LAB 03

CSE 4308
Database Management Systems Lab

Hasin Mahtab Alvee
210042174
Department of CSE
B.Sc in Software Engineering

August 31, 2023

Contents

Introduction

In DBMS lab 03, we were tasked to create 3 tables, Doctor, Patient and Appointment. We must insert data of Instructors in the table as defined. And then we needed to return some data according to the Queries. Once all commands were done, we were told to run the .sql file as a script in the SQL Terminal to execute the code in file.

1 Create Doctor, Patient and Appointment tables

First we create the Doctor and Patients table with both having primary keys as constraints. For Doctor table it is the name and specialization. For the Patient table, it is the Patient No.

Then we create the Appointment table, where we set all three attributes, Patient No, Name, Specialization - as the composite primary keys.

```
CREATE TABLE DOCTOR(
NAME VARCHAR(20),
SPECIALIZATION CHAR(2),
FEE NUMBER(6),
CONSTRAINT PK_DOCTOR PRIMARY KEY(NAME, SPECIALIZATION)
);

CREATE TABLE PATIENT(
PATIENT_NO CHAR(5),
NAME VARCHAR(20) NOT NULL,
ADDRESS VARCHAR(20),
CONSTRAINT PK_PATIENT PRIMARY KEY(PATIENT_NO)
);

CREATE TABLE APPOINTMENT(
PATIENT_NO CHAR(5),
NAME VARCHAR(20),
SPECIALIZATION CHAR(2),
CONSTRAINT PK_APPOINTMENT PRIMARY KEY(PATIENT_NO, NAME,
    SPECIALIZATION)
);
```

1.1 Difficulties

While creating the user, some minor inconvenience were faced, such as -

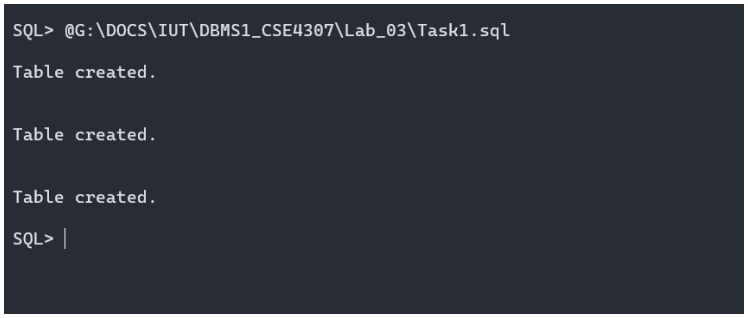


Figure 1:

- Settings the composite constraints was difficult for me.

2 Altering Table attributes

In the second task, we needed to Alter the tables and some attributes. First we added Appointment date attribute to Appointment table. We also changed the constraints for the appointment table, for which we first needed to Drop the previous constraint to assign a new composite constraint. We renamed columns and appointment table to Appointment Info. We also had to set some foreign key constraints to the Appointment Info table.

```
ALTER TABLE APPOINTMENT ADD APPOINTMENT_DATE DATE;

ALTER TABLE APPOINTMENT DROP CONSTRAINT PK_APPOINTMENT;

ALTER TABLE APPOINTMENT ADD CONSTRAINT PK_APPOINTMENT PRIMARY
    KEY(PATIENT_NO, NAME, SPECIALIZATION, APPOINTMENT_DATE);

ALTER TABLE APPOINTMENT RENAME COLUMN PATIENT_NO TO P_NO;

ALTER TABLE APPOINTMENT RENAME COLUMN NAME TO D_NAME;

RENAME APPOINTMENT TO APPOINTMENT_INFO;

ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT FK_APPOINTMENT_DOCTOR
    FOREIGN KEY(D_NAME, SPECIALIZATION) REFERENCES DOCTOR(NAME,
    SPECIALIZATION);

ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT FK_APPOINTMENT_PATIENT
    FOREIGN KEY(P_NO) REFERENCES PATIENT(PATIENT_NO);
```

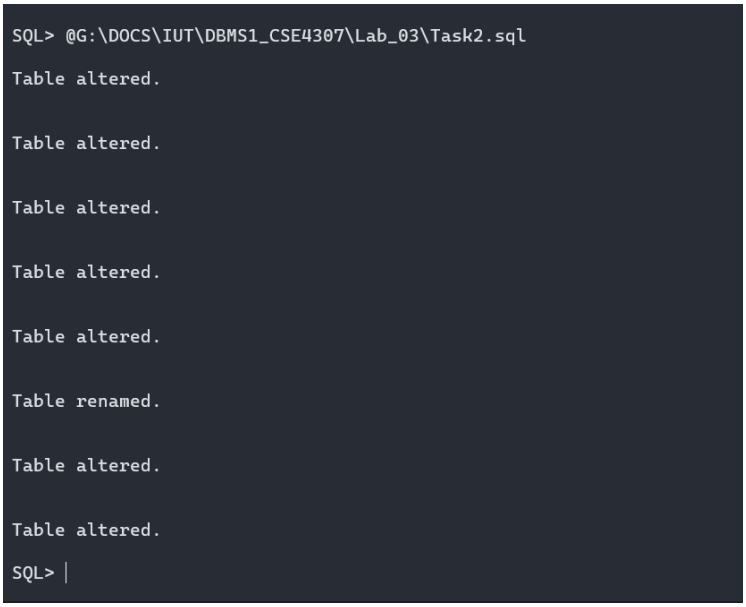


Figure 2:

2.1 Difficulties

While creating the table structure, some minor inconvenience were faced, such as -

- Altering the composite constrains for the Appointment table was confusing as I had to first drop the previous constraint before setting the new constraint.
- I forgot to mention the "COLUMN" attribute before the column name while renaming the columns at first, which was a bit difficult for me.

3 Inserting data into Table

Inserting data to the Doctor, Patient and Appointment Info tables.

```
-- DOCTORS

INSERT INTO DOCTOR VALUES ( 'MR. X', 'CS', 1500 );

INSERT INTO DOCTOR VALUES ( 'MR. Y', 'GS', 2000 );

INSERT INTO DOCTOR VALUES ( 'MR. A', 'IM', 1000 );

INSERT INTO DOCTOR VALUES ( 'MR. B', 'ER', 1200 );

-- PATIENTS
```

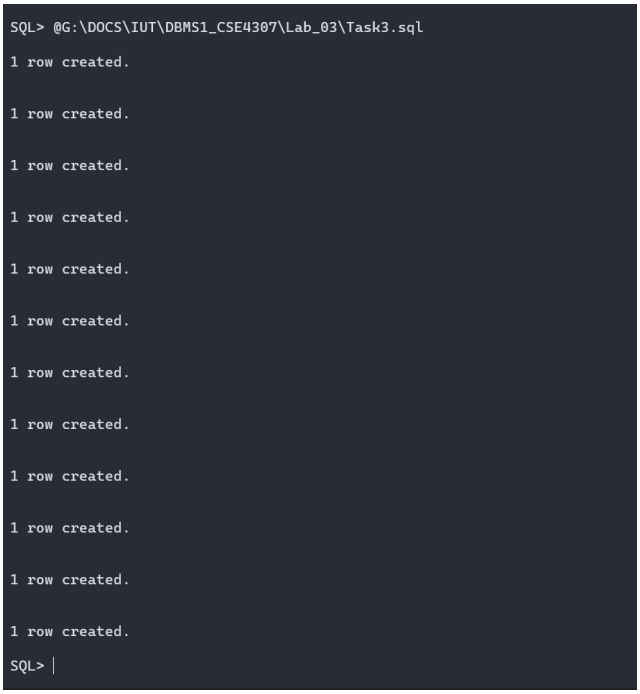


Figure 3:

```
INSERT INTO PATIENT VALUES ( 'P-101', 'A', 'DHK' );

INSERT INTO PATIENT VALUES ( 'P-102', 'B', 'KHL' );

INSERT INTO PATIENT VALUES ( 'P-103', 'C', 'DHK' );

INSERT INTO PATIENT VALUES ( 'P-104', 'D', 'KHL' );

-- APPOINTMENT_INFO

INSERT INTO APPOINTMENT_INFO VALUES ( 'P-101', 'MR. A', 'IM',
    TO_DATE ('2023-08-29', 'YYYY-MM-DD') );

INSERT INTO APPOINTMENT_INFO VALUES ( 'P-102', 'MR. B', 'ER',
    TO_DATE ('2023-08-29', 'YYYY-MM-DD') );

INSERT INTO APPOINTMENT_INFO VALUES ( 'P-103', 'MR. X', 'CS',
    TO_DATE ('2023-08-28', 'YYYY-MM-DD') );

INSERT INTO APPOINTMENT_INFO VALUES ( 'P-104', 'MR. Y', 'GS',
    TO_DATE ('2023-08-28', 'YYYY-MM-DD') );
```

3.1 Difficulties

While inserting the data, some minor inconvenience were faced, such as -

- Inserting the appointment date we had to convert it to date using the To Date method.

4 Returning the Data

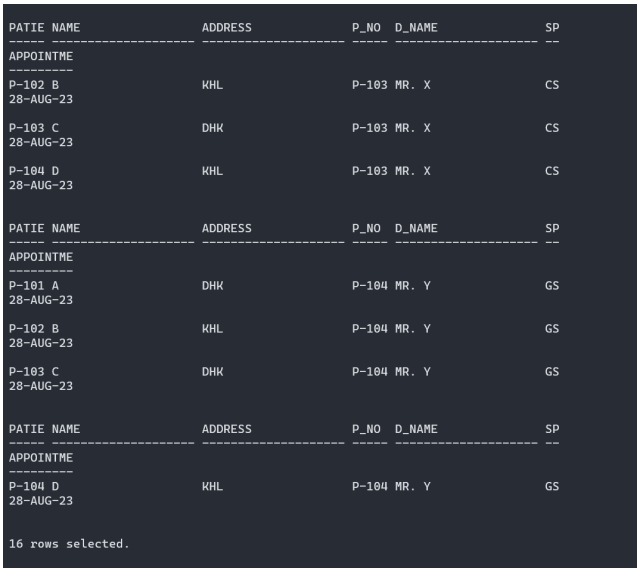
Once the data is inserted, we use specific queries to show the data. Some of the Queries are shown in the Figure.

```
SELECT NAME FROM DOCTOR WHERE FEE < 1500;

SELECT NAME FROM PATIENT WHERE ADDRESS = 'KHL';

SELECT * FROM PATIENT, APPOINTMENT_INFO;

SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO;

SELECT P.PATIENT_NO, P.ADDRESS FROM PATIENT     P WHERE
    P.PATIENT_NO IN (
   SELECT A.P_NO FROM APPOINTMENT_INFO A WHERE
       TRUNC(A.APPOINTMENT_DATE) = TRUNC(SYSDATE)
);

SELECT D.* FROM DOCTOR        D WHERE D.NAME IN (
   SELECT A.D_NAME FROM APPOINTMENT_INFO A, PATIENT     P WHERE
   A.P_NO = P.PATIENT_NO AND P.ADDRESS = 'DHK'
);

SELECT * FROM PATIENT WHERE PATIENT_NO IN (
   SELECT A.P_NO FROM APPOINTMENT_INFO A WHERE A.D_NAME IN (
      SELECT NAME FROM DOCTOR WHERE SPECIALIZATION = 'GS' OR FEE >
          1500 )
);
```

4.1 Difficulties

While creating the table structure, some minor inconvenience were faced, such as -

- It was very hard to come up with queries using the IN command and to nest commands.



```
PATIE NAME                ADDRESS             P_NO  D_NAME               SP
----- -------------------- ------------------- ----- -------------------- --
APPOINTME
---------
P-102 B                   KHL                 P-103 MR. X                CS
28-AUG-23

P-103 C                   DHK                 P-103 MR. X                CS
28-AUG-23

P-104 D                   KHL                 P-103 MR. X                CS
28-AUG-23


PATIE NAME                ADDRESS             P_NO  D_NAME               SP
----- -------------------- ------------------- ----- -------------------- --
APPOINTME
---------
P-101 A                   DHK                 P-104 MR. Y                GS
28-AUG-23

P-102 B                   KHL                 P-104 MR. Y                GS
28-AUG-23

P-103 C                   DHK                 P-104 MR. Y                GS
28-AUG-23


PATIE NAME                ADDRESS             P_NO  D_NAME               SP
----- -------------------- ------------------- ----- -------------------- --
APPOINTME
---------
P-104 D                   KHL                 P-104 MR. Y                GS
28-AUG-23


16 rows selected.
```

Figure 4:

5 Updating the Table data

Updating the data queries are simple using the Update and Set command. But due to constraints, this can not be performed. So I had to change the foreign key constraints with Alter.

```
UPDATE PATIENT SET NAME = 'K', ADDRESS = 'RAJ' WHERE NAME = 'A' AND
    ADDRESS = 'DHK';

ALTER TABLE APPOINTMENT_INFO DROP CONSTRAINT FK_APPOINTMENT_DOCTOR;

UPDATE APPOINTMENT_INFO SET APPOINTMENT_INFO.D_NAME= 'MS. Y' WHERE
    APPOINTMENT_INFO.D_NAME= 'MR. Y';

UPDATE DOCTOR SET DOCTOR.NAME = 'MS. Y' WHERE DOCTOR.NAME = 'MR. Y';

ALTER TABLE APPOINTMENT_INFO DROP CONSTRAINT FK_APPOINTMENT_PATIENT;

DELETE FROM APPOINTMENT_INFO WHERE P_NO = 'P-101';

DELETE FROM PATIENT WHERE PATIENT_NO = 'P-101';

ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT FK_APPOINTMENT_PATIENT
    FOREIGN KEY (P_NO) REFERENCES PATIENT(PATIENT_NO);

ALTER TABLE APPOINTMENT_INFO DROP CONSTRAINT FK_APPOINTMENT_PATIENT;
```

```
SQL> @G:\DOCS\IUT\DBMS1_CSE4307\Lab_03\Task5.sql

1 row updated.

Table altered.

1 row updated.

1 row updated.

Table altered.

1 row deleted.

1 row deleted.

Table altered.

Table altered.

3 rows deleted.

3 rows deleted.

Table altered.
SQL>
```

Figure 5:

```sql
DELETE FROM APPOINTMENT_INFO;

DELETE FROM PATIENT;

ALTER TABLE APPOINTMENT_INFO ADD CONSTRAINT FK_APPOINTMENT_PATIENT
    FOREIGN KEY (P_NO) REFERENCES PATIENT(PATIENT_NO);
```

5.1 Difficulties

While creating the table structure, some minor inconvenience were faced, such as -

- Changing the foreign keys were pretty hard.