Report on Lab-03

DATABASE MANAGEMENT SYSTEMS LAB

Submitted by

Adid-Al-Mahamud Shazid

Student Id: 210042172

Department: CSE

Programme: SWE

Course Title: CSE 4308

Submitted to

Zannatun Naim Sristy

Lecturer, Department of CSE

August 31, 2023

Introduction

In the lab class, we were given five tasks to solve using SQL command line to understand the basics of data definition and data manipulation.

Task 1

1. Write SQL statements to create the following tables with the given specifications:

(a) DOCTOR

NAME	VARCHAR2(20)	(e.g.: MR. X, MR. Y)
SPECIALIZATION	CHAR(2)	(e.g. CS, GS, IM, ER)
FEE	NUMBER	(e.g. 2000, 1500)
		Primary Key(NAME, SPECIALIZATION)

(b) PATIENT

PATIENT_NO	CHAR(5)	(e.g.: P-101) Primary Key
NAME	VARCHAR2(20)	(e.g.: A, B, C) Not Null
ADDRESS	VARCHAR2(10)	(e.g.: DHK, KHL, etc.)

(c) APPOINTMENT

PATIENT_NO	CHAR(5)	(e.g.: P-101)
NAME	VARCHAR2(20)	(e.g.: MR. X, MR. Y)
SPECIALIZATION	CHAR(2)	(e.g. CS, GS, IM, ER)
		Primary Key(PATIENT_NO, NAME,SPECIALIZATION)

1.1 Solution

```
CREATE TABLE DOCTOR
(
    NAME VARCHAR2(20),
    SPECIALIZATION CHAR(2),
    FEE NUMBER NOT NULL,
    CONSTRAINT PK_DOCTOR PRIMARY KEY (NAME,SPECIALIZATION)
);
```

```sql
CREATE TABLE PATIENT
(
    PATIENT_NO CHAR(5),
    NAME VARCHAR2(20) NOT NULL,
    ADDRESS VARCHAR2(10) NOT NULL,
    CONSTRAINT PK_PATIENT PRIMARY KEY (PATIENT_NO)
);

CREATE TABLE APPOINTMENT
(
    PATIENT_NO CHAR(5),
    NAME VARCHAR2(20),
    SPECIALIZATION CHAR(2),
    CONSTRAINT PK_APPOINTMENT PRIMARY KEY (PATIENT_NO,NAME,SPECIALIZATION)
);
```

1.2 Analysis and Explanation

This task was easy to complete since we already learned how to create tables in the last lab also we learned about primary key at the previous lab.

1.3 Difficulties

I did not face any difficulties when doing this task and no mentionable issues were encountered.

1.4 Output

```
SQL> @"D:\Study\Semester 3, Winter 2023\CSE 4308 Lab  Database Managemment System I\Lab 3\Task 1.sql";

Table created.

Table created.

Table created.
```

Task 2

2. Write SQL statements to perform the following alteration operations:

   (a) Add a new attribute 'APPOINTMENT_DATE' (DATE type) in APPOINTMENT table.

   (b) Modify the PRIMARY KEY of APPOINTMENT table and add APPOINTMENT_DATE too with the previous ones.

   (c) Rename the attribute PATIENT_NO, NAME from APPOINTMENT table to P_NO and D_NAME respectively.

   (d) Rename the table APPOINTMENT to APPOINTMENT_INFO.

   (e) Add two foreign key constraints FK_APPOINTMENT_DOCTOR and FK_APPOINTMENT_PATIENT that identifies D_NAME,SPECIALIZATION and P_NO as foreign keys.

2.1 Solution

```
ALTER TABLE APPOINTMENT ADD APPOINTMENT_DATE DATE NOT NULL;
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;
ALTER TABLE APPOINTMENT RENAME 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);
```

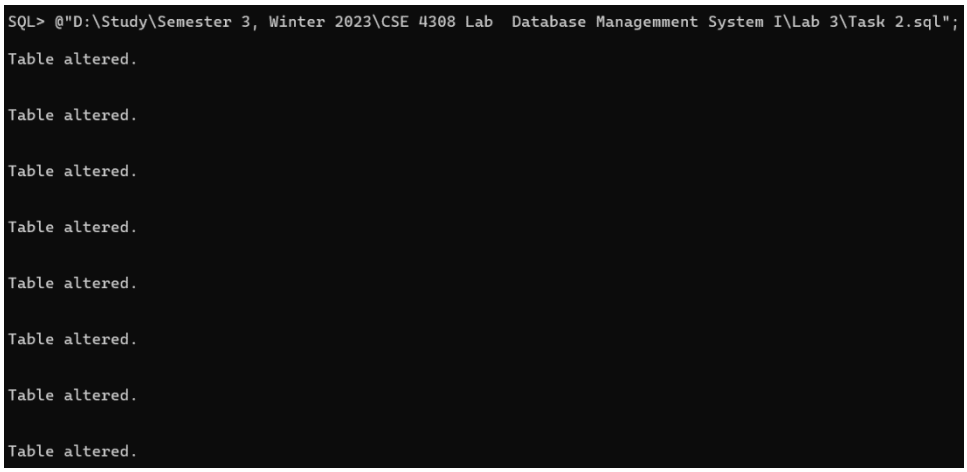
2.2 Analysis and Explanation

The task was easy to be solved following the manual of Lab-2 and Lab-3. Also, the class lectures helped me a lot to understand the basic of foreign key which is actually a referral of the primary key in another table.

2.3 Difficulties

During the first run I faced some difficulties as due to the alteration of table name and column names the name was not as like as it was. So, then I changed the name according to the alteration and my command run successfully.

2.4 Output



Task 3

3. Insert at least 3 records in each table following the example.

3.1 Solution

```
INSERT INTO DOCTOR VALUES('MR. X','CS',2000);
INSERT INTO DOCTOR VALUES('MR. Y','GS',2200);
INSERT INTO DOCTOR VALUES('MR. Z','IM',1000);
INSERT INTO DOCTOR VALUES('MR. P','ER',1800);
INSERT INTO DOCTOR VALUES('MR. Q','PQ',1500);
INSERT INTO PATIENT VALUES('P-101','A','DHK');
INSERT INTO PATIENT VALUES('P-102','B','KHL');
INSERT INTO PATIENT VALUES('P-103','C','RAJ');
INSERT INTO PATIENT VALUES('P-104','D','BAR');
INSERT INTO APPOINTMENT_INFO VALUES('P-101','MR. X','CS','25-AUG-2023');
INSERT INTO APPOINTMENT_INFO VALUES('P-102','MR. Y','GS','26-AUG-2023');
INSERT INTO APPOINTMENT_INFO VALUES('P-103','MR. Z','IM','26-AUG-2023');
INSERT INTO APPOINTMENT_INFO VALUES('P-104','MR. Q','PQ','27-AUG-2023');
```

3.2 Analysis and Explanation

I inserted some records into the tables I created in task 1. This task was also easy to complete.

3.3 Difficulties

I did not face any difficulties when doing this task.

3.4 Output

```
SQL> @"D:\Study\Semester 3, Winter 2023\CSE 4308 Lab  Database Managemment System I\Lab 3\Task 3.sql";
1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.
```

Task 4

4. Write SQL statements to answer the following queries:

(a) Find all the Doctors' names whose fees are less than 1500.

(b) Find all the Patients' names who live in 'KHL' city.

(c) Show the result of Cartesian Product between PATIENT and APPOINTMENT_INFO table.

(d) Show the result of Natural Join between PATIENT and APPOINTMENT_INFO table.

(e) Find all the Patient's names and their address who have an appointment today.

(f) Find all the Doctor-related information who have patients from 'DHK'.

(g) Find all Patient-related information who has an appointment with a doctor of 'GS' specialization or a doctor whose fee is greater than 1500.

4.1 Solution

```
SELECT NAME FROM DOCTOR WHERE FEE<1500;
SELECT NAME FROM PATIENT WHERE ADDRESS='KHL';
SELECT * FROM PATIENT, APPOINTMENT_INFO WHERE PATIENT.PATIENT_NO =
APPOINTMENT_INFO.P_NO;
SELECT * FROM PATIENT NATURAL JOIN APPOINTMENT_INFO WHERE PATIENT.PATIENT_NO =
APPOINTMENT_INFO.P_NO;
SELECT NAME, ADDRESS FROM PATIENT, APPOINTMENT_INFO WHERE PATIENT.PATIENT_NO =
APPOINTMENT_INFO.P_NO AND APPOINTMENT_INFO.APPOINTMENT_DATE= '26-AUG-2023';
SELECT DOCTOR.NAME,DOCTOR.SPECIALIZATION,DOCTOR.FEE FROM PATIENT, DOCTOR,
APPOINTMENT_INFO WHERE PATIENT.PATIENT_NO = APPOINTMENT_INFO.P_NO AND
APPOINTMENT_INFO.D_NAME = DOCTOR.NAME AND APPOINTMENT_INFO.SPECIALIZATION=
DOCTOR.SPECIALIZATION AND PATIENT.ADDRESS='DHK';
SELECT PATIENT.NAME,PATIENT.PATIENT_NO,PATIENT.ADDRESS FROM PATIENT, DOCTOR,
APPOINTMENT_INFO WHERE PATIENT.PATIENT_NO = APPOINTMENT_INFO.P_NO AND
APPOINTMENT_INFO.D_NAME = DOCTOR.NAME AND APPOINTMENT_INFO.SPECIALIZATION=
DOCTOR.SPECIALIZATION AND DOCTOR.SPECIALIZATION='GS' AND DOCTOR.FEE>1500;
```

4.2 Analysis and Explanation

(a) And (b) are simple queries. (c) shows the cartesian product of cartesian product of Patient table and Appointment table. (d) shows the natural join of Patient table and Appointment table. The difference cartesian join and natural join is that cartesian join matches every element of between table 1 with every element of table 2 whereas natural join is only consisting of the matching elements of tables. In (e), (f), (g) I had to run query to find information from cartesian product of tables.

Difficulties

Rather selecting table_name.column_name, I selected only column_name, thus didn't get expected output but later I resolved that.

4.3 Output

```
SQL> @"D:\Study\Semester 3, Winter 2023\CSE 4308 Lab  Database Managemment System I\Lab 3\Task 4.sql";

NAME
--------------------
MR. Z


NAME
--------------------
B


PATIE NAME                 ADDRESS    P_NO  D_NAME               SP APPOINTME
----- -------------------- ---------- ----- -------------------- -- ---------
P-101 A                    DHK        P-101 MR. X                CS 25-AUG-23
P-102 B                    KHL        P-102 MR. Y                GS 26-AUG-23
P-103 C                    RAJ        P-103 MR. Z                IM 26-AUG-23
P-104 D                    BAR        P-104 MR. Q                PQ 27-AUG-23


PATIE NAME                 ADDRESS    P_NO  D_NAME               SP APPOINTME
----- -------------------- ---------- ----- -------------------- -- ---------
P-101 A                    DHK        P-101 MR. X                CS 25-AUG-23
P-102 B                    KHL        P-102 MR. Y                GS 26-AUG-23
P-103 C                    RAJ        P-103 MR. Z                IM 26-AUG-23
P-104 D                    BAR        P-104 MR. Q                PQ 27-AUG-23
```

```
NAME                 ADDRESS
-------------------- ----------
B                    KHL
C                    RAJ


NAME                 SP        FEE
-------------------- -- ----------
MR. X                CS       2000


NAME                 PATIE ADDRESS
-------------------- ----- ----------
B                    P-102 KHL
```

Task 5

5. Write following DML statements:

(a) Update the NAME and ADDRESS of a tuple from ’A’ and ’DHK ’ to ’K’ and ’RAJ’ accordingly.

(b) Update the NAME of table DOCTOR from ’MR. Y’ as ’Ms. Y’.

(c) Delete Patient with PATIENT_NO P-101.

(d) Delete all the information without deleting the table structure.

5.1 Solution

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

5.2 Analysis and Explanation

In this task I had to update tables by performing query. I also Deleted all the information of the Patient table without deleting the table structure.

5.3 Difficulties

I was unable to update or delete tables due to the referencing of foreign key of another table. Then by discussing with the teacher I learnt that to update or delete such table, first I have to delete the query from the child table and then from the parent table. So, at first I removed the constraints of appointment table and updated the table, then updating the parent table I again assigned the constraints.

5.4 Output

```
SQL> @"D:\Study\Semester 3, Winter 2023\CSE 4308 Lab  Database Managemment System I\Lab 3\Task 5.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.
```

Conclusion

As shown in the report, I have solved and tested the solutions for all five tasks given in the lab. All the commands used were written in visual studio code which was then saved with .sql extension. The .sql file was then run through the SQL command line to execute all the commands.