AIN SHAMS UNIVERSITY

FACULTY OF ENGINEERING

INTERNATIONAL CREDIT HOURS PROGRAM

SOCIAL MEDIA PLATFORM DATABASE

CSE333 - Database Systems

Submitted by:

Youssef Mohamed El Nagy	20P8842
Seif Eldin Ashraf Mahmoud	20P7101
Youssef Mohamed Ahmed	20P2810
Tarek Khaled Ezzat	20P1087
Adham Hatem Hanafy	20P8384

Submitted to:

Dr. Hoda Korashy

Eng. Lara Wahby

Table of Contents

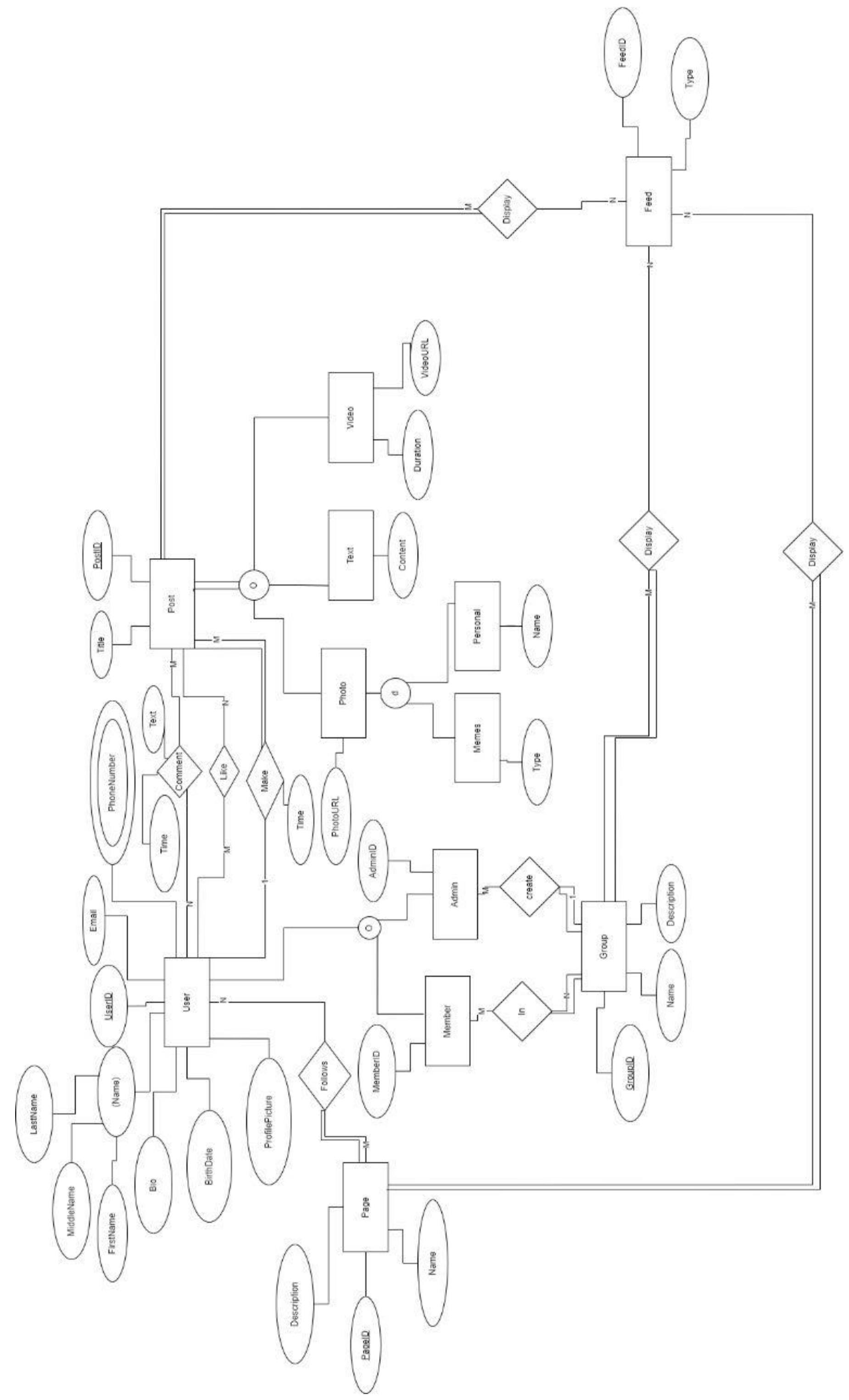
Introduction

This database is created for a social networking platform. It contains user. Each user has a user ID, first name, middle name, last name, email, password, bio, a profile picture, and each user can have multiple phone numbers. Users which are also admins to groups have an admin ID. Users can be admins to many groups. Each user and group have a specific feed, and multiple users can join multiple groups, and each group has members. Each user can also follow multiple pages. Users can also make posts. A post can be text, photo or video or a combination of any of them. Each photo can be either be a personal photo or a meme or neither. Users can also like many posts. Users can message each other.

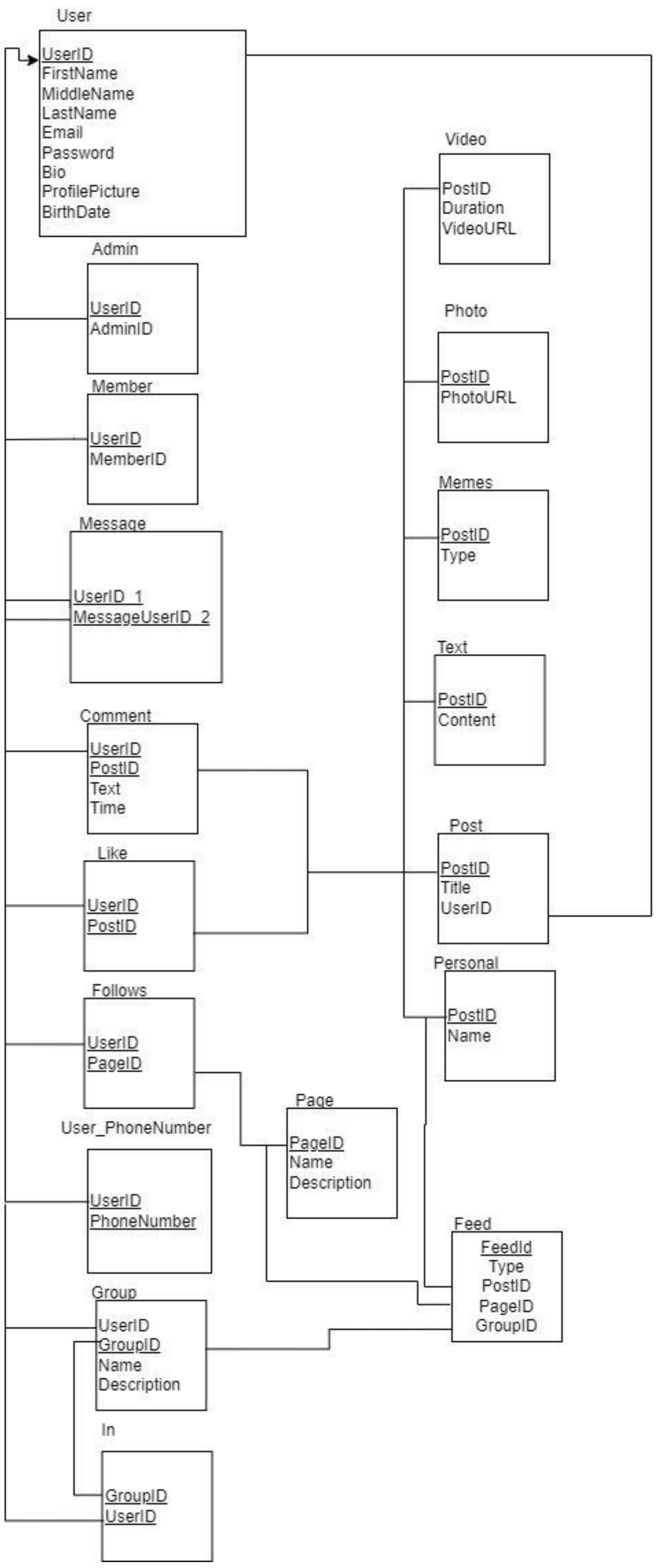
Assumptions

- User has attributes (userID, password, email, name, bio, birthdate, profile picture, phone number(s))
- UserID is unique (primary key).
- Phone number is a multivalue attribute.
- Name is a composite attribute; consists of First, Middle, Last Names.
- Users can be Member, Admin, etc. or could be assigned multiple types.
- Member has memberID attribute while Admin has adminID attribute.
- Page entity has pageID, description, Name.
- PageID is unique (primary key).
- Group entity has groupID, Name, Description.
- GroupID is unique (primary key).
- Feed entity has FeedID, type.
- FeedID is unique (primary key).
- Post entity has postID (unique), title.
- Posts could be photo, text, video, or a combination of any of them.
- Photo entity has photoURL as an attribute.
- Text entity has content attribute.
- Video entity has videoURL, duration as attributes.
- A photo could be classified as personal, memes, etc. but can't be classified as more than one type.
- Meme has attribute Type, Personal has attribute Name.
- A Group can display more than 1 feeds.
- A feed can display 0 or 1 groups.
- A page can display multiple feeds (0+).
- A feed can display 0 or 1 pages.
- A feed contains multiple posts.
- A feed can display 0 or more posts.
- User may message many users, each one separately.
- Members may be in many groups.
- Group must have many members.
- Group must be created by 1 Admin.
- Admin may create many groups.
- Users may follow many pages.
- Pages must have many followers.
- User may comment on many posts.
- Posts may have many comments by many users.
- Comments have time and text attributes.
- User may like many posts.
- A post may have many likes.
- Users may make many posts.
- Each post must have 1 user, and its time is recorded.

Enhanced Entity Relation Diagram

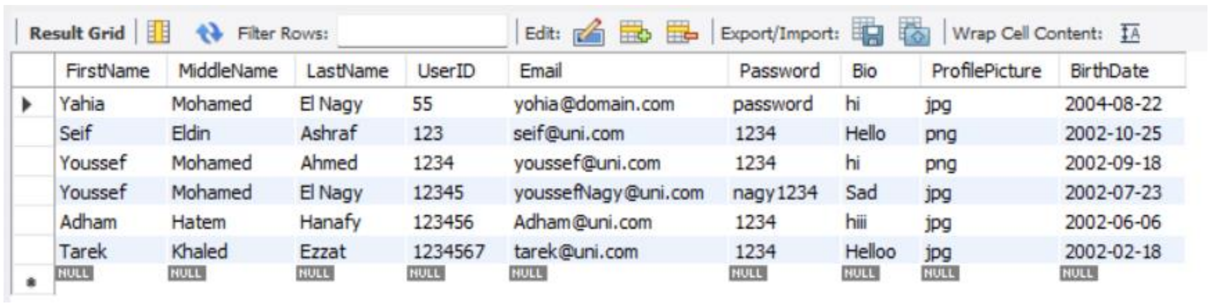


Database Schema



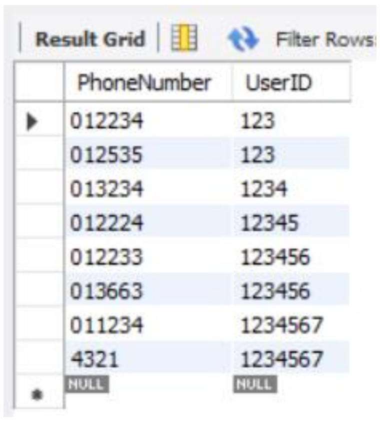
Important data and reports

Tables & Relations



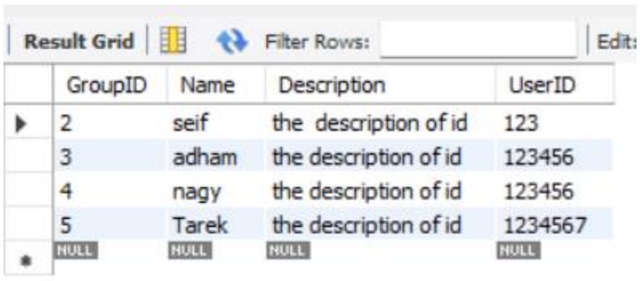
FirstName	MiddleName	LastName	UserID	Email	Password	Bio	ProfilePicture	BirthDate
Yahia	Mohamed	El Nagy	55	yohia@domain.com	password	hi	jpg	2004-08-22
Seif	Eldin	Ashraf	123	seif@uni.com	1234	Hello	png	2002-10-25
Youssef	Mohamed	Ahmed	1234	youssef@uni.com	1234	hi	png	2002-09-18
Youssef	Mohamed	El Nagy	12345	youssefNagy@uni.com	nagy1234	Sad	jpg	2002-07-23
Adham	Hatem	Hanafy	123456	Adham@uni.com	1234	hiii	jpg	2002-06-06
Tarek	Khaled	Ezzat	1234567	tarek@uni.com	1234	Helloo	jpg	2002-02-18
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 1 User Table



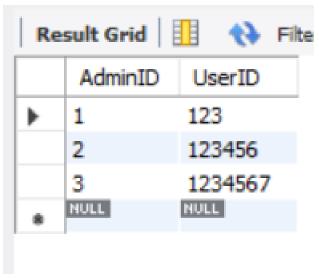
PhoneNumber	UserID
012234	123
012535	123
013234	1234
012224	12345
012233	123456
013663	123456
011234	1234567
4321	1234567
NULL	NULL

Figure 2 Phone numbers



GroupID	Name	Description	UserID
2	seif	the description of id	123
3	adham	the description of id	123456
4	nagy	the description of id	123456
5	Tarek	the description of id	1234567
NULL	NULL	NULL	NULL

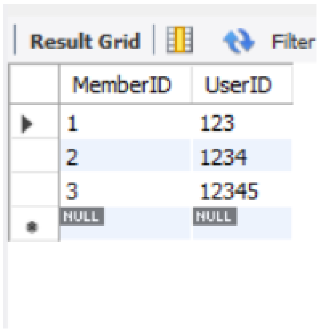
Figure 3 Group Table



Result Grid | Filte

	AdminID	UserID
▶	1	123
	2	123456
	3	1234567
*	NULL	NULL

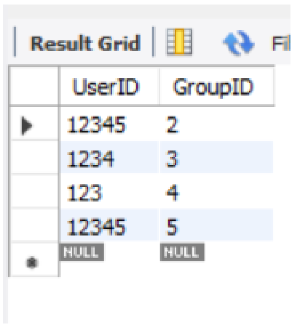
Figure 4 Admin Table



Result Grid | Filter

	MemberID	UserID
▶	1	123
	2	1234
	3	12345
*	NULL	NULL

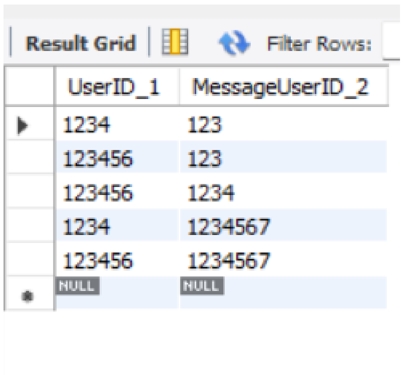
Figure 5 Member Table



Result Grid | Fi

	UserID	GroupID
▶	12345	2
	1234	3
	123	4
	12345	5
*	NULL	NULL

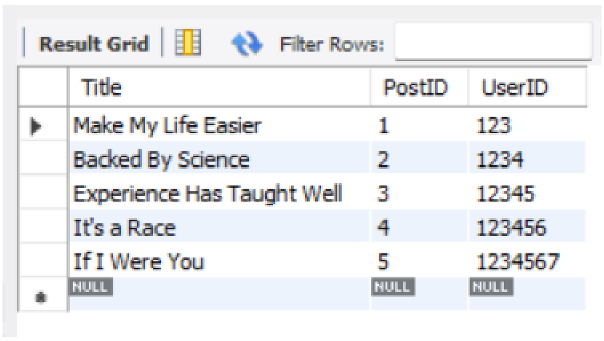
Figure 6 In Relationship (Member in group)



Result Grid | Filter Rows:

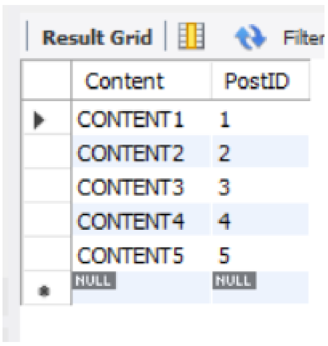
	UserID_1	MessageUserID_2
▶	1234	123
	123456	123
	123456	1234
	1234	1234567
	123456	1234567
*	NULL	NULL

Figure 7 Message Relationship (User-User)



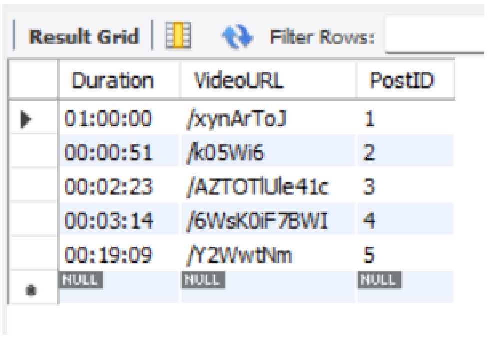
Title	PostID	UserID
Make My Life Easier	1	123
Backed By Science	2	1234
Experience Has Taught Well	3	12345
It's a Race	4	123456
If I Were You	5	1234567
NULL	NULL	NULL

Figure 8 Posts Table



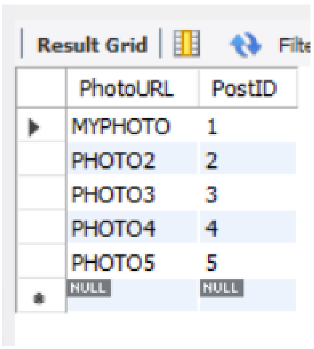
Content	PostID
CONTENT1	1
CONTENT2	2
CONTENT3	3
CONTENT4	4
CONTENT5	5
NULL	NULL

Figure 9 Text Table



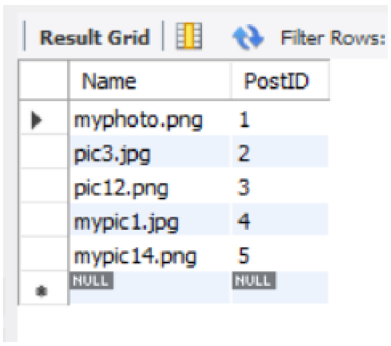
Duration	VideoURL	PostID
01:00:00	/xynArToJ	1
00:00:51	/k05Wi6	2
00:02:23	/AZTOTlUle41c	3
00:03:14	/6WsK0iF7BWI	4
00:19:09	/Y2WwtNm	5
NULL	NULL	NULL

Figure 10 Video Table



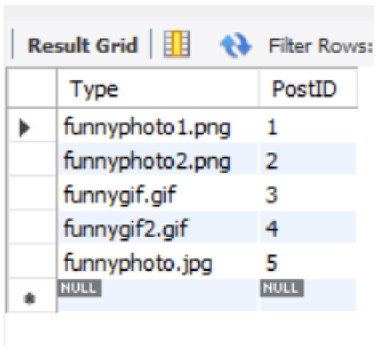
PhotoURL	PostID
MYPHOTO	1
PHOTO2	2
PHOTO3	3
PHOTO4	4
PHOTO5	5
NULL	NULL

Figure 11 Photo Table



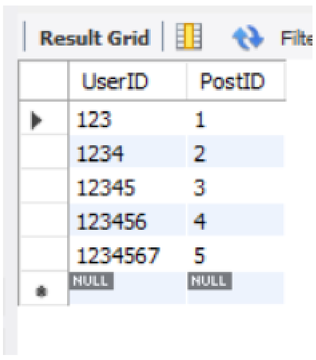
Name	PostID
myphoto.png	1
pic3.jpg	2
pic12.png	3
mypic1.jpg	4
mypic14.png	5
NULL	NULL

Figure 12 Personal Table



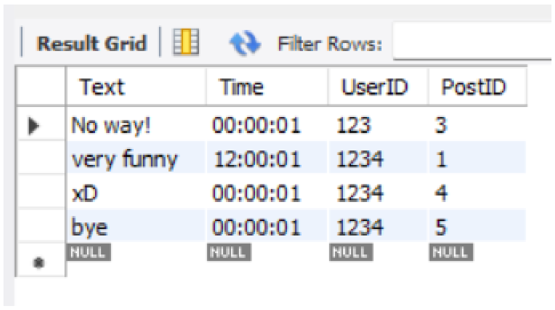
Type	PostID
funnyphoto1.png	1
funnyphoto2.png	2
funnygif.gif	3
funnygif2.gif	4
funnyphoto.jpg	5
NULL	NULL

Figure 13 Meme Table



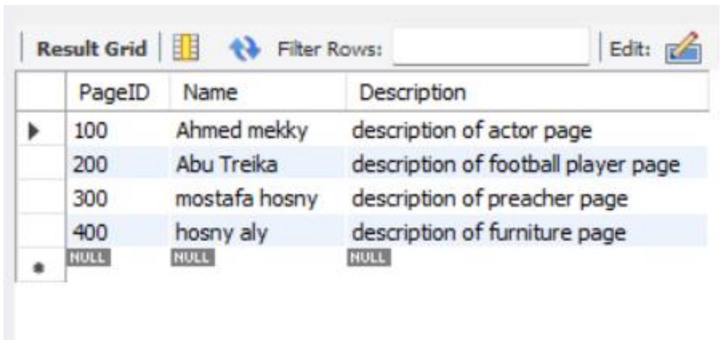
UserID	PostID
123	1
1234	2
12345	3
123456	4
1234567	5
NULL	NULL

Figure 14 Like Relationship (User likes post)



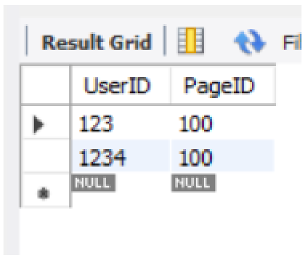
Text	Time	UserID	PostID
No way!	00:00:01	123	3
very funny	12:00:01	1234	1
xD	00:00:01	1234	4
bye	00:00:01	1234	5
NULL	NULL	NULL	NULL

Figure 15 Comment Table



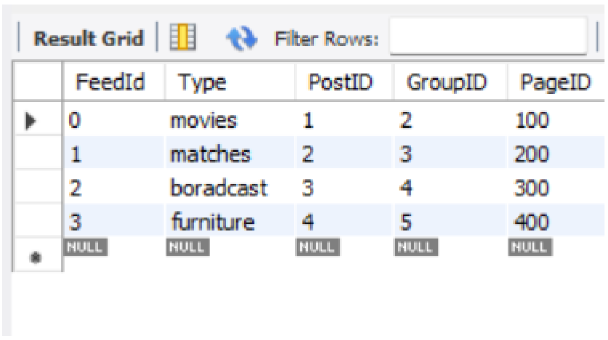
PageID	Name	Description
100	Ahmed mekky	description of actor page
200	Abu Treika	description of football player page
300	mostafa hosny	description of preacher page
400	hosny aly	description of furniture page
NULL	NULL	NULL

Figure 16 Page Table



UserID	PageID
123	100
1234	100
NULL	NULL

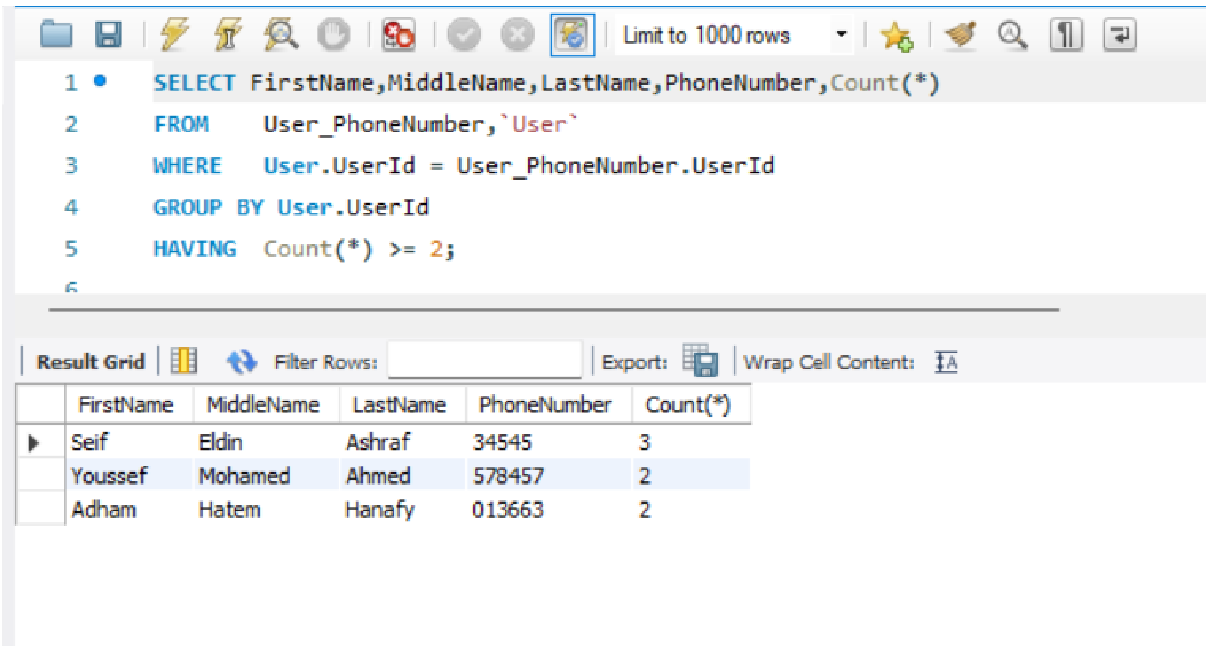
Figure 17 Follows Relation (User follows page)



FeedId	Type	PostID	GroupID	PageID
0	movies	1	2	100
1	matches	2	3	200
2	boradcast	3	4	300
3	furniture	4	5	400
NULL	NULL	NULL	NULL	NULL

Figure 18 Feed Table

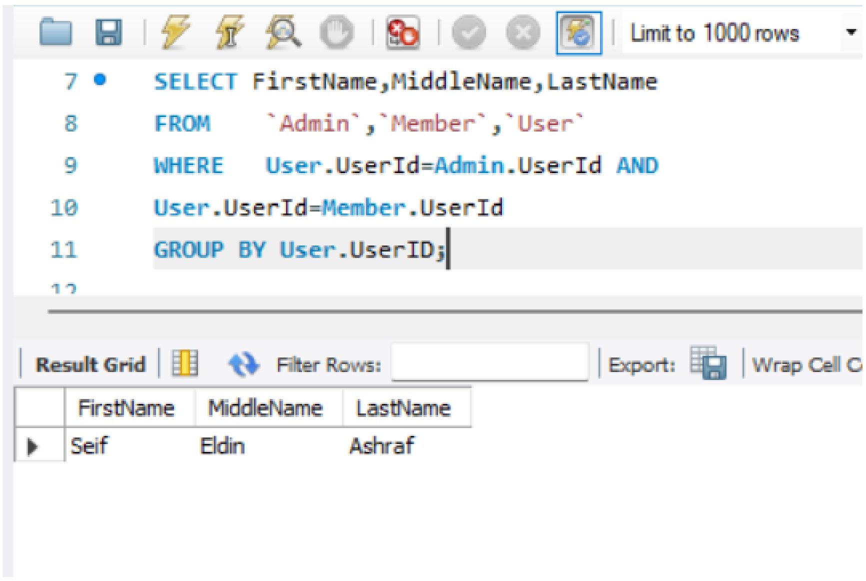
Queries



```sql
SELECT FirstName,MiddleName,LastName,PhoneNumber,Count(*)
FROM    User_PhoneNumber,`User`
WHERE   User.UserId = User_PhoneNumber.UserId
GROUP BY User.UserId
HAVING  Count(*) >= 2;
```

FirstName	MiddleName	LastName	PhoneNumber	Count(*)
Seif	Eldin	Ashraf	34545	3
Youssef	Mohamed	Ahmed	578457	2
Adham	Hatem	Hanafy	013663	2

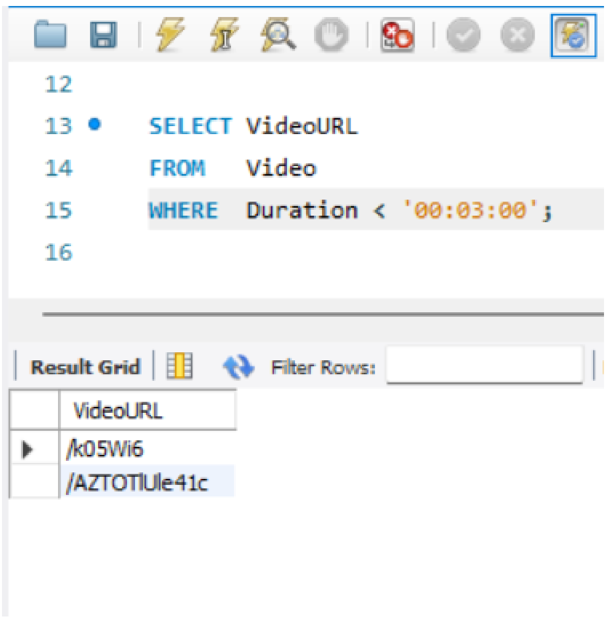
Figure 19 Query that extracts count of phone numbers of users with multiple numbers.



```sql
SELECT FirstName,MiddleName,LastName
FROM    `Admin`,`Member`,`User`
WHERE   User.UserId=Admin.UserId AND
User.UserId=Member.UserId
GROUP BY User.UserID;
```

FirstName	MiddleName	LastName
Seif	Eldin	Ashraf

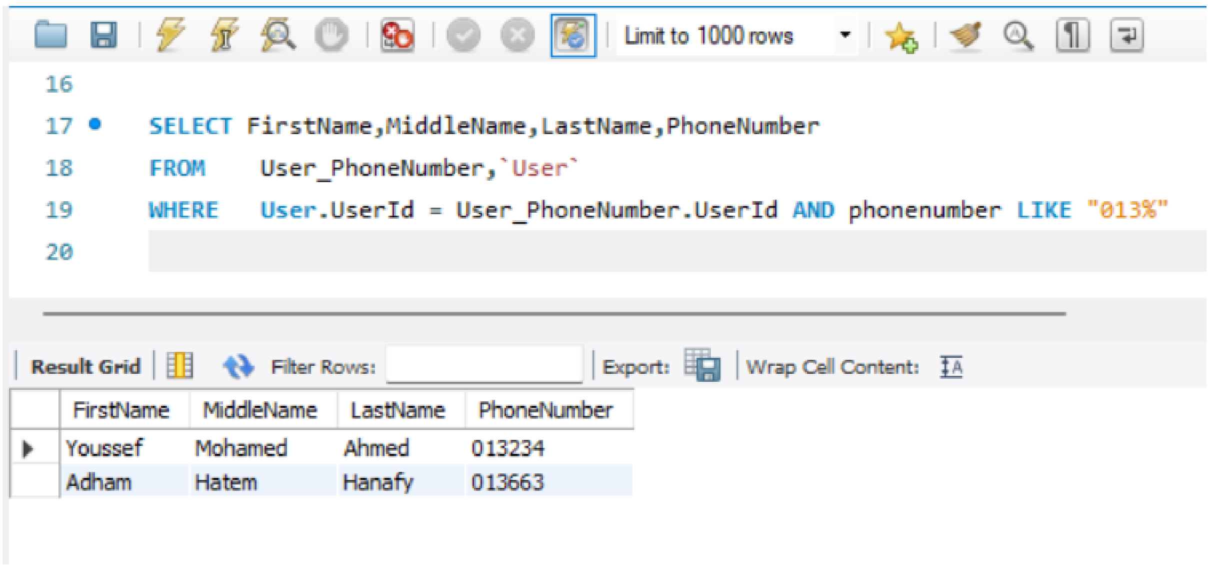
Figure 20 Query that extracts the names of members of a group who are also admins in another.



```sql
SELECT VideoURL
FROM   Video
WHERE  Duration < '00:03:00';
```

VideoURL
/k05Wi6
/AZTOTIUle41c

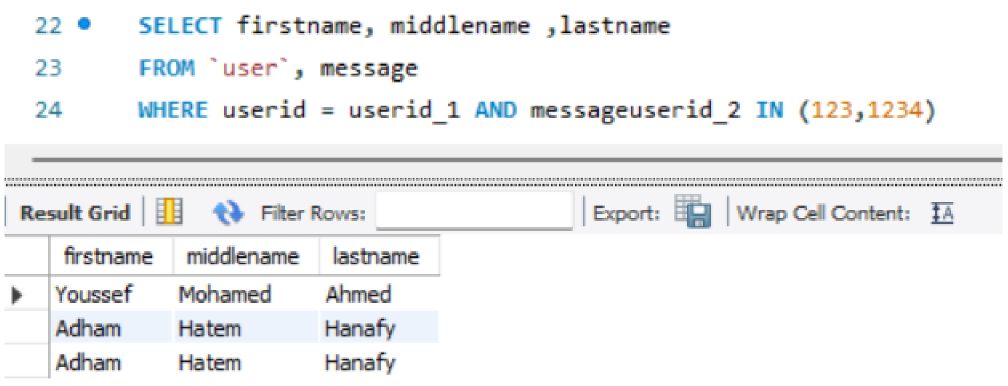
Figure 21 Query that extracts videos that are less than 3 minutes long.



```sql
SELECT FirstName,MiddleName,LastName,PhoneNumber
FROM   User_PhoneNumber,`User`
WHERE  User.UserId = User_PhoneNumber.UserId AND phonenumber LIKE "013%"
```

FirstName	MiddleName	LastName	PhoneNumber
Youssef	Mohamed	Ahmed	013234
Adham	Hatem	Hanafy	013663

Figure 22 Names of users with phone numbers starting with "013”.

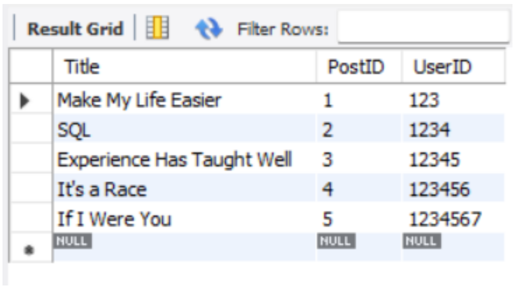


```
SELECT firstname, middlename ,lastname
FROM `user`, message
WHERE userid = userid_1 AND messageuserid_2 IN (123,1234)
```

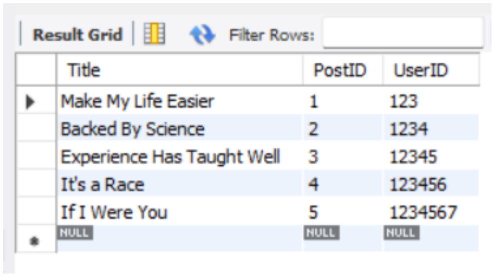
firstname	middlename	lastname
Youssef	Mohamed	Ahmed
Adham	Hatem	Hanafy
Adham	Hatem	Hanafy

Figure 23 Query that displays users that message users with IDs "123" or "1234".

Modifications

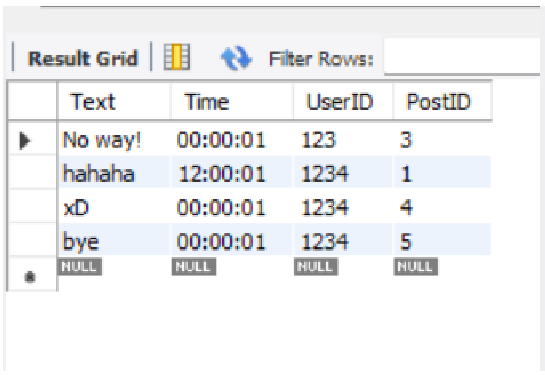


Title	PostID	UserID
Make My Life Easier	1	123
SQL	2	1234
Experience Has Taught Well	3	12345
It's a Race	4	123456
If I Were You	5	1234567
NULL	NULL	NULL

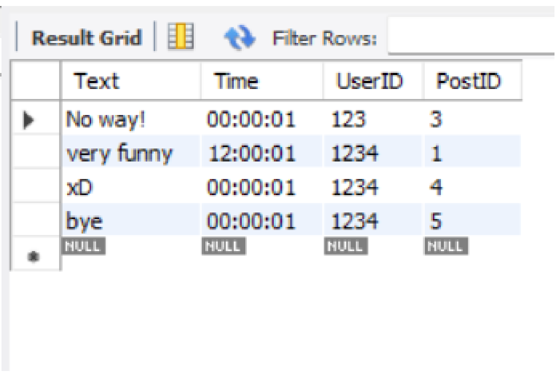


Title	PostID	UserID
Make My Life Easier	1	123
Backed By Science	2	1234
Experience Has Taught Well	3	12345
It's a Race	4	123456
If I Were You	5	1234567
NULL	NULL	NULL

Update Title of Post with ID "2"

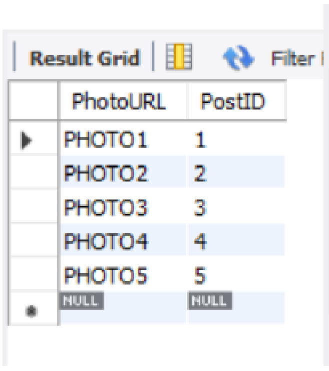


Text	Time	UserID	PostID
No way!	00:00:01	123	3
hahaha	12:00:01	1234	1
xD	00:00:01	1234	4
bye	00:00:01	1234	5
NULL	NULL	NULL	NULL

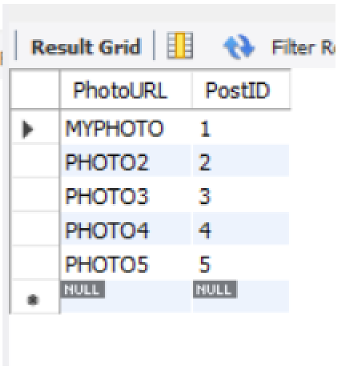


Text	Time	UserID	PostID
No way!	00:00:01	123	3
very funny	12:00:01	1234	1
xD	00:00:01	1234	4
bye	00:00:01	1234	5
NULL	NULL	NULL	NULL

Update comment sent by UserID "1234" on Post "1".

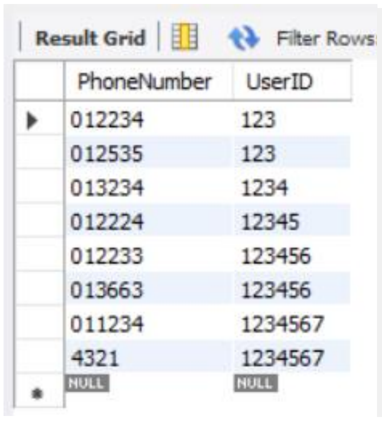

Result Grid | Filter

PhotoURL	PostID
PHOTO1	1
PHOTO2	2
PHOTO3	3
PHOTO4	4
PHOTO5	5
NULL	NULL

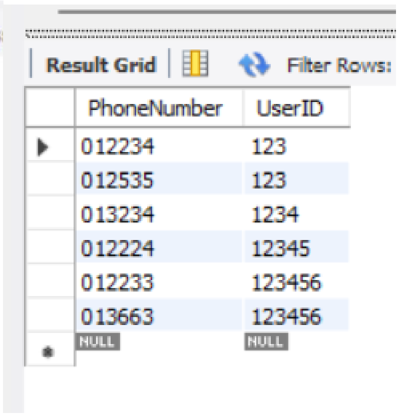

Result Grid | Filter R

PhotoURL	PostID
MYPHOTO	1
PHOTO2	2
PHOTO3	3
PHOTO4	4
PHOTO5	5
NULL	NULL

Update PhotoURL of photo with PostID "1".


Result Grid | Filter Rows

PhoneNumber	UserID
012234	123
012535	123
013234	1234
012224	12345
012233	123456
013663	123456
011234	1234567
4321	1234567
NULL	NULL


Result Grid | Filter Rows:

PhoneNumber	UserID
012234	123
012535	123
013234	1234
012224	12345
012233	123456
013663	123456
NULL	NULL

Deletion of all phone numbers of User with UserID "1234567"

SQL Code

```
CREATE TABLE User

(

  FirstName VARCHAR(100) NOT NULL,

  MiddleName VARCHAR(100) NOT NULL,

  LastName VARCHAR(100) NOT NULL,

  UserID INT NOT NULL,

  Email VARCHAR(100) NOT NULL,

  Password VARCHAR(100) NOT NULL,

  Bio VARCHAR(500) NOT NULL,

  ProfilePicture VARCHAR(50) NOT NULL,

  BirthDate DATE NOT NULL,

  PRIMARY KEY (UserID)


);


CREATE TABLE Admin

(

  AdminID INT NOT NULL,

  UserID INT NOT NULL,

  PRIMARY KEY (UserID),

  FOREIGN KEY (UserID) REFERENCES User(UserID)

  ON UPDATE CASCADE

  ON DELETE RESTRICT

);


CREATE TABLE Member

(

  MemberID INT NOT NULL,

  UserID INT NOT NULL,

  PRIMARY KEY (UserID),

  FOREIGN KEY (UserID) REFERENCES User(UserID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE Post
( Title VARCHAR(50) NOT NULL,
  PostID INT NOT NULL,
  UserID INT NOT NULL,
  PRIMARY KEY (PostID),
  FOREIGN KEY (UserID) REFERENCES `User`(UserID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE Page
(
  PageID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  Description VARCHAR(500) NOT NULL,
  PRIMARY KEY (PageID)
);

CREATE TABLE `group`
(
  GroupID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  Description VARCHAR(500) NOT NULL,
  UserID INT NOT NULL,
  PRIMARY KEY (GroupID),
  FOREIGN KEY (UserID) REFERENCES Admin(UserID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);
```

```sql
CREATE TABLE Feed
(
  FeedId INT NOT NULL,
  Type VARCHAR(50) NOT NULL,
  PostID INT,
  GroupID INT,
  PageID INT,
  PRIMARY KEY (FeedId),
  FOREIGN KEY (PostID) REFERENCES Post(PostID),
  FOREIGN KEY (GroupID) REFERENCES `Group`(GroupID),
  FOREIGN KEY (PageID) REFERENCES Page(PageID)
  ON UPDATE CASCADE
  ON DELETE CASCADE
);
CREATE TABLE Photo
(
  PhotoURL VARCHAR(100) NOT NULL,
  PostID INT NOT NULL,
  PRIMARY KEY (PostID),
  FOREIGN KEY (PostID) REFERENCES Post(PostID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE `Text`
(
  Content VARCHAR(100) NOT NULL,
  PostID INT NOT NULL,
  PRIMARY KEY (PostID),
  FOREIGN KEY (PostID) REFERENCES Post(PostID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE Video
(
  Duration TIME NOT NULL,
  VideoURL VARCHAR(100) NOT NULL,
  PostID INT NOT NULL,
  PRIMARY KEY (PostID),
  FOREIGN KEY (PostID) REFERENCES Post(PostID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE Memes
(
  `Type` VARCHAR(50) NOT NULL,
  PostID INT NOT NULL,
  PRIMARY KEY (PostID),
  FOREIGN KEY (PostID) REFERENCES Photo(PostID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE Personal
(
  `Name` VARCHAR(100) NOT NULL,
  PostID INT NOT NULL,
  PRIMARY KEY (PostID),
  FOREIGN KEY (PostID) REFERENCES Photo(PostID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);
```

```
CREATE TABLE `Like`
(
  UserID INT NOT NULL,
  PostID INT NOT NULL,
  PRIMARY KEY (UserID, PostID),
  FOREIGN KEY (UserID) REFERENCES User(UserID),
  FOREIGN KEY (PostID) REFERENCES Post(PostID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE `Follows`
(
  UserID INT NOT NULL,
  PageID INT NOT NULL,
  PRIMARY KEY (UserID, PageID),
  FOREIGN KEY (UserID) REFERENCES User(UserID),
  FOREIGN KEY (PageID) REFERENCES Page(PageID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE `Comment`
(
  `Text` VARCHAR(500) NOT NULL,
  `Time` TIME NOT NULL,
  UserID INT NOT NULL,
  PostID INT NOT NULL,
  PRIMARY KEY (UserID, PostID),
  FOREIGN KEY (UserID) REFERENCES User(UserID),
  FOREIGN KEY (PostID) REFERENCES Post(PostID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE Message
(
  UserID_1 INT NOT NULL,
  MessageUserID_2 INT NOT NULL,
  PRIMARY KEY (UserID_1, MessageUserID_2),
  FOREIGN KEY (UserID_1) REFERENCES User(UserID),
  FOREIGN KEY (MessageUserID_2) REFERENCES User(UserID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE User_PhoneNumber
(
  PhoneNumber VARCHAR(20) NOT NULL,
  UserID INT NOT NULL,
  PRIMARY KEY (PhoneNumber, UserID),
  FOREIGN KEY (UserID) REFERENCES User(UserID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);

CREATE TABLE `In`
(
  UserID INT NOT NULL,
  GroupID INT NOT NULL,
  PRIMARY KEY (UserID, GroupID),
  FOREIGN KEY (UserID) REFERENCES `Member`(UserID),
  FOREIGN KEY (GroupID) REFERENCES `Group`(GroupID)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);
```

```
Insert INTO User
Values("Youssef","Mohamed","Ahmed",1234,"youssef@uni.com",1234,"hi","png","2002-09-18");

Insert INTO User
Values("Seif","Eldin","Ashraf",123,"seif@uni.com",1234,"Hello","png","2002-10-25");

Insert INTO User
Values("Youssef","Mohamed","El Nagy",12345,"youssef@uni.com",1234,"Sad","jpg","2002-07-23");

Insert INTO User
Values("Adham","Hatem","Hanafy",123456,"Adham@uni.com",1234,"hiii","jpg","2002-06-06");

Insert INTO User
Values("Tarek","Khaled","Ezzat",1234567,"tarek@uni.com",1234,"Helloo","jpg","2002-02-18");

Insert INTO User_PhoneNumber
Values("012234",123);

Insert INTO User_PhoneNumber
Values("012535",123);

Insert INTO User_PhoneNumber
Values("013234",1234);

Insert INTO User_PhoneNumber
Values("012224",12345);

Insert INTO User_PhoneNumber
Values("012233",123456);

Insert INTO User_PhoneNumber
Values("013663",123456);
```

```
Insert INTO User_PhoneNumber
Values("011234",1234567);

INSERT INTO User_PhoneNumber
Values("4321", 1234567);

Insert INTO Member
Values(1,123);

Insert INTO Member
Values(2,1234);

Insert INTO Member
Values(3,12345);

Insert INTO Admin
Values(1,123);

Insert INTO Admin
Values(2,123456);

Insert INTO Admin
Values(3,1234567);

INSERT INTO Post(Title,PostID,UserID)
VALUES("Make My Life Easier",1,123),
              ("Backed By Science",2,1234),
    ("Experience Has Taught Well",3,12345),
    ("It's a Race",4,123456),
    ("If I Were You",5,1234567);

INSERT INTO `Like`
```

```
VALUES  (123,1),
                         (1234,2),
                         (12345,3),
                         (123456,4),
                         (1234567,5);

INSERT INTO Photo
Values  ("PHOTO1",1),
                         ("PHOTO2",2),
                         ("PHOTO3",3),
                         ("PHOTO4",4),
                         ("PHOTO5",5);
INSERT INTO `Text`
VALUES  ("CONTENT1",1),
                         ("CONTENT2",2),
                         ("CONTENT3",3),
                         ("CONTENT4",4),
                         ("CONTENT5",5);



INSERT INTO `Group`(GroupID,name, Description,UserId)
Values (2,"seif","the  description of id",123);

INSERT INTO `Group`(GroupID,name, Description,UserId)
Values (3,"adham","the description of id",123456);



INSERT INTO `Group`(GroupID,`name`, `Description`,UserId)
Values (4,"nagy","the description of id",123456);

INSERT INTO `Group`(GroupID,`name`, `Description`,UserId)
Values (5,"Tarek","the description of id",1234567);
```

```
INSERT INTO `page`(PageID,`Name`, `Description`)

Values (100,"Ahmed mekky","description of actor page");


INSERT INTO `page`(PageID,`Name`, `Description`)

Values (200,"Abu Treika","description of football player page");


INSERT INTO `page`(PageID,`Name`, `Description`)

Values (300,"mostafa hosny","description of preacher page");


INSERT INTO `page`(PageID,`Name`, `Description`)

Values (400,"hosny aly","description of furniture page");


INSERT INTO feed(FeedID,`Type`, PostID, GroupID, PageID)

Values (0,"movies", 1 , 2 , 100);


INSERT INTO feed(FeedID,`Type`, PostID, GroupID, PageID)

Values (1,"matches",  2  , 3 ,200);


INSERT INTO feed(FeedID,`Type`, PostID, GroupID, PageID)

Values (2,"boradcast",  3  , 4 ,300);



 INSERT INTO feed(FeedID,`Type`, PostID, GroupID, PageID)

Values ("3","furniture",  "4"  , "5" ,"400");


Insert into Message

Values(123456,1234),

(1234,123),

(123456,1234567),

(1234,1234567),

(123456,123);
```

```
Insert into Video

Values("01:00","/xynArToJ", 1),

("00:00:51","/k05Wi6", 2),

("00:02:23","/AZTOTlUle41c", 3),

("00:03:14","/6WsK0iF7BWI", 4),

("00:19:09","/Y2WwtNm", 5);


Insert into `Comment`

Values("hahaha","12:00:01", 1234,1),

("No way!","00:00:01", 123,3),

("xD","00:00:01", 1234,4),

("bye","00:00:01", 1234,5);


Insert into `in`

values(12345, 2),

(12345, 5),

(123, 4),

(1234, 3);


insert into personal

values("myphoto.png", 1),

("pic3.jpg", 2),

("pic12.png", 3),

("mypic1.jpg", 4),

("mypic14.png", 5);


insert into memes

values("funnyphoto1.png", 1),

("funnyphoto2.png", 2),

("funnygif.gif", 3),

("funnygif2.gif", 4),

("funnyphoto.jpg", 5);
```

```
insert into user

values("Yahia", "Mohamed", "El Nagy", 55, "yohia@domain.com", "password", "hi", "jpg", "2004-08-22")


insert into follows

values("123", "100"), ("12345", "400"), ("1234", "100")
```

Implementation

ERD Tool

ERDPlus is an online tool for creating Entity Relationship Diagrams (ERDs), a popular modelling technique used in software development and database design. With ERDPlus, users can create clear and concise visual representations of their data models, which can help to improve communication and collaboration among team members.

The tool offers a simple and intuitive interface, making it easy to create and modify ERDs quickly and efficiently. ERDPlus also offers a range of features, including the ability to generate SQL scripts and export diagrams in a variety of formats.

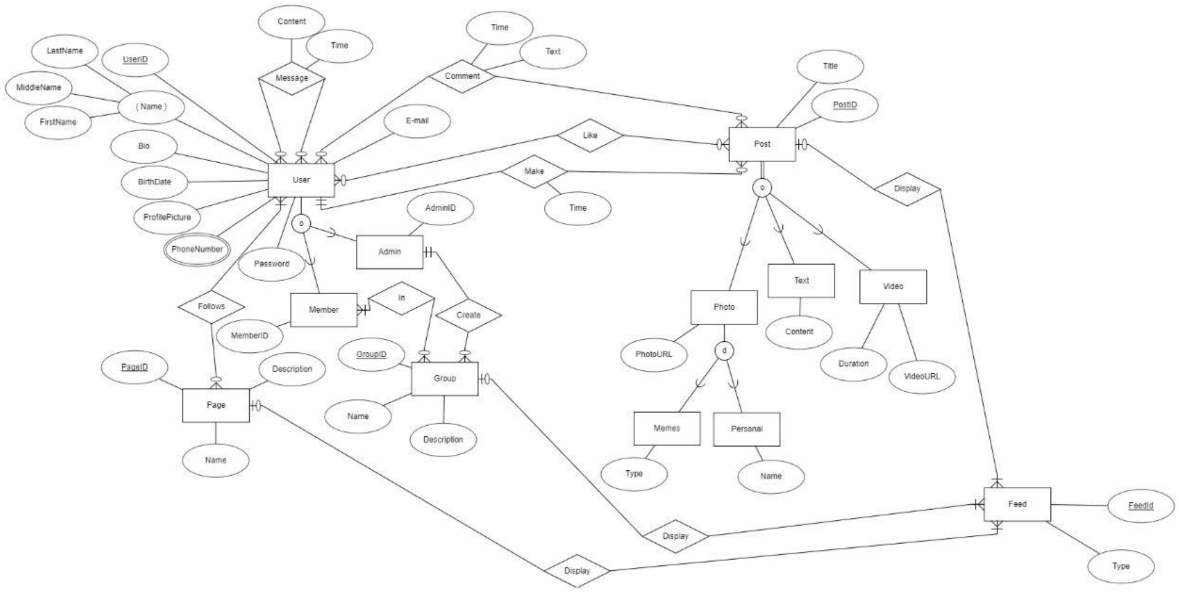


Figure 24 ERD

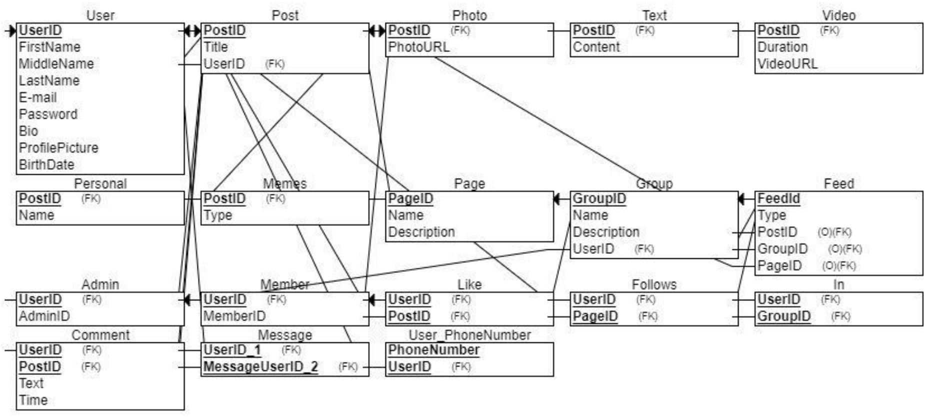
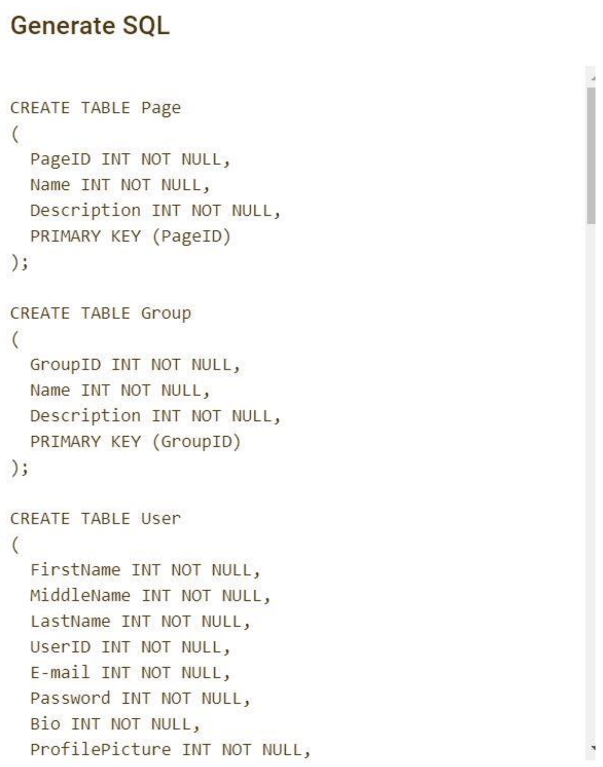


Figure 25 Relational Schema


Generate SQL

```
CREATE TABLE Page
(
  PageID INT NOT NULL,
  Name INT NOT NULL,
  Description INT NOT NULL,
  PRIMARY KEY (PageID)
);

CREATE TABLE Group
(
  GroupID INT NOT NULL,
  Name INT NOT NULL,
  Description INT NOT NULL,
  PRIMARY KEY (GroupID)
);

CREATE TABLE User
(
  FirstName INT NOT NULL,
  MiddleName INT NOT NULL,
  LastName INT NOT NULL,
  UserID INT NOT NULL,
  E-mail INT NOT NULL,
  Password INT NOT NULL,
  Bio INT NOT NULL,
  ProfilePicture INT NOT NULL,
```

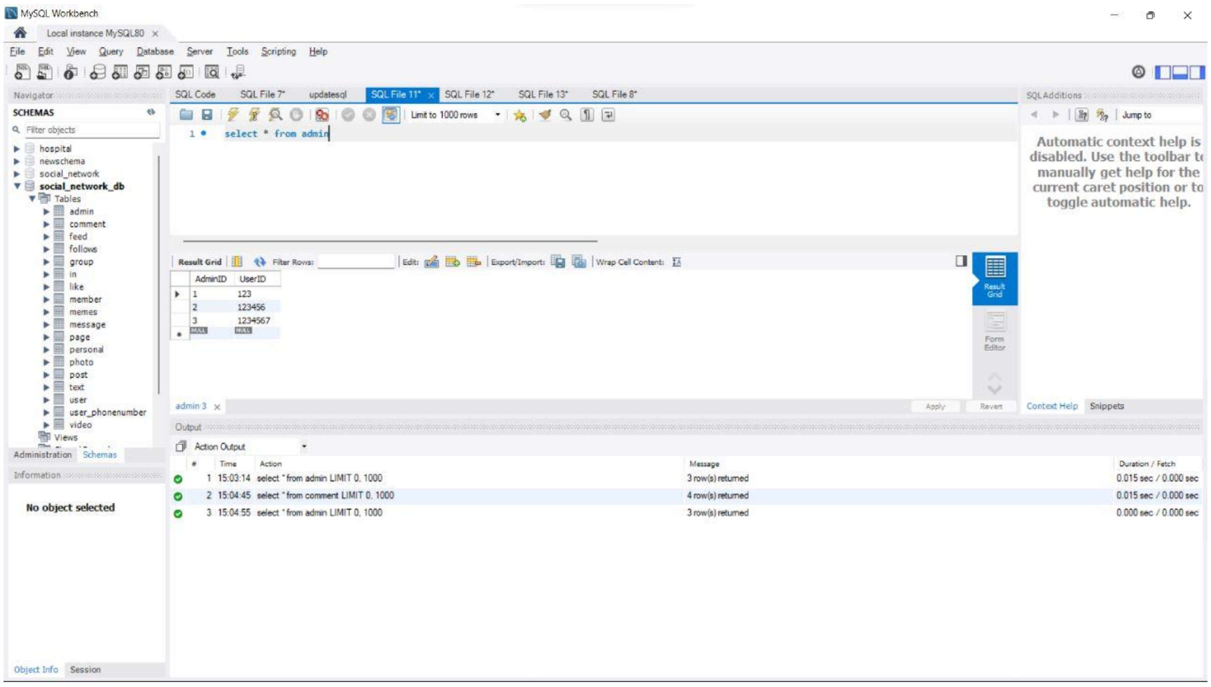
Figure 26 SQL Sample

SQL Tool

MySQL Workbench is a popular open-source relational database management system (RDBMS) that is widely used in web development and other applications. It is known for its scalability, security, and ease of use, making it a popular choice for many organizations and businesses.

MySQL uses a client-server model, where the client sends requests to the server, which then processes those requests and sends back the results. It supports a variety of programming languages and platforms and offers a range of features including support for transactions, stored procedures, triggers, and more.

With its robust and reliable architecture, MySQL is a powerful tool for managing large amounts of data efficiently and securely.

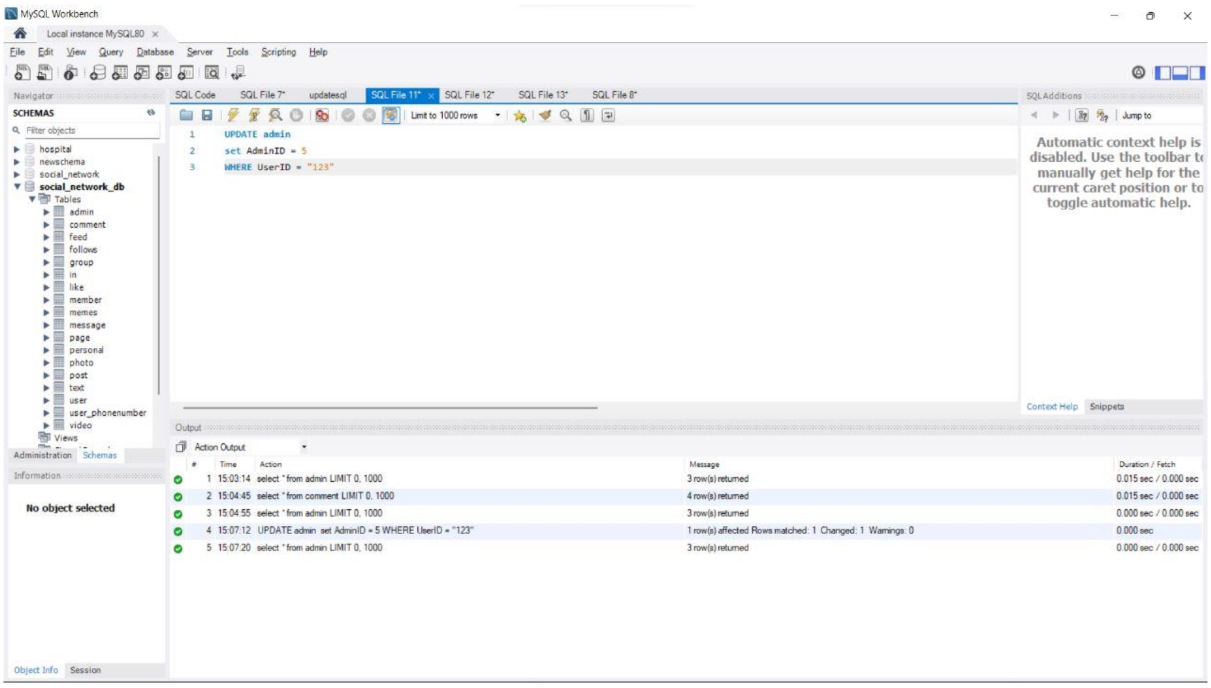


```sql
select * from admin
```

AdminID	UserID
1	123
2	123456
3	1234567
NULL	NULL

```
#  Time      Action                                  Message           Duration / Fetch
1  15:03:14  select * from admin LIMIT 0, 1000       3 row(s) returned 0.015 sec / 0.000 sec
2  15:04:45  select * from comment LIMIT 0, 1000     4 row(s) returned 0.015 sec / 0.000 sec
3  15:04:55  select * from admin LIMIT 0, 1000       3 row(s) returned 0.000 sec / 0.000 sec
```

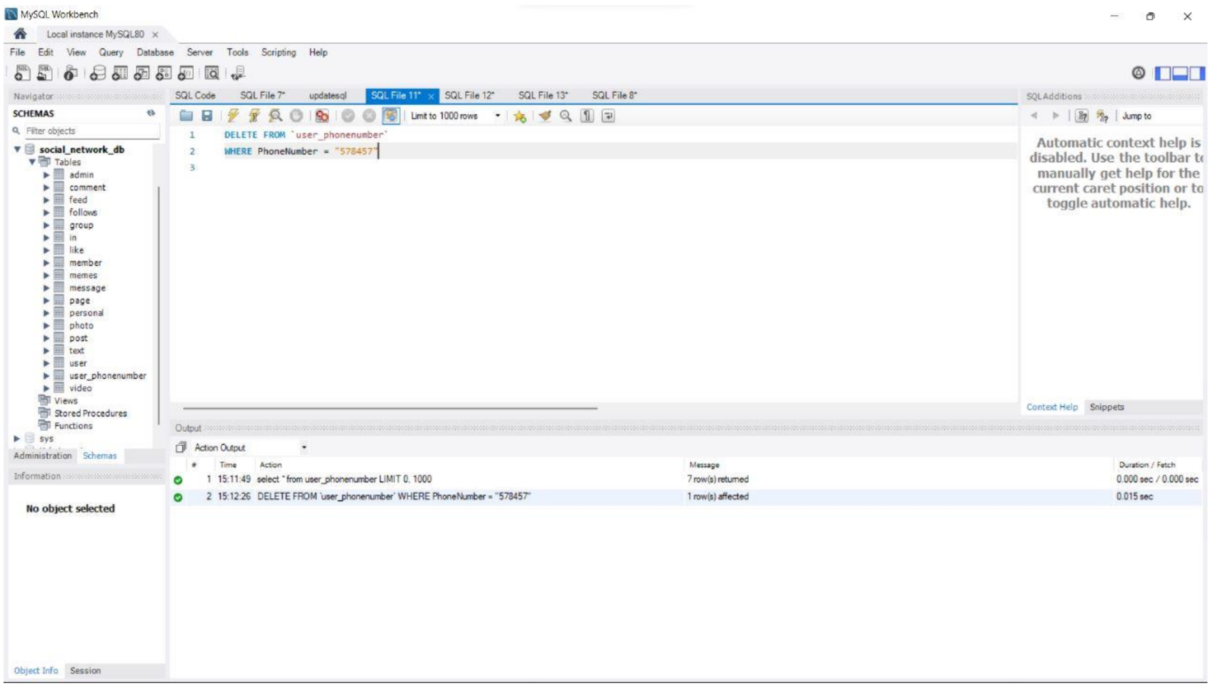
Figure 27 Using MySQL to display all values from Admin Table.



```sql
UPDATE admin
set AdminID = 5
WHERE UserID = "123"
```

```
#  Time      Action                                          Message                                                         Duration / Fetch
1  15:03:14  select * from admin LIMIT 0, 1000               3 row(s) returned                                               0.015 sec / 0.000 sec
2  15:04:45  select * from comment LIMIT 0, 1000             4 row(s) returned                                               0.015 sec / 0.000 sec
3  15:04:55  select * from admin LIMIT 0, 1000               3 row(s) returned                                               0.000 sec / 0.000 sec
4  15:07:12  UPDATE admin  set AdminID = 5 WHERE UserID = "123"  1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0  0.000 sec
5  15:07:20  select * from admin LIMIT 0, 1000               3 row(s) returned                                               0.000 sec / 0.000 sec
```

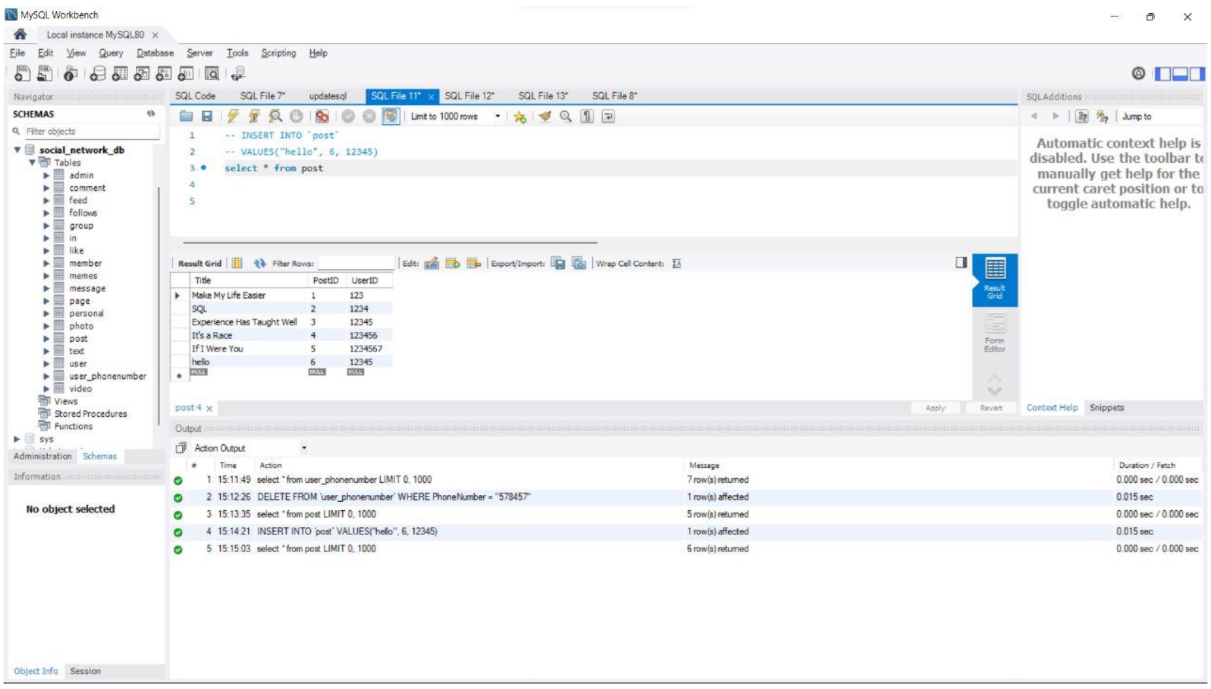
Figure 28 Using MySQL to update Admin Table.



```sql
DELETE FROM `user_phonenumber`
WHERE PhoneNumber = "578457"
```

```
#  Time      Action                                                                 Message             Duration / Fetch
1  15:11:49  select * from user_phonenumber LIMIT 0, 1000                           7 row(s) returned   0.000 sec / 0.000 sec
2  15:12:26  DELETE FROM `user_phonenumber` WHERE PhoneNumber = "578457"            1 row(s) affected   0.015 sec
```

Figure 29 Using MySQL to delete a user phone number.



```sql
-- INSERT INTO `post`
-- VALUES("hello", 6, 12345)
select * from post
```

Title	PostID	UserID
Make My Life Easier	1	123
SQL	2	1234
Experience Has Taught Well	3	12345
It's a Race	4	123456
If I Were You	5	1234567
hello	6	12345
NULL	NULL	NULL

```
#  Time      Action                                                                 Message             Duration / Fetch
1  15:11:49  select * from user_phonenumber LIMIT 0, 1000                           7 row(s) returned   0.000 sec / 0.000 sec
2  15:12:26  DELETE FROM `user_phonenumber` WHERE PhoneNumber = "578457"            1 row(s) affected   0.015 sec
3  15:13:35  select * from post LIMIT 0, 1000                                       5 row(s) returned   0.000 sec / 0.000 sec
4  15:14:21  INSERT INTO `post` VALUES("hello", 6, 12345)                           1 row(s) affected   0.015 sec
5  15:15:03  select * from post LIMIT 0, 1000                                       6 row(s) returned   0.000 sec / 0.000 sec
```

Figure 30 Using my SQL to insert a new post.