**In your queries, please use below fundamentals, make sure all below topics are covered while writing the queries in any of the queries:**

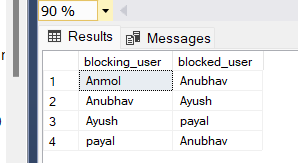
--- Self Join: Find all users who have blocked other users.

SELECT u1.usersname AS blocking\_user, u2.usersname AS blocked\_user

FROM Users\_Block\_List ubl

INNER JOIN Users u1 ON ubl.users\_id = u1.users\_id

INNER JOIN Users u2 ON ubl.blocked\_user\_id = u2.users\_id;



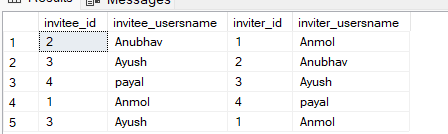
--- Inner Join: Find all users who have been invited to a chat and the inviter's username.

SELECT ci.invitee\_id, u.usersname AS invitee\_usersname, ci.inviter\_id, u2.usersname AS inviter\_usersname

FROM Chat\_Invite ci

INNER JOIN Users u ON ci.invitee\_id = u.users\_id

INNER JOIN Users u2 ON ci.inviter\_id = u2.users\_id;

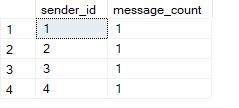


--- Group by: Count the number of messages sent by each user.

SELECT sender\_id, COUNT(\*) AS message\_count

FROM Message

GROUP BY sender\_id;



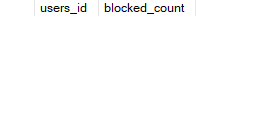
---having clause: Find users who have blocked more than one user.

SELECT users\_id, COUNT(blocked\_user\_id) AS blocked\_count

FROM Users\_Block\_List

GROUP BY users\_id

HAVING COUNT(blocked\_user\_id) > 1;

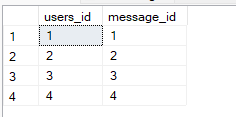


--- Sub Query: Find users who have favorited their own messages using a subquery.

SELECT users\_id, message\_id

FROM Message\_Favorite

WHERE users\_id IN (SELECT message\_id FROM Message\_Favorite);



---window function: Rank users based on the number of messages they've sent.

SELECT u.users\_id, u.usersname, m.message\_count,

RANK() OVER (ORDER BY m.message\_count DESC) AS message\_rank

FROM (

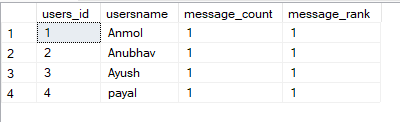
SELECT sender\_id AS users\_id, COUNT(\*) AS message\_count

FROM Message

GROUP BY sender\_id

) AS m

JOIN Users u ON m.users\_id = u.users\_id;

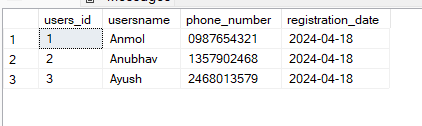


--- String Function: Find all users whose usernames start with 'A'.

SELECT \*

FROM Users

WHERE LEFT(usersname, 1) = 'A';



**Identify 4 Analytical queries for the database as per your understanding**

--\*Top Active Users\*: Find the top 10 users who have sent the most messages.

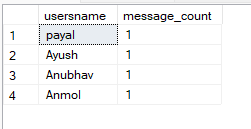
SELECT TOP 10 u.usersname, COUNT(m.message\_id) AS message\_count

FROM Users u

INNER JOIN Message m ON u.users\_id = m.sender\_id

GROUP BY u.usersname

ORDER BY message\_count DESC;



--Call Rating Analysis\*: Calculate the average call rating given by users and identify users with the highest and lowest average ratings.

SELECT TOP 1 users\_id, AVG(rating) AS avg\_rating

FROM Call\_Rating

GROUP BY users\_id

ORDER BY avg\_rating DESC;



--Find out which user\_id has the Lowest Avergae Call Rating

SELECT top 1 users\_id, AVG(rating) AS avg\_rating

FROM Call\_Rating

GROUP BY users\_id

ORDER BY avg\_rating ASC



--Most Popular Chat\*: Find the chat with the highest number of messages.

SELECT TOP 1 c.chat\_name, COUNT(m.message\_id) AS message\_count

FROM Chat c

INNER JOIN Message m ON c.chat\_id = m.chat\_id

GROUP BY c.chat\_name

ORDER BY message\_count DESC;



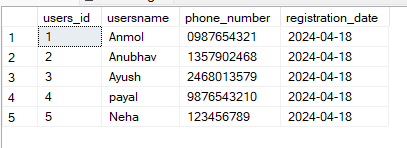
**Identify 4 queries which is required for regular operation in any OLTP environment**

1--Insert Data

INSERT INTO Users VALUES

(5, 'Neha', 123456789, '2024-04-18')

select \* from Users

****

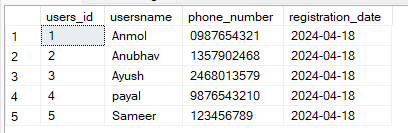
--2. Update Data

UPDATE Users

SET usersname = 'Sameer'

WHERE users\_id = 5

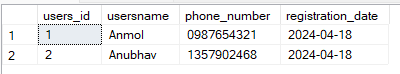
select \* from Users

****

--3. Retrieve Data

SELECT \* from Users

WHERE users\_id <3

****

--4. Delete Data

DELETE FROM Users

WHERE users\_id = 5

select \* from Users

