

Q1. Create the tables for User, Circle, User_Circle, Messages and User_Inbox (to be created with help of Triggers)

And

Q2. Insert the rows into the created tables (User, Circle, User_Circle, and Messages).

a.User Table

create table User(

User_Name VARCHAR(40) NOT NULL,

User_ID integer not null ,

Password VARCHAR(100) NOT NULL,

Gender CHAR(1) NOT NULL,

Email_ID VARCHAR(100) NOT NULL,

UserContact integer(10) NOT NULL,

CreatedBy VARCHAR(40) NOT NULL,

Creation_date DATE,

PRIMARY KEY (User_ID)

);

INSERT INTO User values

('Booma',101,'PwdBooma','F','booma@gmail.com',9876543765,'Admin1','2019-03-03');

INSERT INTO User values

('Devi',102,'PwdDevi','F','devi@gmail.com',9652788225,'Admin1','2019-03-03');

INSERT INTO User values

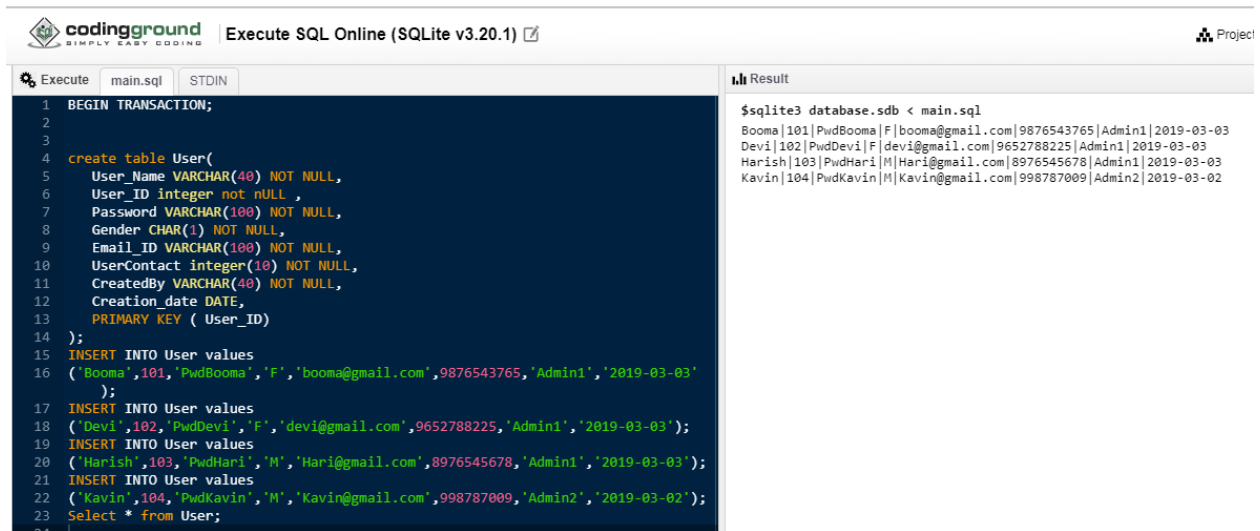
('Harish',103,'PwdHari','M','Hari@gmail.com',8976545678,'Admin1','2019-03-03');

INSERT INTO User values

('Kavin',104,'PwdKavin','M','Kavin@gmail.com',998787009,'Admin2','2019-03-02');

Select * from User;

Output:



The screenshot shows the codingground SQL execution interface. The left pane displays the SQL code, and the right pane shows the results of the query.

```
1 BEGIN TRANSACTION;
2
3
4 create table User(
5   User_Name VARCHAR(40) NOT NULL,
6   User_ID integer not null ,
7   Password VARCHAR(100) NOT NULL,
8   Gender CHAR(1) NOT NULL,
9   Email_ID VARCHAR(100) NOT NULL,
10  UserContact integer(10) NOT NULL,
11  CreatedBy VARCHAR(40) NOT NULL,
12  Creation_date DATE,
13  PRIMARY KEY ( User_ID)
14 );
15 INSERT INTO User values
16 ('Booma','101','PwDBooma','F','booma@gmail.com','9876543765','Admin1','2019-03-03'
17 );
18 INSERT INTO User values
19 ('Devi','102','PwDDevi','F','devi@gmail.com','9652788225','Admin1','2019-03-03');
20 INSERT INTO User values
21 ('Harish','103','PwDHari','M','Hari@gmail.com','8976545678','Admin1','2019-03-03');
22 INSERT INTO User values
23 ('Kavin','104','PwDKavin','M','Kavin@gmail.com','998787009','Admin2','2019-03-02');
24 Select * from User;
```

The results pane shows the output of the query:

```
$sqlite3 database.sdb < main.sql
Booma|101|PwDBooma|F|booma@gmail.com|9876543765|Admin1|2019-03-03
Devi|102|PwDDevi|F|devi@gmail.com|9652788225|Admin1|2019-03-03
Harish|103|PwDHari|M|Hari@gmail.com|8976545678|Admin1|2019-03-03
Kavin|104|PwDKavin|M|Kavin@gmail.com|998787009|Admin2|2019-03-02
```

b.Circle Table

create table Circle(

Circle_Name VARCHAR(40) NOT NULL,

Circle_ID INT NOT NULL,

CreatedBy VARCHAR(40) NOT NULL,

Creation_date DATE,

PRIMARY KEY (Circle_ID)

);

INSERT INTO Circle values('Circle1','9001','Admin1','2019-03-03');

INSERT INTO Circle values('Circle2','9002','Admin2','2019-03-03');

INSERT INTO Circle values('Circle3','9003','Admin1','2019-03-03');

INSERT INTO Circle values('Circle4','9004','Admin2','2019-03-03');

Output:

Select * from Circle;

Execute	main.sql	STDIN	Result
<pre>1 BEGIN TRANSACTION; 2 3 /* CIRCLE TABLE */ 4 create table Circle(5 Circle_Name VARCHAR(40) NOT NULL, 6 Circle_ID INT NOT NULL, 7 CreatedBy VARCHAR(40) NOT NULL, 8 Creation_date DATE, 9 PRIMARY KEY (Circle_ID) 10); 11 12 INSERT INTO Circle values('Circle1','9001','Admin1','2019-03-03'); 13 INSERT INTO Circle values('Circle2','9002','Admin2','2019-03-03'); 14 INSERT INTO Circle values('Circle3','9003','Admin1','2019-03-03'); 15 INSERT INTO Circle values('Circle4','9004','Admin2','2019-03-03'); 16 17 Select * from Circle; 18</pre>			<pre>\$sqlite3 database.sdb < main.sql Circle1 9001 Admin1 2019-03-03 Circle2 9002 Admin2 2019-03-03 Circle3 9003 Admin1 2019-03-03 Circle4 9004 Admin2 2019-03-03</pre>

c. User Circle Table

create table User_Circle(User_ID INT NOT NULL, Circle_ID INT NOT NULL,
Subscribe_Unsubscribe CHAR(1) NOT NULL, FOREIGN KEY (User_ID) REFERENCES User(User_ID),
FOREIGN KEY (Circle_ID) REFERENCES Circle(Circle_ID));

INSERT INTO User_Circle values('101','9004','Y');

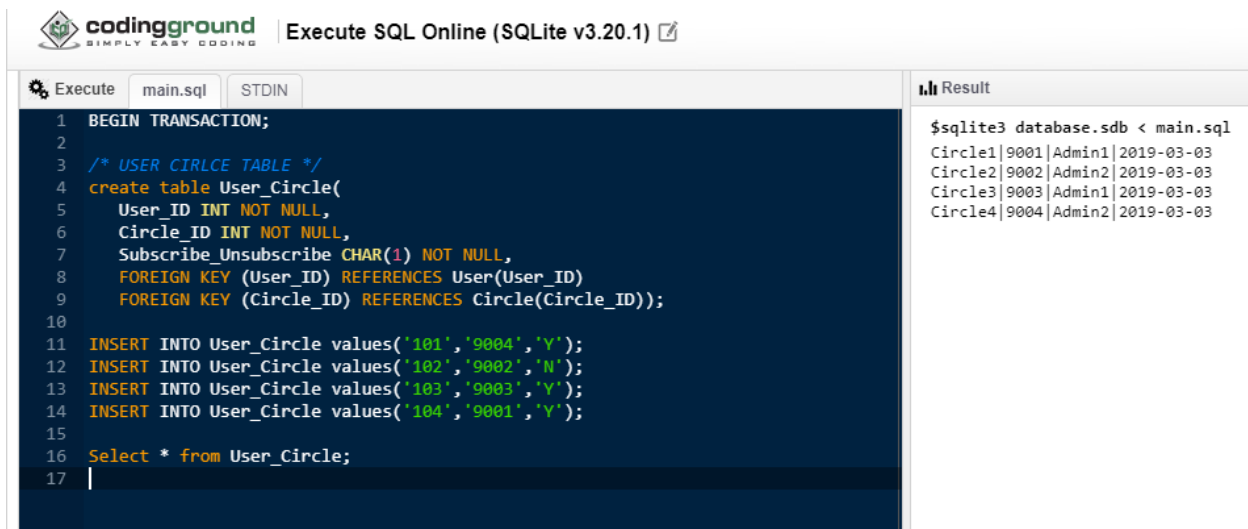
INSERT INTO User_Circle values('102','9002','N');

INSERT INTO User_Circle values('103','9003','Y');

INSERT INTO User_Circle values('104','9001','Y');

Output:

Select * from User_Circle;



The screenshot shows a web-based SQL execution interface. The top bar includes the 'codingground' logo and the text 'Execute SQL Online (SQLite v3.20.1)'. Below this, there are tabs for 'Execute', 'main.sql', and 'STDIN'. The main area displays the SQL code being executed, which includes creating the 'User_Circle' table with foreign key constraints and inserting four rows of data. The 'Result' tab on the right shows the output of the SQL execution, displaying the four rows of data in a table format.

```
1 BEGIN TRANSACTION;
2
3 /* USER CIRCLE TABLE */
4 create table User_Circle(
5     User_ID INT NOT NULL,
6     Circle_ID INT NOT NULL,
7     Subscribe_Unsubscribe CHAR(1) NOT NULL,
8     FOREIGN KEY (User_ID) REFERENCES User(User_ID)
9     FOREIGN KEY (Circle_ID) REFERENCES Circle(Circle_ID));
10
11 INSERT INTO User_Circle values('101','9004','Y');
12 INSERT INTO User_Circle values('102','9002','N');
13 INSERT INTO User_Circle values('103','9003','Y');
14 INSERT INTO User_Circle values('104','9001','Y');
15
16 Select * from User_Circle;
17
```

CircleID	UserID	Subscribe_Unsubscribe	CircleName	UserName	CircleDate
9001	104	Y	Circle1	Admin2	2019-03-03
9002	102	N	Circle2	Admin2	2019-03-03
9003	103	Y	Circle3	Admin1	2019-03-03
9004	101	Y	Circle4	Admin1	2019-03-03

d. Message Table

create table Message(msgID Varchar primary key, SentByUser_ID INT NOT NULL, RecByUser_ID INT,

Msg varchar(200), SentOn date not null, RecOn date not null ,CircleFlag char(1) not null, Circle_ID INT NOT NULL);

INSERT INTO Message values

('MSG001','101','104','Hi','2019-03-02','2019-03-03','N','');

INSERT INTO Message values

('MSG002','104','103','Hello','2019-03-03','2019-03-04','N','');

INSERT INTO Message values

('MSG003','103','','Welcome','2019-03-04','2019-03-05','Y','9002');

INSERT INTO Message values

('MSG004','102','101','Congrats','2019-03-03','2019-03-03','N','');

Output:

Select * from Message;



The screenshot shows a web interface for executing SQL queries. The top bar includes the 'codingground' logo and the text 'Execute SQL Online (SQLite v3.20.1)'. Below this, there are tabs for 'Execute', 'main.sql', and 'STDIN'. The main area displays the following SQL code:

```
1 BEGIN TRANSACTION;
2
3 /* Message TABLE */
4 create table Message(msgID Varchar primary key,
5   SentByUser_ID INT NOT NULL,
6   RecByUser_ID INT,
7   Msg varchar(200), SentOn date not null, RecOn date not null ,CircleFlag char(1)
8     ) not null, Circle_ID INT NOT NULL);
9
10 INSERT INTO Message values
11 ('MSG001','101','104','Hi','2019-03-02','2019-03-03','N','');
12 INSERT INTO Message values
13 ('MSG002','104','103','Hello','2019-03-03','2019-03-04','N','');
14 INSERT INTO Message values
15 ('MSG003','103','','Welcome','2019-03-04','2019-03-05','Y','9002');
16 INSERT INTO Message values
17 ('MSG004','102','101','Congrats','2019-03-03','2019-03-03','N','');
18
19 Select * from Message;
```

On the right side, under the 'Result' tab, the output is displayed as a table:

```
$sqlite3 database.sdb < main.sql
MSG001|101|104|Hi|2019-03-02|2019-03-03|N|
MSG002|104|103|Hello|2019-03-03|2019-03-04|N|
MSG003|103|||Welcome|2019-03-04|2019-03-05|Y|9002
MSG004|102|101|Congrats|2019-03-03|2019-03-03|N|
```

e. User Inbox table:

Create table User_Inbox(messageId Varchar, Message varchar, From_user varchar);

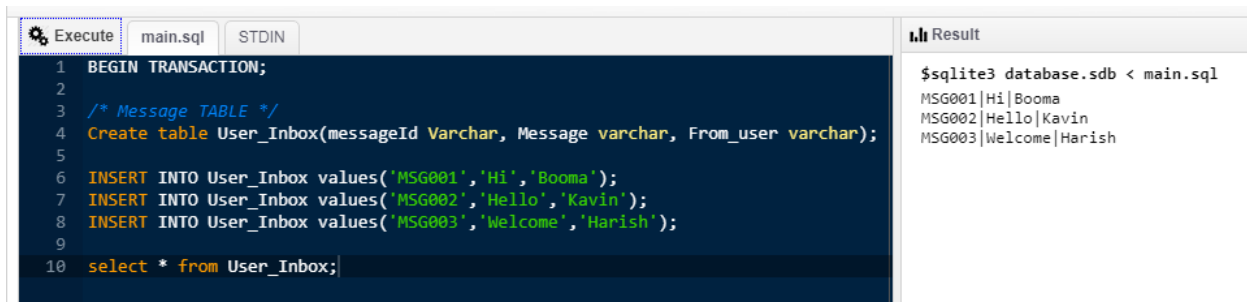
INSERT INTO User_Inbox values('MSG001','Hi','Booma');

INSERT INTO User_Inbox values('MSG002','Hello','Kavin');

INSERT INTO User_Inbox values('MSG003','Welcome','Harish');

Output:

select * from User_Inbox;



The screenshot shows a SQL execution interface with a dark theme. The 'main.sql' tab is active, displaying the following SQL code:

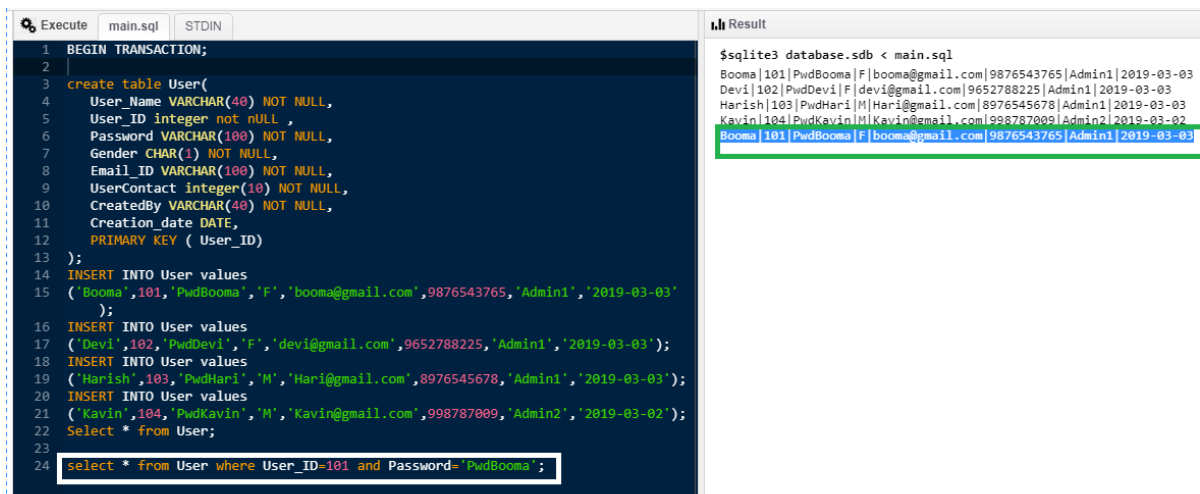
```
1 BEGIN TRANSACTION;
2
3 /* Message TABLE */
4 Create table User_Inbox(messageId Varchar, Message varchar, From_user varchar);
5
6 INSERT INTO User_Inbox values('MSG001','Hi','Booma');
7 INSERT INTO User_Inbox values('MSG002','Hello','Kavin');
8 INSERT INTO User_Inbox values('MSG003','Welcome','Harish');
9
10 select * from User_Inbox;
```

The 'Result' tab on the right shows the output of the \$sqlite3 database.sdb < main.sql command:

```
MSG001|Hi|Booma
MSG002|Hello|Kavin
MSG003|Welcome|Harish
```

Q3. Fetch the row from User table based on Id and Password.

select * from User where User_ID=101 and Password='PwdBooma';



The screenshot shows a SQL execution interface with a dark theme. The 'main.sql' tab is active, displaying the following SQL code:

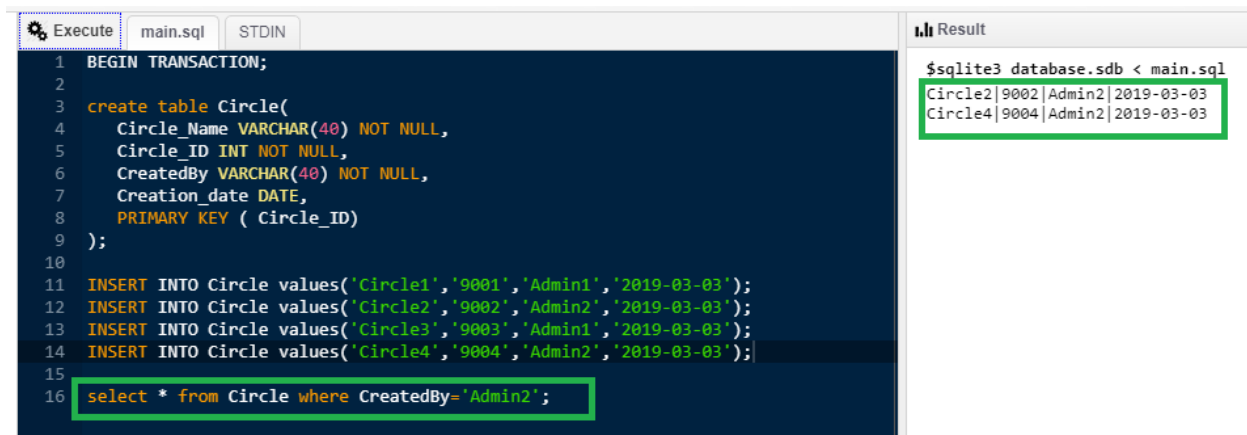
```
1 BEGIN TRANSACTION;
2
3 create table User(
4   User_Name VARCHAR(40) NOT NULL,
5   User_ID integer not null,
6   Password VARCHAR(100) NOT NULL,
7   Gender CHAR(1) NOT NULL,
8   Email_ID VARCHAR(100) NOT NULL,
9   UserContact integer(10) NOT NULL,
10  CreatedBy VARCHAR(40) NOT NULL,
11  Creation_date DATE,
12  PRIMARY KEY ( User_ID)
13 );
14 INSERT INTO User values
15 ('Booma',101,'PwdBooma','F','booma@gmail.com',9876543765,'Admin1','2019-03-03');
16 INSERT INTO User values
17 ('Devi',102,'PwdDevi','F','devi@gmail.com',9652788225,'Admin1','2019-03-03');
18 INSERT INTO User values
19 ('Harish',103,'PwdHari','M','Hari@gmail.com',8976545678,'Admin1','2019-03-03');
20 INSERT INTO User values
21 ('Kavin',104,'PwdKavin','M','Kavin@gmail.com',998787009,'Admin2','2019-03-02');
22 Select * from User;
23
24 select * from User where User_ID=101 and Password='PwdBooma';
```

The 'Result' tab on the right shows the output of the \$sqlite3 database.sdb < main.sql command. The result is a table with 7 columns: User_Name, User_ID, Password, Gender, Email_ID, UserContact, and Creation_date. The row for User_ID 101 is highlighted with a green box:

```
Booma|101|PwdBooma|F|booma@gmail.com|9876543765|Admin1|2019-03-03
```

Q4. Fetch all the rows from Circle table based on the field created by.

select * from Circle where CreatedBy='Admin2';



The screenshot shows an online SQL editor interface. The left pane contains the following SQL code:

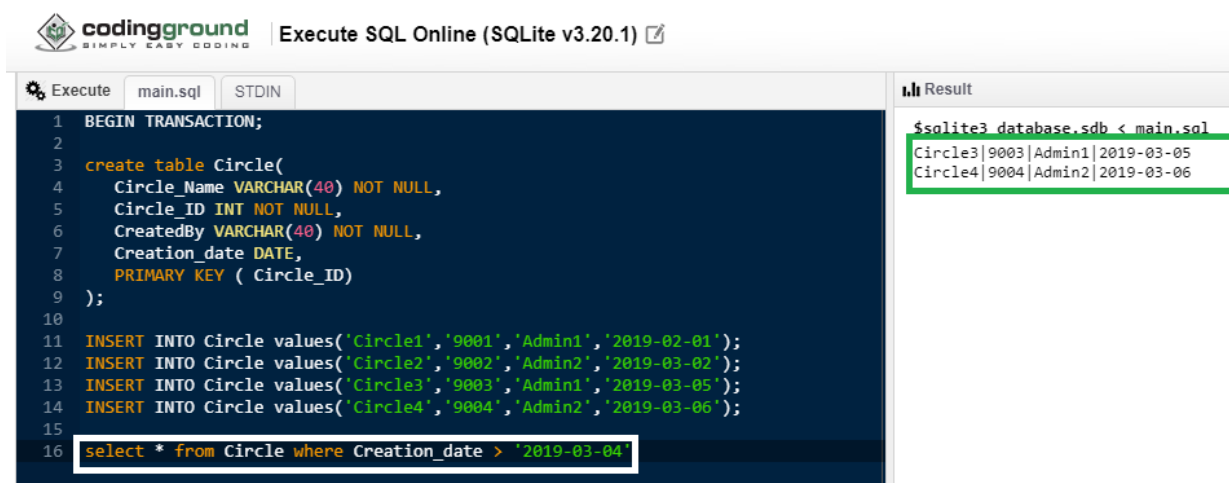
```
1 BEGIN TRANSACTION;
2
3 create table Circle(
4     Circle_Name VARCHAR(40) NOT NULL,
5     Circle_ID INT NOT NULL,
6     CreatedBy VARCHAR(40) NOT NULL,
7     Creation_date DATE,
8     PRIMARY KEY ( Circle_ID)
9 );
10
11 INSERT INTO Circle values('Circle1','9001','Admin1','2019-03-03');
12 INSERT INTO Circle values('Circle2','9002','Admin2','2019-03-03');
13 INSERT INTO Circle values('Circle3','9003','Admin1','2019-03-03');
14 INSERT INTO Circle values('Circle4','9004','Admin2','2019-03-03');
15
16 select * from Circle where CreatedBy='Admin2';
```

The right pane shows the result of the query:

```
$sqlite3 database.sdb < main.sql
Circle2|9002|Admin2|2019-03-03
Circle4|9004|Admin2|2019-03-03
```

Q5. Fetch all the Circles created after the particular Date.

select * from Circle where Creation_date > '2019-03-04'



The screenshot shows an online SQL editor interface. The left pane contains the following SQL code:


```
1 BEGIN TRANSACTION;
2
3 create table Circle(
4     Circle_Name VARCHAR(40) NOT NULL,
5     Circle_ID INT NOT NULL,
6     CreatedBy VARCHAR(40) NOT NULL,
7     Creation_date DATE,
8     PRIMARY KEY ( Circle_ID)
9 );
10
11 INSERT INTO Circle values('Circle1','9001','Admin1','2019-02-01');
12 INSERT INTO Circle values('Circle2','9002','Admin2','2019-03-02');
13 INSERT INTO Circle values('Circle3','9003','Admin1','2019-03-05');
14 INSERT INTO Circle values('Circle4','9004','Admin2','2019-03-06');
15
16 select * from Circle where Creation_date > '2019-03-04';
```

The right pane shows the result of the query:

```
$sqlite3 database.sdb < main.sql
Circle3|9003|Admin1|2019-03-05
Circle4|9004|Admin2|2019-03-06
```

Q6. Fetch all the User Ids from UserCircle table subscribed to particular Circle.

Select User_ID from User_Circle where Subscribe_Unsubscribe='Y' and Circle_ID='9004';

 Execute SQL Online (SQLite v3.20.1)

Execute main.sql STDIN


```
1 BEGIN TRANSACTION;
2
3 create table User_Circle(
4     User_ID INT NOT NULL,
5     Circle_ID INT NOT NULL,
6     Subscribe_Unsubscribe CHAR(1) NOT NULL,
7     FOREIGN KEY (User_ID) REFERENCES User(User_ID)
8     FOREIGN KEY (Circle_ID) REFERENCES Circle(Circle_ID));
9
10 INSERT INTO User_Circle values('101','9004','Y');
11 INSERT INTO User_Circle values('102','9002','N');
12 INSERT INTO User_Circle values('103','9004','Y');
13 INSERT INTO User_Circle values('104','9001','Y');
14
15 Select User_ID from User_Circle where Subscribe_Unsubscribe='Y' and Circle_ID
    ='9004';
```

Result

```
$sqlite3 database.sdb < main.sql
101
103
```

Q7. Write UpDate query to unsubscribe to particular Circle for the given User Id.

Update User_Circle set Subscribe_Unsubscribe='N' where User_ID='101';

 Execute SQL Online (SQLite v3.20.1)

Execute main.sql STDIN

```
1 BEGIN TRANSACTION;
2
3 create table User_Circle(
4     User_ID INT NOT NULL,
5     Circle_ID INT NOT NULL,
6     Subscribe_Unsubscribe CHAR(1) NOT NULL,
7     FOREIGN KEY (User_ID) REFERENCES User(User_ID)
8     FOREIGN KEY (Circle_ID) REFERENCES Circle(Circle_ID));
9
10 INSERT INTO User_Circle values('101','9004','Y');
11 INSERT INTO User_Circle values('102','9002','N');
12 INSERT INTO User_Circle values('103','9004','Y');
13 INSERT INTO User_Circle values('104','9001','Y');
14
15 Select * from User_Circle;
16
17 Update User_Circle set Subscribe_Unsubscribe='N' where User_ID='101';
18 Select * from User_Circle;
```

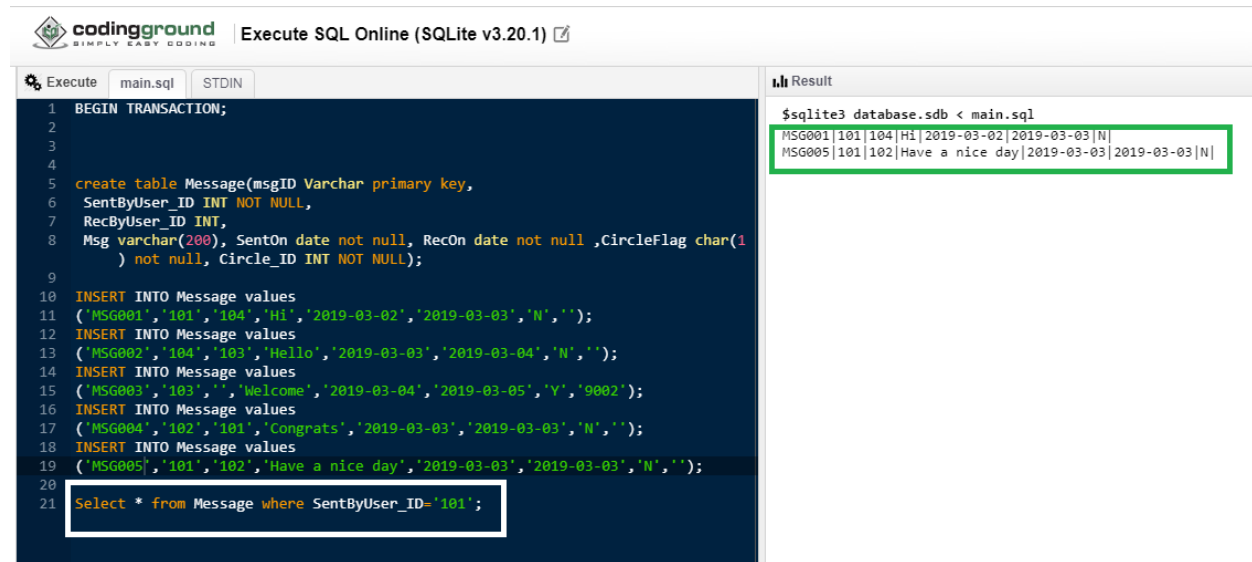
Result

```
$sqlite3 database.sdb < main.sql
101|9004|Y
102|9002|N
103|9004|Y
104|9001|Y
101|9004|N
102|9002|N
103|9004|Y
104|9001|Y
```


Q8. Fetch all the Messages from the Messages table sent by a particular User

SQL Query: `Select * from Message where SentByUser_ID='101';`

Output:



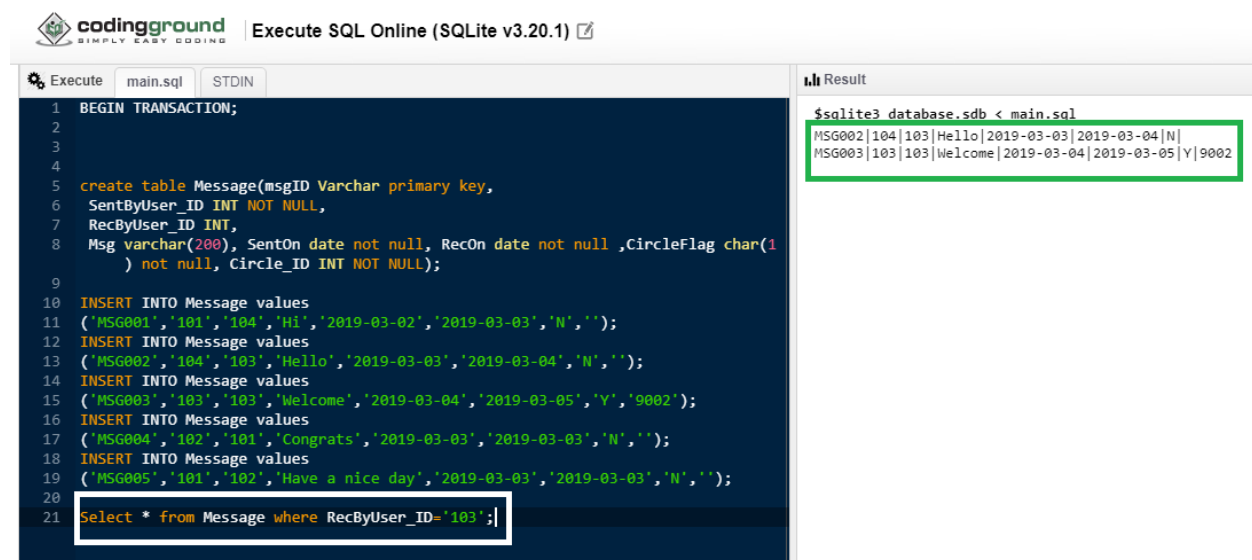
The screenshot shows the codingground SQL execution interface. The SQL query is: `SELECT * FROM Message WHERE SentByUser_ID='101';`. The results are displayed in a table with 7 columns: msgID, SentByUser_ID, RecByUser_ID, Msg, SentOn, RecOn, and CircleFlag. The results are:

msgID	SentByUser_ID	RecByUser_ID	Msg	SentOn	RecOn	CircleFlag
MSG001	101	104	Hi	2019-03-02	2019-03-03	N
MSG005	101	102	Have a nice day	2019-03-03	2019-03-03	N

Q9. Fetch all the Messages from the Messages table received by a particular User

SQL Query: `Select * from Message where RecByUser_ID='103';`

Output:



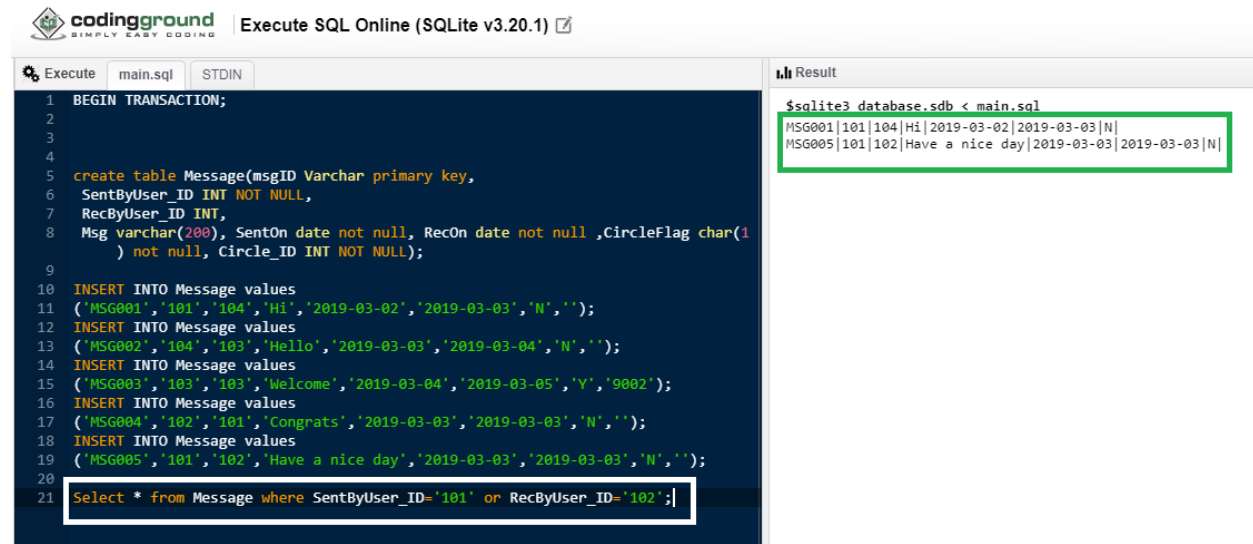
The screenshot shows the codingground SQL execution interface. The SQL query is: `SELECT * FROM Message WHERE RecByUser_ID='103';`. The results are displayed in a table with 7 columns: msgID, SentByUser_ID, RecByUser_ID, Msg, SentOn, RecOn, and CircleFlag. The results are:

msgID	SentByUser_ID	RecByUser_ID	Msg	SentOn	RecOn	CircleFlag
MSG002	104	103	Hello	2019-03-03	2019-03-04	N
MSG003	103	103	Welcome	2019-03-04	2019-03-05	Y 9002

Q10. Fetch all the Messages from the Messages table sent/received by a particular User. (both the Messages)

SQL Query: Select * from Message where SentByUser_ID='101' or RecByUser_ID='102';

Output:



The screenshot shows a web-based SQL editor interface. The left pane contains a SQL script with the following content:

```
1 BEGIN TRANSACTION;
2
3
4
5 create table Message(msgID Varchar primary key,
6   SentByUser_ID INT NOT NULL,
7   RecByUser_ID INT,
8   Msg varchar(200), SentOn date not null, RecOn date not null ,CircleFlag char(1
9     ) not null, Circle_ID INT NOT NULL);
10
11 INSERT INTO Message values
12   ( 'MSG001','101','104','Hi','2019-03-02','2019-03-03','N','');
13 INSERT INTO Message values
14   ( 'MSG002','104','103','Hello','2019-03-03','2019-03-04','N','');
15 INSERT INTO Message values
16   ( 'MSG003','103','103','Welcome','2019-03-04','2019-03-05','Y','0002');
17 INSERT INTO Message values
18   ( 'MSG004','102','101','Congrats','2019-03-03','2019-03-03','N','');
19 INSERT INTO Message values
20   ( 'MSG005','101','102','Have a nice day','2019-03-03','2019-03-03','N','');
21
22 Select * from Message where SentByUser_ID= '101' or RecByUser_ID='102';
```

The right pane shows the results of the query, titled "Result". It displays the following data:

```
$sqlite3 database.sdb < main.sql
MSG001|101|104|Hi|2019-03-02|2019-03-03|N|
MSG005|101|102|Have a nice day|2019-03-03|2019-03-03|N|
```

Q11. Write a query to send Message from particular User to another User (Use Message table - insert statement)

Message table structure:

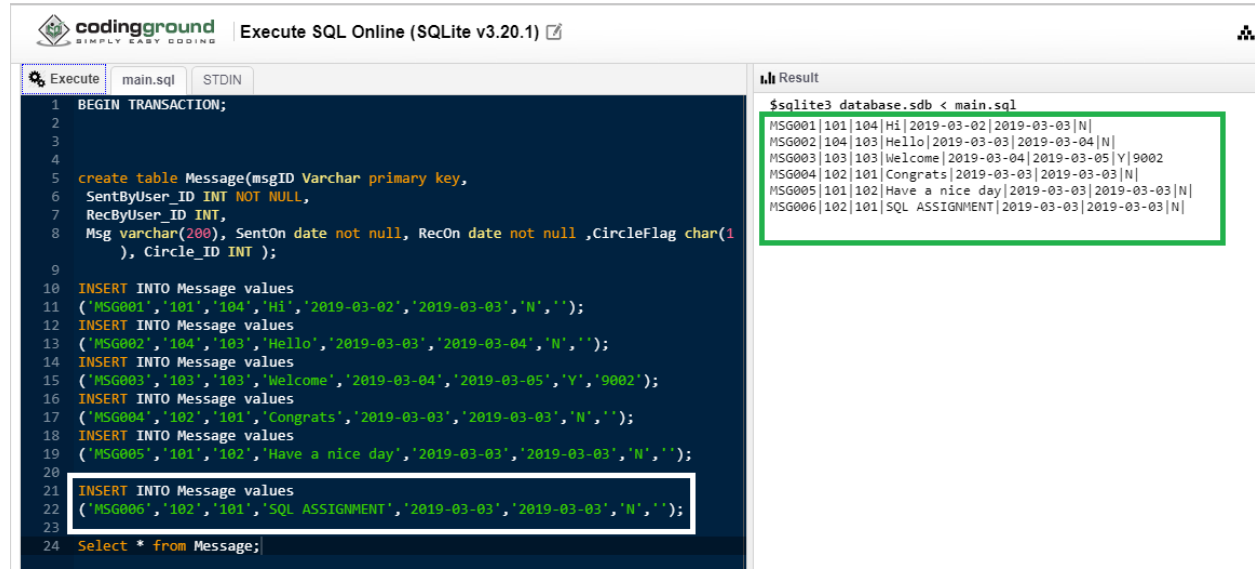
create table Message(msgID Varchar primary key, SentByUser_ID INT NOT NULL, RecByUser_ID INT,

Msg varchar(200), SentOn date not null, RecOn date not null ,CircleFlag char(1), Circle_ID INT);

SQL Query:

INSERT INTO Message values ('MSG006','102','101','SQL ASSIGNMENT','2019-03-03','2019-03-03','N','');

Output:



The screenshot shows the codingground SQL execution interface. The left pane displays the SQL code being executed, and the right pane shows the results of the transaction.

```
1 BEGIN TRANSACTION;
2
3
4
5 create table Message(msgID Varchar primary key,
6   SentByUser_ID INT NOT NULL,
7   RecByUser_ID INT,
8   Msg varchar(200), SentOn date not null, RecOn date not null ,CircleFlag char(1
9   ), Circle_ID INT );
10
11 INSERT INTO Message values
12   ('MSG001','101','104','Hi','2019-03-02','2019-03-03','N','');
13 INSERT INTO Message values
14   ('MSG002','104','103','Hello','2019-03-03','2019-03-04','N','');
15 INSERT INTO Message values
16   ('MSG003','103','103','Welcome','2019-03-04','2019-03-05','Y','9002');
17 INSERT INTO Message values
18   ('MSG004','102','101','Congrats','2019-03-03','2019-03-03','N','');
19 INSERT INTO Message values
20   ('MSG005','101','102','Have a nice day','2019-03-03','2019-03-03','N','');
21
22 INSERT INTO Message values
23   ('MSG006','102','101','SQL ASSIGNMENT','2019-03-03','2019-03-03','N','');
24
25 Select * from Message;
```

The results pane shows the output of the transaction:

```
sqlite3 database.sdb < main.sql
MSG001|101|104|Hi|2019-03-02|2019-03-03|N|
MSG002|104|103|Hello|2019-03-03|2019-03-04|N|
MSG003|103|103|Welcome|2019-03-04|2019-03-05|Y|9002
MSG004|102|101|Congrats|2019-03-03|2019-03-03|N|
MSG005|101|102|Have a nice day|2019-03-03|2019-03-03|N|
MSG006|102|101|SQL ASSIGNMENT|2019-03-03|2019-03-03|N|
```

Q12. Write a query to send Message from particular User to particular Circle(Use Message table - insert statement)

```
create table User_Circle(
    User_ID INT NOT NULL,
    Circle_ID INT NOT NULL,
    Subscribe_Unsubscribe CHAR(1) NOT NULL,
    FOREIGN KEY (User_ID) REFERENCES User(User_ID)
    FOREIGN KEY (Circle_ID) REFERENCES Circle(Circle_ID));
```

```
INSERT INTO User_Circle values('101','9004','Y');
```

```
INSERT INTO User_Circle values('102','9002','N');
```

```
INSERT INTO User_Circle values('103','9004','Y');
```

```
INSERT INTO User_Circle values('104','9001','Y');
```

```
Select * from User_Circle;
```

```
/*Message Table*/
```

create table Message(msgID Varchar primary key,

SentByUser_ID INT NOT NULL, RecByUser_ID INT, Msg varchar(200), SentOn date not null, RecOn date not null);

INSERT INTO Message values

('MSG001','101','104','Hi','2019-03-02','2019-03-03');

INSERT INTO Message values

('MSG002','104','103','Hello','2019-03-03','2019-03-04');

INSERT INTO Message values

('MSG003','103','103','Welcome','2019-03-04','2019-03-05');

INSERT INTO Message values

('MSG004','102','101','Congrats','2019-03-03','2019-03-03');

INSERT INTO Message values

('MSG006','102','101','SQL ASSIGNMENT','2019-03-03','2019-03-03');

Select * from Message where RecByUser_ID IN(select User_ID from User_Circle where Circle_ID='9004')

SQL Query:

Select * from Message where **RecByUser_ID** IN(select User_ID from User_Circle where Circle_ID='9004')

Output:

```
2 BEGIN TRANSACTION;
3 /*USER Circle Table*/
4 create table User_Circle(
5     User_ID INT NOT NULL,
6     Circle_ID INT NOT NULL,
7     Subscribe_Unsubscribe CHAR(1) NOT NULL,
8     FOREIGN KEY (User_ID) REFERENCES User(User_ID)
9     FOREIGN KEY (Circle_ID) REFERENCES Circle(Circle_ID));
10 INSERT INTO User_Circle values('101','9004','Y');
11 INSERT INTO User_Circle values('102','9002','N');
12 INSERT INTO User_Circle values('103','9004','Y');
13 INSERT INTO User_Circle values('104','9001','Y');
14 Select * from User_Circle;
15
16 /*Message Table*/
17 create table Message(msgID Varchar primary key,
18     SentByUser_ID INT NOT NULL, RecByUser_ID INT, Msg varchar(200), SentOn date
19     not null, RecOn date not null);
20 INSERT INTO Message values
21 ('MSG001','101','104','Hi','2019-03-02','2019-03-03');
22 INSERT INTO Message values
23 ('MSG002','104','103','Hello','2019-03-03','2019-03-04');
24 INSERT INTO Message values
25 ('MSG003','103','103','Welcome','2019-03-04','2019-03-05');
26 INSERT INTO Message values
27 ('MSG004','102','101','Congrats','2019-03-03','2019-03-03');
28 INSERT INTO Message values
29 ('MSG006','102','101','SQL ASSIGNMENT','2019-03-03','2019-03-03');
30 Select * from Message where RecByUser_ID IN( select User_ID from User_Circle
    where Circle_ID=9004)
```

```
$sqlite3 database.sdb < main.sql
101|9004|Y
102|9002|N
103|9004|Y
104|9001|Y
MSG002|104|103|Hello|2019-03-03|2019-03-04
MSG003|103|103|Welcome|2019-03-04|2019-03-05
MSG004|102|101|Congrats|2019-03-03|2019-03-03
MSG006|102|101|SQL ASSIGNMENT|2019-03-03|2019-03-03
```

Q13. Write a query to delete particular Message which you received from User(User Inbox table - delete statement)

```
/* USER INBOX*/
```

```
Create table User_Inbox(messageId Varchar, Message varchar, From_user varchar);
```

```
INSERT INTO User_Inbox values('MSG001','Hi','Booma');
```

```
INSERT INTO User_Inbox values('MSG002','Hello','Kavin');
```

```
INSERT INTO User_Inbox values('MSG003','Welcome','Harish');
```

```
INSERT INTO User_Inbox values('MSG004','Have a good day!!!','Swetha');
```

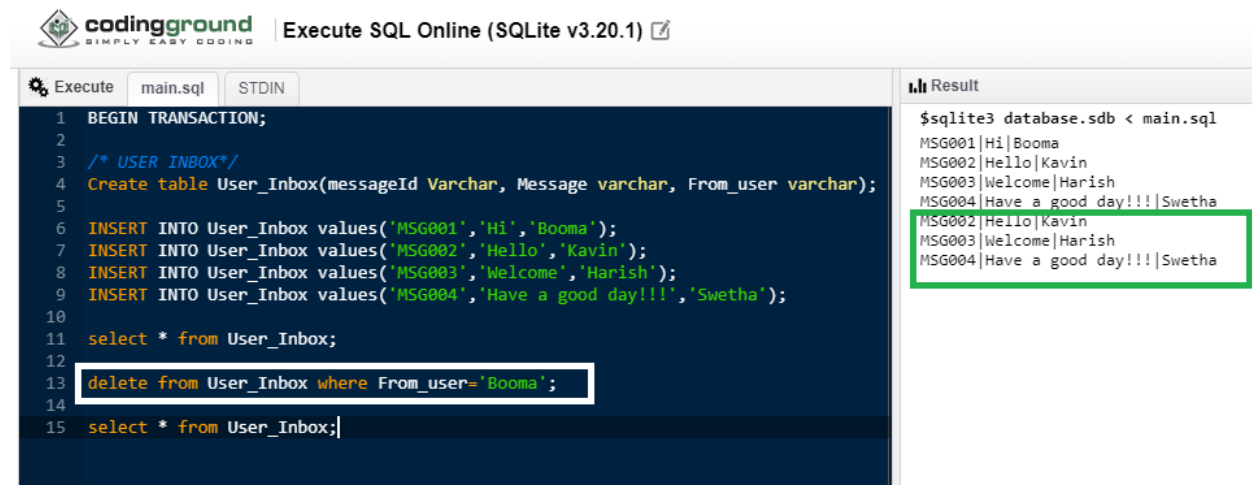
```
select * from User_Inbox;
```

```
delete from User_Inbox where From_user='Booma';
```

```
select * from User_Inbox;
```

SQL Query: DELETE FROM USER_INBOX WHERE FROM_USER='BOOMA';

Output:



The screenshot shows the codingground SQL Online interface. The SQL editor on the left contains the following code:

```
1 BEGIN TRANSACTION;
2
3 /* USER_INBOX*/
4 Create table User_Inbox(messageId Varchar, Message varchar, From_user varchar);
5
6 INSERT INTO User_Inbox values('MSG001','Hi','Booma');
7 INSERT INTO User_Inbox values('MSG002','Hello','Kavin');
8 INSERT INTO User_Inbox values('MSG003','Welcome','Harish');
9 INSERT INTO User_Inbox values('MSG004','Have a good day!!!','Swetha');
10
11 select * from User_Inbox;
12
13 delete from User_Inbox where From_user='Booma';
14
15 select * from User_Inbox;
```

The 'Result' pane on the right shows the output of the SQL execution:

```
$sqlite3 database.sdb < main.sql
MSG001|Hi|Booma
MSG002|Hello|Kavin
MSG003|Welcome|Harish
MSG004|Have a good day!!!|Swetha
```

The result shows that the message from 'Booma' (MSG001) has been successfully deleted, leaving only the messages from 'Kavin', 'Harish', and 'Swetha'.

Q14. Write a query to delete particular Message which you received from particular Circle(User Inbox table - delete statement)

Delete FROM [User_Inbox] WHERE From_user IN(SELECT circle FROM CIRCLE WHERE CIRCLE='circle2')

BEGIN TRANSACTION;

create table Circle(

Circle_Name VARCHAR(40) NOT NULL,

Circle_ID INT NOT NULL,

CreatedBy VARCHAR(40) NOT NULL,

Creation_date DATE,

PRIMARY KEY (Circle_ID)

);

INSERT INTO Circle values('Circle1','9001','Admin1','2019-03-03');

INSERT INTO Circle values('Circle2','9002','Admin2','2019-03-03');

INSERT INTO Circle values('Circle3','9003','Admin1','2019-03-03');

```
INSERT INTO Circle values('Circle4','9004','Admin2','2019-03-03');
```

```
Select * from Circle;
```

```
/* USER INBOX*/
```

```
Create table User_Inbox(messageId Varchar, Message varchar, From_user varchar);
```

```
INSERT INTO User_Inbox values('MSG001','Hi','Booma');
```

```
INSERT INTO User_Inbox values('MSG002','Hello','Kavin');
```

```
INSERT INTO User_Inbox values('MSG003','Welcome','Harish');
```

```
INSERT INTO User_Inbox values('MSG004','Have a good day!!!','Swetha');
```

```
select * from User_Inbox;
```

Query for the question:

```
delete from User_Inbox where From_user='Booma';
```

```
Delete FROM User_Inbox WHERE From_user IN(SELECT Circle_ID FROM Circle WHERE  
Circle_Name='Circle2');
```

```
select * from User_Inbox;
```