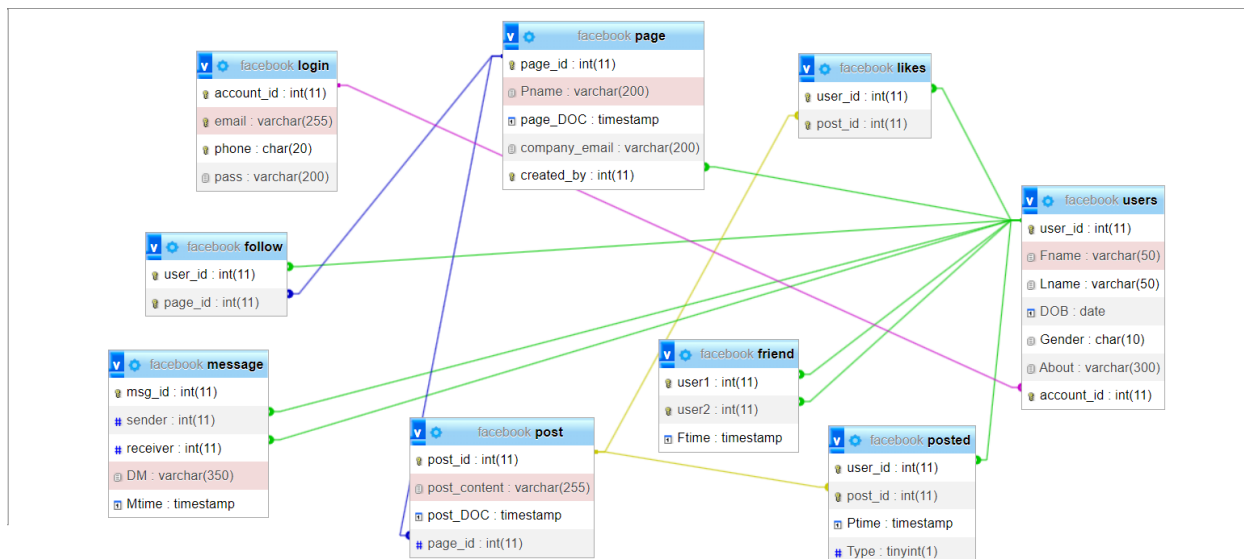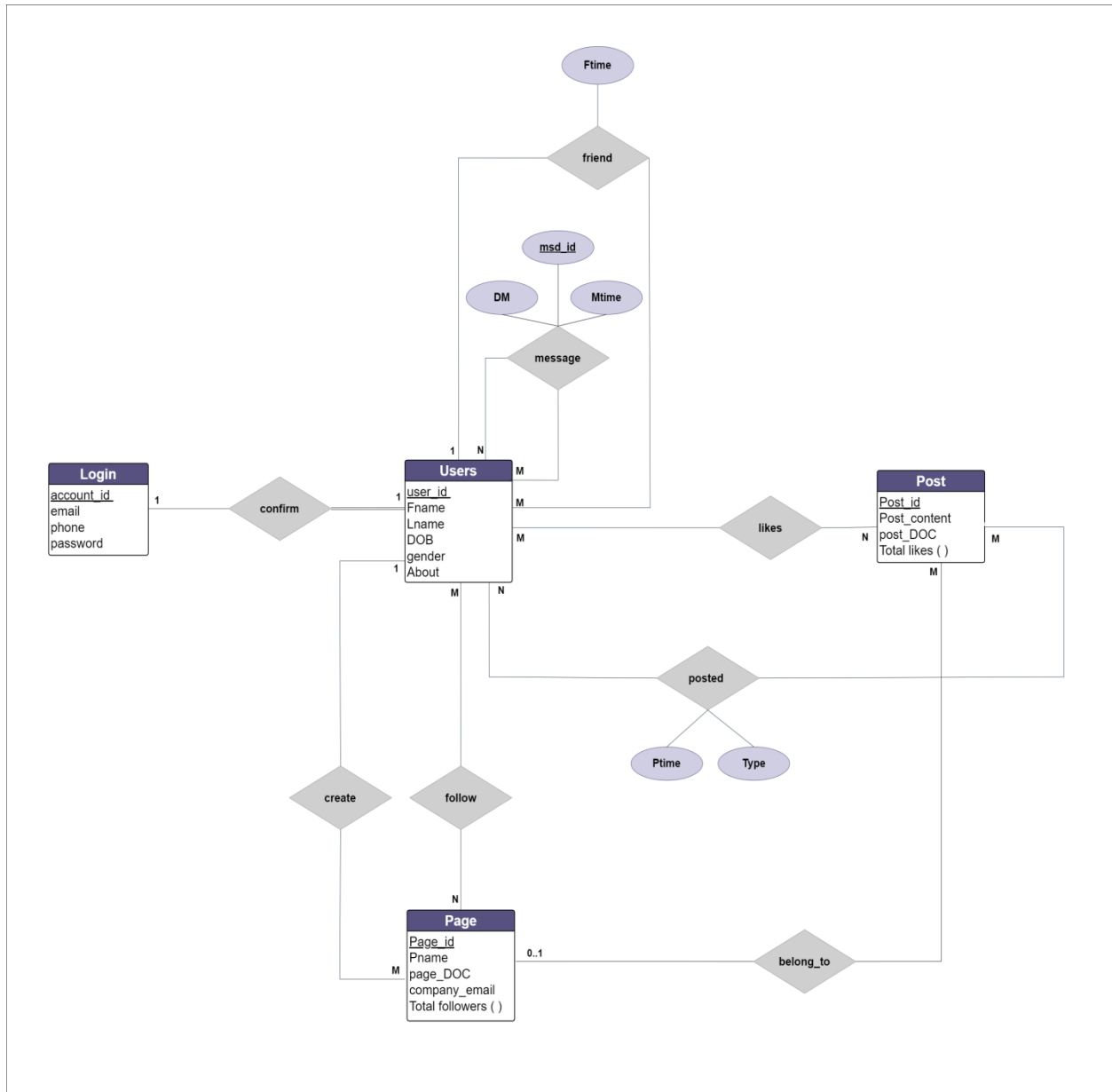# 1) Introduction:

The Facebook database has main entities such that (Login ,User, Page ,Post ,Friend ,Chat).

❖ These entities has relations between each other:
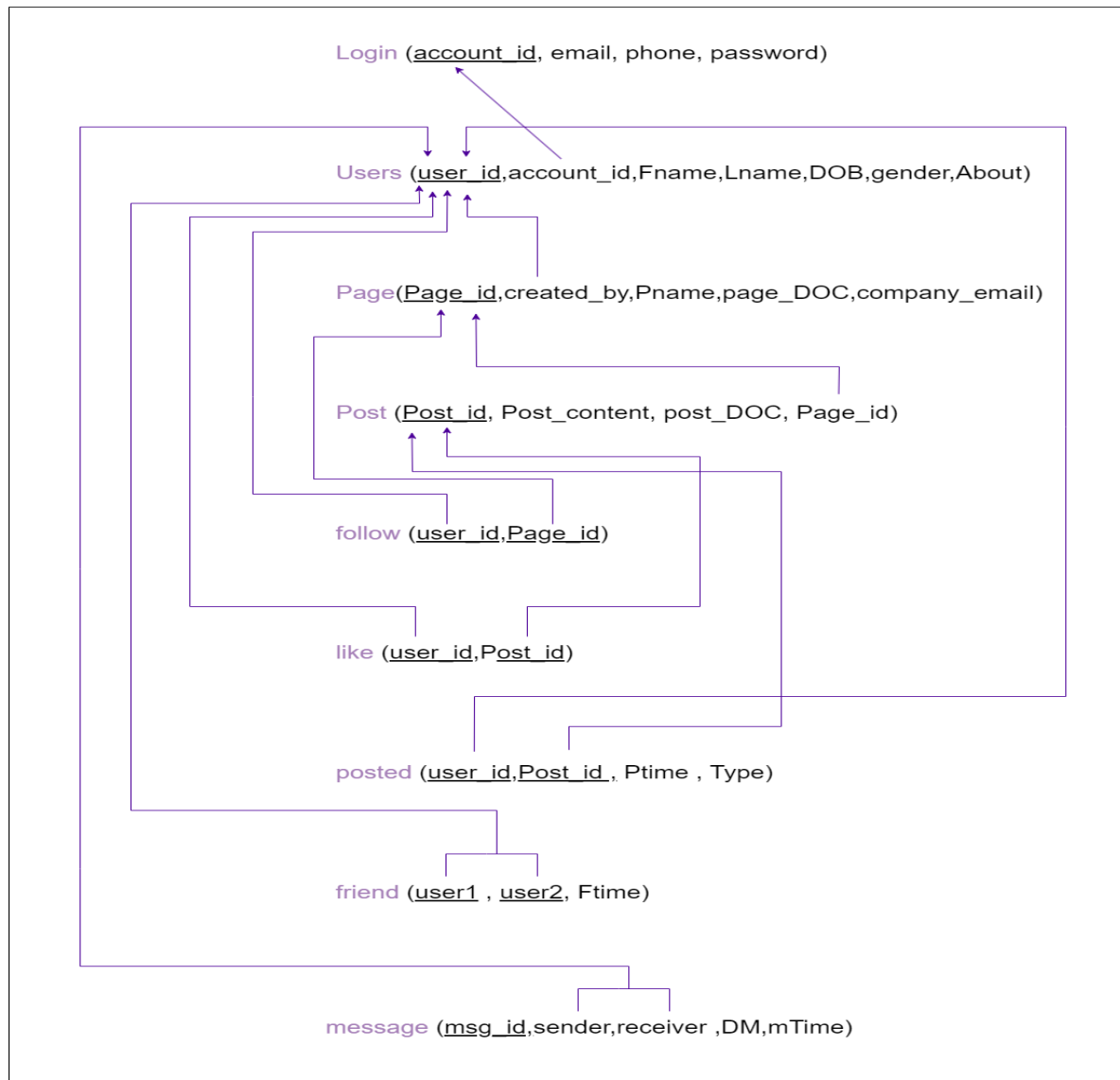
➢ Login has a Strong relation with User

➢ One User can create many Pages

➢ Many Users can Follow many Pages and many Pages can Followed by many Users

➢ One User can be a Friend for many Users

➢ Many Users can send message to many Users

➢ Many Users can Like many Posts and many Posts can liked be many Users

➢ Many Users can Post many Posts and many Posts can posted be many Users

➢ Many Posts can belong to Page or Profile.

2) ER Diagram for Facebook database :

3) Schema for Facebook database:

Login (account_id, email, phone, password)

Users (user_id, account_id, Fname, Lname, DOB, gender, About)

Page (Page_id, created_by, Pname, page_DOC, company_email)

Post (Post_id, Post_content, post_DOC, Page_id)

follow (user_id, Page_id)

like (user_id, Post_id)

posted (user_id, Post_id, Ptime, Type)

friend (user1, user2, Ftime)

message (msg_id, sender, receiver, DM, mTime)

4) Apply schema in SQL:

❖ Create Tables in SQL:

CREATE DATABASE Facebook;

create table Users(user_id INT PRIMARY KEY AUTO_INCREMENT,
        Fname Varchar(50) not null,
        Lname Varchar(50) not null,
        DOB DATE not null,
        Gender CHAR(10) not null,
        About Varchar(300),
     account_id int not null UNIQUE,
FOREIGN key(account_id) REFERENCES Login(account_id));

CREATE table Page(page_id INT PRIMARY KEY AUTO_INCREMENT,
     Pname varchar(200) not null,
       page_DOC TIMESTAMP DEFAULT
CURRENT_TIMESTAMP() not null,
       company_email varchar(200),
       created_by INT Not Null UNIQUE,
FOREIGN key(created_by) REFERENCES Users(user_id));

create table Post(post_id INT PRIMARY KEY AUTO_INCREMENT,
       post_content varchar(255) not null,
       post_DOC timestamp DEFAULT CURRENT_TIMESTAMP()
not null,
       page_id int ,
FOREIGN KEY(page_id) REFERENCES Page(page_id));

```sql
create table follow(user_id int,
            page_id int,
FOREIGN key(user_id) REFERENCES Users(user_id),
FOREIGN key(page_id) REFERENCES Page(page_id ),
PRIMARY key(user_id , page_id));


create table likes(user_id int,
            post_id int,
FOREIGN key(user_id) REFERENCES Users(user_id),
FOREIGN key(post_id) REFERENCES Post(post_id ),
PRIMARY key(user_id , post_id));


create table posted(user_id int,
            post_id int,
            Ptime timestamp DEFAULT CURRENT_TIMESTAMP() not
null,
            Type TINYINT(1) DEFAULT 0 not null,
FOREIGN key(user_id) REFERENCES Users(user_id),
FOREIGN key(post_id ) REFERENCES Post(post_id ),
PRIMARY key (user_id , post_id));


CREATE TABLE friend(user1 INT ,
            user2 INT ,
            Ftime TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
not null,
FOREIGN KEY (user1) REFERENCES Users (user_id),
FOREIGN KEY (user2) REFERENCES Users (user_id),
PRIMARY KEY (user1 , user2));
```

CREATE TABLE message(msg_id INT PRIMARY KEY AUTO_INCREMENT,

        sender INT not null,

        receiver INT not null,

        DM VARCHAR(350) not null,

        Mtime TIMESTAMP DEFAULT CURRENT_TIMESTAMP() not null,

FOREIGN KEY (sender) REFERENCES Users (user_id),

FOREIGN KEY (receiver) REFERENCES Users (user_id));

❖ **Insert data into Login Table and set id to be counted in the Users Table :**

INSERT INTO Login (email, phone, pass)

VALUES ('8888@example.com', '0000-1234', '8887778888');

SET @account_id = LAST_INSERT_ID();

INSERT INTO Users (Fname, Lname, DOB, Gender, About, account_id)

VALUES ('John', 'Doe', '1990-01-01', 'Male', 'About me', @account_id);

✓ Repeat this step to add another data for user.

	user_id	Fname	Lname	DOB	Gender	About	account_id
☐ 🖉 Edit ⁝ Copy ⊖ Delete	1	John	Doe	1990-01-01	Male	About me	1
☐ 🖉 Edit ⁝ Copy ⊖ Delete	2	ali	ahmed	1950-01-01	Male	About me	2
☐ 🖉 Edit ⁝ Copy ⊖ Delete	3	mai	ali	1999-03-01	Female	About me	3
☐ 🖉 Edit ⁝ Copy ⊖ Delete	4	aya	ahmed	1990-11-01	Female	About me	4
☐ 🖉 Edit ⁝ Copy ⊖ Delete	5	max	Doe	1993-01-09	Male	About me	5

↑   ☐ Check all   With selected: 🖉 Edit   ⁝ Copy   ⊖ Delete   🖳 Export

❖ Insert post in Page created by user whose name is mai

INSERT INTO post (post_content,page_id) VALUES('bts forever',(SELECT page_id FROM page WHERE created_by =(SELECT user_id FROM users WHERE Fname ='mai')));

←T→			post_id	post_content	post_DOC	page_id
☐	🖉 Edit ⧉ Copy	⊖ Delete	1	bts forever	2023-05-19 15:43:41	1
☐	🖉 Edit ⧉ Copy	⊖ Delete	2	welcome!!	2023-05-19 15:51:53	NULL
☐	🖉 Edit ⧉ Copy	⊖ Delete	3	hello world!!	2023-05-19 15:51:54	NULL

↑__ ☐ Check all   With selected: 🖉 Edit   ⧉ Copy   ⊖ Delete   📰 Export

❖ Insert post in page which created by user and set id for this post

INSERT INTO page (Pname,company_email,created_by) VALUES('bts','bighit@gmail.com',(SELECT user_id FROM users WHERE Fname ='mai'));

SET @post _id = LAST_INSERT_ID();

INSERT INTO posted (post_id,user_id) VALUES(@post_id,(SELECT user_id FROM users WHERE Fname ='ali'));

INSERT INTO posted (post_id,user_id) VALUES(1,(SELECT user_id FROM users WHERE Fname ='mai'));

←T→			page_id	Pname	page_DOC	company_email	created_by
☐	🖉 Edit ⧉ Copy	⊖ Delete	1	bts	2023-05-19 15:39:42	bighit@gmail.com	3

↑__ ☐ Check all   With selected: 🖉 Edit   ⧉ Copy   ⊖ Delete   📰 Export

❖ **User Follow the page which created by another user and count that how many users follow each page**

INSERT INTO follow (page_id,user_id) VALUES ((SELECT page_id FROM page WHERE created_by =(SELECT user_id FROM users WHERE Fname ='mai')),(SELECT user_id FROM users WHERE Fname ='John'));

✓ Repeat this step with the same or with another Fname

			user_id	page_id
☐	🖊 Edit ᴷ Copy ⊖ Delete		1	1
☐	🖊 Edit ᴷ Copy ⊖ Delete		2	1
☐	🖊 Edit ᴷ Copy ⊖ Delete		4	1
☐	🖊 Edit ᴷ Copy ⊖ Delete		5	1

SELECT COUNT(user_id) AS Total_Followers ,page_id FROM follow GROUP BY page_id;

Total_Followers	page_id
4	1

❖ **one user likes a post which created by another user and count a total likes for each post**

INSERT INTO likes (post_id,user_id) VALUES ((SELECT post_id FROM posted WHERE user_id =(SELECT user_id FROM users

WHERE Fname ='mai')),(SELECT user_id FROM users WHERE Fname ='John'));

- ✓ Repeat this step with the same or with another Fname and post_id

|				user_id	post_id
☐	✐ Edit	⊒ Copy	⊖ Delete	1	1
☐	✐ Edit	⊒ Copy	⊖ Delete	1	3
☐	✐ Edit	⊒ Copy	⊖ Delete	4	1
☐	✐ Edit	⊒ Copy	⊖ Delete	5	3

SELECT COUNT(user_id) AS Total_Likes ,post_id FROM likes GROUP BY post_id;

Total_Likes	post_id
2	1
2	3

❖ User1 be a friend with User2

INSERT INTO friend (user1,user2) VALUES ((SELECT user_id FROM users WHERE Fname ='John'),(SELECT user_id FROM users WHERE Fname ='ali'));

- ✓ Repeat this step with different Fname

				user1	user2	Ftime
☐	✐ Edit	⊒ Copy	⊖ Delete	1	2	2023-05-19 15:57:27
☐	✐ Edit	⊒ Copy	⊖ Delete	3	4	2023-05-19 15:57:27
⬆		☐ Check all	With selected:	✐ Edit	⊒ Copy	⊖ Delete

❖ Users message each other

INSERT INTO message(DM,sender,receiver) VALUES ('Haw are you ?',(SELECT user_id FROM users WHERE Fname ='mai'),(SELECT user_id FROM users WHERE Fname ='aya'));

INSERT INTO message(DM,sender,receiver) VALUES ('Fine!! and You?',(SELECT user_id FROM users WHERE Fname ='aya'),(SELECT user_id FROM users WHERE Fname ='mai'));

←T→	msg_id	sender	receiver	DM	Mtime
☐ 🖊Edit ⧉Copy ⊖Delete	1	3	4	Haw are you ?	2023-05-19 16:38:50
☐ 🖊Edit ⧉Copy ⊖Delete	2	4	3	Fine!! and You?	2023-05-19 16:42:58

↑ ☐ Check all *With selected:* 🖊Edit ⧉Copy ⊖Delete 🗐Export

❖ Join follow, users, page tables to get the followers of each page

select Pname as PageName , Fname as Follower From page p join follow f USING(page_id) join users u on (u.user_id = f.user_id);

PageName	Follower
bts	John
bts	ali
bts	aya

❖ Join Friends, users tables to get the friends

select u1.Fname as 'User name' , u2.Fname as 'Friend' FROM friend f JOIN users u1 on (u1.user_id = f.user1) JOIN users u2 on (u2.user_id = f.user2);

User name	Friend
John	ali
mai	aya

❖ Join users, post, posted tables to get the posts of the users

SELECT Fname as 'User name' , post_content 'Posted' , Ptime as 'On' FROM users u JOIN posted p USING (user_id) JOIN post USING(post_id);

User name	Posted	On
mai	bts forever	2023-05-19 15:53:11
aya	welcome!!	2023-05-19 15:51:54
ali	hello world!!	2023-05-19 15:51:54

5) Database Design Source
https://shorturl.at/AHOP3