DBMS Assignment 2 - Lab 3 Report

Namisa Najah Raisa
Student ID: 210042112

August 30, 2023

1 Introduction

This is a report about the tasks in the 3rd lab.

2 Tasks

2.1 Task Overview

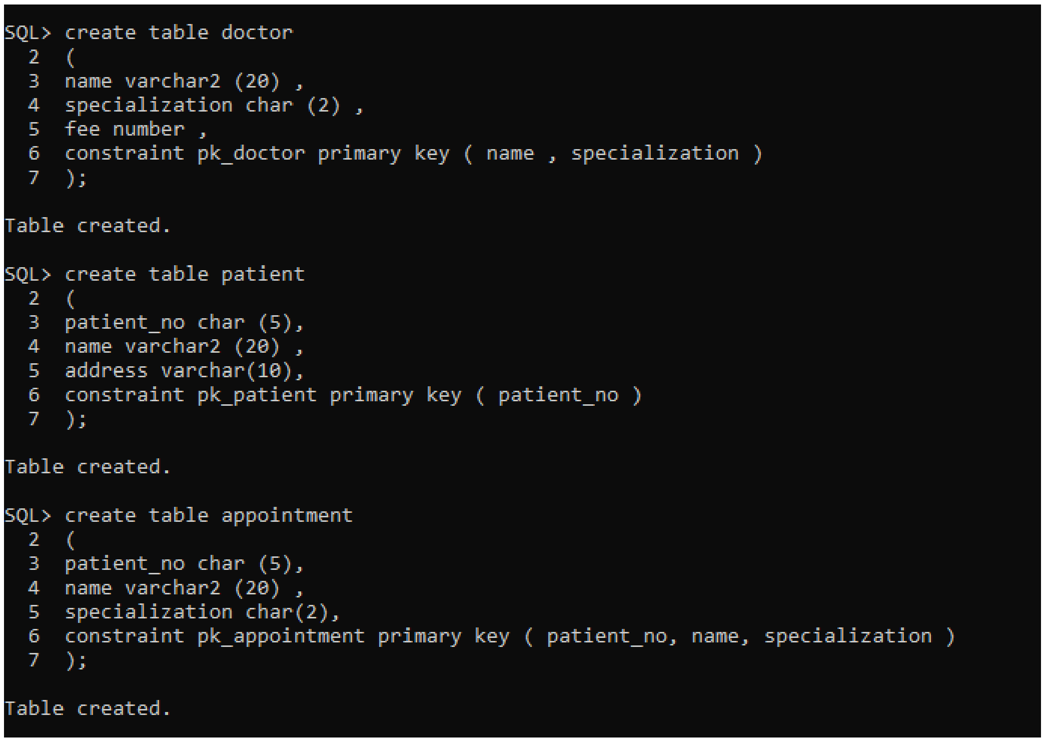
The lab consisted of five tasks:

1. Creating three tables called DOCTOR,PATIENT,APPOINTMENT .These tables had constraints of their own and certain data types.
2. Performing alteration operations, such as adding new attributes, modifying constraints, rename attributes, renaming tables and adding new foreign key constraints.
3. Inserting at least 3 records in each of the tables given as example .
4. Writing SQL query statements. There were seven queries in this task.
5. DML statements. There were 4 statements in this task. At the end we had to delete the table without deleting the table structures.

2.2 Statements

2.2.1 Creating Tables

```sql
--a/creating doctor table
create table doctor
(
name varchar2 (20) ,
specialization char (2) ,
fee number ,
constraint pk_doctor primary key ( name , specialization )
);
--b/patient table
create table patient
(
patient_no char (5),
name varchar2 (20) ,
address varchar(10),
constraint pk_patient primary key ( patient_no )
);
--c/appointment table
create table appointment
(
patient_no char (5),
name varchar2 (20) ,
specialization char(2),
constraint pk_appointment primary key ( patient_no, name,
    specialization )
);
```

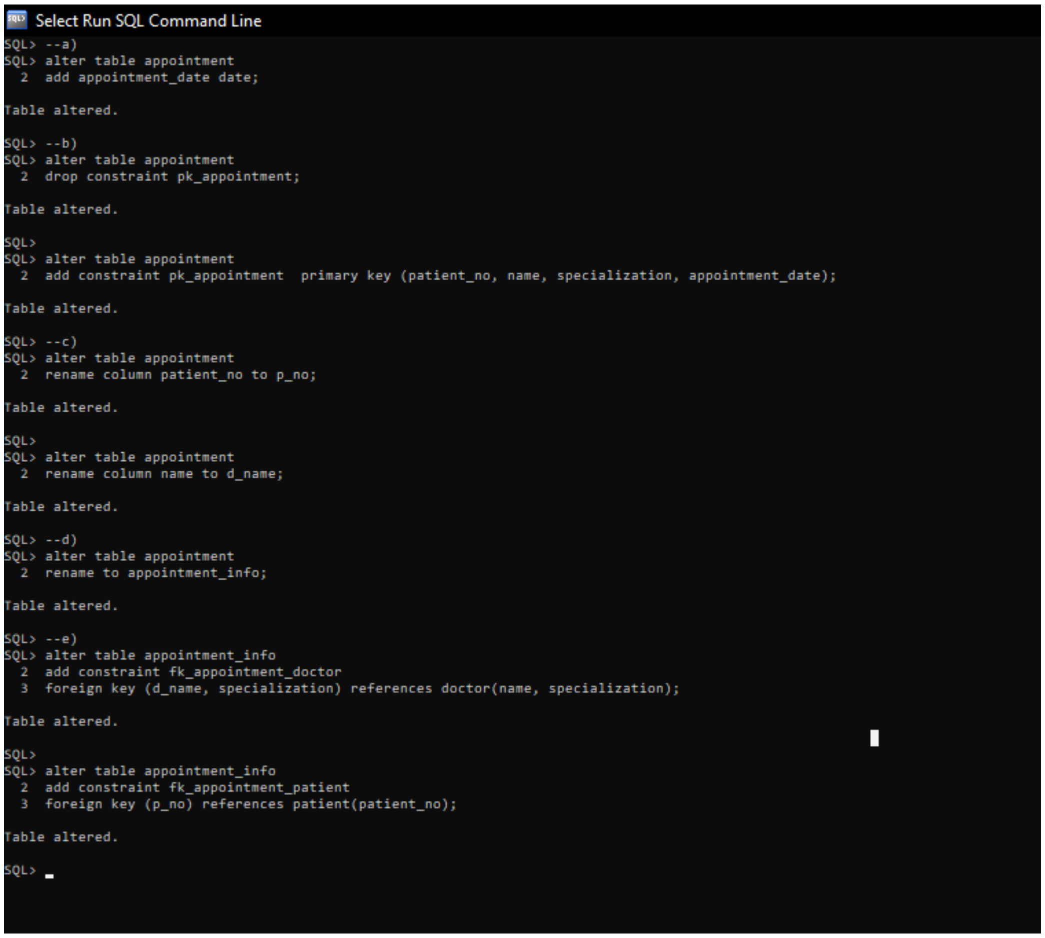


```
SQL> create table doctor
  2  (
  3  name varchar2 (20) ,
  4  specialization char (2) ,
  5  fee number ,
  6  constraint pk_doctor primary key ( name , specialization )
  7  );

Table created.

SQL> create table patient
  2  (
  3  patient_no char (5),
  4  name varchar2 (20) ,
  5  address varchar(10),
  6  constraint pk_patient primary key ( patient_no )
  7  );

Table created.

SQL> create table appointment
  2  (
  3  patient_no char (5),
  4  name varchar2 (20) ,
  5  specialization char(2),
  6  constraint pk_appointment primary key ( patient_no, name, specialization )
  7  );

Table created.
```

2.2.2 SQL statements to perform alteration operations

```
-- a) Add a new attribute APPOINTMENT_DATE (DATE type) in
   APPOINTMENT table
alter table appointment
add appointment_date date;
--b) Modify the PRIMARY KEY of APPOINTMENT table and add
   APPOINTMENT_DATE too with the previous ones.
alter table appointment
drop constraint pk_appointment;

alter table appointment
add constraint pk_appointment  primary key (patient_no, name,
    specialization, appointment_date);
--c) Rename the attribute PATIENT_NO, NAME from APPOINTMENT
   table to P_NO and D_NAME respectively.
alter table appointment
rename column patient_no to p_no;

alter table appointment
rename column name to d_name;
```

```sql
--d) Rename the table APPOINTMENT to APPOINTMENT_INFO.
alter table appointment
rename 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.
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);
```

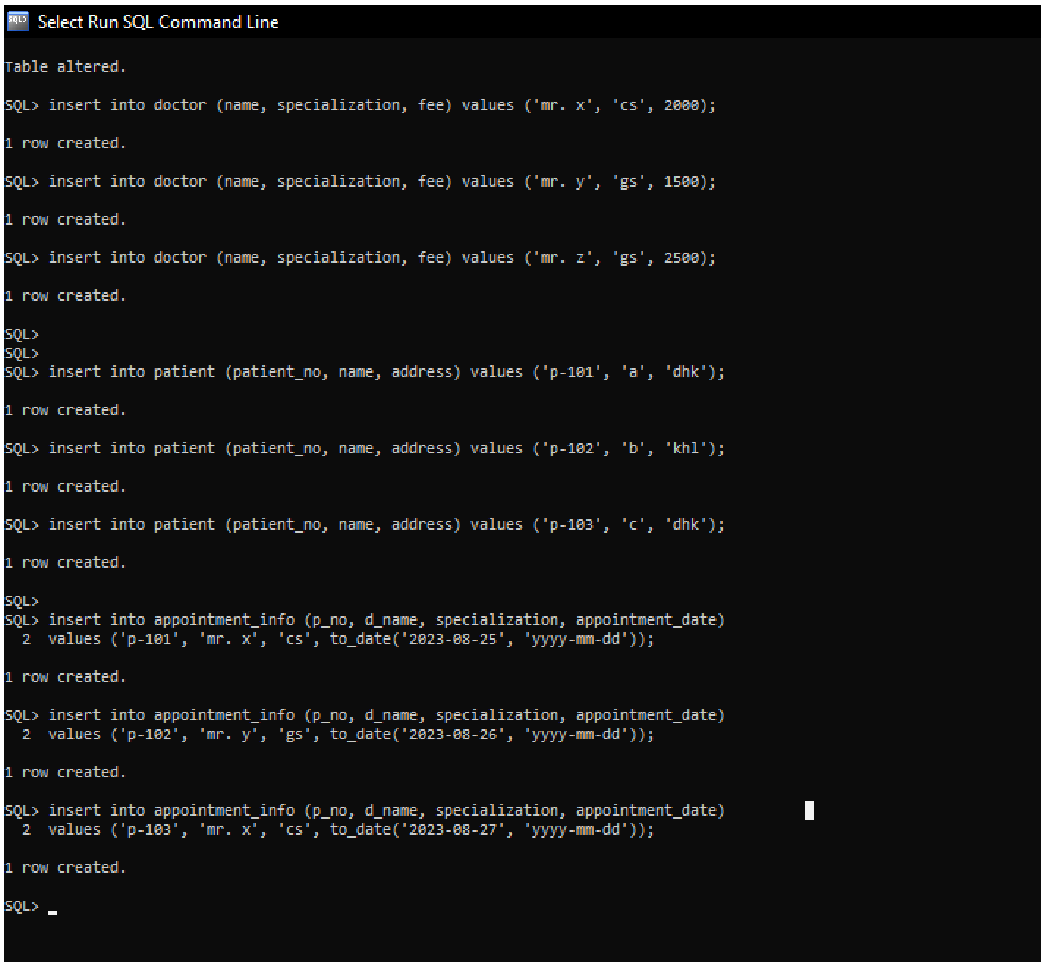


Select Run SQL Command Line

```
SQL> --a)
SQL> alter table appointment
  2  add appointment_date date;

Table altered.

SQL> --b)
SQL> alter table appointment
  2  drop constraint pk_appointment;

Table altered.

SQL>
SQL> alter table appointment
  2  add constraint pk_appointment  primary key (patient_no, name, specialization, appointment_date);

Table altered.

SQL> --c)
SQL> alter table appointment
  2  rename column patient_no to p_no;

Table altered.

SQL>
SQL> alter table appointment
  2  rename column name to d_name;

Table altered.

SQL> --d)
SQL> alter table appointment
  2  rename to appointment_info;

Table altered.

SQL> --e)
SQL> alter table appointment_info
  2  add constraint fk_appointment_doctor
  3  foreign key (d_name, specialization) references doctor(name, specialization);

Table altered.

SQL>
SQL> alter table appointment_info
  2  add constraint fk_appointment_patient
  3  foreign key (p_no) references patient(patient_no);

Table altered.

SQL> _
```

2.2.3 Inserting records in the Tables

```sql
insert into doctor (name, specialization, fee) values ('mr. x', 'cs', 2000);
insert into doctor (name, specialization, fee) values ('mr. y', 'gs', 1500);
insert into doctor (name, specialization, fee) values ('mr. z', 'gs', 2500);


insert into patient (patient_no, name, address) values ('p-101', 'a', 'dhk');
insert into patient (patient_no, name, address) values ('p-102', 'b', 'khl');
insert into patient (patient_no, name, address) values ('p-103', 'c', 'dhk');

insert into appointment_info (p_no, d_name, specialization, appointment_date)
values ('p-101', 'mr. x', 'cs', to_date('2023-08-25', 'yyyy-mm-dd'));
insert into appointment_info (p_no, d_name, specialization, appointment_date)
values ('p-102', 'mr. y', 'gs', to_date('2023-08-26', 'yyyy-mm-dd'));
insert into appointment_info (p_no, d_name, specialization, appointment_date)
values ('p-103', 'mr. x', 'cs', to_date('2023-08-27', 'yyyy-mm-dd'));
```



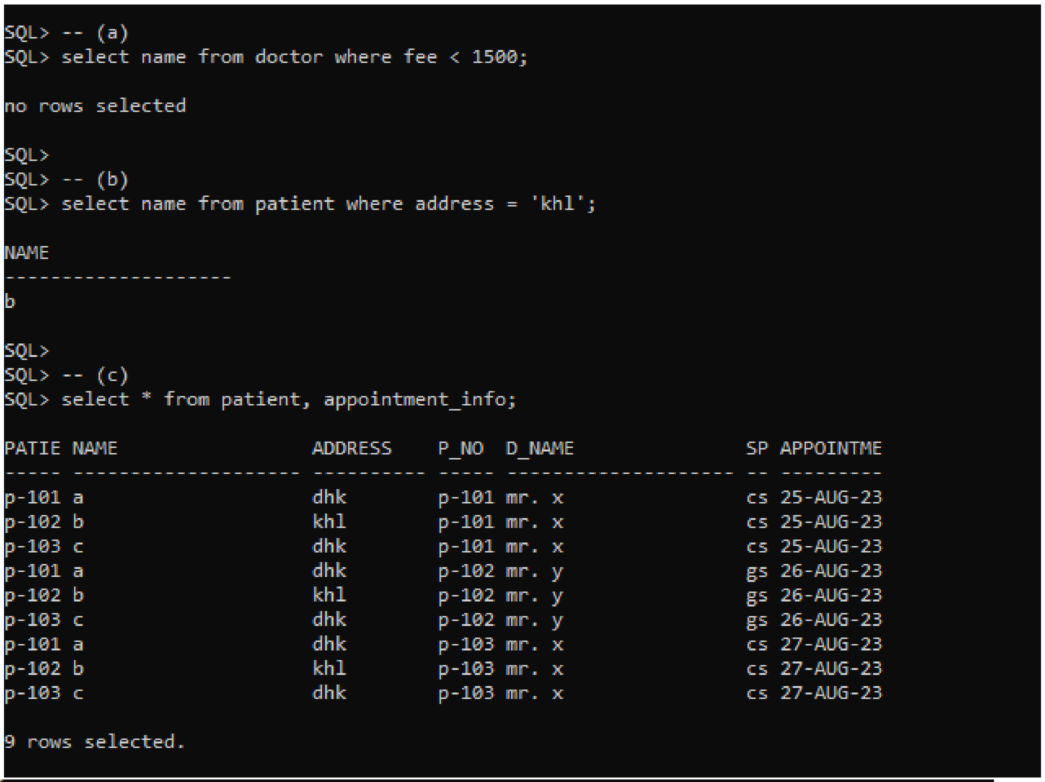
Select Run SQL Command Line

```
Table altered.

SQL> insert into doctor (name, specialization, fee) values ('mr. x', 'cs', 2000);

1 row created.

SQL> insert into doctor (name, specialization, fee) values ('mr. y', 'gs', 1500);

1 row created.

SQL> insert into doctor (name, specialization, fee) values ('mr. z', 'gs', 2500);

1 row created.

SQL>
SQL>
SQL> insert into patient (patient_no, name, address) values ('p-101', 'a', 'dhk');

1 row created.

SQL> insert into patient (patient_no, name, address) values ('p-102', 'b', 'khl');

1 row created.

SQL> insert into patient (patient_no, name, address) values ('p-103', 'c', 'dhk');

1 row created.

SQL>
SQL> insert into appointment_info (p_no, d_name, specialization, appointment_date)
  2  values ('p-101', 'mr. x', 'cs', to_date('2023-08-25', 'yyyy-mm-dd'));

1 row created.

SQL> insert into appointment_info (p_no, d_name, specialization, appointment_date)
  2  values ('p-102', 'mr. y', 'gs', to_date('2023-08-26', 'yyyy-mm-dd'));

1 row created.

SQL> insert into appointment_info (p_no, d_name, specialization, appointment_date)
  2  values ('p-103', 'mr. x', 'cs', to_date('2023-08-27', 'yyyy-mm-dd'));

1 row created.

SQL> _
```

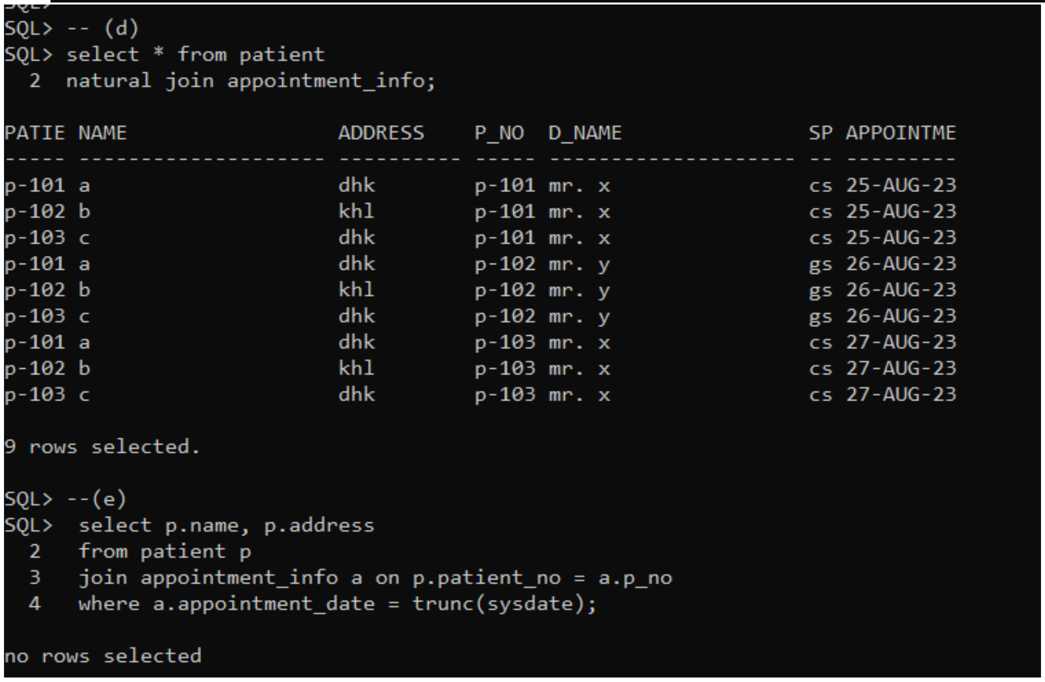
2.2.4 SQL statements to answer queries:

```sql
--a) Find all the Doctors names whose fees are less than 1500.
select name from doctor where fee < 1500;

-- (b) Find all the Patients names who live in   KHL city
select name from patient where address = 'khl';

-- (c) Show the result of Cartesian Product between PATIENT and APPOINTMENT_INFO table.
select * from patient, appointment_info;

```

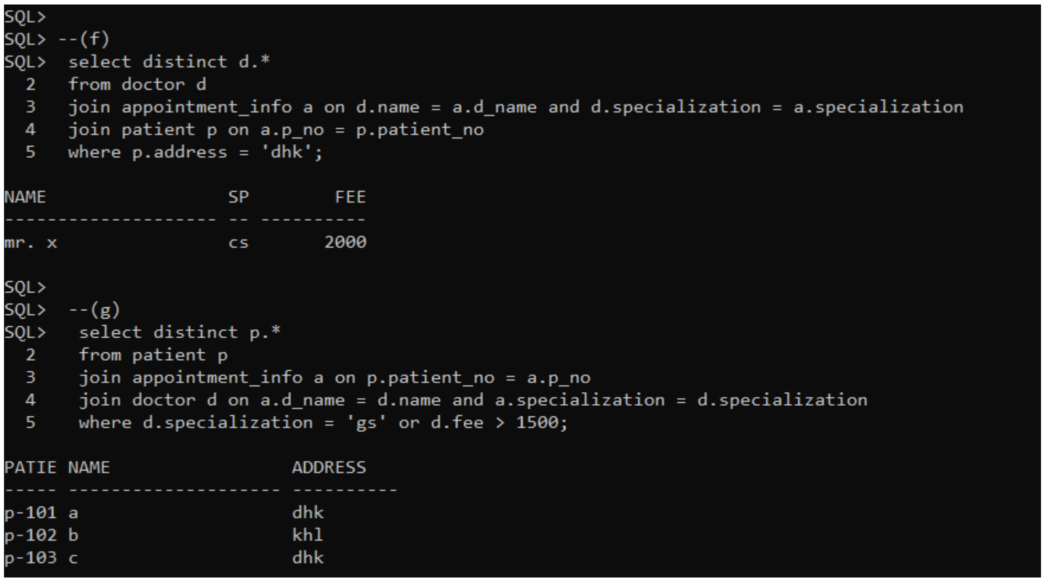
```sql
-- (d) Show the result of Natural Join between PATIENT and APPOINTMENT_INFO table.
select * from patient
natural join appointment_info;
--(e) Find all the Patients names and their address who have an appointment today.
 select p.name, p.address
 from patient p
 join appointment_info a on p.patient_no = a.p_no
 where a.appointment_date = trunc(sysdate);

--(f) Find all the Doctor-related information who have patients from   DHK   .
 select distinct d.*
 from doctor d
 join appointment_info a on d.name = a.d_name and d.specialization = a.specialization
 join patient p on a.p_no = p.patient_no
 where p.address = 'dhk';

 --(g) Find all Patient-related information who has an appointment with a doctor of   GS   spe cialization or a doctor whose fee is greater than 1500.
  select distinct p.*
  from patient p
  join appointment_info a on p.patient_no = a.p_no
  join doctor d on a.d_name = d.name and a.specialization = d.specialization
  where d.specialization = 'gs' or d.fee > 1500;
```



```
SQL> -- (a)
SQL> select name from doctor where fee < 1500;

no rows selected

SQL>
SQL> -- (b)
SQL> select name from patient where address = 'khl';

NAME
--------------------
b

SQL>
SQL> -- (c)
SQL> select * from patient, appointment_info;

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-101 mr. x                cs 25-AUG-23
p-103 c                    dhk        p-101 mr. x                cs 25-AUG-23
p-101 a                    dhk        p-102 mr. y                gs 26-AUG-23
p-102 b                    khl        p-102 mr. y                gs 26-AUG-23
p-103 c                    dhk        p-102 mr. y                gs 26-AUG-23
p-101 a                    dhk        p-103 mr. x                cs 27-AUG-23
p-102 b                    khl        p-103 mr. x                cs 27-AUG-23
p-103 c                    dhk        p-103 mr. x                cs 27-AUG-23

9 rows selected.
```



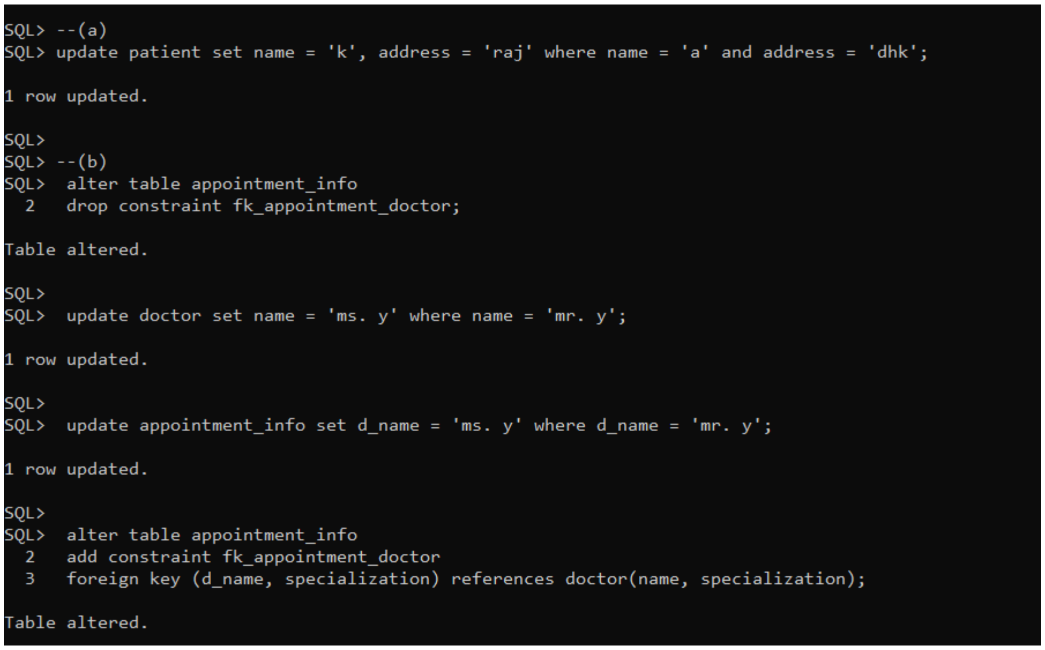
```
SQL> -- (d)
SQL> select * from patient
  2  natural join appointment_info;

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-101 mr. x                cs 25-AUG-23
p-103 c                    dhk        p-101 mr. x                cs 25-AUG-23
p-101 a                    dhk        p-102 mr. y                gs 26-AUG-23
p-102 b                    khl        p-102 mr. y                gs 26-AUG-23
p-103 c                    dhk        p-102 mr. y                gs 26-AUG-23
p-101 a                    dhk        p-103 mr. x                cs 27-AUG-23
p-102 b                    khl        p-103 mr. x                cs 27-AUG-23
p-103 c                    dhk        p-103 mr. x                cs 27-AUG-23

9 rows selected.

SQL> --(e)
SQL>  select p.name, p.address
  2   from patient p
  3   join appointment_info a on p.patient_no = a.p_no
  4   where a.appointment_date = trunc(sysdate);

no rows selected
```


```
SQL>
SQL> --(f)
SQL>  select distinct d.*
  2   from doctor d
  3   join appointment_info a on d.name = a.d_name and d.specialization = a.specialization
  4   join patient p on a.p_no = p.patient_no
  5   where p.address = 'dhk';

NAME                 SP        FEE
-------------------- -- ----------
mr. x                cs       2000

SQL>
SQL>  --(g)
SQL>   select distinct p.*
  2    from patient p
  3    join appointment_info a on p.patient_no = a.p_no
  4    join doctor d on a.d_name = d.name and a.specialization = d.specialization
  5    where d.specialization = 'gs' or d.fee > 1500;

PATIE NAME                 ADDRESS
----- -------------------- ----------
p-101 a                    dhk
p-102 b                    khl
p-103 c                    dhk
```

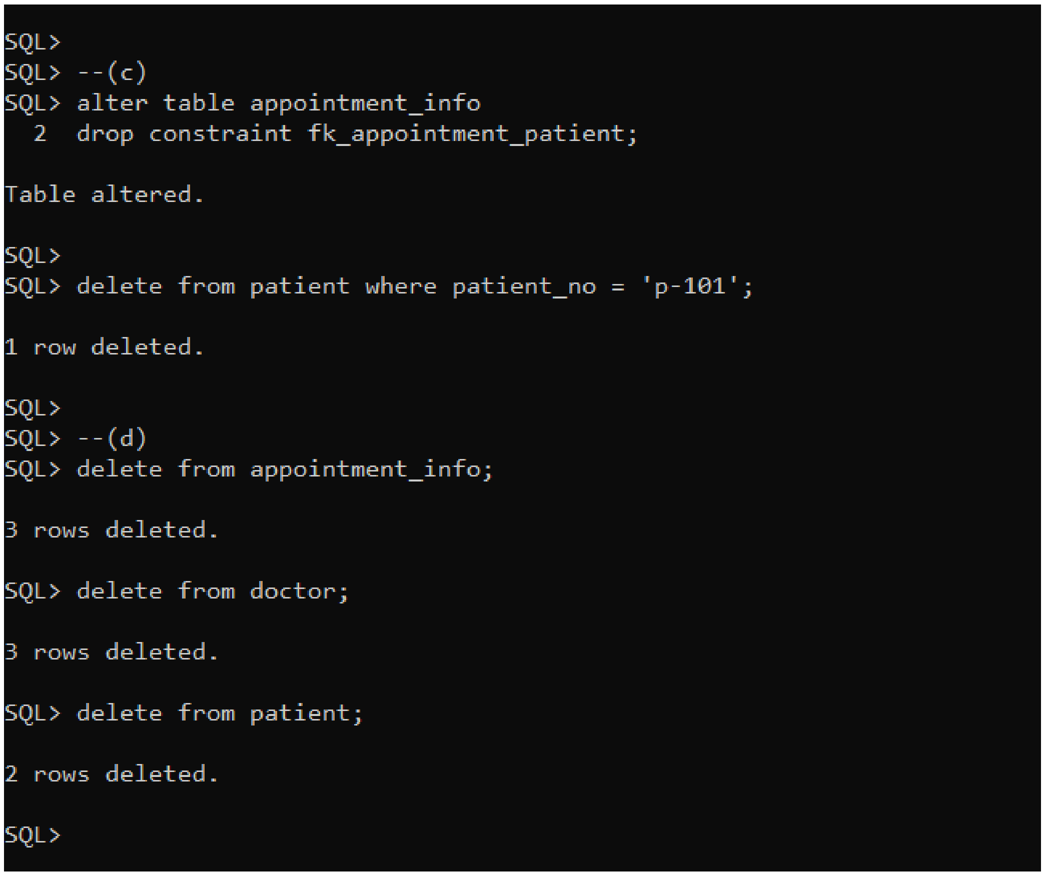
2.2.5 DML statements

```sql
--(a) Update the NAME and ADDRESS of a tuple from  A  and
      DHK     to   K   and   RAJ   accordingly.
update patient set name = 'k', address = 'raj' where name = '
   a' and address = 'dhk';

--(b) Update the NAME of table DOCTOR from  MR  .  Y   as
      Ms  .  Y  .
 alter table appointment_info
 drop constraint fk_appointment_doctor;

 update doctor set name = 'ms. y' where name = 'mr. y';

 update appointment_info set d_name = 'ms. y' where d_name =
   'mr. y';

 alter table appointment_info
 add constraint fk_appointment_doctor
 foreign key (d_name, specialization) references doctor(name,
    specialization);
--(c) Delete Patient with PATIENT_NO P-101.
alter table appointment_info
drop constraint fk_appointment_patient;

delete from patient where patient_no = 'p-101';
```

```sql

--(d) Delete all the information without deleting the table
    structure.
delete from appointment_info;
delete from doctor;
delete from patient;
```



```
SQL> --(a)
SQL> update patient set name = 'k', address = 'raj' where name = 'a' and address = 'dhk';

1 row updated.

SQL>
SQL> --(b)
SQL>  alter table appointment_info
  2   drop constraint fk_appointment_doctor;

Table altered.

SQL>
SQL>  update doctor set name = 'ms. y' where name = 'mr. y';

1 row updated.

SQL>
SQL>  update appointment_info set d_name = 'ms. y' where d_name = 'mr. y';

1 row updated.

SQL>
SQL>  alter table appointment_info
  2   add constraint fk_appointment_doctor
  3   foreign key (d_name, specialization) references doctor(name, specialization);

Table altered.
```



```
SQL>
SQL> --(c)
SQL> alter table appointment_info
  2  drop constraint fk_appointment_patient;

Table altered.

SQL>
SQL> delete from patient where patient_no = 'p-101';

1 row deleted.

SQL>
SQL> --(d)
SQL> delete from appointment_info;

3 rows deleted.

SQL> delete from doctor;

3 rows deleted.

SQL> delete from patient;

2 rows deleted.

SQL>
```

3 Challenges

I faced a lot of challenges while doing these tasks.

1. In the last 3 queries of task 4 I wasn't sure of how to join tables and their information to retrieve data that included things from multiple tables. After encountering a lot of errors, I could finally understand how to solve them.

2. I also miswrote constraint names and then faced errors because of them.

3. While inserting records in the appointment_info table omitting the date attribute gave me an error.

4. while updating,deleting or altering in the independent tables I had to drop constraints in the dependent tables which was unknown to me at first. It led to a lot of confusing errors.