LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC5051NA Database System

Assessment Weightage & Type

50% Individual Coursework

Year and Semester

2018-19 Autumn

Student Name: Basant Tamang

London Met ID: 18029742

College ID: NP01CP4A180053

Assignment Due Date: 30/12/2019

Assignment Submission Date: 30/12/2019

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

Table of Figure

Table of Table

1. Introduction

The Annapurna Neurological Institute & Allied Sciences Pvt. Ltd. (ANIAS), a multi-specialty hospital was established in the year 2009. The objective of the hospital has been to deliver & disseminate accessible high-quality medical service to the people of Nepal and which can be duplicated through standardization of the process. To achieve this service, the hospital has a motto of "The vision of ANIAS is to make the state of art premises which will have facilities to provide best services to the patients, to care for patients, to doctors, to other health service providers and to other professionals involved in the health service sector." It has been groomed from the start with the ambition of incorporating treatment, academics, research and public health activities and within a short period of time, we have achieved a functional wing in all these four areas.

The hospital has National Collaboration with various Medical institutes affiliated with Tribhuwan University and Pokhara University. It also runs various kinds of training and internship program for national and international students including CCNT (Critical Care Nursing Training), Elective internship for MBBS students, and elective postings for nursing students. International Collaboration Hiroshima / ANT / Kagoshima / ONISHI Neurosurgical Institute.

1.1 Current Business Activities and Operation

- Our hospital provides services for patients every day.
- The hospital conducts surgical case and most of them are neurosurgery and non-surgical inpatient cases in a month.
- Appointment is conducted in a ward and there are different treatments available in the hospital.
- Treatment have their own charges and staff get their own charge for conducting appointment.
- People are assigned address to know where they came from.
- Patient need to take appointment charge for treatment.

1.2 Business Rule.

- Both patients and staff can have many appointments as they are person.
- Person can have many addresses.
- Patient is divided into two categories. They are New and Regular.
- Staff is divided into two categories. They are Certified and Uncertified.
- An appointment is conducted in one particular ward.
- On one appointment only one treatment can be done.
- One staff is assigned to one appointment.
- Staff can be patient as well and their treatment cost is free if they are certified.

1.3 Identification of Entities and attributes

The entities and attributes for this patient record system are listed below:

	Person
PK	Person Id First Name Last Name Age Gender Contact Id Mobile Number Email

Table 1 Attributes for Person Entity

	Address
PK	Address Id Country State City Street Street Number Fax No Telephone Number

Table 2 Attributes for Address Entity

	Patient
PK	Patient Id
	Patient Type

Table 3 Attributes for Patient Entity

	Staff
PK	Staff Id
	Staff Type
	Staff Description
	Staff Charge

Table 4 Attributes for Staff Entity

	Appointment
PK	Appointment Id
	Appointment Date
	Appointment Time
	Appointment Charge
	Assigned By

Table 5 Attributes for Appointment Entity

	Treatment
PK	Treatment Id Treatment Treatment Charge

Table 6 Attributes for Treatment Entity

	Ward
PK	Ward No Ward Name

Table 7 Attributes for Ward Entity

1.4 Initial Entity Diagram

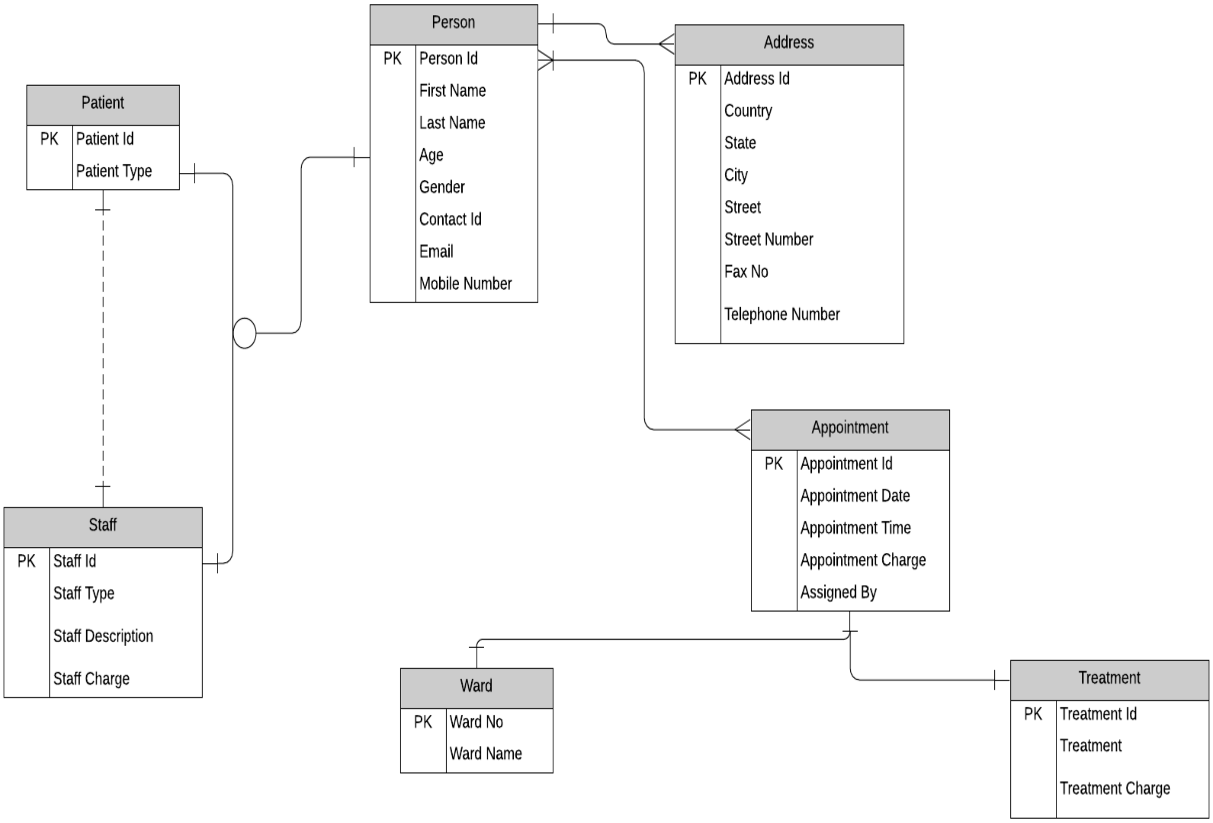


Figure 1 Initial Entity Relation Diagram

Assumptions

- A person can be both patient and staff
- Assigned By is staff who will check the patients.
- An appointment id will have same date but different time.
- Person can have many mobile numbers.
- Treatment charge is different according to treatment.
- Staff cannot do treatment by itself.

2. Normalization

A database design method that manages tables which reduces redundancy and dependency of data is known as normalization. It is used to remove problems of update, intersection and deletion anomalies in database. Normalization make easy to update, insert and delete a data without affection another data.

The steps of normalization are listed below:

- Un – normalized form (UNF)
- First normalized form (1NF)
- Second normalized form (2NF)
- Third normalized form (3NF)

2.1 Un- normalized form (UNF)

In this step, all the attributes are listed without repeating them. The repeating group is placed inside { } after identifying repeating and identify the primary key, represent them with underline.

Patient Record(<u>Person Id</u>, First Name, Last Name, Age, Gender {Contact Id, Mobile Number, Email} Patient Id, Patient Type, Staff Id, Staff Type, Staff Description, Staff Charge {Address Id, Address Type, Country, State, City, Street, Street Number, Fax No, Telephone Number} {Appointment Id, Appointment Date, Appointment Time, Assigned By, Appointment Charge, Treatment, Treatment Id, Treatment Charge, Ward No, Ward Name}).

Here, Person Id is primary key and all the repeating groups is placed inside { }. The relation is named as Patient Record.

2.2 First Normal Form (1NF)

In this step, repeating groups are removed and separate new relation is made. The primary key for new relation is identified and add foreign key to make connection between two relation. Foreign key is denoted by *.

Person - 1 (<u>Person Id</u>, First Name, Last Name, Age, Gender, Patient Id, Patient Type, Staff Id, Staff Type, Staff Description, Staff Charge)

Contact -1 (Person Id*, <u>Contact Id</u>, Email, Mobile Number)

This contact is new relation after removing repeating group and Contact Id is primary key and Person Id is added as foreign key.

Address - 1 (<u>Address Id</u>, Person Id*, Country, State, City, Street, Street Number, Fax No, Telephone Number)

This Address is another new relation. The primary key is Address Id and Foreign key is Person Id for this relation.

Appointment -1 (<u>Appointment Id</u>, Person Id*, Assigned By, Appointment Date, Appointment Time, Appointment Charge, Treatment, Treatment Id, Treatment Charge, Ward No, Ward Name)

This Appointment is also new relation after removing repeating group. The primary key is Appointment Id and foreign key is Person Id for this relation.

2.3 Second Normal Form (2NF)

In this step, partial dependency is removed. Partial dependency means non-key depends on only a part of primary key. Identify the composite key and check every non-key is depend on part of composite key or all of composite key.

Person Id → First Name, Last Name, Age, Gender, Patient Id, Patient Type, Staff Id, Staff Type, Staff Description, Staff Charge

It have only one primary key so it is already in 2NF.

For Contact -1

Person Id →

Contact Id → Email, Mobile Number

Contact Id, Person Id → Null

The composite for this relation is Person Id and Contact Id. The attributes do not depend upon Person Id and Contact Id, Person Id. All the attributes depend up on Contact Id. So, the partial dependency is removed.

For Address -1

Person Id →

Address Id → Country, State, City, Street, Street Number, Fax No, Telephone Number

Person Id, Address Id →

The composite key for this relation is Person Id and Address Id. The attributes do not depend upon Person Id, Address Id and Person Id. All the attributes depend up on Address Id. Partial dependency is removed.

For Appointment -1

Person Id →

Appointment Id → Appointment Date, Appointment Time, Appointment Charge, Treatment, Ward No, Ward Name, Treatment Id, Treatment, Treatment Charge

Person Id, Appointment Id → Null

The composite key for this relation is Person Id and Appointment Id. The attributes do not depend on Person Id and Person Id, Appointment Id. All the attributes depend on Appointment Id. Partial dependency is removed

Applying 2NF

All the partial dependency is removed and new relation are generated.

Person – 2 (Person Id, First Name, Last Name, Age, Gender, Patient Id, Patient Type, Staff Id, Staff Type, Staff Description, Staff Charge)

Contact -2 (Person Id*, Contact Id*)

Contact Detail -2 (Contact Id, Email, Mobile Number)

Address detail -2 (Address Id, Country, State, City, Street, Street Number, Fax No, Telephone Number)

Address -2 (Person Id*, Address Id*)

Appointment Detail -2 (Appointment Id, Appointment Date, Appointment Time, Appointment Charge, Assigned By, Ward No, Ward Name, Treatment Id, Treatment, Treatment Charge)

Appointment -2 (Appointment Id*, Person Id*)

2.4 Third Normal Form (3NF)

In this step, non- key should not be depending with non- key and separate relation is made and new table is formed.

For Person -2

Person Id → First Name, Last Name, Age, Gender, Person Type,

Person Id → Patient Id

Patient Id → Patient Type

Here, non-key Patient type id depend upon Patient Id. So, it is separated into new relation but Patient type does not depend on Person Id. So, Patient Id and Person Id relation is made.

Person Id → Staff Id

Staff Id → Staff Type, Staff Description, Staff Charge

Here, non-key staff type, staff description, staff charge depends upon Staff Id. So, it is separated into new relation but staff type, staff description, staff charge does not depend on Person Id. So, Staff Id and Person Id relation is made.

For Contact -2

Contact Id, Person Id →

Both are foreign key.

For Contact detail-2

Contact Id → Email, Mobile Number

Here, non-key does nor depend up on non - key

For address -2

Person Id, Address Id → Null

For Address Detail -2

Address Id → Country, State, City, Street, Street Number, Fax No, Telephone Number

Here, non – key does not depend upon non-key.

For Appointment -2

Appointment Id, Person Id → Null

Both of them are foreign key.

For Appointment -2

Appointment Id → Appointment Date, Appointment Time, Appointment Charge, Assigned By, Treatment Id, Ward No

Treatment Id → Treatment, Treatment Charge

Ward No → Ward Name

Treatment Id depend upon treatment Id and Treatment charge. Ward depends upon Ward Name. So, they are separated and new relation is made. Treatment Id and Ward No became Primary key for new relation and foreign key on previous relation because both of depend up on the Appointment Id.

Applying 3NF

After removing all transitive dependency, following relation are generated:

Person -3 (Person Id, First Name, Last Name, Age, Gender, Person Type)

Contact Detail-3 (Contact Id, Email, Mobile Number)

Contact -3 (Contact Id*, Person Id*)

Patient Detail -3 (Patient Id, Patient Type)

Patient -3 (Person Id*, Patient Id*)

Staff Detail -3 (Staff Id, Staff Type, Staff Description, Staff Charge)

Staff -3 (Person Id*, Staff Id*)

Address detail -3 (Address Id, Country, State, City, Street, Street Number, Fax No, Telephone Number)

Address -3 (Person Id*, Address Id*)

Appointment detail –3 (Appointment Id, Appointment Date, Appointment Time, Appointment Charge, Assigned By, Treatment Id*, Ward No*)

Appointment -3 (Appointment Id*, Person Id*)

Treatment -3 (Treatment Id, Treatment, Treatment Charge)

Ward-3 (Ward No, Ward Name)

3. Entity Relationship Diagram

A graphical representation of entities (tables) and their relationships between those entities is known as entity relationship diagram. It is also known as ER diagram. It is considered as blueprint of database. ER diagram helps to user to organize the data and helps to understand core of the database. There are three important elements of ER diagram. They are:

- Entities
  The objects that we want to store information about is known as Entity. It is also known as table in database.
- Attributes
  The describing characteristic of entity is known as attributes. It is also called column in database.
- Relationship
  In database, one table has foreign key that reference the primary key of another table which helps to connect two table is known as relationship. There are three type of relationship one to one, one to many and many to many.

After doing normalization, the relationship between entities and attributes are given below:

- The person can be staff or patient or both of them so it will have one to one relation.
- One person can have many contacts so it will have one to many relations.
- A person also can have one or many addresses, it will have one to many relations.
- A person can have one or many appointments, it will have one to many relations.
- An appointment can have only one treatment and referred to one ward, it will have one to one relation to ward and treatment.

The normalized ER diagram is shown below:

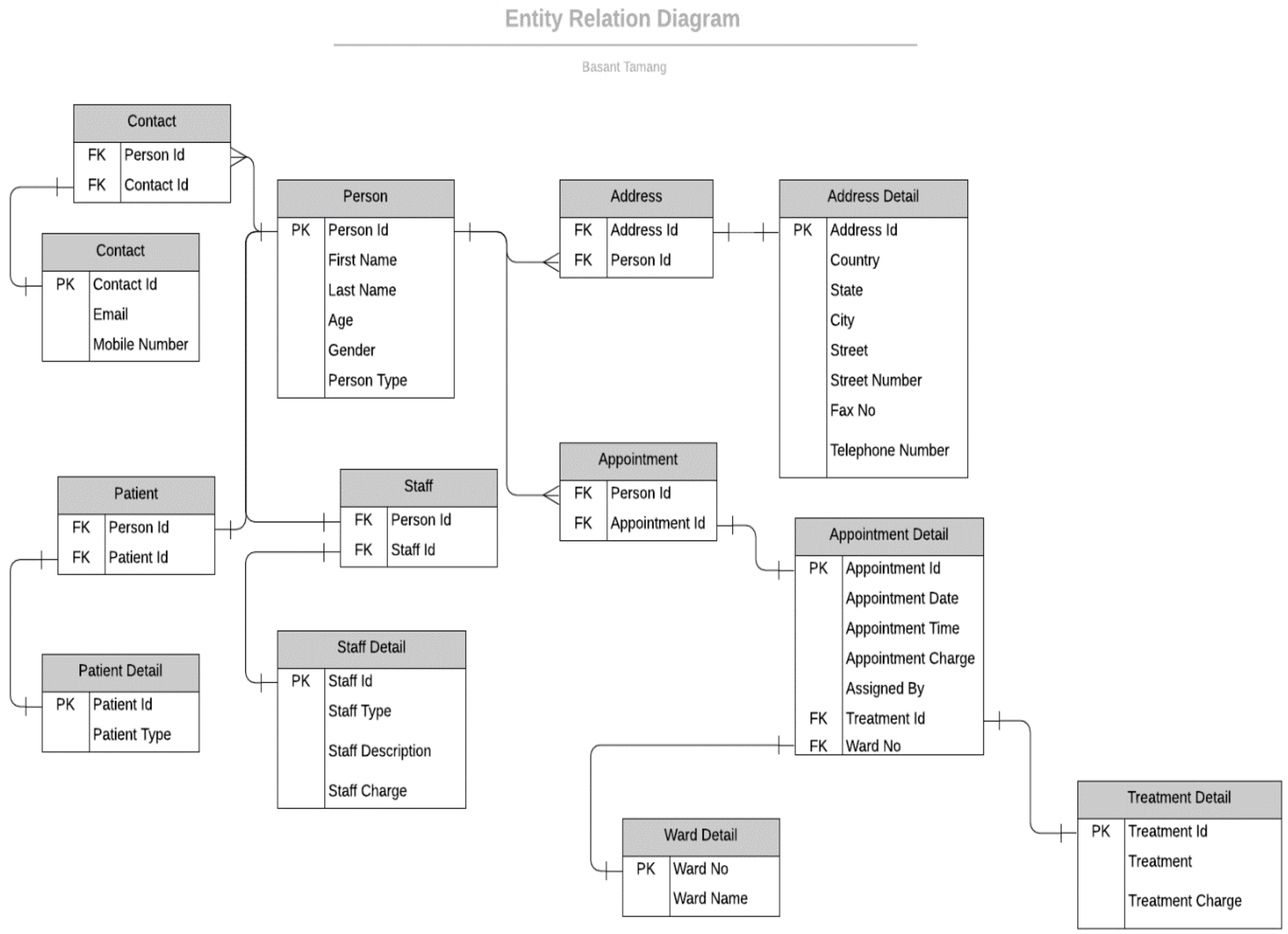


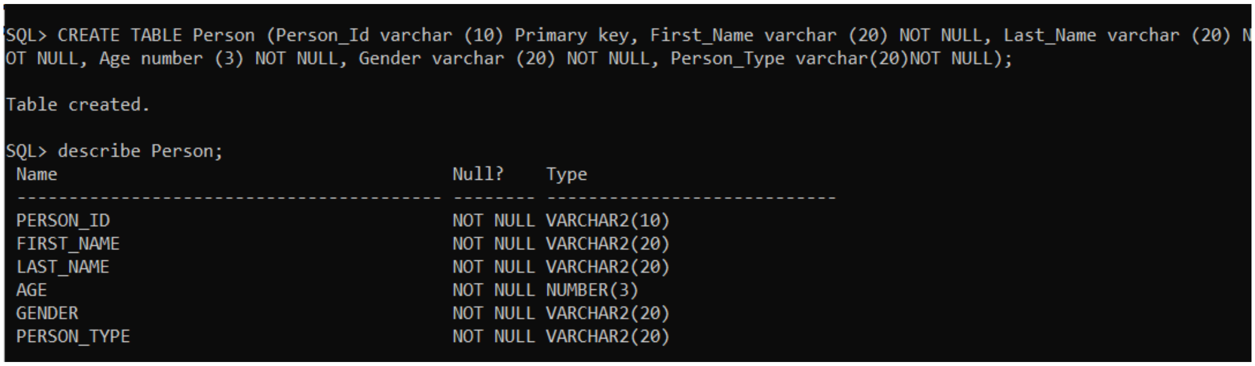
Figure 2 Normalized Entity Relation Diagram

4. Database Implementation

4.1 Table Generation

Person Table

This table contains six attributes Person Id, First Name, Last Name, Age, Gender and Person Type. The primary key is Person Id.



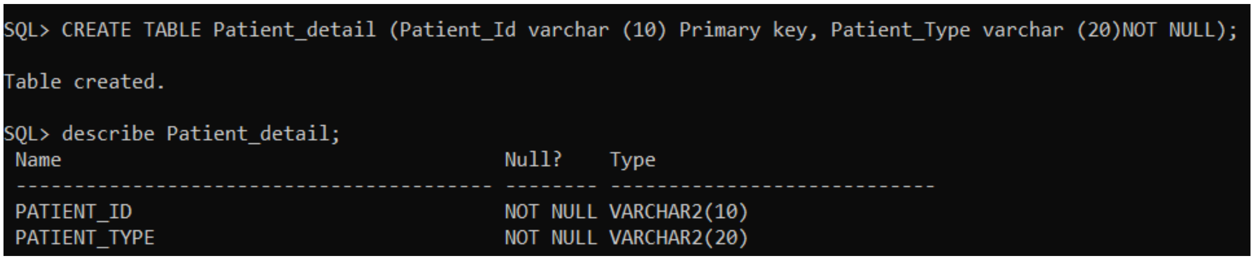
```
SQL> CREATE TABLE Person (Person_Id varchar (10) Primary key, First_Name varchar (20) NOT NULL, Last_Name varchar (20) N
OT NULL, Age number (3) NOT NULL, Gender varchar (20) NOT NULL, Person_Type varchar(20)NOT NULL);

Table created.

SQL> describe Person;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PERSON_ID                                 NOT NULL VARCHAR2(10)
 FIRST_NAME                                NOT NULL VARCHAR2(20)
 LAST_NAME                                 NOT NULL VARCHAR2(20)
 AGE                                       NOT NULL NUMBER(3)
 GENDER                                    NOT NULL VARCHAR2(20)
 PERSON_TYPE                               NOT NULL VARCHAR2(20)
```

Figure 3 Creating Person Table

CREATE TABLE Person (Person_Id varchar (10) Primary key, First_Name varchar (20) NOT NULL, Last_Name varchar (20) NOT NULL, Age number (3) NOT NULL, Gender varchar (20) NOT NULL, Person_Type varchar(20)NOT NULL).

Patient Detail

This Table contact two attributes Patient Id and Patient type. The primary key is Patient Id.



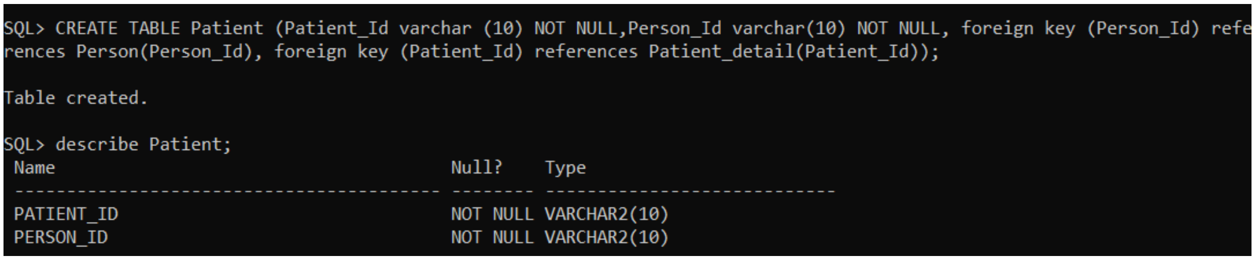
```
SQL> CREATE TABLE Patient_detail (Patient_Id varchar (10) Primary key, Patient_Type varchar (20)NOT NULL);

Table created.

SQL> describe Patient_detail;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PATIENT_ID                                NOT NULL VARCHAR2(10)
 PATIENT_TYPE                              NOT NULL VARCHAR2(20)
```

Figure 4 Creating Patient Detail Table

CREATE TABLE Patient_detail (Patient_Id varchar (10) Primary key, Patient_Type varchar (20) NOT NULL)

Patient Table

This table contains Patient Id and Person Id attributes and helps to link with person table and patient detail table. Both attributes are foreign key.



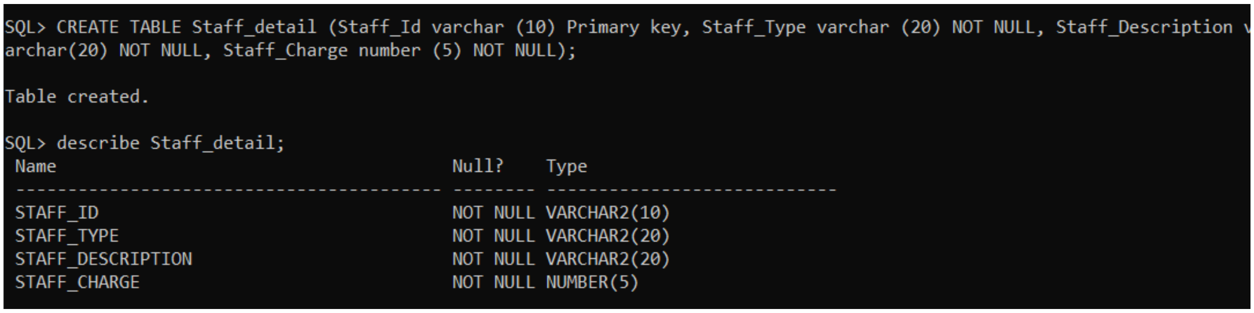
```
SQL> CREATE TABLE Patient (Patient_Id varchar (10) NOT NULL,Person_Id varchar(10) NOT NULL, foreign key (Person_Id) refe
rences Person(Person_Id), foreign key (Patient_Id) references Patient_detail(Patient_Id));

Table created.

SQL> describe Patient;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PATIENT_ID                                NOT NULL VARCHAR2(10)
 PERSON_ID                                 NOT NULL VARCHAR2(10)
```

Figure 5 Creating Patient Table

CREATE TABLE Patient (Patient_Id varchar (10) NOT NULL,Person_Id varchar(10) NOT NULL, foreign key (Person_Id) references Person(Person_Id), foreign key (Patient_Id) references Patient_detail(Patient_Id))

Staff Detail Table

This table contains Staff Id, Staff Type, Staff Description and Staff Charge. The primary key is Staff Id.



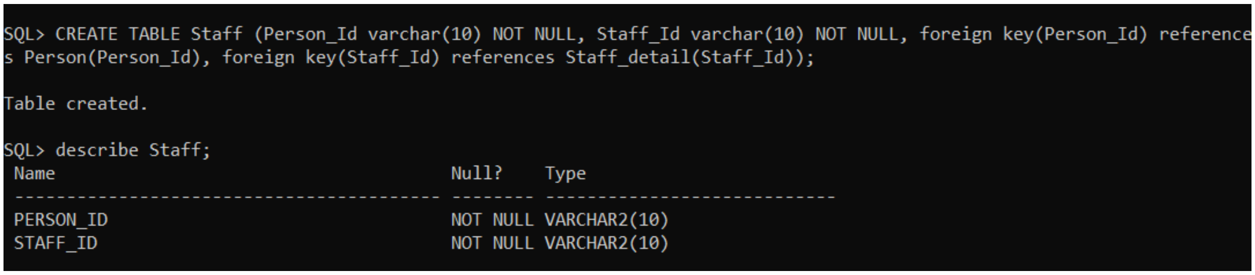
```
SQL> CREATE TABLE Staff_detail (Staff_Id varchar (10) Primary key, Staff_Type varchar (20) NOT NULL, Staff_Description v
archar(20) NOT NULL, Staff_Charge number (5) NOT NULL);

Table created.

SQL> describe Staff_detail;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STAFF_ID                                  NOT NULL VARCHAR2(10)
 STAFF_TYPE                                NOT NULL VARCHAR2(20)
 STAFF_DESCRIPTION                         NOT NULL VARCHAR2(20)
 STAFF_CHARGE                              NOT NULL NUMBER(5)
```

Figure 6 Creating Staff Detail Table

CREATE TABLE Staff_detail (Staff_Id varchar (10) Primary key, Staff_Type varchar (20) NOT NULL, Staff_Description varchar(20) NOT NULL, Staff_Charge number (5) NOT NULL)

Staff Table

This table contains two attributes Person Id, Staff Id. Both the attributes are foreign key and helps to connect staff detail and person table.



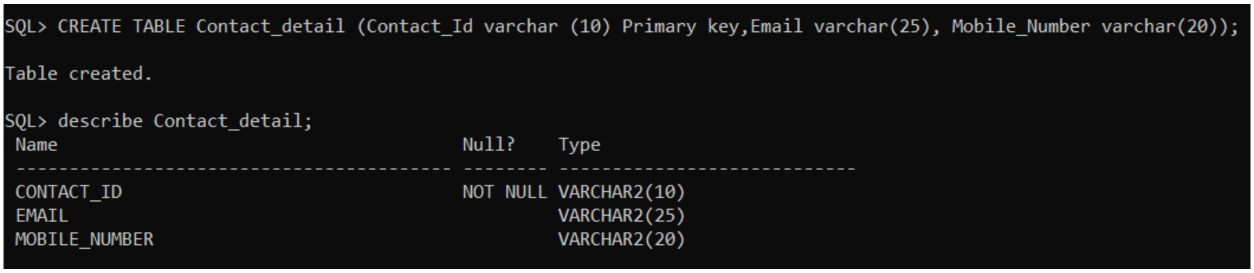
```
SQL> CREATE TABLE Staff (Person_Id varchar(10) NOT NULL, Staff_Id varchar(10) NOT NULL, foreign key(Person_Id) reference
s Person(Person_Id), foreign key(Staff_Id) references Staff_detail(Staff_Id));

Table created.

SQL> describe Staff;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PERSON_ID                                 NOT NULL VARCHAR2(10)
 STAFF_ID                                  NOT NULL VARCHAR2(10)
```

Figure 7 Creating Staff Table

CREATE TABLE Staff (Person_Id varchar(10) NOT NULL, Staff_Id varchar(10) NOT NULL, foreign key(Person_Id) references Person(Person_Id), foreign key(Staff_Id) references Staff_detail(Staff_Id))

Contact Detail Table

This table contains Contact Id, Email and Mobile Number attributes. The primary key is Contact Id.



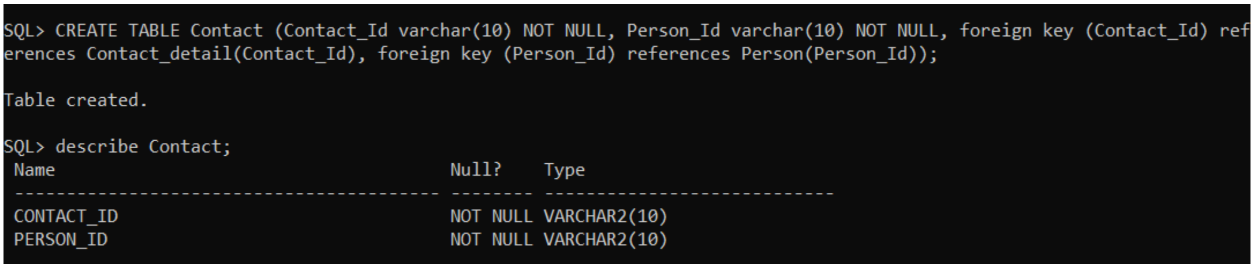
```
SQL> CREATE TABLE Contact_detail (Contact_Id varchar (10) Primary key,Email varchar(25), Mobile_Number varchar(20));

Table created.

SQL> describe Contact_detail;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CONTACT_ID                                NOT NULL VARCHAR2(10)
 EMAIL                                              VARCHAR2(25)
 MOBILE_NUMBER                                      VARCHAR2(20)
```

Figure 8 Creating Contact Detail Table

CREATE TABLE Contact_detail (Contact_Id varchar (10) Primary key,Email varchar(25), Mobile_Number varchar(20))

Contact Table

The table contains two attributes Contact Id and Person Id. Both are foreign key and helps to connect with Person and Contact Detail table.



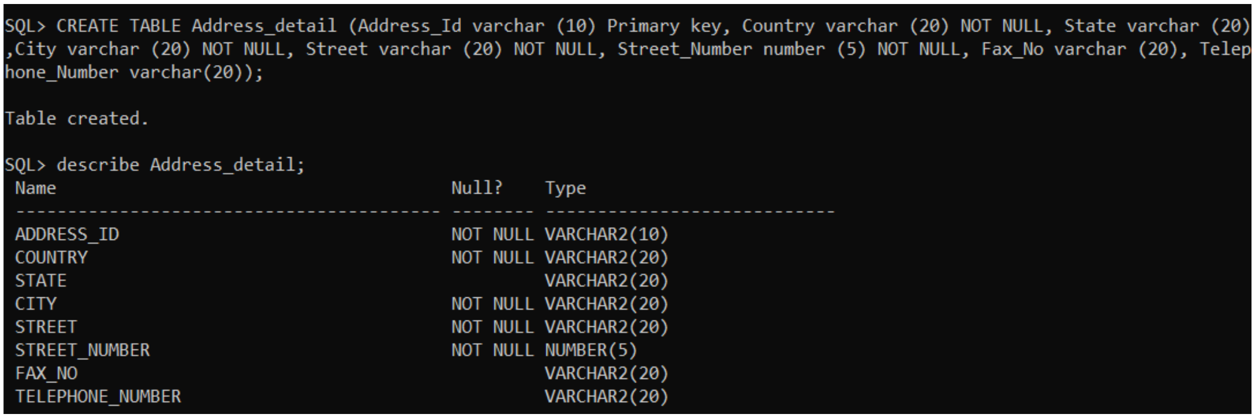
```
SQL> CREATE TABLE Contact (Contact_Id varchar(10) NOT NULL, Person_Id varchar(10) NOT NULL, foreign key (Contact_Id) ref
erences Contact_detail(Contact_Id), foreign key (Person_Id) references Person(Person_Id));

Table created.

SQL> describe Contact;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CONTACT_ID                                NOT NULL VARCHAR2(10)
 PERSON_ID                                 NOT NULL VARCHAR2(10)
```

Figure 9 Creating Contact Table

CREATE TABLE Contact (Contact_Id varchar(10) NOT NULL, Person_Id varchar(10) NOT NULL, foreign key (Contact_Id) references Contact_detail(Contact_Id), foreign key (Person_Id) references Person(Person_Id))

Address Detail Table

The table contains eight attributes Address Id, Country, State, City, Street, Street Number, Fax No and Telephone Number. The primary key is Address Id.



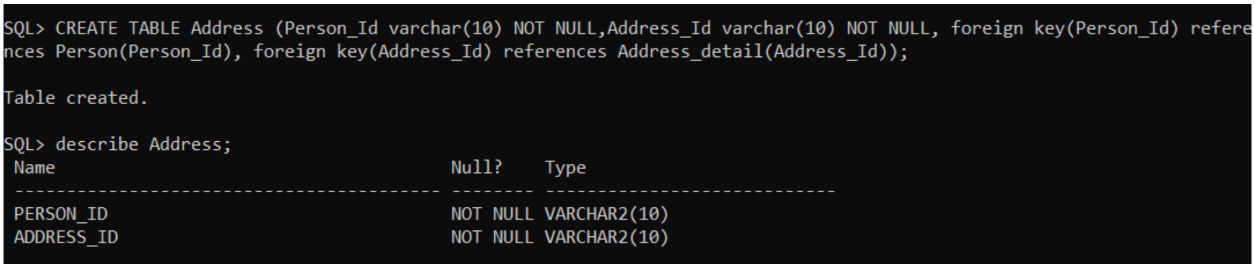
```
SQL> CREATE TABLE Address_detail (Address_Id varchar (10) Primary key, Country varchar (20) NOT NULL, State varchar (20)
,City varchar (20) NOT NULL, Street varchar (20) NOT NULL, Street_Number number (5) NOT NULL, Fax_No varchar (20), Telep
hone_Number varchar(20));

Table created.

SQL> describe Address_detail;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ADDRESS_ID                                NOT NULL VARCHAR2(10)
 COUNTRY                                   NOT NULL VARCHAR2(20)
 STATE                                              VARCHAR2(20)
 CITY                                      NOT NULL VARCHAR2(20)
 STREET                                    NOT NULL VARCHAR2(20)
 STREET_NUMBER                             NOT NULL NUMBER(5)
 FAX_NO                                             VARCHAR2(20)
 TELEPHONE_NUMBER                                   VARCHAR2(20)
```

Figure 10 Creating Address Detail Table

CREATE TABLE Address_detail (Address_Id varchar (10) Primary key, Country varchar (20) NOT NULL, State varchar (20) , City varchar (20) NOT NULL, Street varchar (20) NOT NULL, Street_Number number (5) NOT NULL, Fax_No varchar (20), Telephone_Number varchar(20))

Address Table

The table contains Person Id and Address Id attributes. Both are foreign keys which connects Person and Address Table.



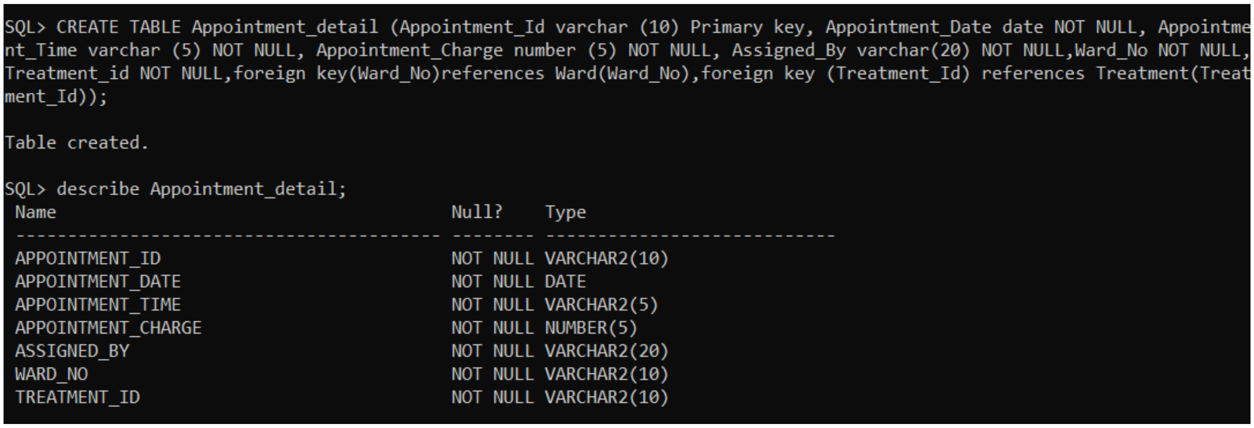
```
SQL> CREATE TABLE Address (Person_Id varchar(10) NOT NULL,Address_Id varchar(10) NOT NULL, foreign key(Person_Id) refere
nces Person(Person_Id), foreign key(Address_Id) references Address_detail(Address_Id));

Table created.

SQL> describe Address;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PERSON_ID                                 NOT NULL VARCHAR2(10)
 ADDRESS_ID                                NOT NULL VARCHAR2(10)
```

Figure 11 Creating Address Table

CREATE TABLE Address (Person_Id varchar(10) NOT NULL,Address_Id varchar(10) NOT NULL, foreign key(Person_Id) references Person(Person_Id), foreign key(Address_Id) references Address_detail(Address_Id))

Appointment Detail Table

The table contains Appointment Id, Appointment Date, Appointment Time, Appointment Charge, Assigned By, Treatment Id and Ward No attributes. The Appointment Id is primary key, Treatment Id and Ward No is foreign key in this table.



```
SQL> CREATE TABLE Appointment_detail (Appointment_Id varchar (10) Primary key, Appointment_Date date NOT NULL, Appointme
nt_Time varchar (5) NOT NULL, Appointment_Charge number (5) NOT NULL, Assigned_By varchar(20) NOT NULL,Ward_No NOT NULL,
Treatment_id NOT NULL,foreign key(Ward_No)references Ward(Ward_No),foreign key (Treatment_Id) references Treatment(Treat
ment_Id));

Table created.

SQL> describe Appointment_detail;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 APPOINTMENT_ID                            NOT NULL VARCHAR2(10)
 APPOINTMENT_DATE                          NOT NULL DATE
 APPOINTMENT_TIME                          NOT NULL VARCHAR2(5)
 APPOINTMENT_CHARGE                        NOT NULL NUMBER(5)
 ASSIGNED_BY                               NOT NULL VARCHAR2(20)
 WARD_NO                                   NOT NULL VARCHAR2(10)
 TREATMENT_ID                              NOT NULL VARCHAR2(10)
```

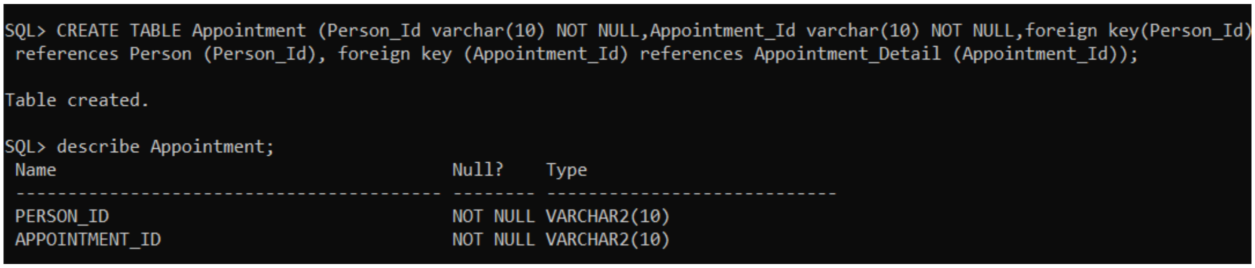
Figure 12 Creating Appointment Detail Table

CREATE TABLE Appointment_detail (Appointment_Id varchar (10) Primary key, Appointment_Date date NOT NULL, Appointment_Time varchar (5) NOT NULL, Appointment_Charge number (5) NOT NULL, Assigned_By varchar(20) NOT

NULL,Ward_No NOT NULL,Treatment_id NOT NULL,foreign key(Ward_No)references Ward(Ward_No),foreign key (Treatment_Id) references Treatment(Treatment_Id))

Appointment Table

The table contains two attributes Person Id and Appointment Id. Both of them are foreign key which helps to connect Person and Appointment detail table



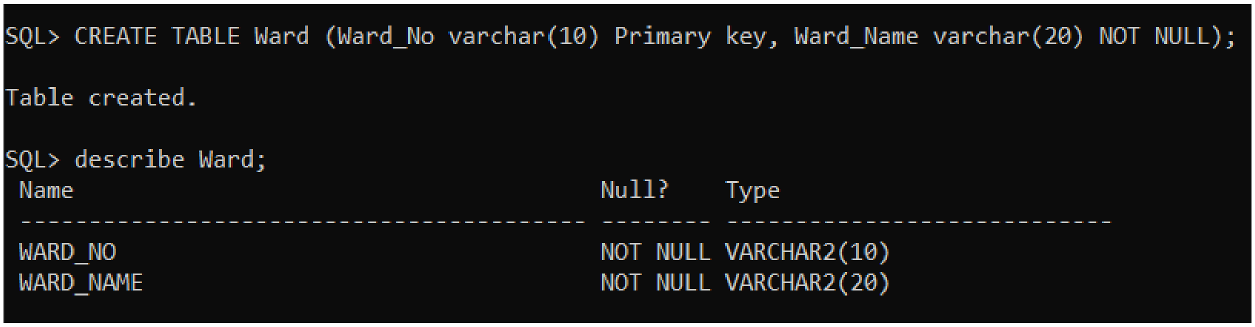
```
SQL> CREATE TABLE Appointment (Person_Id varchar(10) NOT NULL,Appointment_Id varchar(10) NOT NULL,foreign key(Person_Id)
 references Person (Person_Id), foreign key (Appointment_Id) references Appointment_Detail (Appointment_Id));

Table created.

SQL> describe Appointment;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PERSON_ID                                 NOT NULL VARCHAR2(10)
 APPOINTMENT_ID                            NOT NULL VARCHAR2(10)
```

Figure 13 Creating Appointment Table

CREATE TABLE Appointment (Person_Id varchar(10) NOT NULL,Appointment_Id varchar(10) NOT NULL,foreign key(Person_Id) references Person (Person_Id), foreign key (Appointment_Id) references Appointment_Detail (Appointment_Id))

Ward Table

The table contains Ward No and Ward Name attributes. The primary key is Ward No.



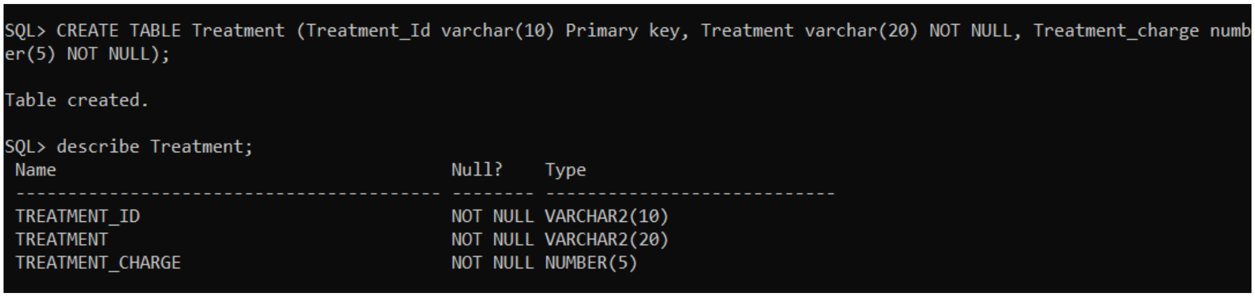
```
SQL> CREATE TABLE Ward (Ward_No varchar(10) Primary key, Ward_Name varchar(20) NOT NULL);

Table created.

SQL> describe Ward;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 WARD_NO                                   NOT NULL VARCHAR2(10)
 WARD_NAME                                 NOT NULL VARCHAR2(20)
```

Figure 14 Creating Ward Table

CREATE TABLE Ward_detail (Ward_No varchar (10) Primary key, Ward_Name varchar (20) NOT NULL)

Treatment Table

The table contain Treatment Id, Treatment and Treatment Charge attributes. The primary key is Treatment Id in this table.



```
SQL> CREATE TABLE Treatment (Treatment_Id varchar(10) Primary key, Treatment varchar(20) NOT NULL, Treatment_charge numb
er(5) NOT NULL);

Table created.

SQL> describe Treatment;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TREATMENT_ID                              NOT NULL VARCHAR2(10)
 TREATMENT                                 NOT NULL VARCHAR2(20)
 TREATMENT_CHARGE                          NOT NULL NUMBER(5)
```

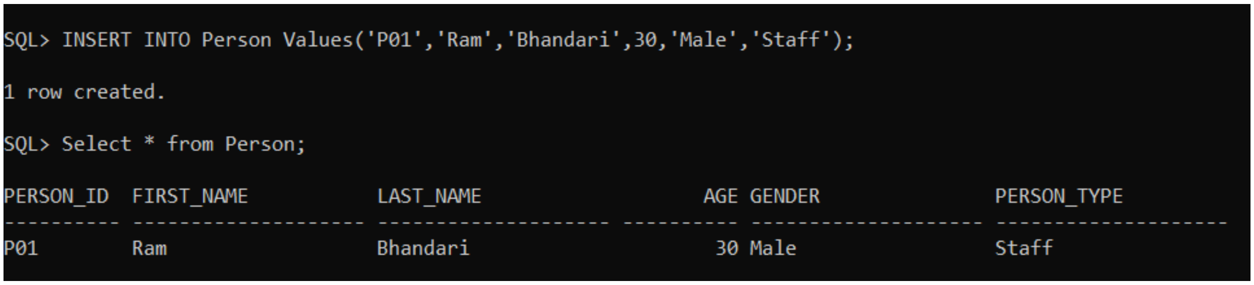
Figure 15 Creating Treatment Table

CREATE TABLE Treatment_detail (Treatment_Id varchar(10) Primary key, Treatment varchar(20) NOT NULL, Treatment_charge number(5) NOT NULL)

4.2 Populating Database

The data is inserted in tables which we created before.

Person

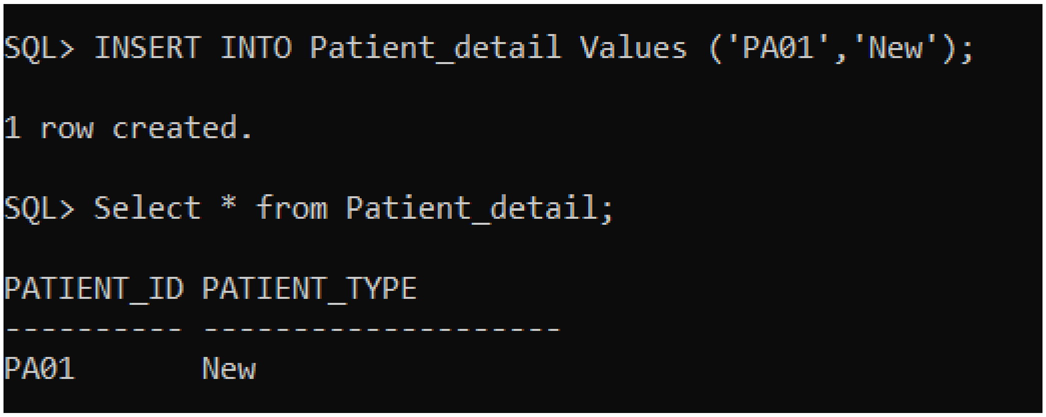


```
SQL> INSERT INTO Person Values('P01','Ram','Bhandari',30,'Male','Staff');

1 row created.

SQL> Select * from Person;

PERSON_ID  FIRST_NAME           LAST_NAME                   AGE GENDER               PERSON_TYPE
---------- -------------------- -------------------- ---------- -------------------- --------------------
P01        Ram                  Bhandari                     30 Male                 Staff
```

Figure 16 Inserting data in Person Table

The data for Person Id is P01, First Name is Ram, Last Name is Bandari, age is 30, gender is male and person type is staff in Person table.

Patient Detail

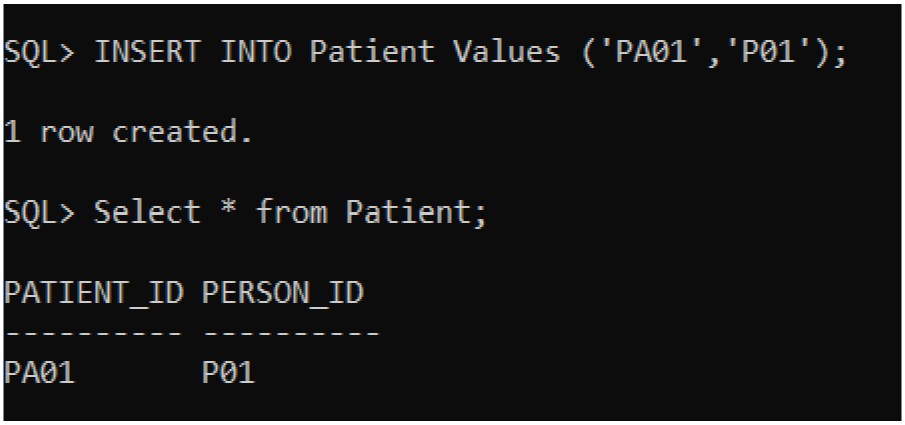


```
SQL> INSERT INTO Patient_detail Values ('PA01','New');

1 row created.

SQL> Select * from Patient_detail;

PATIENT_ID PATIENT_TYPE
---------- --------------------
PA01       New
```

Figure 17 Inserting data in Patient Detail Table

The data for Patient Id is PA01 and Patient type is New in a Patient Detail table.

Patient

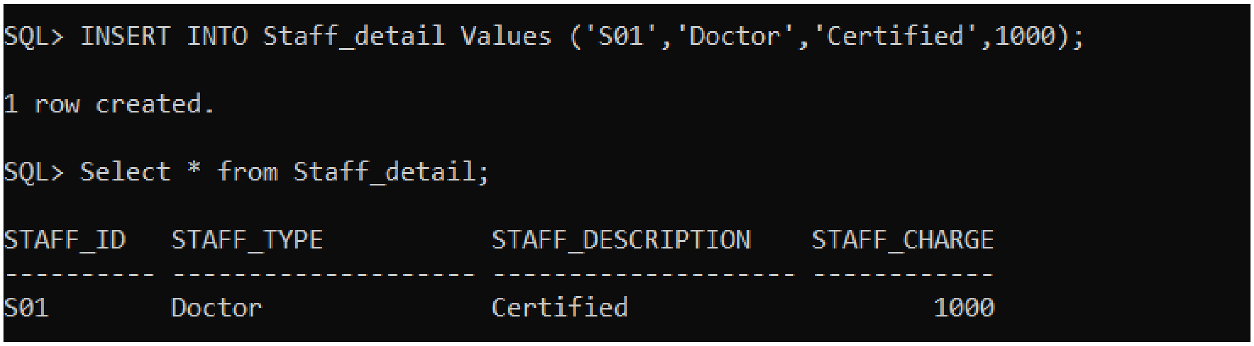


```
SQL> INSERT INTO Patient Values ('PA01','P01');

1 row created.

SQL> Select * from Patient;

PATIENT_ID PERSON_ID
---------- ----------
PA01       P01
```

Figure 18 Inserting data in Patient Table

The data for Patient Id is PA01 and Person Id is P01 in Patient table.

Staff Detail

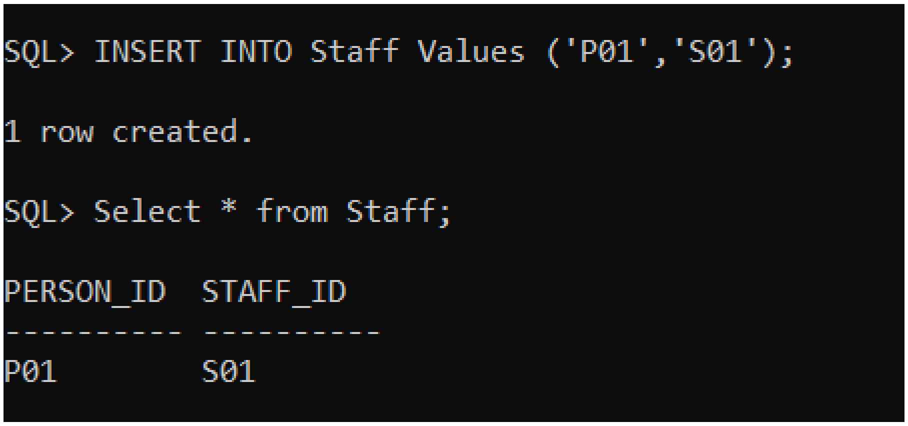


```
SQL> INSERT INTO Staff_detail Values ('S01','Doctor','Certified',1000);

1 row created.

SQL> Select * from Staff_detail;

STAFF_ID   STAFF_TYPE           STAFF_DESCRIPTION    STAFF_CHARGE
---------- -------------------- -------------------- ------------
S01        Doctor               Certified                    1000
```

Figure 19 Inserting data in Staff Detail Table

The data for Staff Id is S01, Staff type is Doctor, Staff description is Certified and Staff charge is 1000 in Staff detail table.

Staff

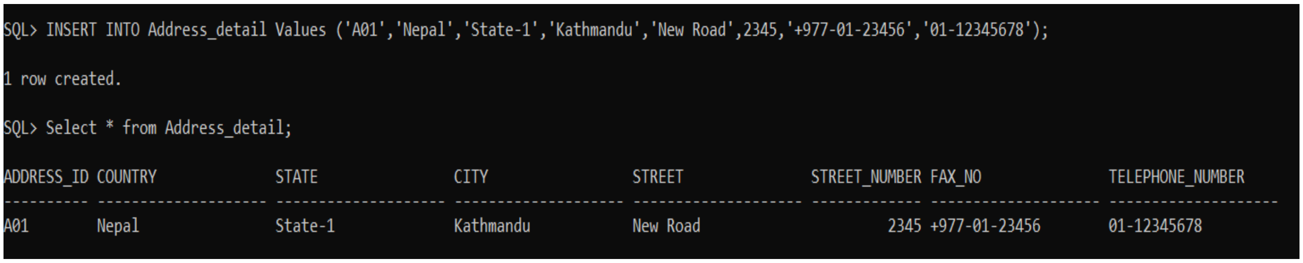


```
SQL> INSERT INTO Staff Values ('P01','S01');

1 row created.

SQL> Select * from Staff;

PERSON_ID  STAFF_ID
---------- ----------
P01        S01
```

Figure 20 Inserting data in Staff Table

The data for Person Id is P01 and Staff Id is S01 in Staff table.

Address Detail

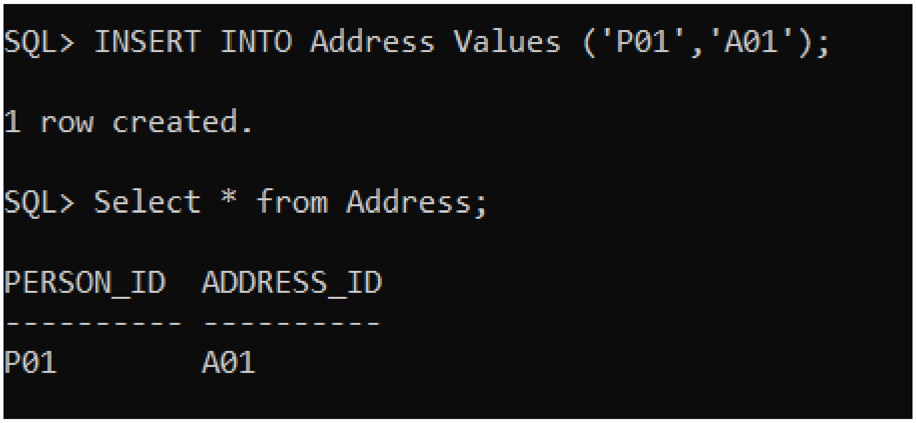


```
SQL> INSERT INTO Address_detail Values ('A01','Nepal','State-1','Kathmandu','New Road',2345,'+977-01-23456','01-12345678');

1 row created.

SQL> Select * from Address_detail;

ADDRESS_ID COUNTRY             STATE               CITY                STREET              STREET_NUMBER FAX_NO              TELEPHONE_NUMBER
---------- ------------------- ------------------- ------------------- ------------------- ------------- ------------------- -------------------
A01        Nepal               State-1             Kathmandu           New Road                     2345 +977-01-23456       01-12345678
```

Figure 21 Inserting data in Address Detail Table

The data for Address Id is A01, Country is Nepal, State is State-1, City is Kathmandu, Street is New Road, Street Number is 2345, Fax No is +977-01-23456 and telephone number is 01-12345678 in Address detail table.

Address

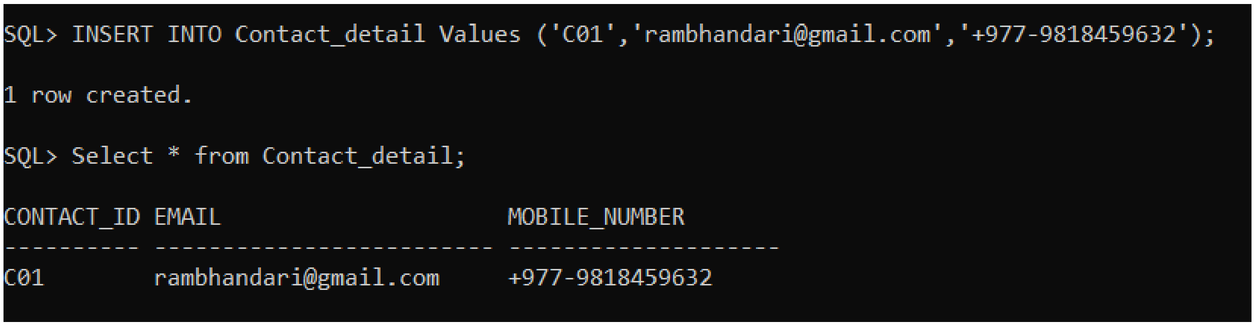


```
SQL> INSERT INTO Address Values ('P01','A01');

1 row created.

SQL> Select * from Address;

PERSON_ID  ADDRESS_ID
---------- ----------
P01        A01
```

Figure 22 Inserting data in Address Table

The data for Person Id is P01 and Address Id is A01 in Address table.

Contact Detail

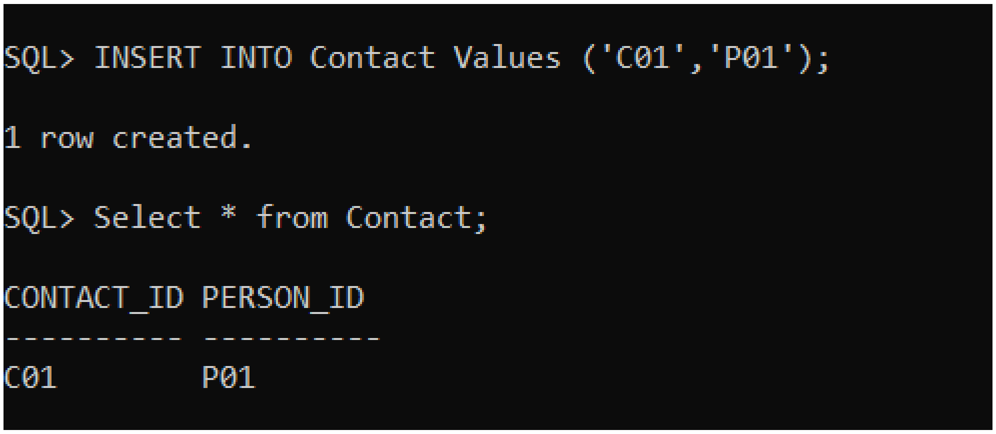


```
SQL> INSERT INTO Contact_detail Values ('C01','rambhandari@gmail.com','+977-9818459632');

1 row created.

SQL> Select * from Contact_detail;

CONTACT_ID EMAIL                     MOBILE_NUMBER
---------- ------------------------- --------------------
C01        rambhandari@gmail.com     +977-9818459632
```

Figure 23 Inserting data in Contact Detail Table

The data for Contact Id is C01, Email is rambhandari@gmail.com and Mobile number is +977-9818459632 in Contact Detail table.

Contact

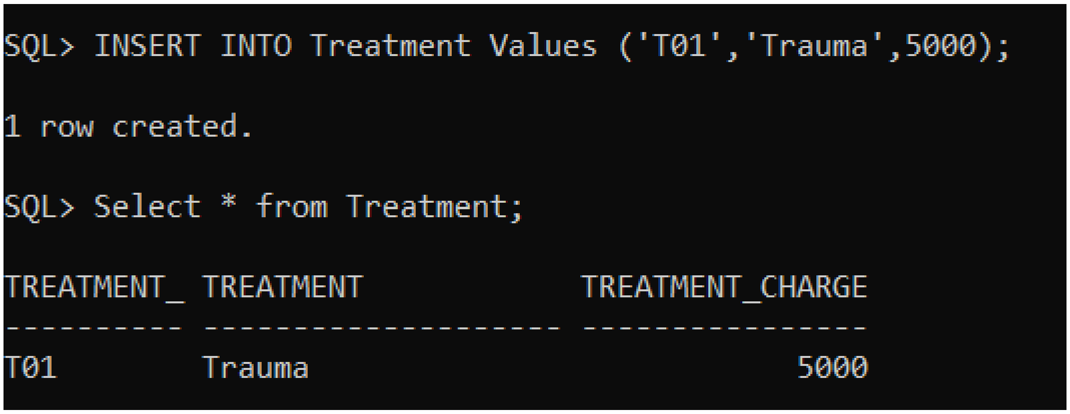


```
SQL> INSERT INTO Contact Values ('C01','P01');

1 row created.

SQL> Select * from Contact;

CONTACT_ID PERSON_ID
---------- ----------
C01        P01
```

Figure 24 Inserting data in Contact Table

The data for Contact Id is C01 and Person Id is P01 in Contact table.

Treatment

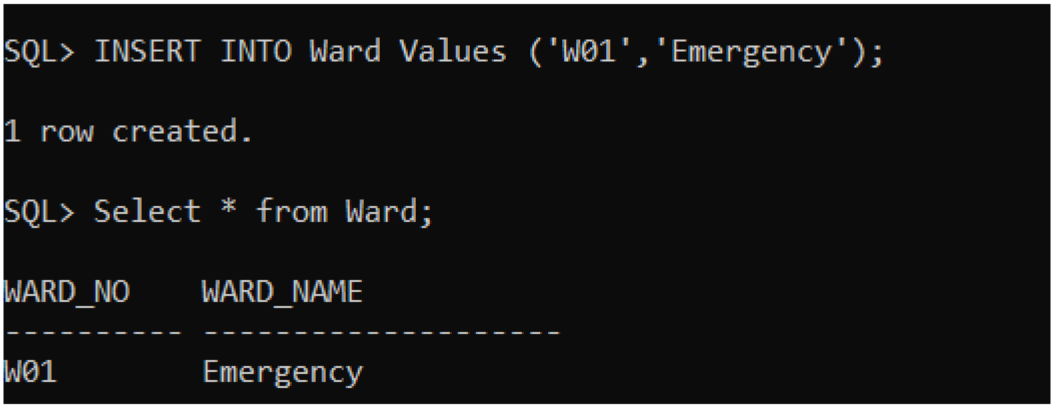


```
SQL> INSERT INTO Treatment Values ('T01','Trauma',5000);

1 row created.

SQL> Select * from Treatment;

TREATMENT_ TREATMENT            TREATMENT_CHARGE
---------- -------------------- ----------------
T01        Trauma                           5000
```

Figure 25 Inserting data in Treatment Table

The data for Treatment Id is T01, Treatment is Trauma and Treatment charge is 5000 in Treatment table.

Ward

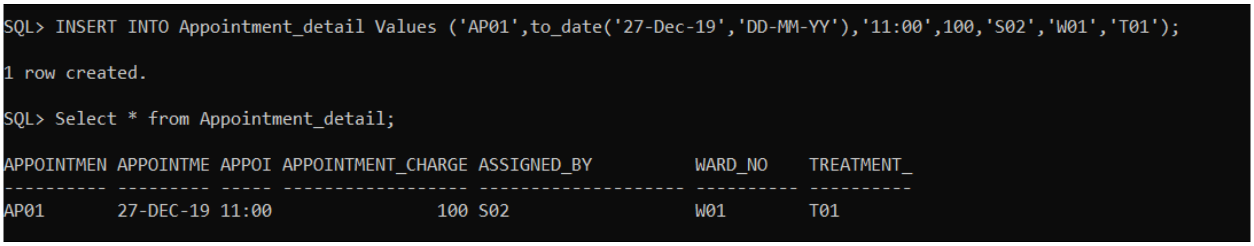


```
SQL> INSERT INTO Ward Values ('W01','Emergency');

1 row created.

SQL> Select * from Ward;

WARD_NO    WARD_NAME
---------- --------------------
W01        Emergency
```

Figure 26 Inserting data in Ward Table

The data for Ward No is W01 and Ward name is Emergency in Ward Table.

Appointment detail

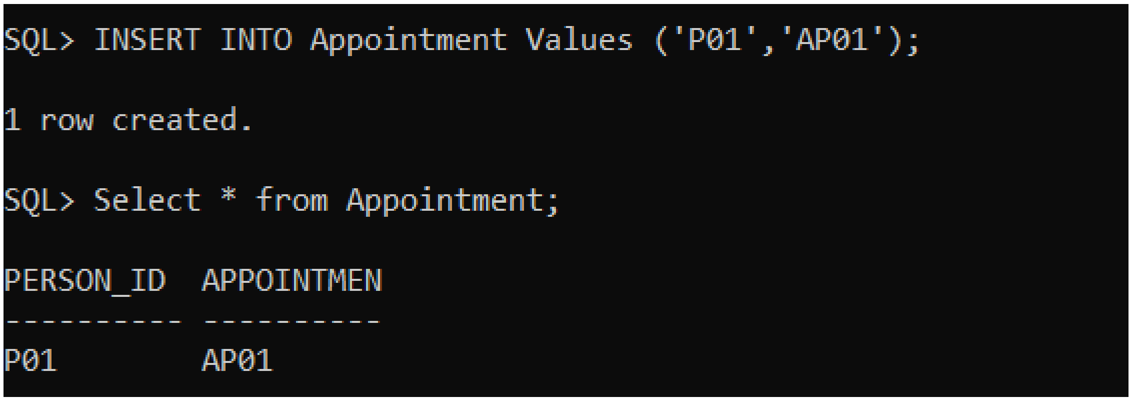


```
SQL> INSERT INTO Appointment_detail Values ('AP01',to_date('27-Dec-19','DD-MM-YY'),'11:00',100,'S02','W01','T01');

1 row created.

SQL> Select * from Appointment_detail;

APPOINTMEN APPOINTME APPOI APPOINTMENT_CHARGE ASSIGNED_BY          WARD_NO    TREATMENT_
---------- --------- ----- ------------------ -------------------- ---------- ----------
AP01       27-DEC-19 11:00                100 S02                  W01        T01
```

Figure 27 Inserting data in Appointment Detail Table

The data for Appointment Id is AP01, Appointment Date is 27-Dec-19, Appointment time is 11:00, Appointment charge id 100, Assigned By is S02, Ward No is W01 and Treatment Id is T01 in Appointment detail table.

Appointment



```
SQL> INSERT INTO Appointment Values ('P01','AP01');

1 row created.

SQL> Select * from Appointment;

PERSON_ID  APPOINTMEN
---------- ----------
P01        AP01
```

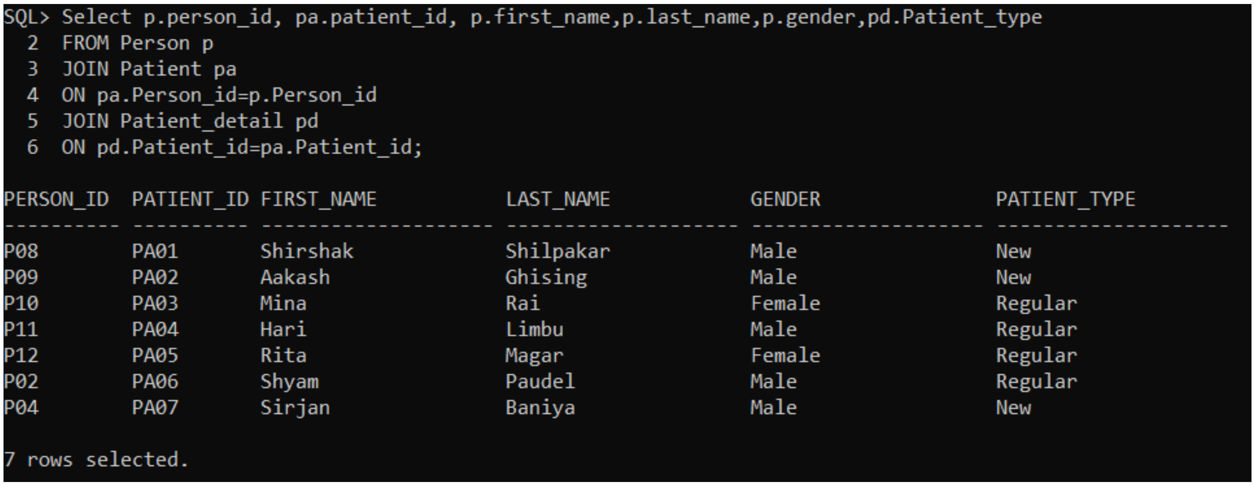
Figure 28 Inserting data in Appointment Table

The data for Person Id is P01 and Appointment Id is AP01 in Appointment table.

5. Database Querying

5.1 Information Queries

i) List all patients, regular and new

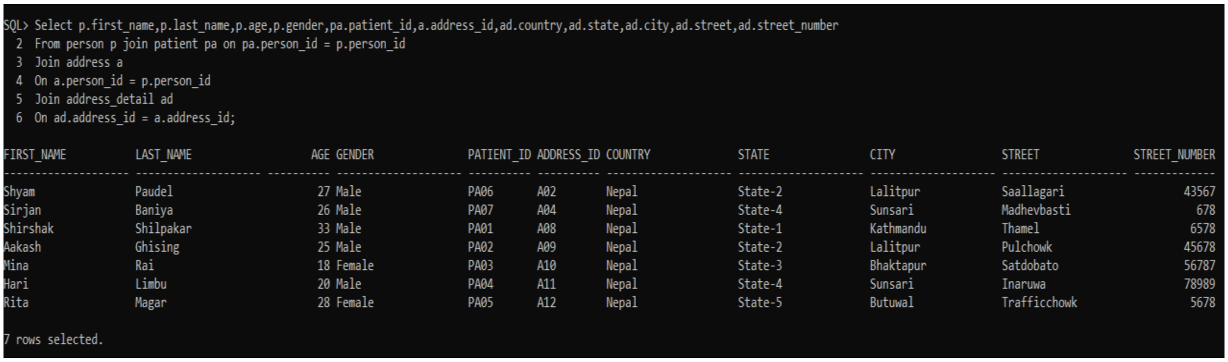


```
SQL> Select p.person_id, pa.patient_id, p.first_name,p.last_name,p.gender,pd.Patient_type
     FROM Person p
     JOIN Patient pa
     ON pa.Person_id=p.Person_id
     JOIN Patient_detail pd
     ON pd.Patient_id=pa.Patient_id;

PERSON_ID  PATIENT_ID FIRST_NAME           LAST_NAME            GENDER               PATIENT_TYPE
---------- ---------- -------------------- -------------------- -------------------- --------------------
P08        PA01       Shirshak             Shilpakar            Male                 New
P09        PA02       Aakash               Ghising              Male                 New
P10        PA03       Mina                 Rai                  Female               Regular
P11        PA04       Hari                 Limbu                Male                 Regular
P12        PA05       Rita                 Magar                Female               Regular
P02        PA06       Shyam                Paudel               Male                 Regular
P04        PA07       Sirjan               Baniya               Male                 New

7 rows selected.
```

Figure 29 Query 1

In above figure, the Patient who are new and regular is shown with Person Id, Patient Id, First Name, Last Name, Gender and Patient Type by joining Person, Patient and Patient Detail table.

ii) List all patients with all their addresses.

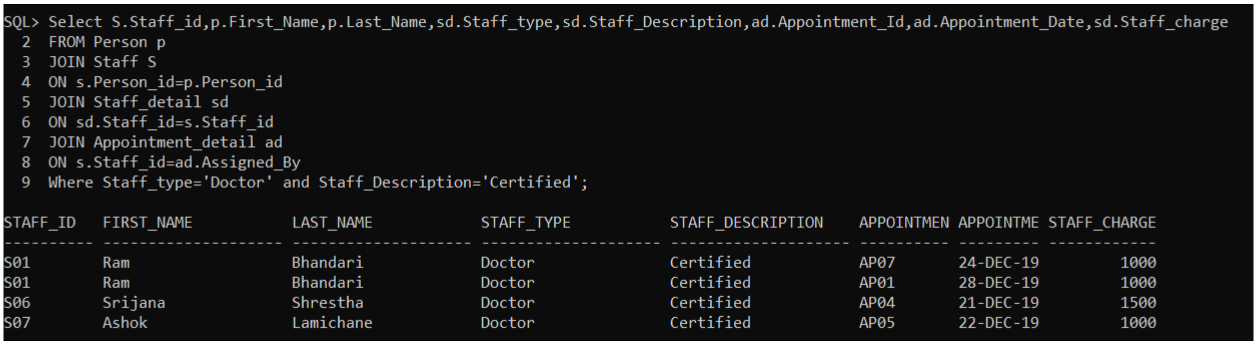


```
SQL> Select p.first_name,p.last_name,p.age,p.gender,pa.patient_id,a.address_id,ad.country,ad.state,ad.city,ad.street,ad.street_number
     From person p join patient pa on pa.person_id = p.person_id
     Join address a
     On a.person_id = p.person_id
     Join address_detail ad
     On ad.address_id = a.address_id;

FIRST_NAME           LAST_NAME                   AGE GENDER               PATIENT_ID ADDRESS_ID COUNTRY              STATE                CITY                 STREET               STREET_NUMBER
-------------------- -------------------- ---------- -------------------- ---------- ---------- -------------------- -------------------- -------------------- -------------------- -------------
Shyam                Paudel                       27 Male                 PA06       A02        Nepal                State-2              Lalitpur             Saallagari                   43567
Sirjan               Baniya                       26 Male                 PA07       A04        Nepal                State-4              Sunsari              Madhevbasti                    678
Shirshak             Shilpakar                    33 Male                 PA01       A08        Nepal                State-1              Kathmandu            Thamel                        6578
Aakash               Ghising                      25 Male                 PA02       A09        Nepal                State-2              Lalitpur             Pulchowk                     45678
Mina                 Rai                          18 Female               PA03       A10        Nepal                State-3              Bhaktapur            Satdobato                    56787
Hari                 Limbu                        20 Male                 PA04       A11        Nepal                State-4              Sunsari              Inaruwa                      78989
Rita                 Magar                        28 Female               PA05       A12        Nepal                State-5              Butuwal              Trafficchowk                  5678

7 rows selected.
```

Figure 30 Query 2

In above figure, the address of all the patient is shown with First Name, Last Name, Age, Gender, Patient Id, Address Id, Country, State, City, Street and Street Number by joining Person, Address and Address table.

iii) For a given certified doctor, find all the appointments he/she have conducted and the amount he/she got for conducting the appointment.

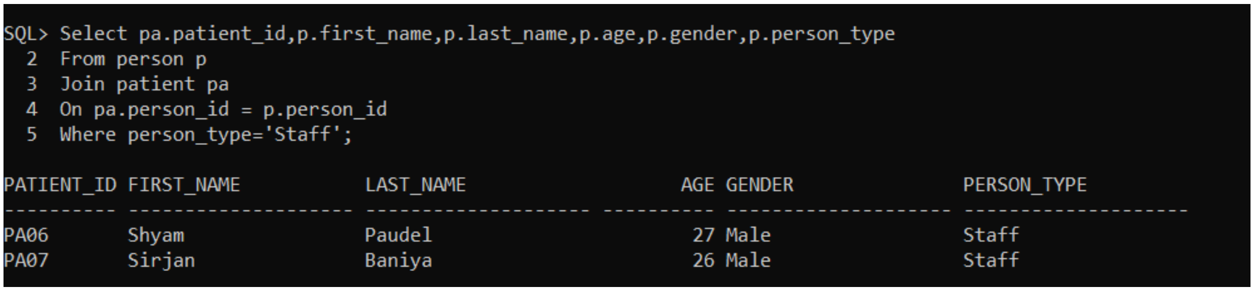


```
SQL> Select S.Staff_id,p.First_Name,p.Last_Name,sd.Staff_type,sd.Staff_Description,ad.Appointment_Id,ad.Appointment_Date,sd.Staff_charge
FROM Person p
JOIN Staff S
ON s.Person_id=p.Person_id
JOIN Staff_detail sd
ON sd.Staff_id=s.Staff_id
JOIN Appointment_detail ad
ON s.Staff_id=ad.Assigned_By
Where Staff_type='Doctor' and Staff_Description='Certified';

STAFF_ID   FIRST_NAME           LAST_NAME            STAFF_TYPE           STAFF_DESCRIPTION    APPOINTMEN APPOINTME STAFF_CHARGE
---------- -------------------- -------------------- -------------------- -------------------- ---------- --------- ------------
S01        Ram                  Bhandari             Doctor               Certified            AP07       24-DEC-19         1000
S01        Ram                  Bhandari             Doctor               Certified            AP01       28-DEC-19         1000
S06        Srijana              Shrestha             Doctor               Certified            AP04       21-DEC-19         1500
S07        Ashok                Lamichane            Doctor               Certified            AP05       22-DEC-19         1000
```

Figure 31 Query 3

In above figure, Doctors who have conducted appointments and amount paid for an appointment is shown with Staff Id, First Name, Last Name, Staff type, Staff Description, Appointment Id, Appointment Date and Staff Charge by joining Person, Staff, Staff Detail and Appointment detail table.

iv) List all staffs that are also a patient.



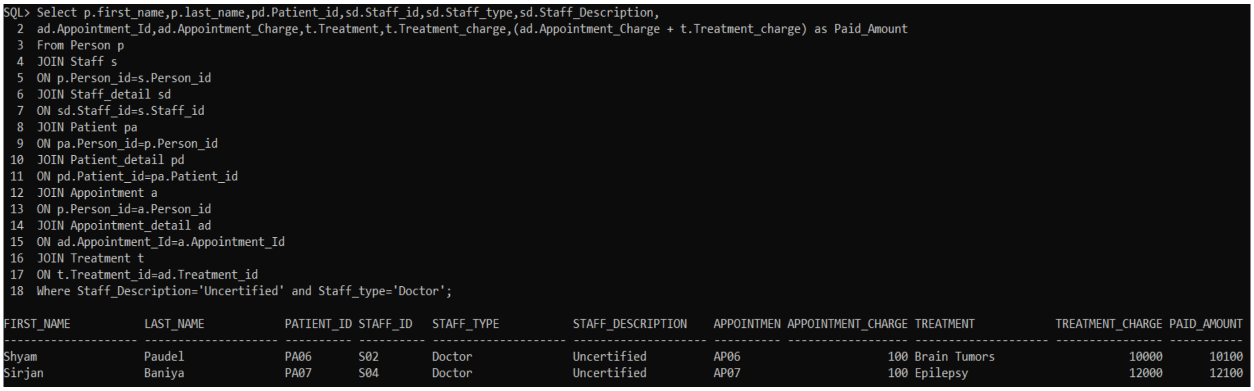
```
SQL> Select pa.patient_id,p.first_name,p.last_name,p.age,p.gender,p.person_type
From person p
Join patient pa
On pa.person_id = p.person_id
Where person_type='Staff';

PATIENT_ID FIRST_NAME           LAST_NAME                   AGE GENDER               PERSON_TYPE
---------- -------------------- -------------------- ---------- -------------------- --------------------
PA06       Shyam                Paudel                       27 Male                 Staff
PA07       Sirjan               Baniya                       26 Male                 Staff
```

Figure 32 Query 4

In above figure, the staff of hospital who are also a patient is shown with their Patient Id, First Name, Last Name, Age, Gender and Person type by joining Person and Patient table.

5.2 Transaction Queries

i) List all uncertified doctors who have been attended an appointment for a treatment and the amount he/she have paid.

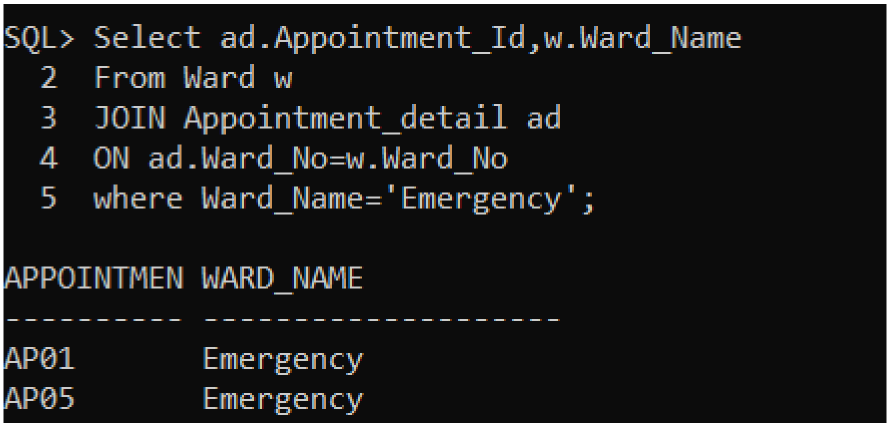


```
SQL> Select p.first_name,p.last_name,pd.Patient_id,sd.Staff_id,sd.Staff_type,sd.Staff_Description,
ad.Appointment_Id,ad.Appointment_Charge,t.Treatment,t.Treatment_charge,(ad.Appointment_Charge + t.Treatment_charge) as Paid_Amount
From Person p
JOIN Staff s
ON p.Person_id=s.Person_id
JOIN Staff_detail sd
ON sd.Staff_id=s.Staff_id
JOIN Patient pa
ON pa.Person_id=p.Person_id
JOIN Patient_detail pd
ON pd.Patient_id=pa.Patient_id
JOIN Appointment a
ON p.Person_id=a.Person_id
JOIN Appointment_detail ad
ON ad.Appointment_Id=a.Appointment_Id
JOIN Treatment t
ON t.Treatment_id=ad.Treatment_id
Where Staff_Description='Uncertified' and Staff_type='Doctor';

FIRST_NAME           LAST_NAME            PATIENT_ID STAFF_ID   STAFF_TYPE           STAFF_DESCRIPTION    APPOINTMEN APPOINTMENT_CHARGE TREATMENT            TREATMENT_CHARGE PAID_AMOUNT
-------------------- -------------------- ---------- ---------- -------------------- -------------------- ---------- ------------------ -------------------- ---------------- -----------
Shyam                Paudel               PA06       S02        Doctor               Uncertified          AP06                      100 Brain Tumors                    10000       10100
Sirjan               Baniya               PA07       S04        Doctor               Uncertified          AP07                      100 Epilepsy                        12000       12100
```

Figure 33 Query 5

In above figure, all uncertified doctors who have attended a appointment for a treatment and the amount they have paid is shown with First Name, Last Name, Patient Id, Staff Id, Staff type, Staff description, Appointment Id, Appointment charge, Treatment, Treatment charge and Paid Amount by joining Person, Staff, Staff detail, Patient, Patient detail, Appointment, Appointment detail and Treatment table.

ii) List the appointments that have been conducted in an emergency ward.

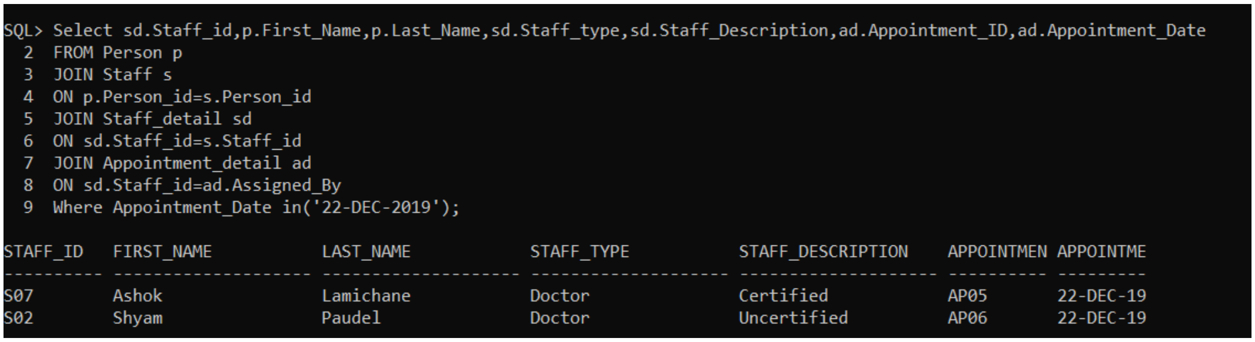


```
SQL> Select ad.Appointment_Id,w.Ward_Name
From Ward w
JOIN Appointment_detail ad
ON ad.Ward_No=w.Ward_No
where Ward_Name='Emergency';

APPOINTMEN WARD_NAME
---------- --------------------
AP01       Emergency
AP05       Emergency
```

Figure 34 Query 6

In the above figure, the appointment that are conducted on emergency ward is shown with Appointment Id and Ward Name by joining Appointment detail and Ward table.

iii) List all staffs (certified and uncertified) who have conducted or will conduct an appointment on a given date.

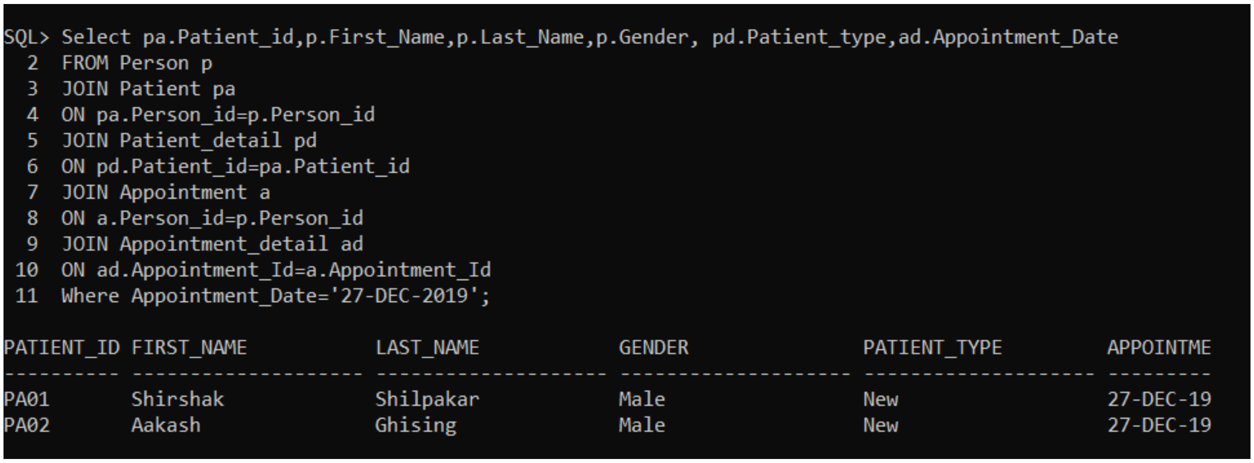


```
SQL> Select sd.Staff_id,p.First_Name,p.Last_Name,sd.Staff_type,sd.Staff_Description,ad.Appointment_ID,ad.Appointment_Date
  2  FROM Person p
  3  JOIN Staff s
  4  ON p.Person_id=s.Person_id
  5  JOIN Staff_detail sd
  6  ON sd.Staff_id=s.Staff_id
  7  JOIN Appointment_detail ad
  8  ON sd.Staff_id=ad.Assigned_By
  9  Where Appointment_Date in('22-DEC-2019');

STAFF_ID   FIRST_NAME           LAST_NAME            STAFF_TYPE           STAFF_DESCRIPTION    APPOINTMEN APPOINTME
---------- -------------------- -------------------- -------------------- -------------------- ---------- ---------
S07        Ashok                Lamichane            Doctor               Certified            AP05       22-DEC-19
S02        Shyam                Paudel               Doctor               Uncertified          AP06       22-DEC-19
```

Figure 35 Query 7

In above figure, all the staff who have conducted or will conduct an appointment on a given date is shown with their staff Id, First Name, Last Name, Staff type, Staff description, Appointment Id and Appointment Date by joining Person, Staff, Staff detail and Appointment detail table.

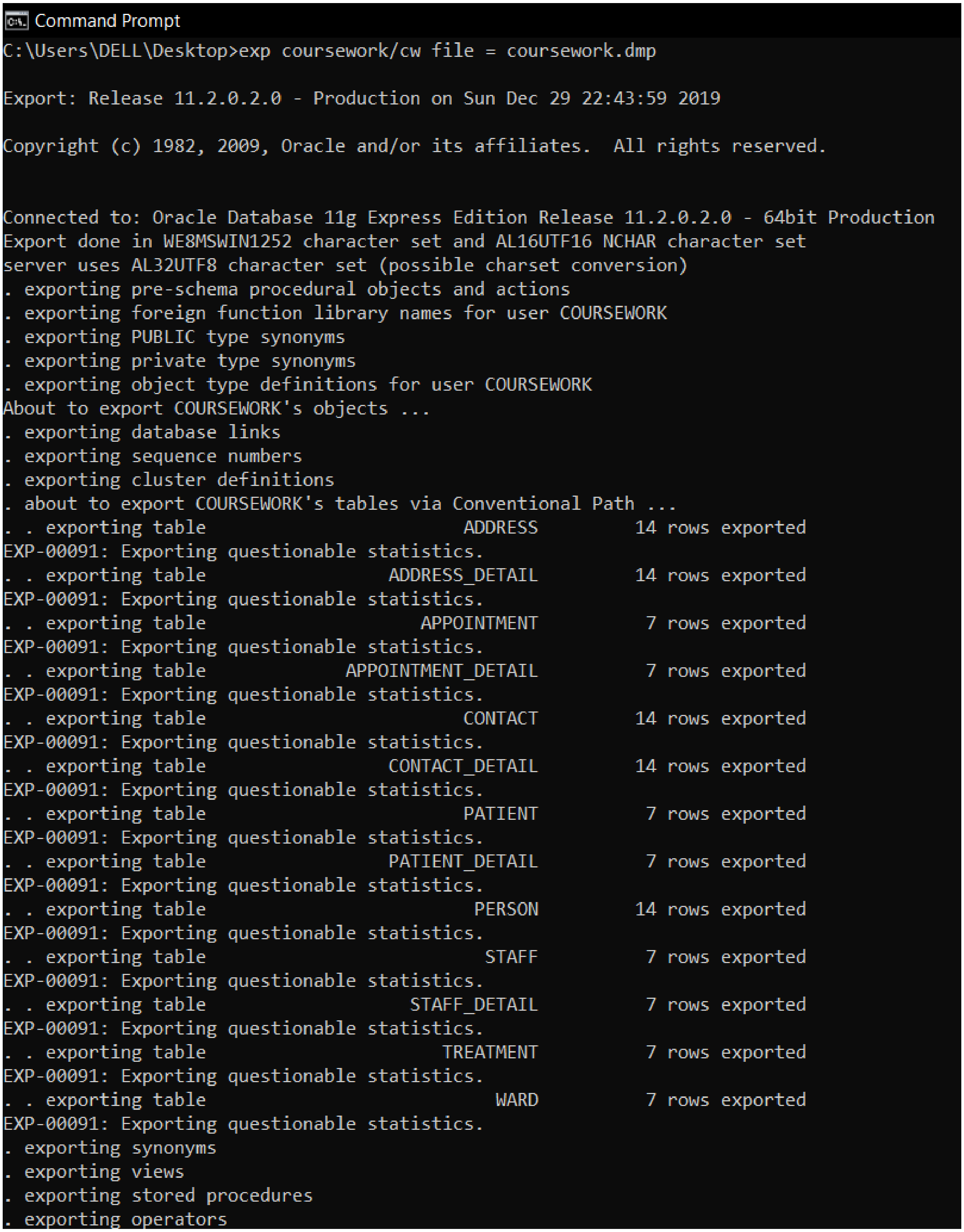
iv) List all patients booked for an appointment on a given date.



```
SQL> Select pa.Patient_id,p.First_Name,p.Last_Name,p.Gender, pd.Patient_type,ad.Appointment_Date
  2  FROM Person p
  3  JOIN Patient pa
  4  ON pa.Person_id=p.Person_id
  5  JOIN Patient_detail pd
  6  ON pd.Patient_id=pa.Patient_id
  7  JOIN Appointment a
  8  ON a.Person_id=p.Person_id
  9  JOIN Appointment_detail ad
 10  ON ad.Appointment_Id=a.Appointment_Id
 11  Where Appointment_Date='27-DEC-2019';

PATIENT_ID FIRST_NAME           LAST_NAME            GENDER               PATIENT_TYPE         APPOINTME
---------- -------------------- -------------------- -------------------- -------------------- ---------
PA01       Shirshak             Shilpakar            Male                 New                  27-DEC-19
PA02       Aakash               Ghising              Male                 New                  27-DEC-19
```

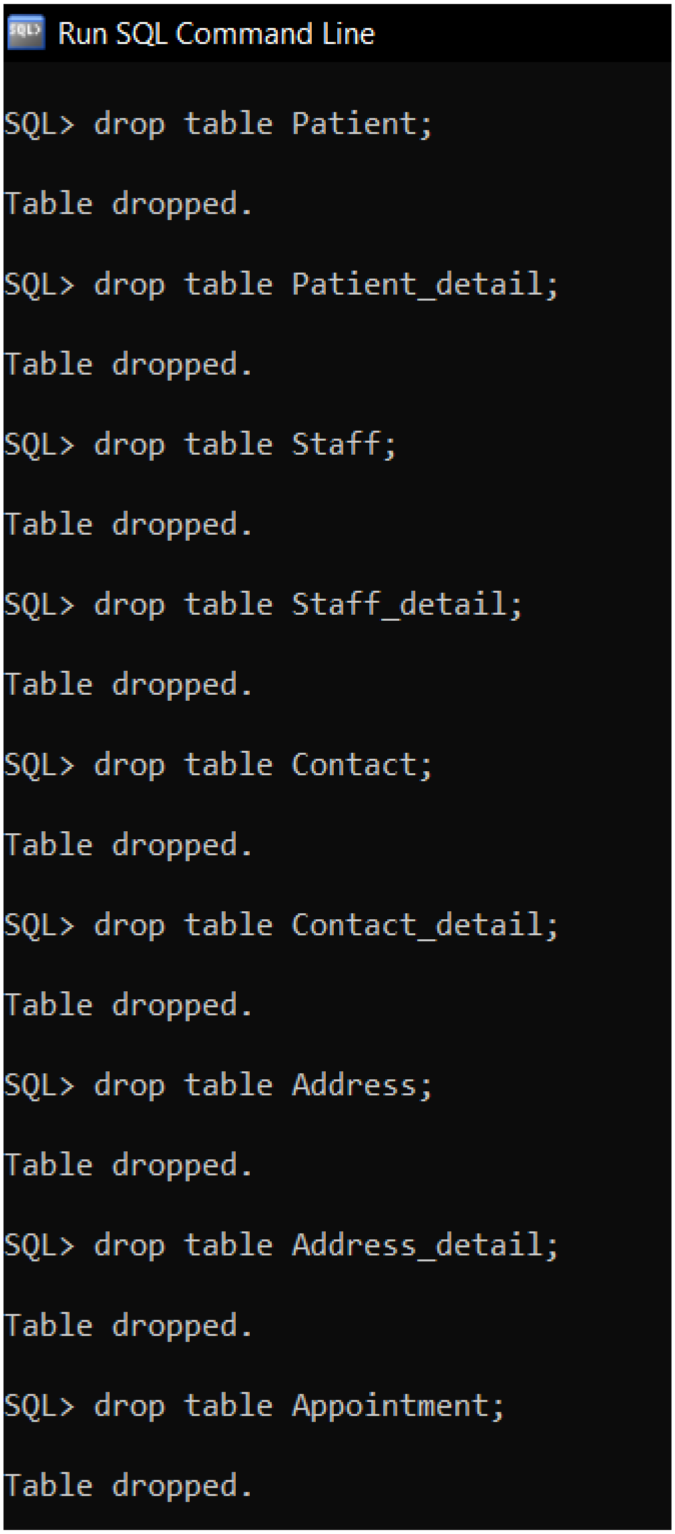
Figure 36 Query 8

In above figure, all patient who have booked an appointment on given date is shown with Patient Id, First Name, Last Name, Gender, Patient type and appointment date by joining Person, Patient, Patient detail, Appointment and Appointment table.



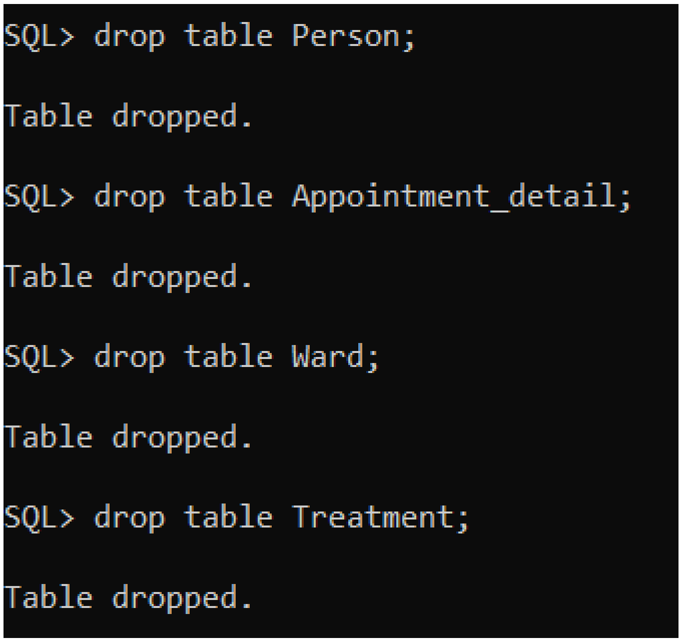
```
C:\Users\DELL\Desktop>exp coursework/cw file = coursework.dmp

Export: Release 11.2.0.2.0 - Production on Sun Dec 29 22:43:59 2019

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights reserved.


Connected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user COURSEWORK
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user COURSEWORK
About to export COURSEWORK's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export COURSEWORK's tables via Conventional Path ...
. . exporting table                        ADDRESS         14 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                 ADDRESS_DETAIL         14 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                    APPOINTMENT          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table             APPOINTMENT_DETAIL          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                        CONTACT         14 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                 CONTACT_DETAIL         14 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                        PATIENT          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                 PATIENT_DETAIL          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                         PERSON         14 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                          STAFF          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                   STAFF_DETAIL          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                      TREATMENT          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                           WARD          7 rows exported
EXP-00091: Exporting questionable statistics.
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
```

Figure 37 Dump File Created



```
SQL> drop table Patient;

Table dropped.

SQL> drop table Patient_detail;

Table dropped.

SQL> drop table Staff;

Table dropped.

SQL> drop table Staff_detail;

Table dropped.

SQL> drop table Contact;

Table dropped.

SQL> drop table Contact_detail;

Table dropped.

SQL> drop table Address;

Table dropped.

SQL> drop table Address_detail;

Table dropped.

SQL> drop table Appointment;

Table dropped.
```

Figure 38 Table dropped 1



```
SQL> drop table Person;

Table dropped.

SQL> drop table Appointment_detail;

Table dropped.

SQL> drop table Ward;

Table dropped.

SQL> drop table Treatment;

Table dropped.
```

Figure 39 Table dropped 2

Note:

Username = coursework

Password = cw

6. Critical Evaluation

The coursework of database was quilt difficult and vast. After many errors and obstacles, the assignment has almost come to an end. There was scenario given about Patient record system of Hospital. We need to create database to keep record of patient for a hospital. So, I took reference of most popular neurological hospital of Nepal ‘Annapurna Neurological Institute & Allied Sciences’.

The creation of entities and attributes was little easier it given in the requirements of coursework. While creating ER diagram, assigning relationship between entities was little confusing. I have to make many ER diagrams because I had to change them time and again due to some mistake while making attributes and normalization. Normalization was most difficult, confusing and adventurous part of this coursework. I need to do normalization many times because it was not correct. So, I had to do many researches and consult with my module teachers. There was not much difficulties doing 1NF, 2NF and 3NF but it was much difficult to UNF. So, I need to normalize from beginning as UNF is not correct. In order to make correct UNF, we have to know all the scenario and have to be well panned. It was hard to separate repeating groups and repeating data. The repeating data and group have to match with our scenario. It requires a lot hard thinking to create UNF. There was not much difficult to insert the data because it was already normalized but inserting data type was little tough as I did not think properly before inserting data. I had to some research while doing given SQL queries on relating things.

This coursework was very fruitful to get knowledge about the database. Now I had clear view of normalization after doing a lot of normalization in this coursework.

7. Critical Assessment of Coursework

After many trials and errors, the coursework was finally completed. The coursework was very hard and challenging but very fruitful and informative. The main core of database management system is normalization because it helps to reduce Data Redundancy. I had to do many researches to get many ideas and techniques for this coursework. After completing this project, I know how to create a database with normalization in detail of Oracle SQL Plus. Database is developed to reduce duplication of data, easy to update, cost and performance efficient. Last year, we worked on MY SQL database which was easier. We made database of company without normalization. But this coursework was very hard compared to last year. We have to plan everything from the beginning and have normalize with the scenario.

To conclude, it was very interesting and challenging experience as it was very fruitful to get knowledge about how to create a database model. It can be very useful for creating a database management system for small and big organization where data is very important. This coursework helped me to improve my skills and knowledge to create a database which can be useful for my future projects.