

GROUP – 29

-Aditya

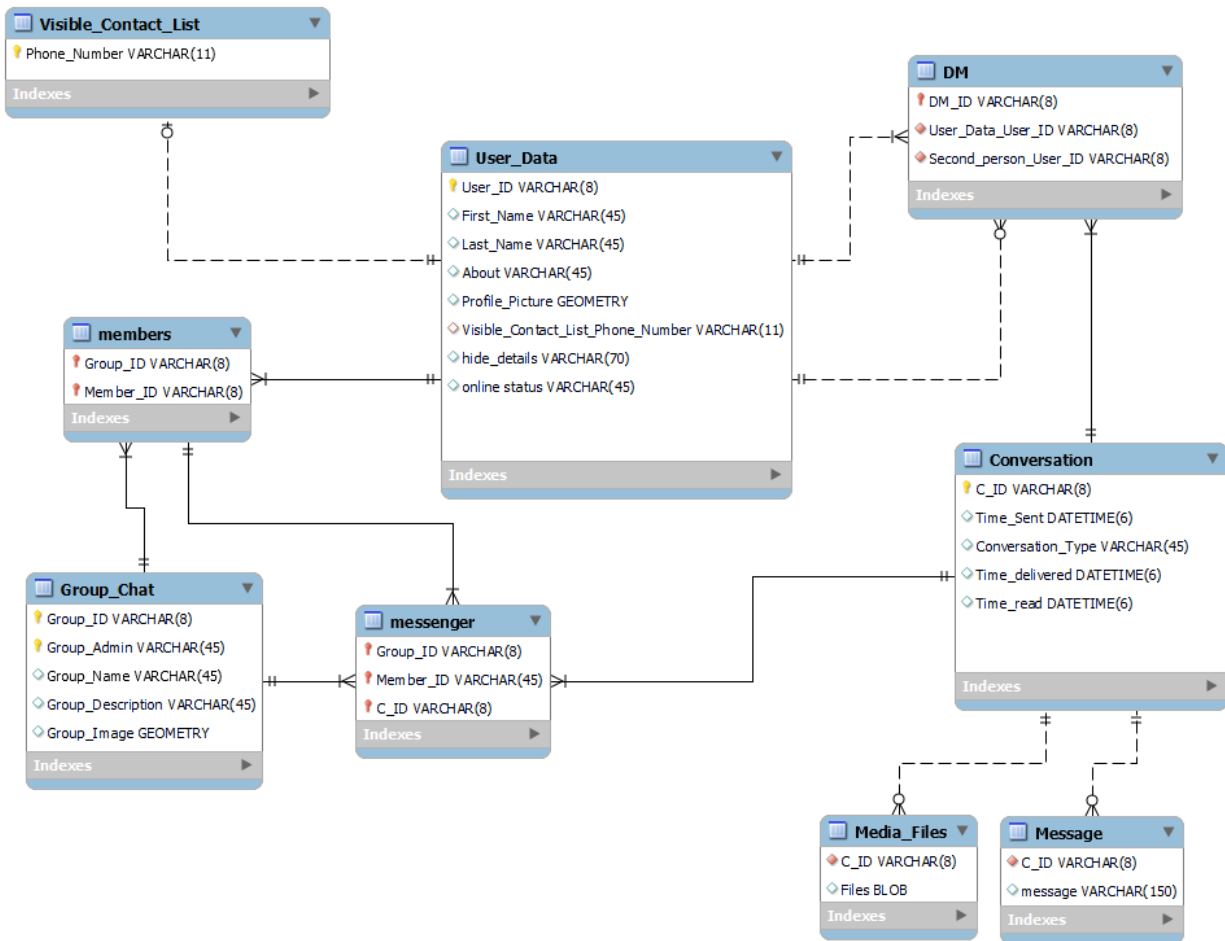
-Anurag

-harsh

-Kushagra

Project Scope

Our project is a chat application that allows users to contact others via text messages in a group chat or a direct message. We have tried to implement this chat application using MySQL, (HTML, tailwind CSS, and javascript) for the User Interface and Flask for the backend.



Stakeholders

Users, Owners, developers, business analysts, project manager, technical team.

SQL Queries

Query 1 – User Name, Status, Profile Picture & Phone Number

```
SELECT u.First_Name, u.Status, u.Profile_Picture, u.Visible_Contact_List_Phone_Number  
FROM User_Data as u
```

Query 2 – Message retrieval of a DM Conversation

```
SELECT u.First_name as `first person` , s.First_Name as `second person`, Message.message  
FROM Message, DM, Conversation, User_Data as u, User_Data as s  
WHERE Conversation.Conversation_Type = "message" and Message.Conversation_C_ID =  
Conversation.C_ID and Conversation.DM_DM_ID = DM.DM_ID and  
DM.User_Data_User_ID = u.User_ID and DM.Second_person_User_ID = s.User_ID
```

Query 3 – Message retrieval of a Group chat Conversation

```
SELECT u.First_name as `first person`, Message.message, group_chat.Group_ID  
FROM Message, group_chat, Conversation, User_Data as u  
WHERE Conversation.Conversation_Type = "message" and Message.Conversation_C_ID =  
Conversation.C_ID and Conversation.Group_Chat_Group_Id = Group_Chat.Group_ID and  
Group_Chat.Group_Id = "8" and group_chat.member = u.user_Id
```

Query 4 – Time Sent, Time delivered and Time read of a Message

```
SELECT conversation.C_ID, message.message, conversation.Time_Sent,  
conversation.Time_delivered, conversation.Time_read
```

```
FROM Message, Conversation
```

```
WHERE Message.C_ID = Conversation.C_ID
```

Query 5 – Participants in a group chat

```
select u.First_Name, g.Group_Name
```

```
from members as gu, user_data as u, group_chat as g
```

```
where gu.Group_ID = g.Group_ID and gu.Member_ID = u.User_ID;
```

Query 6 – Common Groups Between 2 users

```
SELECT m.Member_ID, n.Member_ID
```

```
FROM members as m, members as n
```

```
WHERE m.Group_ID = n.Group_ID and m.Member_ID <>n.Member_ID
```

Query 7 - Update Status/Phone no./Profile Picture/Name

```
UPDATE User_Data as u,visible_contact_list as v
```

```
set v.Phone_Number="9810320617" ,u.`online status` = "Online",  
u.Visible_Contact_List_Phone_Number = "9810320617",u.`First_Name` = "AK"
```

Query 8 - To place the most recent conversation on top.

```
SELECT Time_delivered as chats  
FROM conversation  
ORDER BY Time_delivered DESC
```

Query 9 - To display whether online/offline.

```
SELECT user_data.First_Name  
FROM user_data  
WHERE user_data.`online status` = "Online"
```

Query 10 - To display last seen

```
SELECT recent_messages.g, recent_messages.d, MIN(recent_messages.diff)  
FROM recent_messages  
WHERE  
    (  
        CREATE VIEW [recent_messages] AS  
        SELECT Group_Chat.Group_Name as g, DM.DM_ID as d, TIMESTAMPDIFF(SECOND,  
        NOW(), Conversation.Time_Delievered) AS diff  
        FROM Message, Conversation, Group_Chat, DM
```

```
WHERE Message.Conversation_C_ID = Conversation.C_ID and  
(Conversation.Group_Chat_Group_ID = Group_Chat.Group_ID or  
Conversation.DM_DM_ID = DM.DM_ID.)
```

```
)
```

SQL Triggers

Trigger 1

```
CREATE TRIGGER delete_profile_photo  
AFTER INSERT  
ON  
user_data  
FOR EACH ROW  
SET new.Profile_Picture = null;
```

Trigger 2

```
CREATE TRIGGER block  
AFTER INSERT  
ON  
user_data  
FOR EACH ROW  
SET new.hide_details = 70;
```

SQL Indexing

Index 1

```
create index times on conversation (Time_Sent);
```

Index 2

```
create index messages on message (message);
```

Index 3

create index memb on members (Member_ID);

Index 4

create index user_d on user_data (Visible_Contact_List_Phone_Number);

Index 5

create index group_ch on group_chat (Group_Name);

Views And Grants

```
CREATE USER anurag@localhost  
IDENTIFIED BY 'Secure1Pass!';
```

1.

```
GRANT INSERT, UPDATE, DELETE
```



```
ON visible_contact_list  
TO anurag@localhost;
```

```
-- grants for all users
```

2.

```
GRANT  
    SELECT (First_Name,Last_Name, About,Profile_picture,hide_details,'online status'),  
    UPDATE(First_Name,Last_Name, About,Profile_picture)  
ON user_data  
TO anurag@localhost;
```

```
-- grants for all users
```

3.

```
GRANT INSERT,DELETE  
ON members  
TO anurag@localhost;
```

```
-- grants for group admin
```

```
-- grants for all users
```

4.

```
CREATE VIEW view_message AS  
    SELECT message  
FROM message
```

```
-- SELECT First_Name FROM DetailsView;
```

```
-- SELECT * FROM view_message;
```

5.

```
CREATE VIEW Details_View AS
    SELECT First_Name
FROM user_data
WHERE hide_details = 0;
```

-- or profile photo can be hidden