

## Facebook database

**Create the Facebook database with the following table structure.**

- **create database facebook;**

---

### 1. Users Table

```
create table users(  
  user_id int primary key ,  
  user_name varchar(100) not null,  
  email varchar(100) unique not null,  
  phone varchar(15) unique not null,  
  date_of_birth date not null,  
  gender varchar(10),  
  city varchar(50),  
  created_at datetime default current_timestamp  
);
```

---

### 2. Posts Table

```
create table posts(  
  post_id int primary key ,  
  user_id int not null,  
  content text not null,  
  image_url varchar(255),  
  created_at datetime default current_timestamp,  
  foreign key (user_id) references users(user_id)  
);
```

---

### 3. Likes Table

MUSKAN SINGH

```
create table likes(  
like_id int primary key ,  
user_id int not null,  
post_id int not null,  
created_at datetime default current_timestamp,  
foreign key (user_id) references users (user_id),  
foreign key (post_id) references posts (post_id)  
);
```

---

#### 4. Friends Table

```
create table friends(  
friendship_id int primary key,  
user_id int not null,  
friend_id int not null,  
status varchar(20) default 'pending',  
foreign key (user_id) references users(user_id),  
foreign key (friend_id) references users(user_id)  
);
```

---

#### 5. Comments Table

```
create table comments(  
comment_id int primary key ,  
user_id int not null,  
post_id int not null,  
comment_text text not null,  
created_at datetime default current_timestamp,  
foreign key (user_id) references users (user_id),
```

```
foreign key (post_id) references posts (post_id)  
);
```

---

## 6. Messages Table

```
create table messages(  
message_id int primary key ,  
sender_id int not null,  
receiver_id int not null,  
message_text text not null,  
created_at datetime default current_timestamp,  
foreign key (sender_id) references users(user_id),  
foreign key (receiver_id) references users(user_id)  
);
```

---

## 7. Notifications Table

```
create table notifications(  
notification_id int primary key ,  
user_id int,  
notification_type varchar(50),  
reference_id int,  
is_read boolean default false,  
created_at datetime default current_timestamp,  
foreign key (user_id) references users(user_id)  
);
```

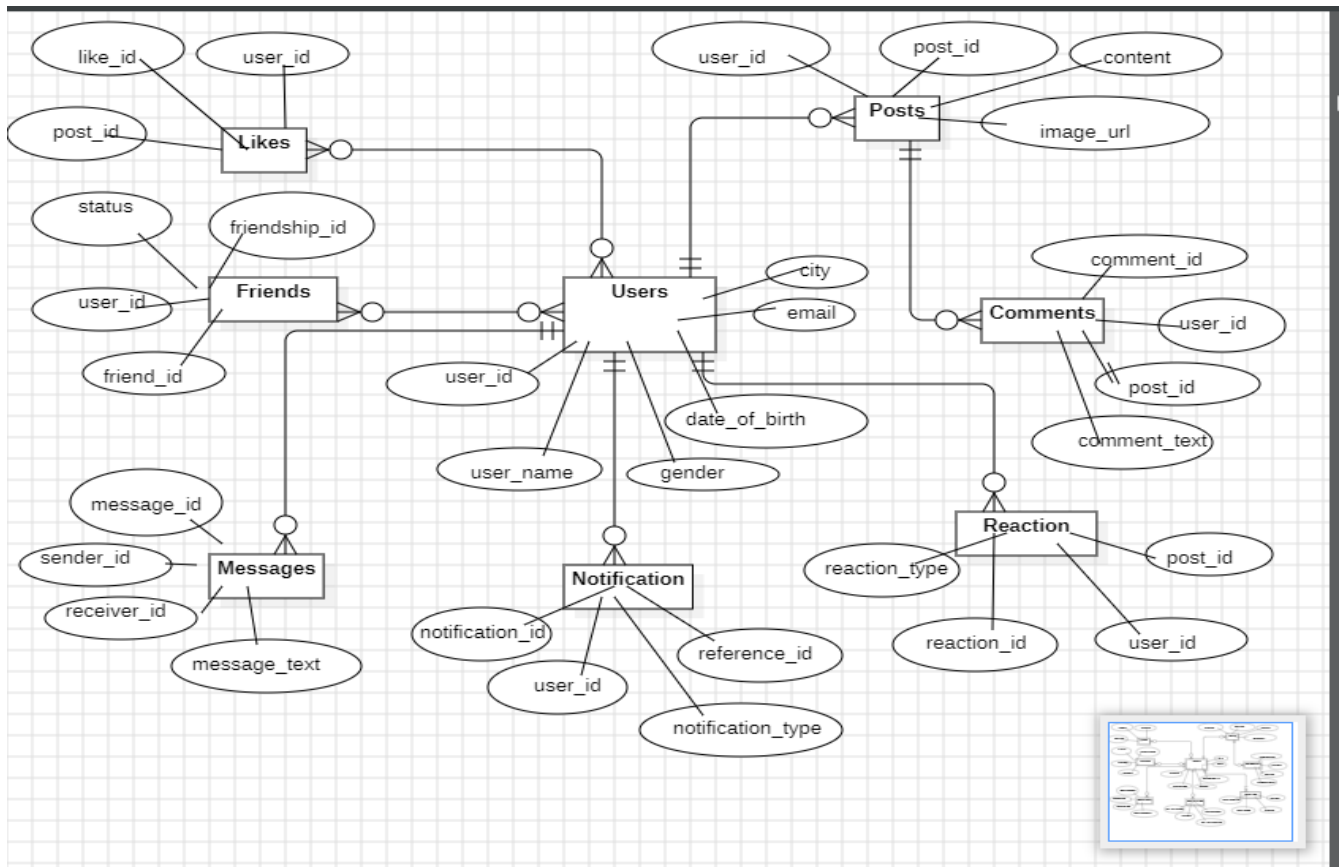
---

## 8. Reactions Table

```
create table reactions(  
  reaction_id int primary key ,  
  user_id int,  
  post_id int,  
  reaction_type varchar(50),  
  created_at datetime default current_timestamp,  
  foreign key (user_id) references users(user_id),  
  foreign key (post_id) references posts(post_id)  
);
```

---

**Draw the appropriate ER diagram to depict the following scenario.**



## Insert appropriate values in the tables (10 values in each table)

### 1. Users

insert into users (user\_id, user\_name, email, phone, date\_of\_birth, gender, city, age)

values

(101, 'Muskan Singh', 'muskan@example.com', '9956231478', '2024-01-01', 'Female', 'New York', '18'),

(102, 'Himu Yadav', 'himu@example.com', '9324877891', '1989-02-02', 'Male', 'Los Angeles', '25'),

(103, 'Animesh Gupta', 'animesh@example.com', '8865467892', '1988-03-03', 'Male', 'Chicago', '17'),

(104, 'Shraddha Ahire', 'shraddha@example.com', '9369478193', '2024-04-04', 'Female', 'Houston', '21'),

(105, 'Dinesh Jaiswar', 'dinesh@example.com', '7712486394', '1992-05-05', 'Male', 'Phoenix', '18'),

(106, 'Samruddhi Chaudhari', 'samu@example.com', '9148726895', '2024-06-06', 'Female', 'Philadelphia', '18'),

(107, 'Aniket Shirsat', 'aniket@example.com', '9845217963', '2024-07-07', 'Male', 'New York', '18'),

(108, 'Simran Gaikwad', 'simmi@example.com', '9756214785', '1993-08-08', 'Female', 'San Diego', '18'),

(109, 'Siddharth Pagare', 'sid@example.com', '9523478619', '2024-09-01', 'Male', 'Dallas', '17'),

(110, 'Tanu Korde', 'tanu@example.com', '9874124365', '1991-10-10', 'Female', 'San Francisco', '62');

---

### 2. Posts

insert into posts (post\_id, user\_id, content, image\_url)

values

(1, 101, 'Hello, world!', 'https://example.com/image1.jpg'),

(2, 102, 'Today was a great day!', 'https://example.com/image2.jpg'),

(3, 103, 'Just finished a book!', NULL),

```
(4, 104, 'Loving the sunny weather!', 'https://example.com/image3.jpg'),  
(5, 105, 'Excited for the weekend!', 'https://example.com/image4.jpg'),  
(6, 106, 'Trying out a new recipe.', NULL),  
(7, 107, 'Can't wait for the concert!', 'https://example.com/image5.jpg'),  
(8, 108, 'Went hiking today!', 'https://example.com/image6.jpg'),  
(9, 109, 'Caught up with an old friend.', NULL),  
(10, 110, 'Just got a new job!', 'https://example.com/image7.jpg');
```

---

### 3. Likes

```
insert into likes (like_id, user_id, post_id)
```

```
values
```

```
(201, 101, 1),  
(202, 102, 2),  
(203, 103, 3),  
(204, 104, 4),  
(205, 105, 5),  
(206, 106, 6),  
(207, 107, 7),  
(208, 108, 8),  
(209, 109, 9),  
(210, 110, 10);
```

---

### 4. Friends

```
insert into friends(friendship_id,user_id, friend_id, status)
```

```
values
```

```
(1,101, 101, 'accepted'),  
(2,102, 102, 'pending'),
```

MUSKAN SINGH

```
(3,103, 103, 'accepted'),  
(4,104, 104, 'declined'),  
(5,105, 105, 'pending'),  
(6,106, 106, 'accepted'),  
(7,107, 107, 'accepted'),  
(8,108, 108, 'pending'),  
(9,109, 109, 'declined'),  
(10,110, 110, 'accepted');
```

---

## 5. Comments

insert into comments (comment\_id, user\_id, post\_id, comment\_text)

values

```
(1, 101, 1, 'Great post!'),  
(2, 102, 1, 'Thanks for sharing!'),  
(3, 103, 2, 'Very informative.'),  
(4, 104, 2, 'I completely agree!'),  
(5, 105, 3, 'This is awesome!'),  
(6, 106, 3, 'Can you elaborate more?'),  
(7, 107, 1, 'Loved this!'),  
(8, 108, 2, 'Interesting perspective.'),  
(9, 109, 3, 'What a fantastic read!'),  
(10, 110, 1, 'I found this really helpful.');
```

---

## 6. Messages

insert into messages (message\_id, sender\_id, receiver\_id, message\_text)

values

```
(1, 101, 102, 'Hey, how are you?'),
```

MUSKAN SINGH

```
(2, 102, 101, 'I'm good, thanks! How about you?'),  
(3, 103, 104, 'Did you finish the project?'),  
(4, 104, 103, 'Yes, I just sent it over.'),  
(5, 105, 106, 'Let's catch up soon!'),  
(6, 106, 105, 'Definitely, when are you free?'),  
(7, 107, 108, 'Happy birthday!'),  
(8, 108, 107, 'Thank you!'),  
(9, 109, 110, 'Can you help me with the assignment?'),  
(10, 110, 109, 'Sure, I'll send you the notes.');
```

---

## 7. Notification

```
insert into notifications (notification_id, user_id, notification_type, reference_id, is_read)  
values
```

```
(1, 101, 'like', 201, false),  
(2, 102, 'comment', 1, false),  
(3, 103, 'friend_request', 104, false),  
(4, 104, 'message', 1, true),  
(5, 105, 'like', 202, false),  
(6, 106, 'friend_request', 107, false),  
(7, 107, 'comment', 3, false),  
(8, 108, 'message', 2, true),  
(9, 109, 'like', 203, false),  
(10, 110, 'mention', 4, false);
```

---



MUSKAN SINGH

## 8. Reactions

```
insert into reactions (reaction_id, user_id, post_id, reaction_type)
```

```
values
```

```
(1, 101, 1, 'like'),
```

```
(2, 102, 1, 'love'),
```

```
(3, 103, 2, 'wow'),
```

```
(4, 104, 2, 'sad'),
```

```
(5, 105, 3, 'like'),
```

```
(6, 106, 3, 'angry'),
```

```
(7, 107, 1, 'haha'),
```

```
(8, 108, 2, 'love'),
```

```
(9, 109, 3, 'like'),
```

```
(10, 110, 1, 'wow');
```

MUSKAN SINGH

Question :

1. Get all users' details from the Users table.

ANS:

**select \* from users;**

2. Get all posts created by a user with the user\_id = 101.

ANS:

**select \* from posts**

**where user\_id = 101;**

3. Find all users from the city of "New York".

ANS:

**select \* from users**

**where city = 'New York';**

4. Get all friends of user with user\_id = 101 whose friendship status is 'Accepted'.

ANS:

**select \* from friends**

**where user\_id = 101 and status = 'accepted';**

5. Find all posts created between 2024-01-01 and 2024-09-01.

ANS:

**select \* from posts**

**where created\_at between 2024-01-01 and 2024-09-01;**

6. Get details of users from the city of "Chicago", "Los Angeles", or "San Francisco".

ANS:

**select city from users**

MUSKAN SINGH

**where city in ('chicago', 'Los Angeles', 'San Francisco');**

7. Find all users whose email address ends with "@gmail.com".

ANS:

```
select * from users  
where email like '@gmail.com';
```

8. Get the number of posts created by each user.

ANS:

```
select user_id, count(post_id) as number_of_post  
from posts  
group by user_id;
```

9. Get the total number of likes on each post.

ANS:

```
select post_id count(like_id) as number of_like  
from likes  
group by post_id;
```

10. Get all comments on a specific post along with the user name of the commenter.

ANS:

```
select users.user_name, comments.comment_text  
from comments  
inner join users on comments.user_id = users.user_id  
where comments.post_id = 1;
```

11. Get all posts liked by a specific user along with the post content and the user who posted it.

ANS:

```
select posts.content, users.user_name as author
```

MUSKAN SINGH

**from likes**

**inner join posts on likes.post\_id = posts.post\_id**

**inner join users on posts.user\_id = users.user\_id**

**where likes.user\_id in (101,102) ;**

12. Get the most liked post.

ANS:

**select post\_id, count(like\_id) as most\_liked**

**from likes**

**group by post\_id;**

13. Find the users who haven't posted anything yet.

ANS:

**select users.user\_id, users.user\_name**

**from users**

**left join posts on users.user\_id = posts.user\_id**

**where posts.post\_id is null ;**

14. Get the first 10 characters of each post's content.

ANS:

**select post\_id, substring(content, 1, 10) as short\_content from posts;**

15. Convert all user names to uppercase.

ANS:

**select user\_id, upper(user\_name) as name from users;**

16. Get the top 5 users who have received the most likes on their posts.

ANS:

**select user\_id, post\_id from likes**

MUSKAN SINGH

**limit 5;**

17. Get the users who have more than 100 friends.

ANS:

```
select user_id, friend_id from friends  
where friend_id > 100;
```

18. Show the friendship status as 'Friends' if the status is 'Accepted', 'Pending' if it's pending, and 'Not Friends' for any other status.

ANS:

```
select user_id, friendship_id,status,  
case  
when status = 'accepted' then 'friends'  
when status = 'pending' then 'pending'  
else 'not friends'  
end as friend_status  
from friends;
```

19. Get all private messages between two users (user\_id = 101 and user\_id = 102).

ANS:

```
select sender_id, receiver_id, message_text, created_at from messages  
where (select sender_id = 101 and receiver_id = 102)  
or (select sender_id = 102 and receiver_id = 101)  
order by created_at;
```

20. Categorize the post length as 'Short', 'Medium', or 'Long' based on character count (e.g., < 50 characters = 'Short', 50-100 = 'Medium', > 100 = 'Long').

ANS:

```
select user_id, post_id, content,
```

MUSKAN SINGH

```
case
when content < 50 then 'short'
when content between 50 and 100 then 'medium'
when content > 100 then 'long'
end as post_length
from posts;
```

21. Display the user's age category as 'Minor' if their age is less than 18, 'Adult' if their age is between 18 and 60, and 'Senior' if their age is greater than 60.

ANS:

```
select user_id, age,
case
when age < 18 then 'Minor'
when age between 18 and 60 then 'Adults'
when age > 60 then 'Senior'
end as age_category
from users;
```

22. Get the total number of reactions for a specific post and categorizes them by reaction type.

ANS:

```
select reaction_type, count(*) as total_reaction
from reactions
where post_id = 3
group by reaction_type;
```

23. To retrieve all unread notifications for a specific user, ordered by the date created.

ANS:

```
select * from notifications
where user_id = 103 and is_read = 'false'
```

MUSKAN SINGH

**order by created\_at desc;**

24. Get reaction\_type from reactions with user\_id = 107 .

ANS:

**select reaction\_id, reaction\_type from reactions  
where user\_id = 107;**

25. Get notification\_type from notifications with reference\_id = 1.

ANS:

**select user\_id, notification\_type, reference\_id from notifications  
where reference\_id = 1;**

26. Convert all reaction type to name length.

ANS:

**select reaction\_type, length(reaction\_type) as name\_length from reactions;**

MUSKAN SINGH