Business case: Create a database management system for the Integrated Hospital and Pharmacy (IHP)

Problem: There is lack of structure in the organization and storing of data between the hospital and pharmacy causing multiple issues such as mishandling of patient records and medications

Aim: To synchronize the operations and data administration of IHP

A comprehensive database project called the Integrated Hospital and Pharmacy administration System aims to synchronize the operations and data administration of a hospital and a connected pharmacy. Healthcare databases help individual medical organizations understand their daily activities and their place within the larger healthcare industry. This allows healthcare professionals to make decisions abouthow they run their businesses, the work they do, and the systems they use to manage their operations. Due to the rapid advancement of healthcare technology and post-COVID-19 changes in healthcareoperations, it is more vital than ever for healthcare databases to be organized, well-maintained, and simple to use. With the aim of enhancing patient care, streamlining administrative procedures, and

optimizing inventory management, this project spans the whole database project lifecycle.

A healthcare database management system (DBMS) project has a large scope due to the complex and crucial nature of healthcare data. When defining the scope of a healthcare DBMS project, consider the following critical issues and areas:

1. Registration and demographic information: Which are part of patient information management. Medical and health records. Allergies, medications, and immunization records. Insurance information.

2. Scheduling and managing appointments: Making appointments. Keeping track of and managingappointment schedules.

3. Patient health records are stored and retrieved via electronic health records (EHR): For bothorganized and unstructured data support. Integration of diagnostic instruments and tools.

4. Billing and payment management: Billing insurance companies and patients. Taking care of receiptsand bills. Managing insurance reimbursements and claims.

5. Management of Laboratory and Diagnostic Tests: Ordering and tracking lab tests. Preserving testoutcome data. Integration with systems and equipment in the lab.

6. Staff and Resource Management: Organizing the schedules of medical specialists. Keeping track ofemployee qualifications and certifications. Allocating and managing resources.

7.Medical supply and equipment inventory management: Keeping track of stock levels and placing neworders. Item expiration warnings.

8. Feedback and Iteration:

- Planning for continuous improvement based on user feedback and evolving healthcare needs.

Based on the needs of the healthcare organization, the size of operations, and the available resources, the scope of a healthcare DBMS project might vary significantly. To ensure the project's success, it is crucial to work closely with administrators, IT specialists, and healthcare professionals to define and hone the project scope. Keeping current with industry standards and best practices for healthcare data management is also essential.

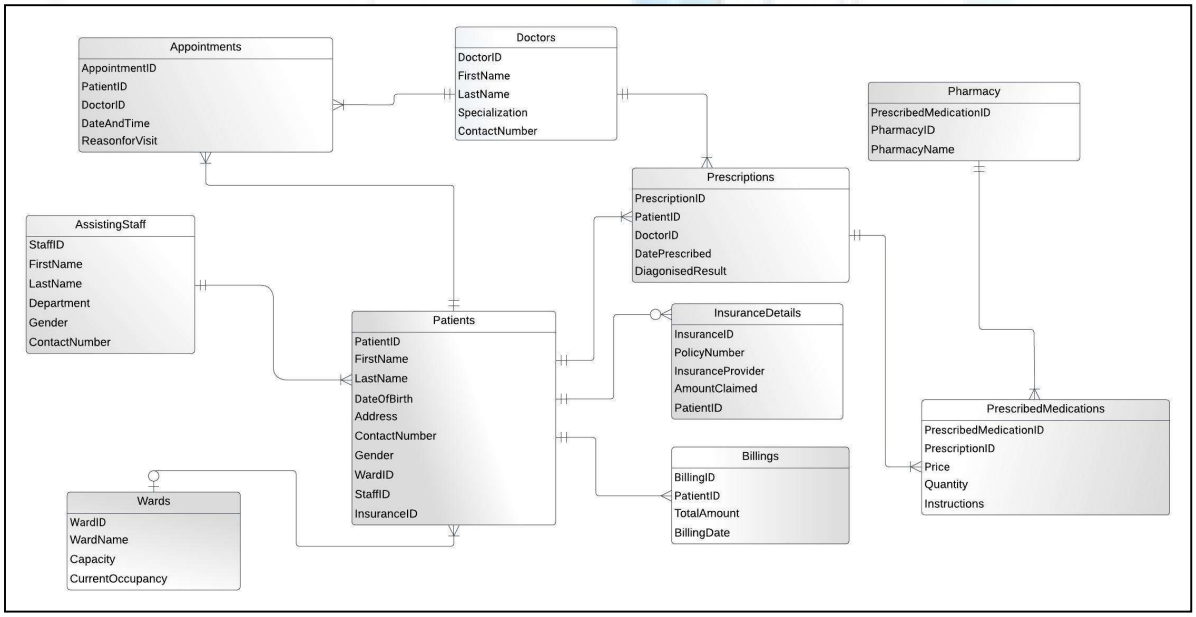
GP2- Detailed Business Rules & ER Diagram

Project: Ensure effective access and integration for better patient care by enabling seamlesssynchronization between the pharmacy's system and the hospital's patient care delivery system.

Entities:

- Doctors
- Patients
- Assisting Staff
- Pharmacy
- Prescriptions
- Prescribed Medications
- Appointments
- Insurance Details
- Wards
- Billings

ERD:
Click here→ERD



Business Rules:

Based on how things are associated in the environment, relationship cardinalities are defined.

- **Doctors:**
  - Relationships:
    - One-to-Many with Appointments.
    - ZOne-to-Many with Prescriptions.
  - Attributes: Doctor ID, FirstName, LastName, Specialization, and ContactNumber
- **Patients:**
  - Relationships:
    - One-to-Many with Appointments
    - One-to-Many with Prescriptions.
    - One-to-Many with InsuranceDetails
    - Many-to-Optional One with Wards
    - One-to-Many with PatientMedicalHistories
    - One-to-Many with Billings
    - Many-to-One with AssistingStaff
  - Attributes: PatientID, FirstName, LastName, DateOfBirth, Address,ContactNumber, Gender, WardID, StaffID, and InsuranceID
- **AssistingStaff:**
  - Relationships:
    - One-to-Many with Patients.
  - Attributes: StaffID, FirstName, LastName, Department, Gender, and ContactNumber.
- **Pharmacy:**
  - Relationships:
    - One-to-Many with PrescribedMedications
  - Attributes: PrescribedMedicationID (FK), PharmacyID and PharmacyName
- **Prescriptions:**
  - Relationships:
    - Many-to-One with Doctors.
    - Many-to-One with Patients.
    - One-to-Many with PrescribedMedication
  - Attributes: PrescriptionID, PatientID (FK), DoctorID (FK), DatePrescribed, and DiagonisedResult.

- **Prescribed Medications:**
  - Relationships:
    - Many-to-One with Prescriptions.
    - Many-to-One with Pharmacy.
  - Attributes: PrescribedMedicationId (PK), PrescriptionID (FK), Price, Quantity, and Instructions.
- **Appointments:**
  - Relationships:
    - Many-to-One with Doctors.
    - Many-to-One with Patients.
  - Attributes: AppointmentID, PatientID (FK), DoctorID (FK), DateAndTime, and ReasonForVisit.
- **InsuranceDetails:**
  - Relationships:
    - Many-to-One with Patients.
  - Attributes: InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed and PatientID (FK).
- **Wards:**
  - Relationships:
    - One-to-Many with Patients
  - Attributes: WardID, WardName, Capacity, and CurrentOcupancy.
- **Billings:**
  - Relationships:
    - Many-to-One with Patients
  - Attributes: BillingID, PatientID (FK), TotalAmount, and BillingDate

GP3-DDL AND PHYSICAL DIAGRAM

1. DDL of Hospital and Pharmacy Management

Doctors table:

```
CREATE TABLE Doctors (
   DoctorID NUMBER(10) PRIMARY KEY,
   FirstName VARCHAR2(200),
   LastName VARCHAR2(200),
   Specialization VARCHAR2(200),
   ContactNumber NUMBER(10)
);
```

AssistingStaff table:

```
CREATE TABLE AssistingStaff (
   StaffID NUMBER(10) PRIMARY KEY,
   FirstName VARCHAR2(200),
   LastName VARCHAR2(200),
   Department VARCHAR2(200),
   Gender VARCHAR2(10),
   ContactNumber NUMBER(10)
);
```

Wards table:

```
CREATE TABLE Wards (
   WardID NUMBER(10) PRIMARY KEY,
   WardName VARCHAR2(20),
   Capacity NUMBER(10),
   CurrentOccupancy NUMBER(10)
);
```

Patients table:

```
CREATE TABLE Patients (
   PatientID NUMBER(10) PRIMARY KEY,
   FirstName VARCHAR2(200),
```

```
    LastName VARCHAR2(200),
    DateOfBirth DATE,
    Address VARCHAR2(200),
    ContactNumber NUMBER(10),
    Gender VARCHAR2(10),
    InsuranceID NUMBER(10),
    StaffID NUMBER(10),
    WardID NUMBER(10),


    FOREIGN KEY (InsuranceID) REFERENCES InsuranceDetails(InsuranceID),
    FOREIGN KEY (StaffID) REFERENCES AssistingStaff(StaffID),
    FOREIGN KEY (WardID) REFERENCES Wards(WardID)
);
```

InsuranceDetails table:

```
CREATE TABLE InsuranceDetails (
    InsuranceID NUMBER(10) PRIMARY KEY,
    PolicyNumber NUMBER(10),
    InsuranceProvider VARCHAR2(20),
    AmountClaimed NUMBER(10),
    PatientID NUMBER(10)
);
```

Appointments table:

```
CREATE  TABLE  Appointments (
    AppointmentID NUMBER(10) PRIMARY KEY,
    PatientID NUMBER(10),
    DoctorID NUMBER(10),
    DateAndTime TIMESTAMP,
    ReasonForVisit VARCHAR2(200),
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
);
```

Prescriptions table:

```
CREATE TABLE Prescriptions (
   PrescriptionID NUMBER(10) PRIMARY KEY,
   PatientID NUMBER(10),
   DoctorID NUMBER(10),
   DatePrescribed DATE,
   DiagnosedResult VARCHAR2(200),
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
   FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
);
```

Billings table:

```
CREATE TABLE Billings (
   BillingID NUMBER(10) PRIMARY KEY,
   PatientID  NUMBER(10),
   TotalAmount NUMBER(10),
   BillingDate DATE,
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)
);
```

PrescribedMedications table:

```
CREATE TABLE PrescribedMedications (
   PrescribedMedicationID NUMBER(10) PRIMARY KEY,
   PrescriptionID NUMBER(10),
   Price NUMBER(10),
   Quantity NUMBER(10),
   Instructions VARCHAR2(200),
   FOREIGN KEY (PrescriptionID) REFERENCES Prescriptions(PrescriptionID)
);
```

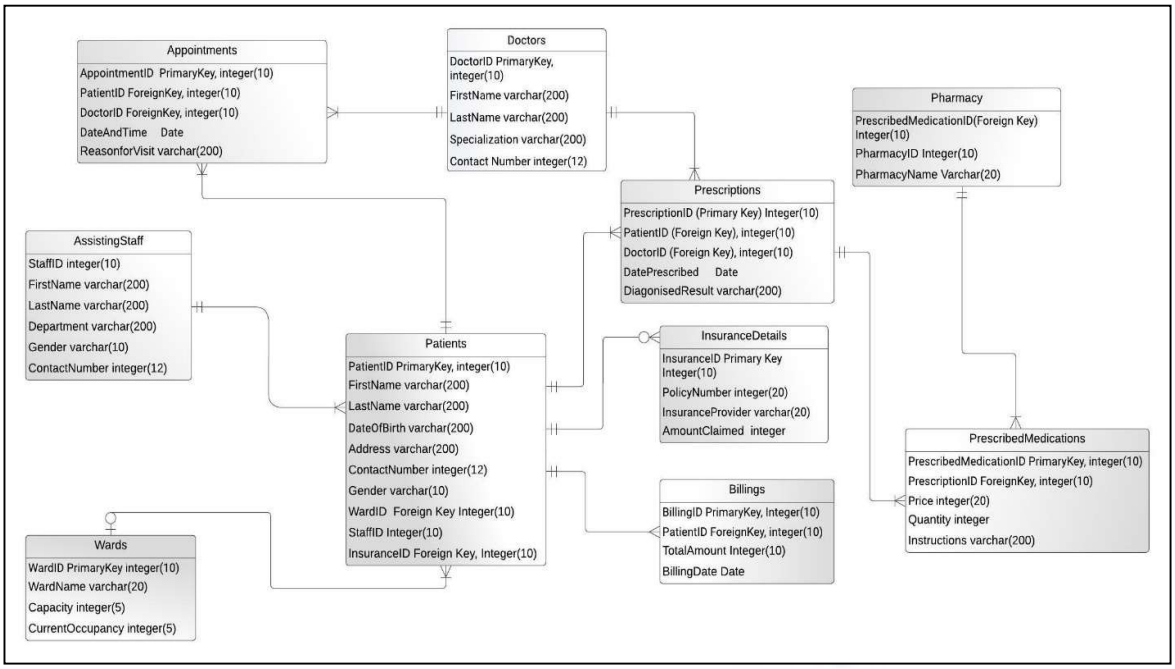
Pharmacy table:

```
CREATE TABLE Pharmacy (
   PrescribedMedicationID  NUMBER(10),
   PharmacyID NUMBER(10),
   PharmacyName VARCHAR2(30),
```

```
    FOREIGN KEY (PrescribedMedicationID) REFERENCES
PrescribedMedications(PrescribedMedicationID)
);
```

2. Physical ERD for Hospital and Pharmacy Management system

Click here→Physical ERD



GP4- Populated Data & Query Output

- **Creation of the Tables:**

```
-- Create the Doctors table
CREATE TABLE Doctors (
    DoctorID NUMBER(10) PRIMARY KEY,
    FirstName VARCHAR2(200),
    LastName VARCHAR2(200),
    Specialization VARCHAR2(200),
    ContactNumber NUMBER(10)
)
```

```
Table created.
```

```
-- Create the AssistingStaff table
CREATE TABLE AssistingStaff (
    StaffID NUMBER(10) PRIMARY KEY,
    FirstName VARCHAR2(200),
    LastName VARCHAR2(200),
    Department VARCHAR2(200),
    Gender VARCHAR2(10),
    ContactNumber NUMBER(10)
)
```

```
Table created.
```

```
-- Create the Wards table
CREATE TABLE Wards (
    WardID NUMBER(10) PRIMARY KEY,
    WardName VARCHAR2(20),
    Capacity NUMBER(10),
    CurrentOccupancy NUMBER(10)
)
```

```
Table created.
```

```
CREATE TABLE InsuranceDetails (
    InsuranceID NUMBER(10) PRIMARY KEY,
    PolicyNumber NUMBER(10),
    InsuranceProvider VARCHAR2(20),
    AmountClaimed NUMBER(10),
    PatientID NUMBER(10)
)

Table created.
```

```sql
-- Create the Patients table
CREATE TABLE Patients (
    PatientID NUMBER(10) PRIMARY KEY,
    FirstName VARCHAR2(200),
    LastName VARCHAR2(200),
    DateOfBirth DATE,
    Address VARCHAR2(200),
    ContactNumber NUMBER(10),
    Gender VARCHAR2(10),
    InsuranceID NUMBER(10),
    StaffID NUMBER(10),
    WardID NUMBER(10),
    FOREIGN KEY (InsuranceID) REFERENCES InsuranceDetails(InsuranceID),
    FOREIGN KEY (StaffID) REFERENCES AssistingStaff(StaffID),
    FOREIGN KEY (WardID) REFERENCES Wards(WardID)
)
```

```
Table created.
```

```sql
-- Create the Appointments table
CREATE TABLE Appointments (
    AppointmentID NUMBER(10) PRIMARY KEY,
    PatientID NUMBER(10),
    DoctorID NUMBER(10),
    DateAndTime TIMESTAMP,
    ReasonForVisit VARCHAR2(200),
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
)
```

```
Table created.
```

```sql
-- Create the Prescriptions table
CREATE TABLE Prescriptions (
    PrescriptionID NUMBER(10) PRIMARY KEY,
    PatientID NUMBER(10),
    DoctorID NUMBER(10),
    DatePrescribed DATE,
    DiagnosedResult VARCHAR2(200),
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
)
```

```
Table created.
```

```sql
-- Create the Billings table
CREATE TABLE Billings (
    BillingID NUMBER(10) PRIMARY KEY,
    PatientID NUMBER(10),
    TotalAmount NUMBER(10),
    BillingDate DATE,
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)
)
```

```
Table created.
```

```sql
-- Create the PrescribedMedications table
CREATE TABLE PrescribedMedications (
    PrescribedMedicationID NUMBER(10) PRIMARY KEY,
    PrescriptionID NUMBER(10),
    Price NUMBER(10),
    Quantity NUMBER(10),
    Instructions VARCHAR2(200),
    FOREIGN KEY (PrescriptionID) REFERENCES Prescriptions(PrescriptionID)
)
```

```
Table created.
```

```sql
-- Create the Pharmacy table
CREATE TABLE Pharmacy (
    PrescribedMedicationID NUMBER(10),
    PharmacyID NUMBER(10),
    PharmacyName VARCHAR2(30),
    FOREIGN KEY (PrescribedMedicationID) REFERENCES PrescribedMedications(PrescribedMedicationID)
)
```

```
Table created.
```

- **Inserting Values to the tables:**

```
-- Insert into the Doctors table
INSERT ALL
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2001, 'Michael', 'Smith', 'Cardiologist', '+1234567890')
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2002, 'Jennifer', 'Johnson', 'Dermatologist', '+1987654321')
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2003, 'William', 'Williams', 'Endocrinologist', '+1122334455')
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2004, 'Jessica', 'Jones', 'Gastroenterologist', '+1555098765')
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2005, 'Christopher', 'Brown', 'Hematologist', '+1357924680')
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2006, 'Amanda', 'Davis', 'Immunologist', '+2468135790')
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2007, 'Matthew', 'Miller', 'Nephrologist', '+1122334466')
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2008, 'Emily', 'Wilson', 'Neurologist', '+1987654322')
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2009, 'Daniel', 'Anderson', 'Oncologist', '+1555098777')
  INTO Doctors (DoctorID, FirstName, LastName, Specialization, ContactNumber)
  VALUES (2010, 'Hannah', 'Martinez', 'Orthopedist', '+1357924699')
  SELECT 1 FROM DUAL

10 row(s) inserted.
```

```
-- Insert into the AssistingStaff table
INSERT ALL
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5001, 'John', 'Doe', 'Cardiology', 'Male', 1234567890)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5002, 'Jane', 'Smith', 'Neurology', 'Female', 1987654321)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5003, 'Michael', 'Johnson', 'Oncology', 'Male', 1122334455)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5004, 'Emily', 'Williams', 'Pediatrics', 'Female', 1555098765)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5005, 'William', 'Brown', 'Emergency', 'Male', 1357924680)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5006, 'Sarah', 'Miller', 'Surgery', 'Female', 2468135790)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5007, 'David', 'Wilson', 'Orthopedics', 'Male', 1122334455)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5008, 'Jessica', 'Garcia', 'Radiology', 'Female', 1987654321)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5009, 'Andrew', 'Lee', 'Dermatology', 'Male', 1555098765)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5010, 'Amanda', 'Martinez', 'Gastroenterology', 'Female', 135792468
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5011, 'James', 'Robinson', 'Urology', 'Male', 2468135790)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5012, 'Lauren', 'Hill', 'Ophthalmology', 'Female', 1234567890)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5013, 'Ryan', 'Cooper', 'Psychiatry', 'Male', 1987654321)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5014, 'Sophia', 'Cox', 'ENT', 'Female', 1122334455)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5015, 'Daniel', 'Howard', 'Endocrinology', 'Male', 1555098765)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5016, 'Olivia', 'Bell', 'Nephrology', 'Female', 1357924680)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5017, 'Matthew', 'Ward', 'Hematology', 'Male', 2468135790)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5018, 'Isabella', 'Cole', 'Pulmonology', 'Female', 1234567890)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5019, 'Christopher', 'Diaz', 'Rheumatology', 'Male', 1987654321)
  INTO AssistingStaff (StaffID, FirstName, LastName, Department, Gender, ContactNumber) VALUES (5020, 'Samantha', 'Richardson', 'Oncology', 'Female', 1122334455)
SELECT 1 FROM DUAL

10 row(s) inserted.
```

```
-- Insert into the Wards table
INSERT ALL
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (101, 'General', 50, 30)
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (102, 'Emergency', 20, 15)
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (103, 'ICU', 10, 8)
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (104, 'Pediatrics', 30, 20)
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (105, 'Surgery', 25, 18)
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (106, 'Maternity', 15, 10)
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (107, 'Oncology', 20, 12)
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (108, 'Cardiology', 15, 9)
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (109, 'Orthopedics', 25, 20)
  INTO Wards (WardID, WardName, Capacity, CurrentOccupancy)
  VALUES (110, 'Neurology', 15, 11)
  SELECT 1 FROM DUAL

10 row(s) inserted.
```

```
-- Insert into the Patients table
INSERT ALL
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1001, 'John', 'Doe', TO_DATE('1985-05-12', 'YYYY-MM-DD'), '123 Main St, CityA, CountryA', 1234567890, 'Male', 2001, 5001, 101)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1002, 'Jane', 'Smith', TO_DATE('1990-07-18', 'YYYY-MM-DD'), '456 Elm St, CityB, CountryB', 1987654321, 'Female', 2002, 5002, 102)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1003, 'Michael', 'Johnson', TO_DATE('1982-12-03', 'YYYY-MM-DD'), '789 Oak Ave, CityC, CountryC', 1122334455, 'Male', 2003, 5003, 103)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1004, 'Emily', 'Davis', TO_DATE('1988-09-25', 'YYYY-MM-DD'), '234 Maple St, CityD, CountryD', 1555098765, 'Female', 2004, 5004, 104)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1005, 'William', 'Wilson', TO_DATE('1995-03-08', 'YYYY-MM-DD'), '567 Pine Rd, CityE, CountryE', 1357924680, 'Male', 2005, 5005, 105)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1006, 'Sarah', 'Brown', TO_DATE('1980-11-15', 'YYYY-MM-DD'), '890 Cedar Ave, CityF, CountryF', 2468135790, 'Female', 2006, 5006, 106)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1007, 'David', 'Garcia', TO_DATE('1987-06-30', 'YYYY-MM-DD'), '123 Birch St, CityG, CountryG', 1122334455, 'Male', 2007, 5007, 107)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1008, 'Jessica', 'Lee', TO_DATE('1993-01-20', 'YYYY-MM-DD'), '456 Spruce Ave, CityH, CountryH', 1987654321, 'Female', 2008, 5008, 108)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1009, 'Andrew', 'Martinez', TO_DATE('1984-08-14', 'YYYY-MM-DD'), '789 Oak St, CityI, CountryI', 1555098765, 'Male', 2009, 5009, 109)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1010, 'Amanda', 'Robinson', TO_DATE('1991-04-29', 'YYYY-MM-DD'), '234 Pine Ave, CityJ, CountryJ', 1357924680, 'Female', 2010, 5010, 110)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1011, 'James', 'Wright', TO_DATE('1986-10-11', 'YYYY-MM-DD'), '567 Cedar Rd, CityK, CountryK', 2468135790, 'Male', 2011, 5011, 101)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1012, 'Lauren', 'Hill', TO_DATE('1983-07-23', 'YYYY-MM-DD'), '890 Birch Ave, CityL, CountryL', 1234567890, 'Female', 2012, 5012, 102)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1013, 'Ryan', 'Cooper', TO_DATE('1989-02-17', 'YYYY-MM-DD'), '123 Spruce St, CityM, CountryM', 1987654321, 'Male', 2013, 5013, 103)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1014, 'Sophia', 'Cox', TO_DATE('1994-06-05', 'YYYY-MM-DD'), '456 Elm Ave, CityN, CountryN', 1122334455, 'Female', 2014, 5014, 104)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1015, 'Daniel', 'Howard', TO_DATE('1981-09-07', 'YYYY-MM-DD'), '789 Oak Ave, CityO, CountryO', 1555098765, 'Male', 2015, 5015, 105)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1016, 'Olivia', 'Bell', TO_DATE('1988-12-01', 'YYYY-MM-DD'), '234 Maple St, CityP, CountryP', 1357924680, 'Female', 2016, 5016, 106)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1017, 'Matthew', 'Ward', TO_DATE('1993-04-14', 'YYYY-MM-DD'), '567 Cedar Ave, CityQ, CountryQ', 2468135790, 'Male', 2017, 5017, 107)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1018, 'Isabella', 'Cole', TO_DATE('1984-11-27', 'YYYY-MM-DD'), '890 Birch Rd, CityR, CountryR', 1234567890, 'Female', 2018, 5018, 108)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1019, 'Christopher', 'Diaz', TO_DATE('1990-08-19', 'YYYY-MM-DD'), '123 Pine Ave, CityS, CountryS', 1987654321, 'Male', 2019, 5019, 109)
INTO Patients (PatientID, FirstName, LastName, DateOfBirth, Address, ContactNumber, Gender, InsuranceID, StaffID, WardID)
    VALUES (1020, 'Samantha', 'Richardson', TO_DATE('1985-02-02', 'YYYY-MM-DD'), '456 Cedar Rd, CityT, CountryT', 1122334455, 'Female', 2020, 5020, 110)
SELECT 1 FROM DUAL
```

```
-- Insert into the InsuranceDetails table
INSERT ALL
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2001, 12345, 'ABC Insurance', 5000, 1001)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2002, 23456, 'XYZ Insurance', 7000, 1002)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2003, 34567, 'DEF Insurance', 4500, 1003)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2004, 45678, 'PQR Insurance', 6000, 1004)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2005, 56789, 'MNO Insurance', 5500, 1005)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2006, 67890, 'GHI Insurance', 4000, 1006)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2007, 78901, 'JKL Insurance', 6500, 1007)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2008, 89012, 'STU Insurance', 3000, 1008)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2009, 90123, 'VWX Insurance', 4500, 1009)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2010, 11223, 'YZA Insurance', 5500, 1010)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2011, 22334, 'BCD Insurance', 4800, 1011)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2012, 33445, 'EFG Insurance', 5200, 1012)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2013, 44556, 'HIJ Insurance', 4200, 1013)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2014, 55667, 'KLM Insurance', 6700, 1014)
INTO InsuranceDetails (InsuranceID, PolicyNumber, InsuranceProvider, AmountClaimed, PatientID) VALUES (2015, 66778, 'NOP Insurance', 3800, 1015)
SELECT 1 FROM DUAL
```

```
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3016, 1016, 2006, TO_TIMESTAMP('2023-11-16 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Fever')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3017, 1017, 2007, TO_TIMESTAMP('2023-11-17 08:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Stomachache')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3018, 1018, 2008, TO_TIMESTAMP('2023-11-18 09:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Headache')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3019, 1019, 2009, TO_TIMESTAMP('2023-11-19 10:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Back pain')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3020, 1020, 2010, TO_TIMESTAMP('2023-11-20 11:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'General check-up')
SELECT 1 FROM DUAL
```

```
-- Insert into the Prescriptions table
INSERT ALL
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4001, 1001, 2001, TO_DATE('2023-11-01', 'YYYY-MM-DD'), 'Fever')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4002, 1002, 2002, TO_DATE('2023-11-02', 'YYYY-MM-DD'), 'Stomachache')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4003, 1003, 2003, TO_DATE('2023-11-03', 'YYYY-MM-DD'), 'Headache')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4004, 1004, 2004, TO_DATE('2023-11-04', 'YYYY-MM-DD'), 'Back pain')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4005, 1005, 2005, TO_DATE('2023-11-05', 'YYYY-MM-DD'), 'General check-up')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4006, 1006, 2006, TO_DATE('2023-11-06', 'YYYY-MM-DD'), 'Cough')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4007, 1007, 2007, TO_DATE('2023-11-07', 'YYYY-MM-DD'), 'Fever')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4008, 1008, 2008, TO_DATE('2023-11-08', 'YYYY-MM-DD'), 'Stomachache')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4009, 1009, 2009, TO_DATE('2023-11-09', 'YYYY-MM-DD'), 'Headache')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4010, 1010, 2010, TO_DATE('2023-11-10', 'YYYY-MM-DD'), 'Back pain')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4011, 1011, 2001, TO_DATE('2023-11-11', 'YYYY-MM-DD'), 'General check-up')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4012, 1012, 2002, TO_DATE('2023-11-12', 'YYYY-MM-DD'), 'Cough')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4013, 1013, 2003, TO_DATE('2023-11-13', 'YYYY-MM-DD'), 'Fever')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4014, 1014, 2004, TO_DATE('2023-11-14', 'YYYY-MM-DD'), 'Stomachache')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
    VALUES (4015, 1015, 2005, TO_DATE('2023-11-15', 'YYYY-MM-DD'), 'Headache')
INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
```

```
 INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
     VALUES (4016, 1016, 2006, TO_DATE('2023-11-16', 'YYYY-MM-DD'), 'Back pain')
 INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
     VALUES (4017, 1017, 2007, TO_DATE('2023-11-17', 'YYYY-MM-DD'), 'General check-up')
 INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
     VALUES (4018, 1018, 2008, TO_DATE('2023-11-18', 'YYYY-MM-DD'), 'Cough')
 INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
     VALUES (4019, 1019, 2009, TO_DATE('2023-11-19', 'YYYY-MM-DD'), 'Fever')
 INTO Prescriptions (PrescriptionID, PatientID, DoctorID, DatePrescribed, DiagnosedResult)
     VALUES (4020, 1020, 2010, TO_DATE('2023-11-20', 'YYYY-MM-DD'), 'Stomachache')
 SELECT 1 FROM DUAL
```

```sql
-- Insert into the Billings table
INSERT ALL
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5001, 1001, 100, TO_DATE('2023-11-01', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5002, 1002, 150, TO_DATE('2023-11-02', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5003, 1003, 200, TO_DATE('2023-11-03', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5004, 1004, 250, TO_DATE('2023-11-04', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5005, 1005, 300, TO_DATE('2023-11-05', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5006, 1006, 350, TO_DATE('2023-11-06', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5007, 1007, 400, TO_DATE('2023-11-07', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5008, 1008, 450, TO_DATE('2023-11-08', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5009, 1009, 500, TO_DATE('2023-11-09', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5010, 1010, 550, TO_DATE('2023-11-10', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5011, 1011, 600, TO_DATE('2023-11-11', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5012, 1012, 650, TO_DATE('2023-11-12', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5013, 1013, 700, TO_DATE('2023-11-13', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5014, 1014, 750, TO_DATE('2023-11-14', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5015, 1015, 800, TO_DATE('2023-11-15', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5016, 1016, 850, TO_DATE('2023-11-16', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5017, 1017, 900, TO_DATE('2023-11-17', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5018, 1018, 950, TO_DATE('2023-11-18', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5019, 1019, 1000, TO_DATE('2023-11-19', 'YYYY-MM-DD'))
 INTO Billings (BillingID, PatientID, TotalAmount, BillingDate) VALUES (5020, 1020, 1050, TO_DATE('2023-11-20', 'YYYY-MM-DD'))
 SELECT 1 FROM DUAL
```

```sql
-- Insert into the PrescribedMedications table
INSERT ALL
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6001, 4001, 10, 2, 'Take twice daily')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6002, 4002, 15, 1, 'Take after meals')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6003, 4003, 20, 3, 'Take as needed')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6004, 4004, 25, 2, 'Take with water')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6005, 4005, 30, 1, 'Take after meals')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6006, 4006, 35, 2, 'Take as directed')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6007, 4007, 40, 1, 'Take after meals')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6008, 4008, 45, 3, 'Take twice daily')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6009, 4009, 50, 2, 'Take with water')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6010, 4010, 55, 1, 'Take as needed')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6011, 4011, 60, 2, 'Take as directed')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6012, 4012, 65, 1, 'Take after meals')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6013, 4013, 70, 3, 'Take with water')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6014, 4014, 75, 2, 'Take twice daily')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6015, 4015, 80, 1, 'Take as needed')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6016, 4016, 85, 2, 'Take as directed')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6017, 4017, 90, 1, 'Take after meals')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6018, 4018, 95, 3, 'Take with water')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6019, 4019, 100, 2, 'Take twice daily')
 INTO PrescribedMedications (PrescribedMedicationID, PrescriptionID, Price, Quantity, Instructions) VALUES (6020, 4020, 105, 1, 'Take as needed')
 SELECT 1 FROM DUAL
```

```sql
INSERT ALL
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6001, 7001, 'ABC Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6002, 7002, 'XYZ Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6003, 7003, 'MediLife Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6004, 7004, 'Good Health Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6005, 7005, 'Wellness Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6006, 7006, 'Cure Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6007, 7007, 'LifeCare Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6008, 7008, 'Helping Hands Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6009, 7009, 'QuickCure Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6010, 7010, 'Aid Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6011, 7011, 'SaveLife Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6012, 7012, 'MedicAid Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6013, 7013, 'HealWell Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6014, 7014, 'WellBeing Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6015, 7015, 'EasyCure Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6016, 7016, 'QuickAid Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6017, 7017, 'FirstAid Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6018, 7018, 'GoodCare Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6019, 7019, 'MediCure Pharmacy')
 INTO Pharmacy (PrescribedMedicationID, PharmacyID, PharmacyName) VALUES (6020, 7020, 'LifeCure Pharmacy')
SELECT 1 FROM DUAL


20 row(s) inserted.
```

```
-- Insert into the Appointments table
INSERT ALL
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3021, 1001, 2001, TO_TIMESTAMP('2023-11-01 08:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'General check-up')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3022, 1002, 2001, TO_TIMESTAMP('2023-11-02 09:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Fever')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3023, 1003, 2001, TO_TIMESTAMP('2023-11-03 10:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Stomachache')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3024, 1004, 2004, TO_TIMESTAMP('2023-11-04 11:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Headache')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3025, 1005, 2004, TO_TIMESTAMP('2023-11-05 13:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Back pain')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3026, 1006, 2004, TO_TIMESTAMP('2023-11-06 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'General check-up')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3027, 1007, 2007, TO_TIMESTAMP('2023-11-07 15:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Cough')
INTO Appointments (AppointmentID, PatientID, DoctorID, DateAndTime, ReasonForVisit)
    VALUES (3028, 1008, 2008, TO_TIMESTAMP('2023-11-08 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'General check-up')
SELECT 1 FROM DUAL
8 row(s) inserted.
```

- **Queries and Outputs:**

1) Using Case Manipulation Functions, Working with Dates, Conversion Functions, Displaying Data from Multiple Tables, and Aggregating Data Using Group Functions.

```
SELECT UPPER(Doctors.FirstName) AS Uppercase_FirstName,
   TO_CHAR(Appointments.DateAndTime, 'DD-MON-YYYY')
   AS Formatted_Date,
   COUNT(DISTINCT Patients.PatientID) AS Total_Patients,
   MIN(Billings.TotalAmount) AS Min_Billing_Amount,
   AVG(PrescribedMedications.Price) AS Avg_Medication_Price
FROM Appointments
JOIN Doctors ON Appointments.DoctorID = Doctors.DoctorID
JOIN Patients ON Appointments.PatientID = Patients.PatientID
LEFT OUTER JOIN Billings ON Patients.PatientID = Billings.PatientID
LEFT OUTER JOIN Prescriptions ON Patients.PatientID =
Prescriptions.PatientID
LEFT OUTER JOIN PrescribedMedications ON
Prescriptions.PrescriptionID = PrescribedMedications.PrescriptionID
GROUP BY UPPER(Doctors.FirstName),
TO_CHAR(Appointments.DateAndTime,
'DD-MON-YYYY');
```

UPPERCASE_FIRSTNAME	FORMATTED_DATE	TOTAL_PATIENTS	MIN_BILLING_AMOUNT	AVG_MEDICATION_PRICE
MICHAEL	01-NOV-2023	1	100	10
WILLIAM	03-NOV-2023	1	200	20
JESSICA	14-NOV-2023	1	750	75
AMANDA	16-NOV-2023	1	850	85
WILLIAM	13-NOV-2023	1	700	70
HANNAH	20-NOV-2023	1	1050	105
JENNIFER	02-NOV-2023	1	150	15
HANNAH	10-NOV-2023	1	550	55
EMILY	18-NOV-2023	1	950	95
MATTHEW	07-NOV-2023	1	400	40
CHRISTOPHER	15-NOV-2023	1	800	80
EMILY	08-NOV-2023	1	450	45
AMANDA	06-NOV-2023	1	350	35
DANIEL	09-NOV-2023	1	500	50
DANIEL	09-NOV-2023	1	500	50
MATTHEW	17-NOV-2023	1	900	90
MICHAEL	11-NOV-2023	1	600	60
CHRISTOPHER	05-NOV-2023	1	300	30
DANIEL	19-NOV-2023	1	1000	100
JESSICA	04-NOV-2023	1	250	25
JENNIFER	12-NOV-2023	1	650	65

Download CSV

20 rows selected.

2) The concepts used in this query include concatenation, arithmetic operations, the NVL function for handling null values, and date functions for calculating age and retrieving the current date.
Here, we calculate the age of patients based on their "DateOfBirth" and find doctors who have more than 3 appointments.

```
SELECT PatientID,
  FirstName || ' ' || LastName AS Full_Name,
  DateOfBirth, TRUNC((SYSDATE - DateOfBirth) / 365) AS Age,
  NVL(Address, 'Unknown') AS Address
FROM Patients
ORDER BY Age DESC;
```

PATIENTID	FULL_NAME	DATEOFBIRTH	AGE	ADDRESS
1006	Sarah Brown	15-NOV-80	43	890 Cedar Ave, CityF, CountryF
1015	Daniel Howard	07-SEP-81	42	789 Oak Ave, CityO, CountryO
1003	Michael Johnson	03-DEC-82	40	789 Oak Ave, CityC, CountryC
1012	Lauren Hill	23-JUL-83	40	890 Birch Ave, CityL, CountryL
1009	Andrew Martinez	14-AUG-84	39	789 Oak St, CityI, CountryI
1020	Samantha Richardson	02-FEB-85	38	456 Cedar Rd, CityT, CountryT
1001	John Doe	12-MAY-85	38	123 Main St, CityA, CountryA
1018	Isabella Cole	27-NOV-84	38	890 Birch Rd, CityR, CountryR
1011	James Wright	11-OCT-86	37	567 Cedar Rd, CityK, CountryK
1007	David Garcia	30-JUN-87	36	123 Birch St, CityG, CountryG
1004	Emily Davis	25-SEP-88	35	234 Maple St, CityD, CountryD
1013	Ryan Cooper	17-FEB-89	34	123 Spruce St, CityM, CountryM
1016	Olivia Bell	01-DEC-88	34	234 Maple St, CityP, CountryP
1002	Jane Smith	18-JUL-90	33	456 Elm St, CityB, CountryB

1002	Jane Smith	18-JUL-90	33	456 Elm St, CityB, CountryB
1019	Christopher Diaz	19-AUG-90	33	123 Pine Ave, CityS, CountryS
1010	Amanda Robinson	29-APR-91	32	234 Pine Ave, CityJ, CountryJ
1008	Jessica Lee	20-JAN-93	30	456 Spruce Ave, CityH, CountryH
1017	Matthew Ward	14-APR-93	30	567 Cedar Ave, CityQ, CountryQ
1014	Sophia Cox	05-JUN-94	29	456 Elm Ave, CityN, CountryN
1005	William Wilson	08-MAR-95	28	567 Pine Rd, CityE, CountryE

Download CSV

20 rows selected.

3) The functionalities used in this query include the NULLIF function, aggregation functions (MIN, MAX, COUNT), a window function (RANK), and LEFT OUTER JOIN operations to combine and analyze data from multiple tables related to doctors, appointments, prescriptions, and patients.

```
SELECT Doctors.DoctorID,
    NULLIF(Doctors.LastName, 'Unknown') AS Last_Name,
    MIN(Appointments.DateAndTime) AS First_Appointment_Date,
    MAX(Prescriptions.DatePrescribed) AS Last_Prescription_Date,
    RANK() OVER (ORDER BY MAX(Doctors.Specialization))
    AS Specialization_Rank,
    COUNT(DISTINCT Patients.PatientID) AS Total_Patients
FROM Doctors
LEFT OUTER JOIN Appointments ON Doctors.DoctorID =
Appointments.DoctorID
LEFT OUTER JOIN Prescriptions ON Doctors.DoctorID =
Prescriptions.DoctorID
LEFT OUTER JOIN Patients ON Appointments.PatientID =
Patients.PatientID
GROUP BY Doctors.DoctorID, NULLIF(Doctors.LastName, 'Unknown');
```

DOCTORID	LAST_NAME	FIRST_APPOINTMENT_DATE	LAST_PRESCRIPTION_DATE	SPECIALIZATION_RANK	TOTAL_PATIENTS
2001	Smith	01-NOV-23 08.30.00.000000 AM	11-NOV-23	1	2
2002	Johnson	02-NOV-23 09.00.00.000000 AM	12-NOV-23	2	2
2003	Williams	03-NOV-23 10.30.00.000000 AM	13-NOV-23	3	2
2004	Jones	04-NOV-23 11.00.00.000000 AM	14-NOV-23	4	2
2005	Brown	05-NOV-23 01.30.00.000000 PM	15-NOV-23	5	2
2006	Davis	06-NOV-23 02.00.00.000000 PM	16-NOV-23	6	2
2007	Miller	07-NOV-23 03.30.00.000000 PM	17-NOV-23	7	2
2008	Wilson	08-NOV-23 04.00.00.000000 PM	18-NOV-23	8	2
2009	Anderson	09-NOV-23 08.30.00.000000 AM	19-NOV-23	9	2
2010	Martinez	10-NOV-23 09.00.00.000000 AM	20-NOV-23	10	2

Download CSV

10 rows selected.

4) The concepts used in this query include date functions (TRUNC, SYSDATE), a conversion function (TO_CHAR), an aggregation function (COUNT), and JOIN operations to combine and analyze data from multiple tables related to billings, patients, appointments, and doctors.

```
SELECT Patients.Gender,
   TRUNC(SYSDATE) AS Truncated_Current_Date,
   TO_CHAR(Billings.BillingDate, 'YYYY-MM') AS Billing_Month,
   Doctors.Specialization,
   COUNT(DISTINCT Patients.PatientID) AS Total_Patients
FROM Billings
JOIN Patients ON Billings.PatientID = Patients.PatientID
JOIN Appointments ON Patients.PatientID = Appointments.PatientID
JOIN Doctors ON  Appointments.DoctorID =  Doctors.DoctorID
GROUP BY
TRUNC(SYSDATE), Patients.Gender,
TO CHAR(Billings.BillingDate, 'YYYY-MM'), Doctors.Specialization;
```

TRUNCATED_CURRENT_DATE	GENDER	BILLING_MONTH	SPECIALIZATION	TOTAL_PATIENTS
16-NOV-23	Female	2023-11	Immunologist	2
16-NOV-23	Male	2023-11	Nephrologist	2
16-NOV-23	Male	2023-11	Hematologist	2
16-NOV-23	Male	2023-11	Cardiologist	2
16-NOV-23	Male	2023-11	Endocrinologist	2
16-NOV-23	Male	2023-11	Oncologist	2
16-NOV-23	Female	2023-11	Dermatologist	2
16-NOV-23	Female	2023-11	Neurologist	2
16-NOV-23	Female	2023-11	Orthopedist	2
16-NOV-23	Female	2023-11	Gastroenterologist	2

Download CSV

10 rows selected.

5) The concepts used in this query include aggregation functions (MAX, SUM), NULL-related functions (NVL), the modulus function (MOD), a window function (LAG), and LEFT OUTER JOIN operations to combine and analyze data from multiple tables related to patients, billings, appointments, and assisting staff.

```
SELECT Patients.PatientID, MOD(MAX(Patients.ContactNumber), 2)
   AS ContactNumber_Modulus,
   NVL(SUM(NVL(Billings.TotalAmount, 0)), 0)
   AS Total_Billings,
   MAX(AssistingStaff.Department)
   AS Department,
   LAG(MAX(Appointments.AppointmentID))
   OVER (ORDER BY    MAX(Appointments.DateAndTime))
   AS Previous_AppointmentID
FROM Patients
LEFT OUTER JOIN Billings ON Patients.PatientID = Billings.PatientID
```

```
LEFT OUTER JOIN Appointments ON Patients.PatientID =
Appointments.PatientID

LEFT OUTER JOIN AssistingStaff ON Patients.StaffID =
AssistingStaff.StaffID

GROUP BY Patients.PatientID;
```

PATIENTID	CONTACTNUMBER_MODULUS	TOTAL_BILLINGS	DEPARTMENT	PREVIOUS_APPOINTMENTID
1001	0	100	Cardiology	-
1002	1	150	Neurology	3001
1003	1	200	Oncology	3002
1004	1	250	Pediatrics	3003
1005	0	300	Emergency	3004
1006	0	350	Surgery	3005
1007	1	400	Orthopedics	3006
1008	1	450	Radiology	3007
1009	1	500	Dermatology	3008
1010	0	550	Gastroenterology	3009
1011	0	600	Urology	3010
1012	0	650	Ophthalmology	3011
1013	1	700	Psychiatry	3012
1014	1	750	ENT	3013
1015	1	800	Endocrinology	3014
1016	0	850	Nephrology	3015
1017	0	900	Hematology	3016
1018	0	950	Pulmonology	3017
1019	1	1000	Rheumatology	3018
1020	1	1050	Oncology	3019

Download CSV

20 rows selected.

6) The functionalities used in this query include the COALESCE function for handling null values, a comparison operator (>), the NOW() function to retrieve the current date and time, and an INNER JOIN operation to combine and retrieve data from the Patients and Appointments tables. Here, we retrieve patients and their appointments only if the appointment date is in the future and replace NULL values in the "Address" column with "Unknown" for patients.

```
SELECT
    P.FirstName, P.LastName,
    COALESCE(P.Address, 'Unknown') AS Address,
    A.DateAndTime
FROM Patients P
INNER JOIN Appointments A ON P.PatientID = A.PatientID
WHERE A.DateAndTime > NOW();
```

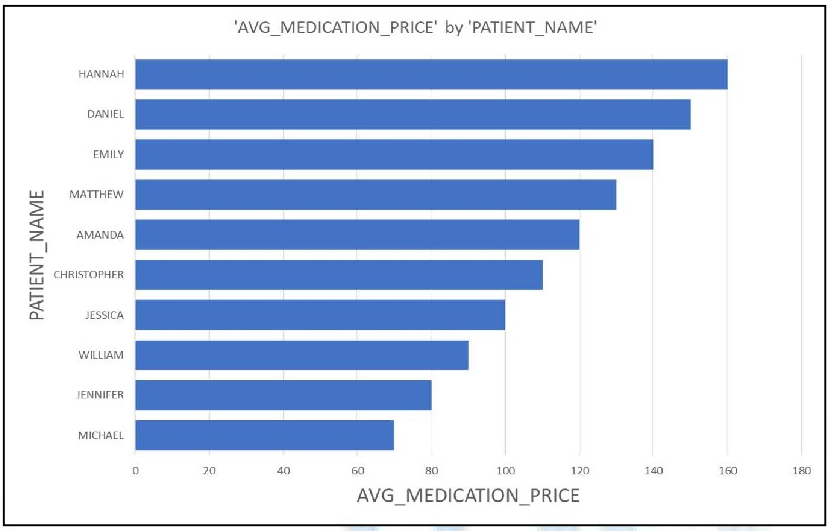
FIRSTNAME	LASTNAME	ADDRESS	DATEANDTIME
Olivia	Bell	234 Maple St, CityP, CountryP	16-NOV-23 04.00.00.000000 PM
Matthew	Ward	567 Cedar Ave, CityQ, CountryQ	17-NOV-23 08.30.00.000000 AM
Isabella	Cole	890 Birch Rd, CityR, CountryR	18-NOV-23 09.00.00.000000 AM
Christopher	Diaz	123 Pine Ave, CityS, CountryS	19-NOV-23 10.30.00.000000 AM
Samantha	Richardson	456 Cedar Rd, CityT, CountryT	20-NOV-23 11.00.00.000000 AM

Download CSV

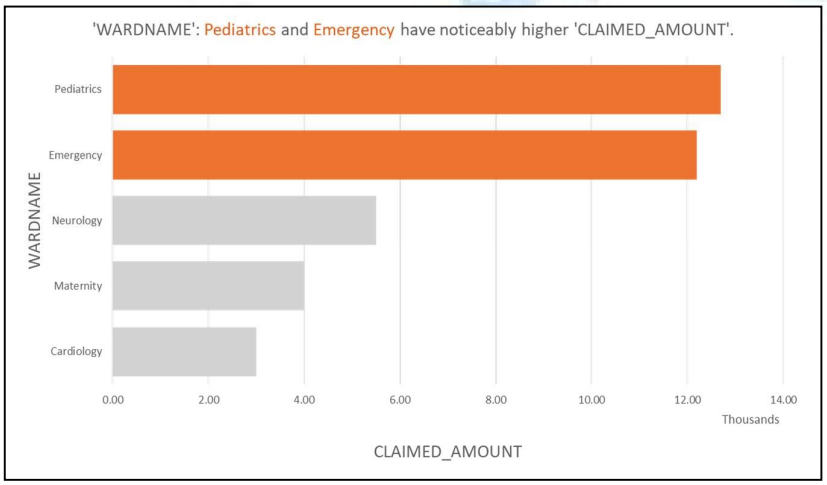
5 rows selected.

GP5- Visualizations

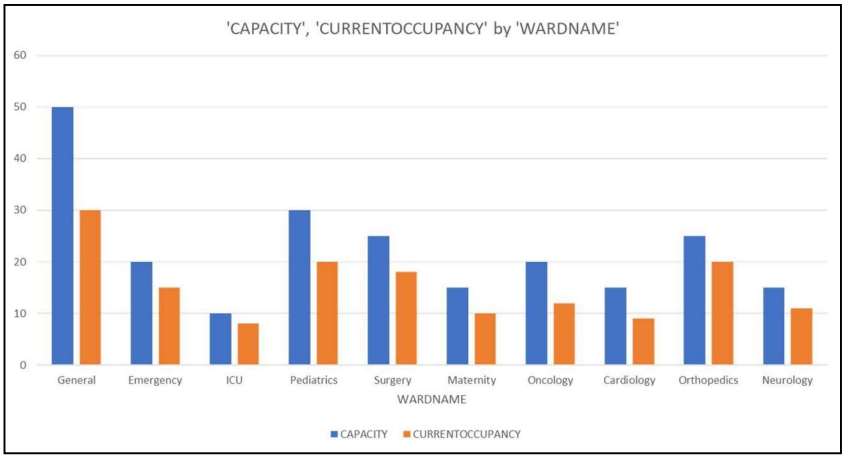
1. To analyze the average medication price of each patient.



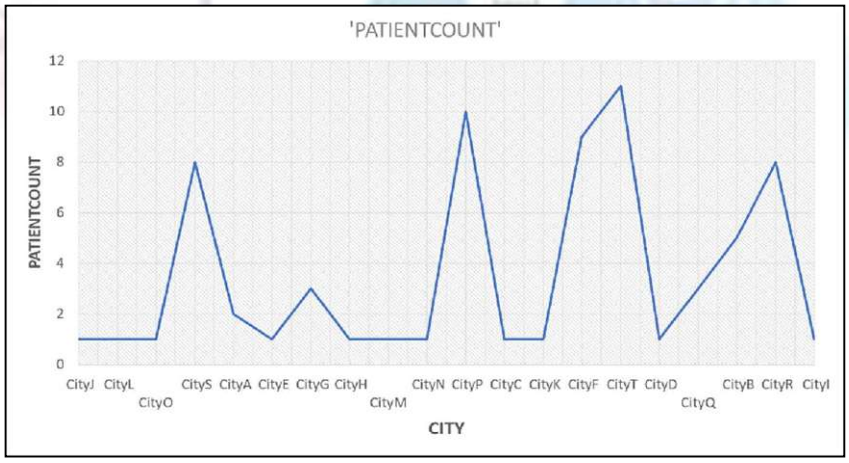
2. To analyze what is the claimed amount for each wards.



3. To analyze the actual capacity and the current occupancy of each ward.



4. Analyzing the distribution of patients across different cities in the dataset.



5. Analyzing the distribution of diagnosed results in the dataset and understanding the frequency of different diagnoses.