attribute PostID keeping it in 2NF.
- Not a single attribute of the groups table transitively depends on the candidate key PostID, making it positive in 3NF.
- Every other attribute directly depends on the candidate key PostID, Satisfying 3NF.
- All determinants in functional dependencies are super keys, hence the table is in BCNF.

10. TopicTag

Candidate keys:
{TagID}

- All of the Attributes of the Topic tags relation are single valued so it’s in 1NF.
- No Partial dependencies are possible in this relation with a single determining attribute TagID keeping it in 2NF. Other attributes directly depends on the candidate key TagID, Satisfying 3NF.
- All determinants in functional dependencies are super keys, hence the table is in BCNF.

11. Likes

Candidate keys:
{PostID,UserID}

- All of the Attributes of the Likes relation are single valued and so it’s in 1NF.
- No Partial dependencies can be spotted as all PostID, UserID together identify a row uniquely keeping it in 2NF.
- Every attribute of the table Directly depends on the determining attributes , making it satisfy 3NF. The determinant here is a super key. So, the table is in BCNF.

12. Comments

Candidate keys:
{PostID,UserID}

- All of the Attributes of the Comments relation are single valued and so it’s in 1NF.
- No Partial dependencies can be spotted as all PostID, UserID together identify a row uniquely keeping it in 2NF.
- Every attribute of the table Directly depends on the determining attributes , making it satisfy 3NF. The determinant here is a super key. So, the table is in BCNF.

13. Recommended

Candidate keys:
{TagID,UserID}

- All of the Attributes of the Recommended relation are single valued and so it’s in 1NF.
- No Partial dependencies can be spotted as all TagID, UserID together identify a row uniquely keeping it in 2NF.
- Every attribute of the table Directly depends on the determining attributes , making it satisfy 3NF.
- The determinant here is a super key. So, the table is in BCNF.

Conclusion

During our database management course we have learnt about the basics of database design. This project gave us the opportunity to bring our new skills to practice. We also gained deeper understanding on database design and how it can be implemented in real life situations. This social media database model is simplistic and has scope for further enhancements. It covers all the basic features of a social media platform and can be extended to develop further nuance.

Tools:

- Oracle Database 11g Express Edition
- Oracle SQL Developer
- Canva Online Doc
- app.diagrams.net - Flowchart Maker for E-R & Relational Model
- MS Office Excel

References:

- Database System Concepts (A. Silberschatz, H.F. Korth, S. Sudarshan)
- javatpoint.com
- GATE Smashers (Youtube channel)
- guru99.com
- stackoverflow.com