INTRODUCTION

Introduction

Twitter Proto Type Database is chosen from the IEEE Paper 2017 Towards Detecting Compromised Accounts On Social Networks. In this project we are trying to find out the users who tweets inappropriately or spreading malicious links and block them from login.

In this Twitter Proto Type Database project we are going to see

1) Why particular project is selected?
2) Software's that are going to be used? Why?
3) Entity-Relationship Diagram (ER-D) with at least having 10 entities and attributes in it.
4) Database relation diagrams showing referential integrity. Such that all relations are in the 3rd normal form.
5) SQL statements to Create tables with primary and foreign key references including all required data integrity constraints.
6) Insertion Of Data Into Tables.
7) Examples of the following queries and their results:

a) INNER JOIN

b) OUTER JOIN

c) sub-query

d) correlated sub-query

8) Outputs of Query results.
9) Java ODBC connection to the database.

1) Why particular project is selected?

Twitter Proto Type Database is selected because of feeling that one can learn and improve their knowledge a lot from the mistakes they made.

In my undergraduate level I have taken the project Towards Detecting Compromised Accounts On Social Networks where I concentrated more on defining the logic and front end of the project where user can see it and use it by taking Twitter as an example of proto type. The Database created is of name sake to store and retire the data. There are no entity relationships between tables and soon.

Once the planning started I want to create the entities like User , Admin ,Tweets , Re_Tweets which holds their particular attributes. A entity named Filter to hold attributes like category of filter and filter words and links. Finally, I have 9 entities to deal with in the database project.

2) Softwares that are going to be used? Why?

After a good work , I came to a final decision about the softwares that are going to be used.

ERD Software : Smart Draw or Lucid Chart

Reason : These are the one of top rated software on internet to draw Entity Relationship Diagrams.

Smart Draw is not a open source software where we need to buy after free trail. Where as Lucid chart is a cloud based software which is free and most of the features are covered in it.

I was planning to use Lucid chart which is very comfortable as it is a cloud based system.

DBMS : MYSQL

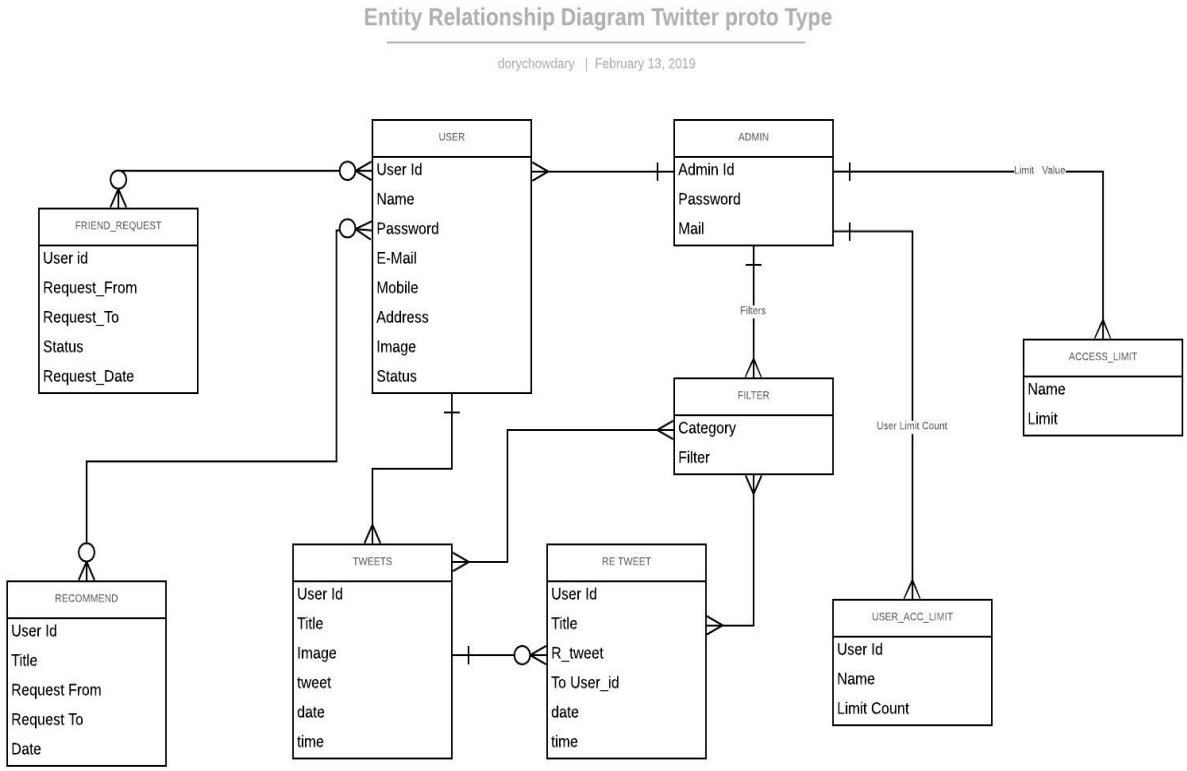
Reason : Being MYSQL an open source Relational Database Management System. It is one of the most efficient and flexible software which can be used to build any application.

3) Entity-Relationship Diagram (ER-D) with at least having 10 entities and attributes in it

It started with 9 entities like User, Tweet, Re_Tweet, Admin, AccessLimit, Filter, Friend_Request, Recommend , User_acc_Limit. Included attributes into each of the entity.

The relationships between the entities is established.

a) One to one
b) One to Many
c) Many to Many



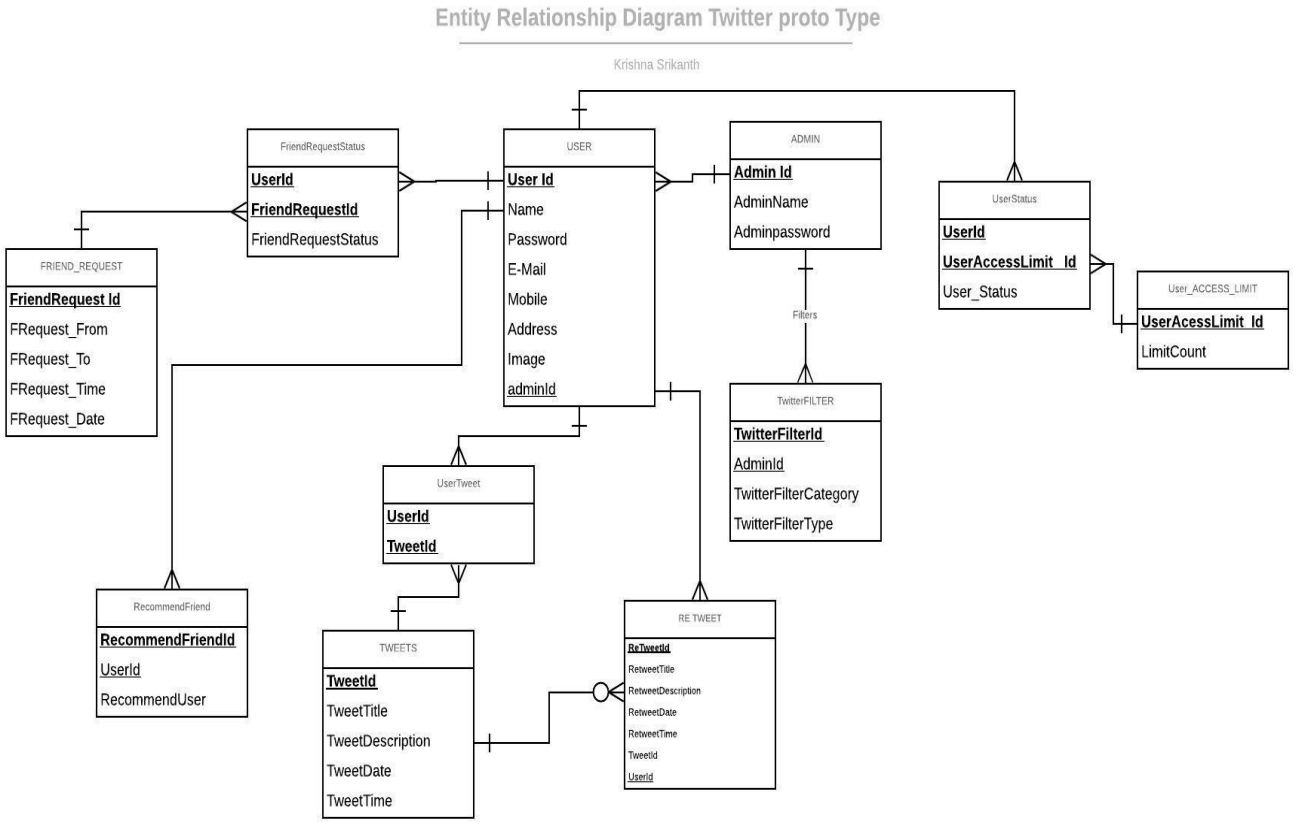
Moving further I understood that there are many mistakes in my ERD.The mistakes that I figured out in first ERD are 1)No associative table for many to many relationship.

2)Some of the relationships established between are wrong and some are not necessary.(Example:User and Friend_Request relationship is many to one not many to many.)

3)The Names of entities and attributes are updated for easy understanding.

In final modification realtionship between User and Re_Tweet was added as I felt it is necessary.

The Final modified Entity Realtionship Diagram is as follows:



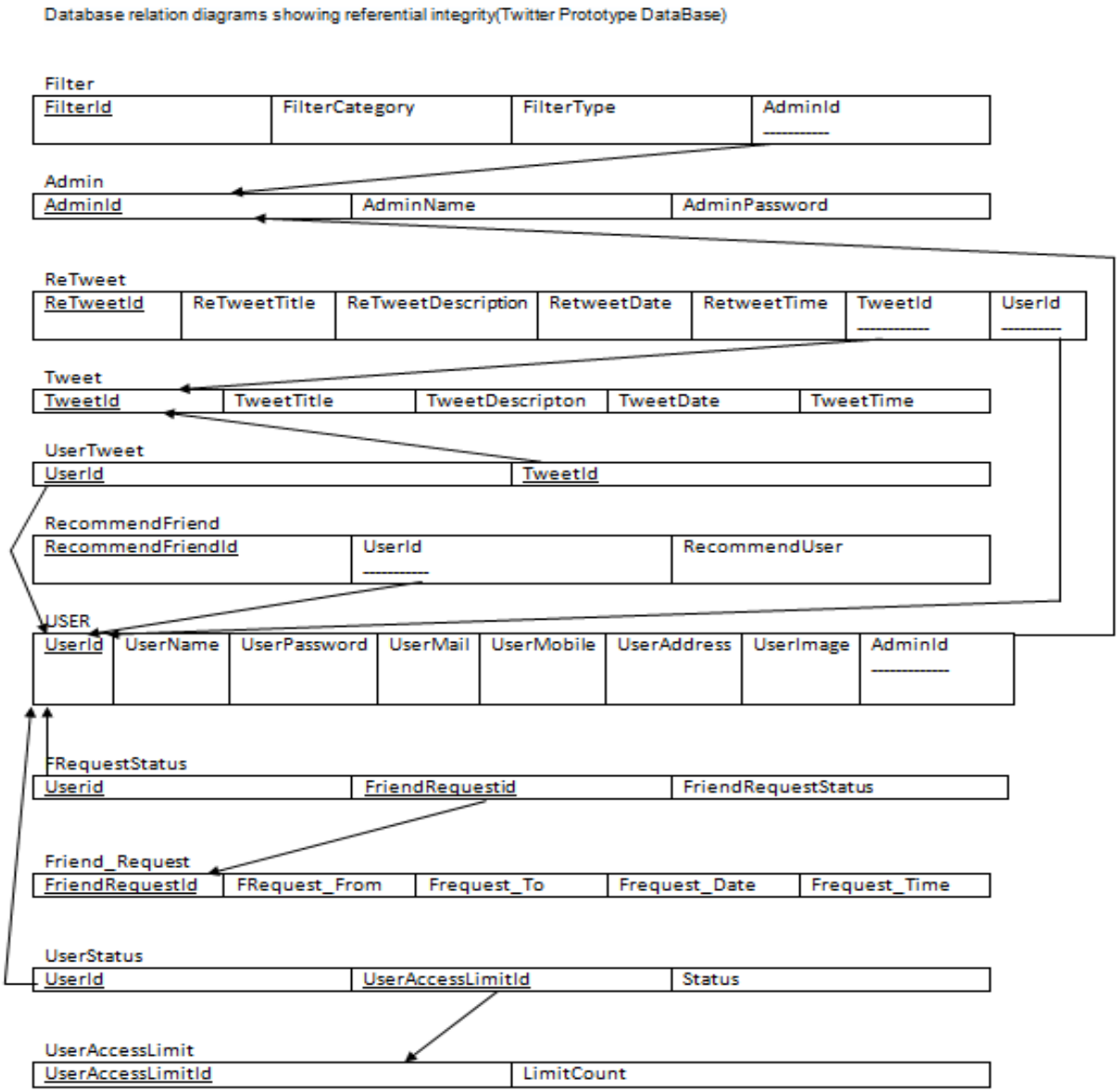
Key Representations:

Primary Key: **Bold and Underline**.

Foreign Key: Underline

4) Database relation diagrams showing referential integrity. Such that all relations are in the 3rd normal form.

Following ERD the Database realtion diagram showing referential integrity is drawn. It is also modified to 3rd Normal form by removing 1)Removing Multivalued Attributes.
2)Removing Partial Dependencies.
3)Removing Transitive Dependencies.
In the diagram the data flow is explained through connecting primary and foreign keys.



5) SQL statements to Create tables with primary and foreign key references including all required data integrity constraints.

To create table the command used is
CREATE table TableName (Attribues for given table);

Here the 11 entities are finally turned into physical tables using create table command.

```
-- ----------------------------
-- Table structure for admin
-- ----------------------------

CREATE TABLE `Admin` (
                `Admin_id` int  NOT NULL  ,
                `Admin_name` VARCHAR(100) NOT NULL,
                `Admin_pass` VARCHAR(100) NOT NULL,

 CONSTRAINT Admin_PK PRIMARY KEY(Admin_id)
) ;


-- ----------------------------
-- Table structure for USER
-- ----------------------------

CREATE TABLE `User` (
                `User_id` int NOT NULL auto_increment,
                `User_name` VARCHAR(100) NOT NULL,
                `User_Password` VARCHAR(100) NOT NULL,
                `User_E-Mail` text,
                `User_mobile` bigint,
                `User_address` text,
                `Admin_id`   int,

 CONSTRAINT User_PK PRIMARY KEY  (User_id),
 CONSTRAINT User_FK FOREIGN KEY (Admin_id )    REFERENCES Admin(Admin_id)
ON DELETE RESTRICT
);
```

```
-- ----------------------------------
-- Table structure for Twitter_Filter
-- ----------------------------------

CREATE TABLE `Twitter_Filter`(
                    `Twitter_Filter_Id` int NOT NULL auto_increment ,
                    `Admin_id` int,
                    `Twitter_Filter_Category` text,
                    `Twitter_Filter_Type` text,

CONSTRAINT Twitter_Filter_PK PRIMARY KEY  (Twitter_Filter_Id),
CONSTRAINT Twitter_Filter_FK FOREIGN KEY (Admin_id) REFERENCES Admin(Admin_id) ON DELETE RESTRICT
);

-- ----------------------------
-- Table structure for Tweets
-- ----------------------------

CREATE TABLE `Tweets`(
                `Tweet_id` int NOT NULL auto_increment,
                `Tweet_Title` VARCHAR(100),
                `Tweet_Description` TEXT,
                `Tweet_Date` DATE,
                `Tweet_Time` TIME,

  CONSTRAINT Tweets_PK PRIMARY KEY (Tweet_id)
);


-- ----------------------------------
-- Table structure for User_Tweet
-- ----------------------------------

CREATE TABLE `User_Tweet`(
                    `User_id` int NOT NULL REFERENCES User(User_id),
                    `Tweet_id` int NOT NULL REFERENCES Tweets(Tweet_id) ,

CONSTRAINT User_Tweet_PK PRIMARY KEY (User_id,Tweet_id)

);
```

```
-- ----------------------------
-- Table structure for Re_Tweet
-- ----------------------------

CREATE TABLE `RE_Tweet`(
                `RE_Tweet_id` int NOT NULL ,
                `RE_Tweet_Title` VARCHAR(100),
                `RE_Tweet_Description` text,
                `RE_Tweet_Date`  date,
                `RE_Tweet_Time` time,
                `Tweet_id` int NOT NULL,
                `User_id` int  NOT NULL,
CONSTRAINT RE_Tweet_PK PRIMARY KEY (Re_Tweet_id),
CONSTRAINT RE_Tweet_FK FOREIGN KEY (Tweet_id) REFERENCES Tweets(Tweet_id)
ON DELETE CASCADE,
CONSTRAINT RE_Tweet_FK1 FOREIGN KEY (User_id) REFERENCES User(User_id)
);

-- ------------------------------------
-- Table structure for Recommend_Friend
-- ------------------------------------

CREATE TABLE `Recommend_Friend`(
                    `Recommend_Friend_Id` int NOT NULL,
                    `Rcommend_Id` int NOT NULL,
                    `User_id` int,
                    `Recommend_Friend_Rating` text,
CONSTRAINT Recommend_Friend_PK PRIMARY KEY (Recommend_Friend_Id),
CONSTRAINT Recommend_Friend_FK FOREIGN KEY (User_id) REFERENCES
User(User_id)
);

-- ------------------------------------
-- Table structure for Friend_Request
-- ------------------------------------

CREATE TABLE `Friend_Request`(
                   `Friend_Request_id` int NOT NULL,
                   `Friend_Request_From` VARCHAR(100),
                   `Friend_Request_TO` VARCHAR(100),
                   `Friend_Request_Time` time,
                   `Friend_Request_Date` date,
CONSTRAINT Friend_Request_PK PRIMARY KEY (Friend_Request_id)

);
```

```
-- ----------------------------------------
-- Table structure for Friend_Request_Status
-- ----------------------------------------

CREATE TABLE `Friend_Request_Status`(
                        `User_id` int NOT NULL REFERENCES User(User_id) ,
                        `Friend_Request_id`  int  NOT  NULL  REFERENCES
Friend_Request(Friend_Request_id) ,
                        `Friend_Request_Status` VARCHAR(50),
CONSTRAINT Friend_Request_Status_PK  PRIMARY KEY (User_id,Friend_Request_id)

);

-- ---------------------------------------
-- Table structure for User_Access_Limit
-- ---------------------------------------

CREATE TABLE `User_Access_Limit`(
                        `User_Access_Limit_id` VARCHAR(20) NOT NULL,
                        `Limit_Count` int NOT NULL,
CONSTRAINT User_Access_Limit_PK PRIMARY KEY (User_Access_Limit_id)
);

-- --------------------------------
-- Table structure for User_Status
-- --------------------------------

CREATE TABLE `User_Status`(
                    `User_id` int NOT NULL  REFERENCES User(User_id)  ,
                            `User_Access_Limit_id` VARCHAR(20) NOT NULL REFERENCES
                  User_Access_Limit(User_Access_Limit_id) ,
                    `User_limit_count` int,
                    `User_Status` VARCHAR(50) NOT NULL,
CONSTRAINT User_Status_PK PRIMARY KEY (User_id,User_Access_Limit_id)

);
```

6) Insertion Of Data Into Tables

Insertion of data into tables is done by using SQL command:

Insert into `tablename` values (attribute values);

```
-- ----------------------------
-- Records
-- ----------------------------
-- ----------------------------
-- Admin
-- ----------------------------

insert into `Admin`values ('6492240','Krishna','99Krishna45@');
insert into `Admin`values ('6492241','John','helloSri&');
insert into `Admin`values ('6492242','Madhu','#Madhu118#');
insert into `Admin`values ('6492243','Delina','!@#845623*()');

-- ----------------------------
-- User
-- ----------------------------

insert into `User`values ('1','Nithin','julakanti123','nithinreddy@gmail.com','8456221430','28 A colonial NewPaltz','6492240');
insert into `User`values ('2','Max','Max@23&*','maxhasmanager@gmail.com','8455222300','25 Kingston NY','6492240');
insert into `User`values ('3','Alton','&beHappy&','Altonsup@gmail.com','6239221430','150  A colonial NewPaltz','6492241');
insert into `User`values ('4','Manasa','luvUpapa90','Manasagopisetty@gmail.com','9703501802','Guntur India','6492242');
insert into `User`values ('5','Pranay','9963730322','Pranaymbbs@gmail.com','9963730322','India','6492243');
insert into `User`values ('6','Sophia','Sophi123','Sophi123@gmail.com','8452211120','Main Street NewPaltz NY','6492243');


-- ----------------------------
-- User Twitter_Filter
-- ----------------------------
insert into `Twitter_Filter` values('1','6492240','Word','bullying');
insert into `Twitter_Filter` values('2','6492241','Word','fuck');
insert into `Twitter_Filter` values('3','6492240','Link','http://www.hacktheuser.com');
insert into `Twitter_Filter` values('4','6492242','Link','http://2www.clickonit.com');
insert into `Twitter_Filter` values('5','6492243','Link','http://www.hacksyou.com');
insert into `Twitter_Filter` values('6','6492242','Word','Mother Fucker');
```

```
-- ----------------------------
-- User Tweets
-- ----------------------------

insert into  `Tweets` values('1','Foot Ball','The football between SUNY teams is going pretty well.','2017/09/09','10:39:08');
insert into  `Tweets` values('2','Politics','The elections are pre-poned due to some problems in the administration.','2017/01/10','10:39:08');
insert into  `Tweets` values('3','Soccer','Soccer is a big thing in United States Of America .','2017/01/11','10:39:08');
insert into  `Tweets` values('4','Silicon Valley','According to the stats the use of artificial intelligence has been increased in Silicon Valley.','2017/09/03','10:39:08');


-- ----------------------------
-- User_Tweet
-- ----------------------------

insert into `User_Tweet` values('1','1');
insert into `User_Tweet` values('2','2');
insert into `User_Tweet` values('1','3');
insert into `User_Tweet` values('2','4');


-- ----------------------------
-- Re_Tweet
-- ----------------------------

insert into `RE_Tweet` values ('1','Politics Reply','The latest update is that they are still discussing about yet to make the final decision','2017/01/10','10:40:08','2','5');
insert into `RE_Tweet` values ('2','Politics Reply','It is expected to take lot of time.it seems they have lot of issues to deal with it','2017/01/10','10:41:08','2','2');
insert into `RE_Tweet` values ('3','Soccer Reply','It is one of the big thing football is also a big thing ','2018/05/11','9:50:08','3','6');
insert into `RE_Tweet` values ('4','Soccer Reply','I also agree with you sophia ','2018/05/11','9:55:08','3','3');
```

```
-- ----------------------------
-- Recommend_Friend
-- ----------------------------
/*`Recommend_Friend_Id` int ,`Rcommend_Id` int ,`User_id` int,`Recommend_Friend_Rating` text, */
insert into `Recommend_Friend` values ('1','1','6','Good');
insert into `Recommend_Friend` values ('2','2','6','Good in Communication');
insert into `Recommend_Friend` values ('3','3','6','Excellent Person');
insert into `Recommend_Friend` values ('4','2','1','Good in Communication');
insert into `Recommend_Friend` values ('5','3','1','Excellent Person');

-- ----------------------------
-- Friend_Request
-- ----------------------------

insert into `Friend_Request` values ('1','sophia','Pranay','12:55:08','2018/12/22');
insert into `Friend_Request` values ('2','sophia','Max','1:50:08','2018/12/22');
insert into `Friend_Request` values ('3','Pranay','Alton','10:55:08','2019/01/22');
insert into `Friend_Request` values ('4','Nithin','Alton','11:55:08','2019/02/22');

-- ----------------------------
-- Friend_Request_Status
-- ----------------------------

insert into `Friend_Request_Status` values('6','1','Pending');
insert into `Friend_Request_Status` values('2','2','ACCEPTED');
insert into `Friend_Request_Status` values('3','3','ACCEPTED');
insert into `Friend_Request_Status` values('3','4','Declined');

-- ----------------------------
-- User_Access_Limit
-- ----------------------------

insert into `User_Access_Limit` values('1','3');

-- ----------------------------
-- UUser_Status
-- ----------------------------

insert into `User_Status`values('1','1','1','Active');
insert into `User_Status`values('2','1','0','Active');
insert into `User_Status`values('3','1','2','Active');
insert into `User_Status`values('4','1','3','Compromise');
insert into `User_Status`values('5','1','1','Active');
insert into `User_Status`values('6','1','3','Compromise');
```

7) Queries And Results

INNER JOIN :-

In this query we are going to combine User and Recommend_Friend tables to get the results of User_id ,User_Name and Recommended User_id to the User and their Rating.

```
-- ----------------------------
-- INNER JOIN
-- ----------------------------
Select ' INNER JOIN ';
 select U.User_id,U.User_name,R.Rcommend_Id,R.Recommend_Friend_Rating
        from User U INNER JOIN Recommend_Friend R ON U.User_id = R.User_id;
```

```
+--------------+
| INNER JOIN   |
+--------------+
|  INNER JOIN  |
+--------------+
1 row in set (0.00 sec)

+---------+-----------+-------------+-------------------------+
| User_id | User_name | Rcommend_Id | Recommend_Friend_Rating |
+---------+-----------+-------------+-------------------------+
|       6 | Sophia    |           1 | Good                    |
|       6 | Sophia    |           2 | Good in Communication   |
|       6 | Sophia    |           3 | Excellent Person        |
|       1 | Nithin    |           2 | Good in Communication   |
|       1 | Nithin    |           3 | Excellent Person        |
+---------+-----------+-------------+-------------------------+
5 rows in set (0.26 sec)
```

Outer Join :-

In this query we are going to combine User and User_Tweet tables using left join to get the results of Users and their respective Tweet_id's.

```
-- ----------------------------
-- Outer JOIN
-- ----------------------------
Select ' Outer JOIN ';
 select U.User_id,U.User_name,T.Tweet_id from User U LEFT JOIN User_Tweet T ON
U.User_id = T.User_id;
```

```
+---------------+
| Outer JOIN    |
+---------------+
|  Outer JOIN   |
+---------------+
1 row in set (0.00 sec)

+---------+-----------+----------+
| User_id | User_name | Tweet_id |
+---------+-----------+----------+
|       1 | Nithin    |        1 |
|       1 | Nithin    |        3 |
|       2 | Max       |        2 |
|       2 | Max       |        4 |
|       3 | Alton     |     NULL |
|       4 | Manasa    |     NULL |
|       5 | Pranay    |     NULL |
|       6 | Sophia    |     NULL |
+---------+-----------+----------+
8 rows in set (0.00 sec)
```

Sub_Query:-

To get the users who are compromised that is User_limit_count is 3.

```
-- ----------------------------
--  SUB QUERY
-- ----------------------------
Select ' SUB QUERY ';
select User_id,User_name from User WHERE User_id IN (SELECT User_id from User_Status WHERE User_limit_count ='3' );
```

Command Prompt - mysql -u root -p

```
+--------------+
| SUB QUERY    |
+--------------+
|  SUB QUERY   |
+--------------+
1 row in set (0.00 sec)

+---------+-----------+
| User_id | User_name |
+---------+-----------+
|       4 | Manasa    |
|       6 | Sophia    |
+---------+-----------+
2 rows in set (0.05 sec)
```

Correlated Sub Query:-

To get all the users information who tweets.

```
-- ----------------------------
--  Correlated Sub Query
-- ----------------------------

select ' Correlated Sub Query ';

 SELECT * FROM user u  WHERE User_id = ANY (SELECT User_id FROM User_Tweet t WHERE t.User_id= u.User_id);
```

```
+-----------------------+
| Correlated Sub Query  |
+-----------------------+
|  Correlated Sub Query |
+-----------------------+
1 row in set (0.00 sec)

+---------+-----------+---------------+-----------------------+-------------+----------------------+----------+
| User_id | User_name | User_Password | User_E-Mail           | User_mobile | User_address         | Admin_id |
+---------+-----------+---------------+-----------------------+-------------+----------------------+----------+
|       1 | Nithin    | julakanti123  | nithinreddy@gmail.com |  8456221430 | 28 A colonial NewPaltz |  6492240 |
|       2 | Max       | Max@23&*      | maxhasmanager@gmail.com |  8455222300 | 25 Kingston NY       |  6492240 |
+---------+-----------+---------------+-----------------------+-------------+----------------------+----------+
2 rows in set (0.00 sec)

mysql>
```

8) Java ODBC connection to the database.

Java Program connecting MySQL Database to Insert data into User Table and also to read data from the Admin table.

```
import java.sql.*;
import java.util.*;
class MysqlCon{
public static void main(String args[]){
//Entering the data
                Scanner k = new Scanner(System.in);
                System.out.println("enter User_id");
                int id = k.nextInt();

                System.out.println("enter name");
                String name = k.next();

                System.out.println("enter User_Password");
                String pass= k.next();

                System.out.println("enter User_E-Mail");
                String mail =k.next();

                System.out.println("enter User_mobile");
                int mob = k.nextInt();

                System.out.println("enter User_address");
                String cls = k.next();

                System.out.println("enter Admin_id");
                int admin = k.nextInt();

                //Inserting data using SQL query
                String sql = "insert into user
values('"+id+"','"+name+"','"+pass+"','"+mail+"','"+mob+"','"+cls+"','"+admin+"')";

        try{
                        Class.forName("com.mysql.jdbc.Driver");

                        Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/project","root","sriDory118");
                        //here project is database name, root is username and password
                        Statement stmt=con.createStatement();
                        System.out.println("");
                        System.out.println("Reading Data from Admin Table");
```

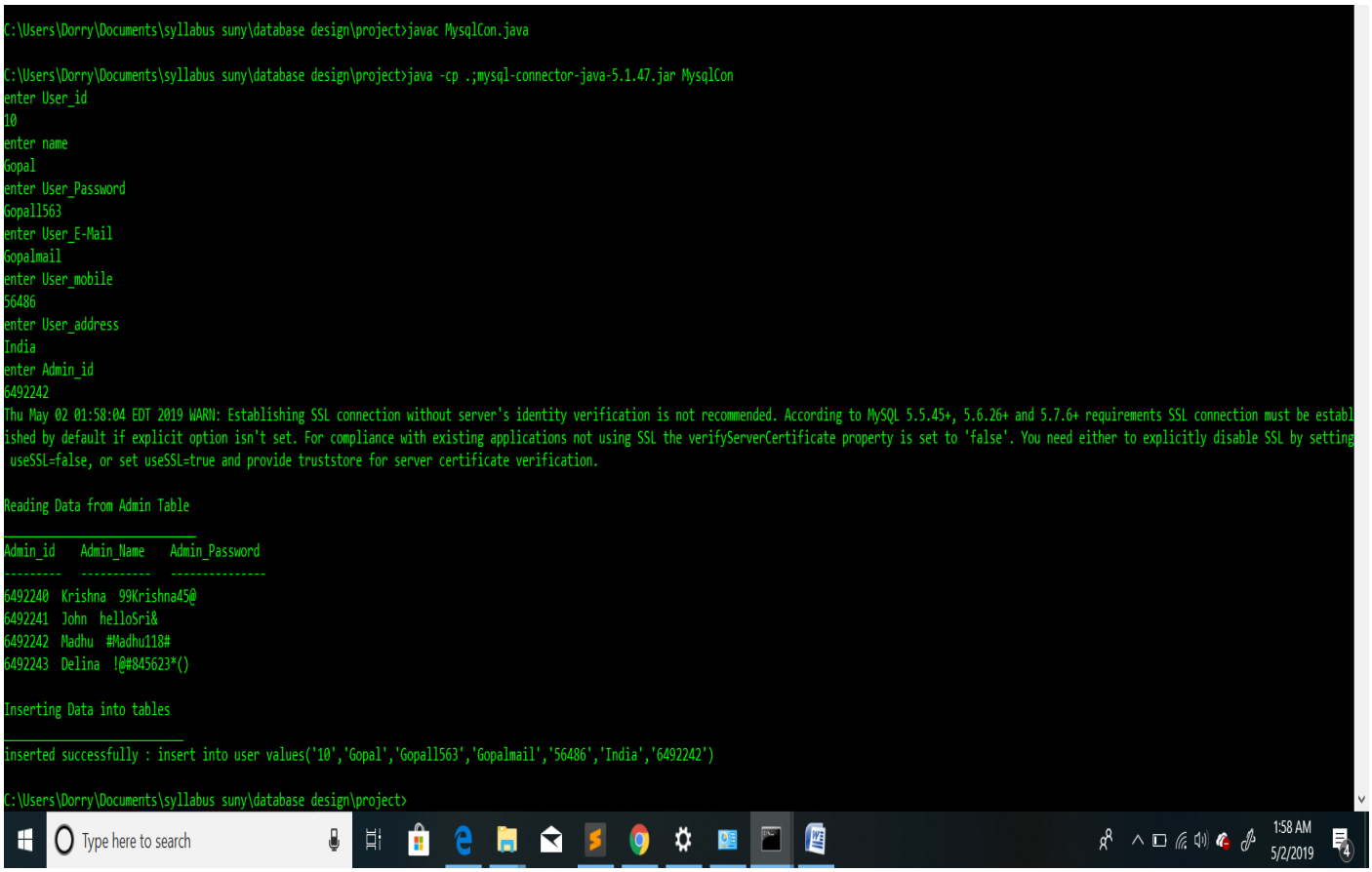
```
                    System.out.println("_____");
                    System.out.println("Admin_id    Admin_Name    Admin_Password");
                    System.out.println("---------  -----------  ---------------");
                    ResultSet rs=stmt.executeQuery("select * from admin         ");

                    while(rs.next())  {
        //Reading from Admin Table
        System.out.println(rs.getInt(1)+""+rs.getString(2)+""+rs.getString(3));
                    }
                    System.out.println("");
                    System.out.println("Inserting Data into tables");
                    System.out.println("_____");
                    int m=stmt.executeUpdate(sql);

                    if (m == 1)
                            System.out.println("inserted successfully : "+sql);
                    else
                            System.out.println("insertion failed");


                    con.close();
            }
            catch(Exception e){ System.out.println(e);}
        }
}
```

Basically the code is built in java with MySQL connector which helps in connecting the database project with the java program.

The commands used while complile : javac MysqlCon.java
(File Name)
Runing : java –cp .;mysql-connector-java-5.1.47.jar MysqlCon

Here the important point is that MySQL new versions were saving the passwords via hashing. So,the connectors which are used may not be compatible to find the passwords.
It can be connected to Database by coverting Password type to old version:

```
a) mysql> update user set password=OLD_PASSWORD('password') WHERE
user='username';
Query OK, 0 rows affected (0.02 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```


C:\Users\Dorry\Documents\syllabus suny\database design\project>javac MysqlCon.java
C:\Users\Dorry\Documents\syllabus suny\database design\project>java -cp .;mysql-connector-java-5.1.47.jar MysqlCon
enter User_id
10
enter name
Gopal
enter User_Password
Gopal1563
enter User_E-Mail
Gopalmail
enter User_mobile
56486
enter User_address
India
enter Admin_id
6492242
Thu May 02 01:58:04 EDT 2019 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertificate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate verification.
Reading Data from Admin Table
Admin_id Admin_Name Admin_Password
6492240 Krishna 99Krishna45@
6492241 John helloSri&
6492242 Madhu #Madhu118#
6492243 Delina !@#845623*()
Inserting Data into tables
inserted successfully : insert into user values('10','Gopal','Gopal1563','Gopalmail','56486','India','6492242')
C:\Users\Dorry\Documents\syllabus suny\database design\project>
Type here to search
1:58 AM
5/2/2019