UNIVERSITI TEKNOLOGI MALAYSIA, JOHOR BAHRU

FACULTY OF COMPUTING

SECD2523-08 DATABASE

Phase 03: Database Conceptual Design (ERD)

Group Members	
Name	***Matric Number***
Ahmed Zaki Ahmed Mohammed Al-Gabaly	***A22EC4003***
Devinda Weerasinghe	***A22EC5005***
Abdulrahman Daney	***A22EC4001***
Sohaib Belaroussi	***A21EC4057***
Adam Amer	***A22EC0002***

Lecturer: Dr. Noor Hidayah Binti Zakaria

List of Contents

INTRODUCTION

PROJECT BACKGROUND

Ajman University Private Dental Clinic has encountered numerous issues in the rapidly evolving healthcare industry, including managing healthcare during the pandemic and dealing with internal and external payment issues. These challenges emphasize the necessity of leaving outdated pen and paper methods for recording patient data and transactions while encouraging to adopt a new approach to its healthcare management.

The COVID-19 pandemic significantly disrupted the health care system. Ajman clinic like many others, had to make immediate adjustments to guarantee the security of its staff and patients. We experience troubles in keeping track of patients, adjusting schedules, and obtaining correct information within a certain time frame.

In addition to pandemic related issues, the university clinic also encounters difficulties in processing payments. Financial hardships and administrative challenges arose from delayed reimbursements due to outdated billing and payment practices. Patients also requested for more accessible payment options pointing out the need for a flexible billing system.

Realizing these challenges, Ajman University Private Dental Clinic decided to enhance its healthcare management system. The clinic's dedication to advancement and improved community service is the source of our creative solution. Our initiative intends to transform healthcare management by transferring from manual methods to a dependable database driven approach promoting accuracy, efficiency, and unparalleled patient care.

OVERVIEW OF THE PROJECT

We are currently in the 3rd Phase of our project. In this phase, we will describe some of what we got in the previous phase, such as updating business rules, conceptual ERD, enhanced ERD, logical enhanced ERD and updating some of them at once. The final logical ERD must be created to achieve the normalization process to the Boyce-Codd Normal Form (BCNF) relation. The data dictionary will also be updated based on the normalized relations that we created to describe the main data entities, attributes, and relationships. At the end, SQL statements will be produced to transform the requirements into complete and detailed system design specifications.

DATABASE CONCEPTUAL DESIGN

UPDATED BUSINESS RULES

1. The dental clinic will open daily from 9:30 AM to 10:00 PM (12 Hours & 30 Minutes)

2. Each patient should register and schedule an appointment via the system.

3. Each appointment must have a unique Identifier, Data & Time.

4. No overlapping appointments are allowed for each patient.

5. The medical history of each patient, along with treatments and prescribed medications, should be documented.

6. All treatments and procedures must be linked to the corresponding patient and the attending dentist.

7. Medical records must comply with privacy regulations for patients.

8. Inventory Supplies and equipment should have unique identifiers, quantities, and reorder thresholds.

9. An automated reorder request should be generated when inventory levels drop below a threshold.

10. Invoices must be generated for every treatment or service provided to patients.

11. Payment records must include details such as payment method, amount paid, date of payment, and the corresponding patient.

12. Insurance claims must be processed and properly linked to patient records.

13. Patient information, including name, contact details, medical history, and insurance information, must be accurately documented.

14. Each patient must have a unique identifier for referencing in appointments, treatments, and billing processes.

15. Information about dentists, hygienists, and administrative staff, including qualifications, schedules, and certifications, must be maintained.

16. Each appointment must have a unique identification along with the patient's ID, date, time, and assigned dentist/hygienist.

17. Updates related to scheduling changes such as rescheduling or cancellations must be recorded.

18. Updates regarding treatments and any changes made to records must be recorded.

19. Changes made to medical records should be tracked for auditing purposes.

20. Information about payment methods used by patients, such as cash or card payments, must be recorded.

21. All aspects of the system, especially medical records, must comply with privacy regulations and access to patient information should be restricted to authorized personnel.

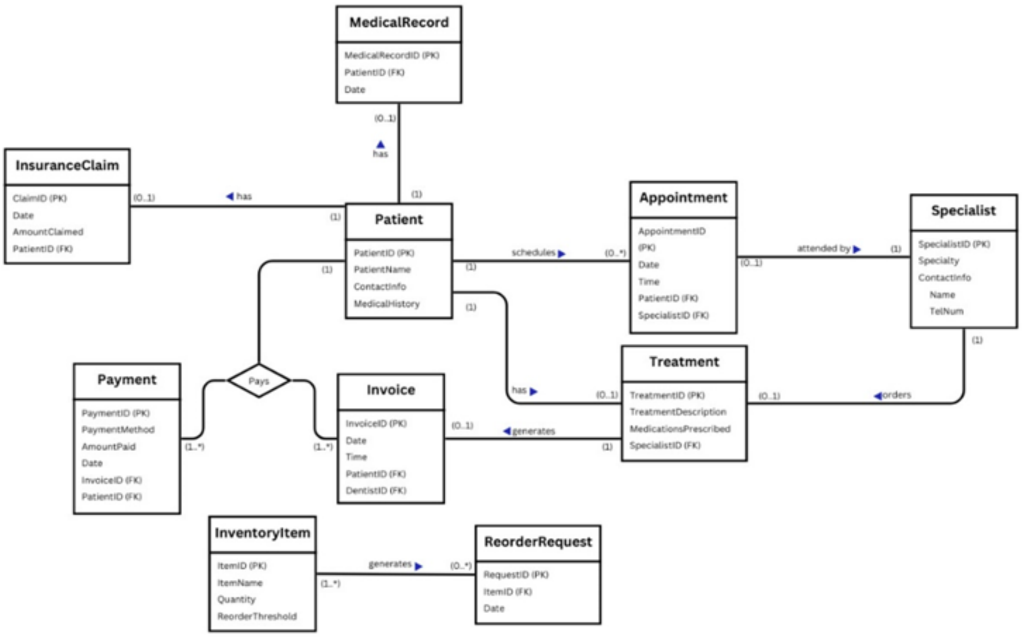
22. The system should enforce data accuracy checks to ensure the integrity of patient information and medical records.

23. Patients should receive confirmation of their scheduled appointments through appropriate communication channels.

24. Confirmation notifications should include date, time, and the assigned dentist/hygienist.

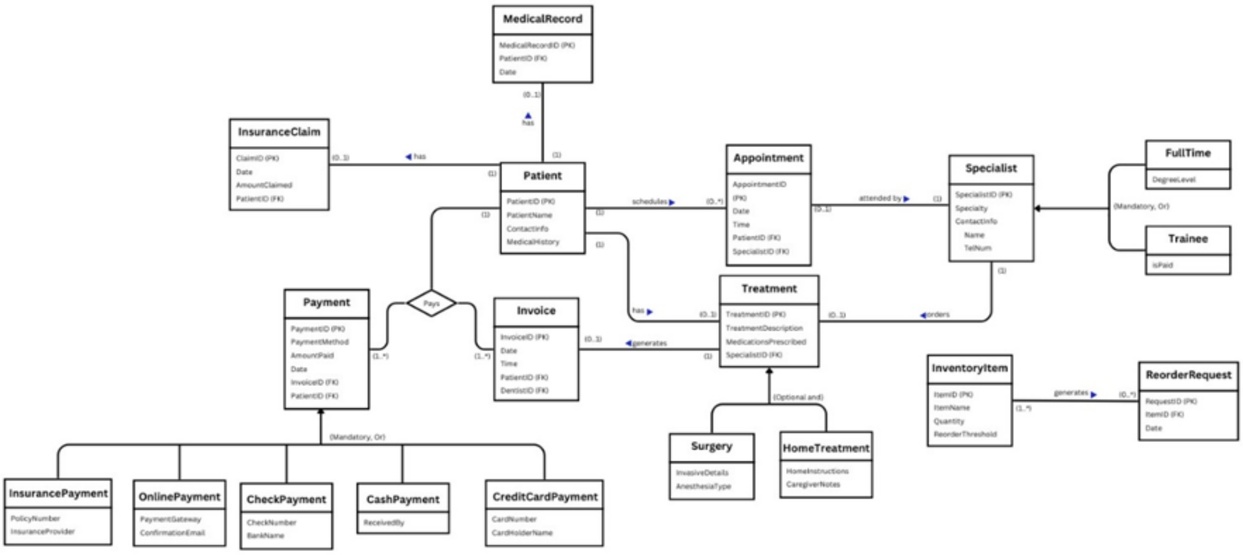
25. Access to different parts of the system should be role-based, ensuring that personnel have access only to the information necessary for their roles.

CONCEPTUAL ERD


MedicalRecord
MedicalRecordID (PK)
PatientID (FK)
Date
(0..1)
has
(1)
InsuranceClaim
ClaimID (PK)
Date
AmountClaimed
PatientID (FK)
(0..1)
has
(1)
Patient
PatientID (PK)
PatientName
ContactInfo
MedicalHistory
schedules
(1)
(0..*)
Appointment
AppointmentID
(PK)
Date
Time
PatientID (FK)
SpecialistID (FK)
(0..1)
attended by
(1)
Specialist
SpecialistID (PK)
Specialty
ContactInfo
Name
TelNum
(1)
(1)
(1)
Payment
PaymentID (PK)
PaymentMethod
AmountPaid
Date
InvoiceID (FK)
PatientID (FK)
(1..*)
Pays
(1..*)
Invoice
InvoiceID (PK)
Date
Time
PatientID (FK)
DentistID (FK)
(0..1)
generates
(1)
has
(0..1)
Treatment
TreatmentID (PK)
TreatmentDescription
MedicationsPrescribed
SpecialistID (FK)
(0..1)
orders
InventoryItem
ItemID (PK)
ItemName
Quantity
ReorderThreshold
(1..*)
generates
(0..*)
ReorderRequest
RequestID (PK)
ItemID (FK)
Date

DB Logical design

DB ENHANCED ERD



LOGICAL ERD

Step 1: Strong Entity

1. Patient(PatientID,PatientName,ContactInfo,MedicalHistory)
2. Appointment(AppointmentID, Date,Time)**
3. Specialist(SpecialistID,Specialty,Name,TelNum)
4. Treatment(TreatmentID, TreatmentDescription,MedicationsPrescribed)**
5. Invoice(InvoiceID,Date,Time)**
6. InsuranceClaim(ClaimID,Date,AmountClaim)**
7. MedicalRecord(MedicalRecordID, Date)**
8. Payment(PaymentID, PaymentMethod, AmountPaid, Date)**
9. InventoryItem(ItemID, ItemName, Quantity, ReorderThreshold)
10. ReorderRequest(RequestID, Date) **

Step 2: Weak Entity

None.

Step 3: One-to-many (1:) Binary Relationship Types*

1. Relationship: Patient schedules appointments.
   Parent → Patient

Child → Appointment

Patient(PatientID, PatientName, ContactInfo, MedicalHistory)

Appointment(AppointmentID, Date, Time, *PatientID*)

2. Relationship: Patient pays invoices.
   Parent → Patient

   Child → Invoice

   Patient(PatientID, PatientName, ContactInfo, MedicalHistory)

   Invoice(InvoiceID, Date, Time, *PatientID*)

3. Relationship: Patient makes payments.
   Parent → Patient

   Child → Payment

   Patient(PatientID, PatientName, ContactInfo, MedicalHistory)

   Payment(PaymentID, PaymentMethod, AmountPaid,Date *PatientID*)

4. Relationship: Treatment generates Invoice.
   Parent → Treatment

   Child → Invoice

   Treatment(TreatmentID, TreatmentDescription, MedicationsPrescribed, *SpecialistID*)

   Invoice(InvoiceID, Date, Time, *PatientID*)

Step 4: One-to-one (1:1) Binary Relationship Types:

1. Relationship: Patient has medical records.

Parent → Patient

Child → MedicalRecord

Patient(PatientID, PatientName, ContactInfo, MedicalHistory)

MedicalRecord(MedicalRecordID, Date, *PatientID*)

2. Relationship: Specialist attended by appointment.
   Parent → Specialist

Child → Appointment

Specialist(SpecialistID, Specialty, ContactInfo, Name, TelNum)

Appointment(AppointmentID, Date, Time, PatientID, *SpecialistID*)

3. Relationship: Specialist orders treatments.
   Parent → Specialist

   Child → Treatment

   Specialist(SpecialistID, Specialty, ContactInfo, Name, TelNum)

   Treatment(TreatmentID, TreatmentDescription, MedicationsPrescribed, *SpecialistID*)

4. Relationship: Patient has treatment.
   Parent → Patient

   Child → Treatment

   Patient(PatientID, PatientName, ContactInfo, MedicalHistory)

   Treatment(TreatmentID, TreatmentDescription, MedicationsPrescribed, SpecialistID)

Step 5: Superclass/Subclass Relationship Types:

1. Specialist
   FullTimeSpecialist(SpecialistID, DegreeLevel, Specialty, Name, TelNum)

   TraineeSpecialist(SpecialistID, isPaid, Specialty, Name, TelNum)

2. Treatment
   Treatment(TreatmentID, TreatmentDescription, MedicationsPrescribed, SpecialistID)

   SubTreatment(TreatmentID, IsSurgery, InvasiveDetails, AnesthesiaType, IsHomeTreatment, HomeInstructions, CaregiverNotes)

3. Payment
   InsurancePayment(PaymentID, PolicyNumber, InsuranceProvider)

   OnlinePayment(PaymentID, PaymentGateway, ConfirmationEmail)

   CheckPayment(PaymentID, CheckNumber, BankName)

   CashPayment(PaymentID, ReceivedBy)

   CreditCardPayment(PaymentID, CardNumber, CardHolderName)

Step 6: Many-to-many (:*) Binary Relationship Types:*

1. Relationship: InventoryItem necessitates reorder requests.
   InventoryItem(ItemID, ItemName, Quantity, ReorderThreshold)

   ReorderRequest(RequestID, Date)

   InventoryReorder(ItemID, RequestID)

Step 7: Recursive Relationship Types:

None.

Step 8: Complex Relationship Types:

1. Relationship: Patient pays the invoice using payment.
   Patient(PatientID, PatientName, ContactInfo, MedicalHistory)

   Invoice(InvoiceID, Date, Time, PatientID)

   Payment(PaymentID, PaymentMethod, AmountPaid, PatientID,InvoiceID)

   PaymentAssociation(PatientID, InvoiceID, PaymentID)

Step 9: Multi-valued attributes:

None.

Finalize:

01. Patient(**PatientID**, PatientName, ContactInfo, MedicalHistory)
02. Appointment(**AppointmentID**, Date, Time, PatientID, SpecialistID)
03. Specialist(**SpecialistID**,Specialty,Name,TelNum)
04. FullTimeSpecialist(**SpecialistID**, DegreeLevel, Specialty, Name, TelNum)
05. TraineeSpecialist(**SpecialistID**, isPaid, Specialty, Name, TelNum)
06. Treatment(**TreatmentID**, TreatmentDescription, MedicationsPrescribed, SpecialistID)
07. SubTreatment(**TreatmentID**, IsSurgery, InvasiveDetails, AnesthesiaType, IsHomeTreatment, HomeInstructions, CaregiverNotes)
08. Invoice(**InvoiceID**, Date, Time, PatientID, TreatmentID)
09. InsuranceClaim(**ClaimID**, Date, AmountClaim, PatientID)
10. MedicalRecord(**MedicalRecordID**, Date, PatientID)
11. Payment(**PaymentID**, PaymentMethod, AmountPaid, Date, InvoiceID, PatientID)
12. InventoryItem(**ItemID**, ItemName, Quantity, ReorderThreshold)
13. ReorderRequest(**RequestID**, Date, ItemID)
14. InventoryReorder(**ItemID**, **RequestID**)
15. InsurancePayment(**PaymentID**, PolicyNumber, InsuranceProvider, PaymentMethod,

AmountPaid, Date)

16. OnlinePayment(**PaymentID**, PaymentGateway, ConfirmationEmail, PaymentMethod, AmountPaid, Date)
17. CheckPayment(**PaymentID**, CheckNumber, BankName, PaymentMethod, AmountPaid, Date)
18. CashPayment(**PaymentID**, ReceivedBy, PaymentMethod, AmountPaid, Date)
19. CreditCardPayment(**PaymentID**, CardNumber, CardHolderName, PaymentMethod, AmountPaid, Date)
20. PaymentAssociation(**PatientID**, **InvoiceID**, **PaymentID**)

NORMALIZATION

1. Patient(PatientID, PatientName, ContactInfo, MedicalHistory)
   fd1: PatientID → PatientName, ContactInfo, MedicalHistory (Primary Key)

   1NF & 2NF & 3NF & BCNF:

   Patient(**PatientID**, PatientName, ContactInfo, MedicalHistory)

   Appointment

2. Appointment(AppointmentID, Date, Time, PatientID, SpecialistID)
   fd1: AppointmentID → Date, Time, PatientID, SpecialistID (Primary Key)

   1NF & 2NF & 3NF & BCNF:

   Appointment(**AppointmentID**, Date, Time, PatientID, SpecialistID)

3. Specialist(SpecialistID,Specialty,Name,TelNum)
   fd1: SpecialistID → Specialty, Name, TelNum (Primary Key)

   1NF & 2NF & 3NF & BCNF:

   Specialist(**SpecialistID**, Specialty, Name, TelNum)

4. FullTimeSpecialist(SpecialistID, DegreeLevel, Specialty, Name, TelNum)
   fd1: SpecialistID → DegreeLevel (Primary Key)

   1NF & 2NF & 3NF & BCNF:

   FullTimeSpecialist(**SpecialistID**, DegreeLevel)

   Specialist(**SpecialistID**, Specialty, Name, TelNum)

5. TraineeSpecialist(SpecialistID, isPaid, Specialty, Name, TelNum)

fd1: SpecialistID → isPaid (Primary Key, Foreign Key)

1NF & 2NF & 3NF & BCNF:
TraineeSpecialist(**SpecialistID**, isPaid)

Specialist(**SpecialistID**, Specialty, Name, TelNum)

6. Payment(**PaymentID**, PaymentMethod, AmountPaid, Date, InvoiceID, PatientID)
   fd1: PaymentID → PaymentMethod, AmountPaid, Date, InvoiceID, PatientID (Primary Key)

   1NF & 2NF & 3NF & BCNF:

   Payment(**PaymentID**, PaymentMethod, AmountPaid, Date, InvoiceID, PatientID)

7. InsurancePayment(PaymentID, PolicyNumber, InsuranceProvider, PaymentMethod, AmountPaid, Date)
   fd1: PaymentID → PolicyNumber, InsuranceProvider (Primary Key, Foreign Key)

   1NF & 2NF & 3NF & BCNF:

   InsurancePayment(**PaymentID**, PolicyNumber, InsuranceProvider)

   Payment(**PaymentID**, PaymentMethod, AmountPaid, Date, InvoiceID, PatientID)

8. OnlinePayment(PaymentID, PaymentGateway, ConfirmationEmail, PaymentMethod, AmountPaid, Date)
   fd1: PaymentID → PaymentGateway, ConfirmationEmail (Primary Key, Foreign Key)

   1NF & 2NF & 3NF & BCNF:

   OnlinePayment(**PaymentID**, PaymentGateway, ConfirmationEmail)

   Payment(**PaymentID**, PaymentMethod, AmountPaid, Date, InvoiceID, PatientID)

9. CheckPayment(PaymentID, CheckNumber, BankName, PaymentMethod, AmountPaid, Date)
   fd1: PaymentID → CheckNumber, BankName (Primary Key, Foreign Key)

   1NF & 2NF & 3NF & BCNF:

   CheckPayment(**PaymentID**, CheckNumber, BankName)

   Payment(**PaymentID**, PaymentMethod, AmountPaid, Date, InvoiceID, PatientID)

10. CashPayment(PaymentID, ReceivedBy, PaymentMethod, AmountPaid, Date)
    fd1: PaymentID → ReceivedBy (Primary Key, Foreign Key)

    1NF & 2NF & 3NF & BCNF:

    CashPayment(**PaymentID**, ReceivedBy)

    Payment(**PaymentID**, PaymentMethod, AmountPaid, Date, InvoiceID, PatientID)

11. CreditCardPayment(PaymentID, CardNumber, CardHolderName, PaymentMethod, AmountPaid, Date)
    fd1: PaymentID → CardNumber, CardHolderName (Primary Key, Foreign Key)

    1NF & 2NF & 3NF & BCNF:

    CreditCardPayment(**PaymentID**, CardNumber, CardHolderName)

    Payment(**PaymentID**, PaymentMethod, AmountPaid, Date, InvoiceID, PatientID)

12. Treatment(TreatmentID, TreatmentDescription, MedicationsPrescribed, SpecialistID)
    fd1: TreatmentID → TreatmentDescription, MedicationsPrescribed, SpecialistID (Primary Key)

    1NF & 2NF & 3NF & BCNF:

    Treatment(**TreatmentID**, TreatmentDescription, MedicationsPrescribed, SpecialistID)

13. SubTreatment(TreatmentID, IsSurgery, InvasiveDetails, AnesthesiaType, IsHomeTreatment, HomeInstructions, CaregiverNotes)
    fd1: TreatmentID → IsSurgery, InvasiveDetails, AnesthesiaType, IsHomeTreatment, HomeInstructions, CaregiverNotes (Primary Key)

    1NF & 2NF & 3NF & BCNF:

    SubTreatment(**TreatmentID**, IsSurgery, InvasiveDetails, AnesthesiaType, IsHomeTreatment, HomeInstructions, CaregiverNotes)

14. Invoice(InvoiceID, Date, Time, PatientID, TreatmentID)
    fd1: InvoiceID → Date, Time, PatientID, TreatmentID (Primary Key)

    1NF & 2NF & 3NF & BCNF:

    Invoice(**InvoiceID**, Date, Time, PatientID, TreatmentID)

15. InsuranceClaim(ClaimID, Date, AmountClaim, PatientID)

fd1: ClaimID → Date, AmountClaim, PatientID (Primary Key)

1NF & 2NF & 3NF & BCNF:

InsuranceClaim(**ClaimID**, Date, AmountClaim, PatientID)

16. MedicalRecord(MedicalRecordID, Date, PatientID)
    fd1: MedicalRecordID → Date, PatientID (Primary Key)

    1NF & 2NF & 3NF & BCNF:

    MedicalRecord(**MedicalRecordID**, Date, PatientID)

17. InventoryItem(ItemID, ItemName, Quantity, ReorderThreshold)
    fd1: ItemID → ItemName, Quantity, ReorderThreshold (Primary Key)

    1NF & 2NF & 3NF & BCNF:

    InventoryItem(**ItemID**, ItemName, Quantity, ReorderThreshold)

18. ReorderRequest(RequestID, Date, ItemID)
    fd1: RequestID → Date, ItemID (Primary Key)

    1NF & 2NF & 3NF & BCNF:

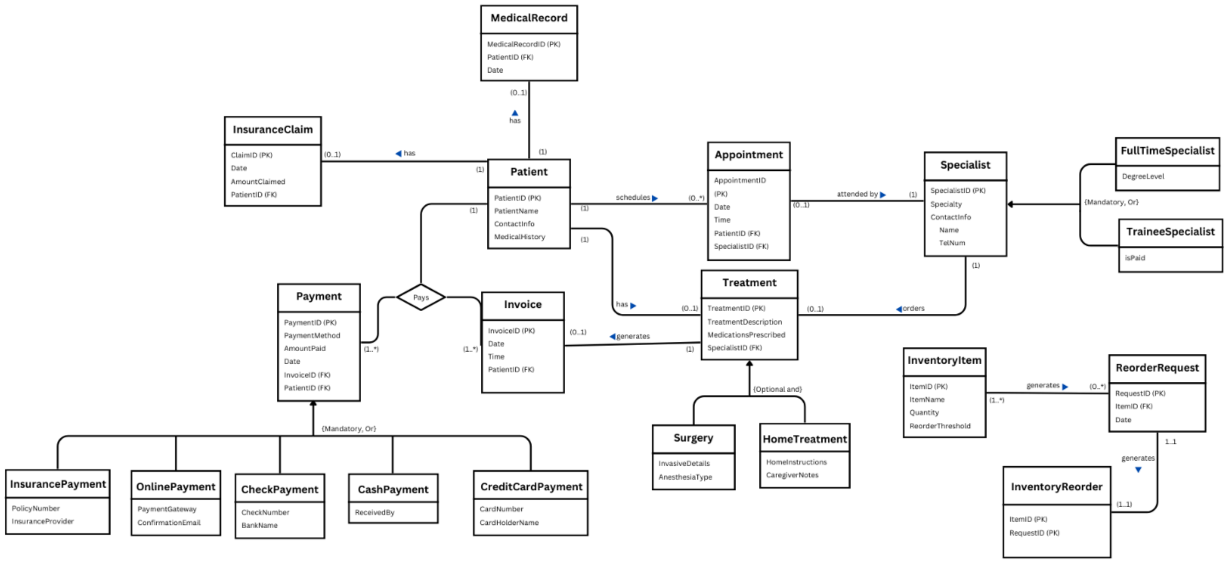
    ReorderRequest(**RequestID**, Date, ItemID)

19. InventoryReorder(ItemID, RequestID)
    fd1: (ItemID, RequestID) → (Composite Primary Key)

    1NF & 2NF & 3NF & BCNF:

    InventoryReorder(**ItemID**, **RequestID**)

LOGICAL ERD FOR BCNF



Updated Data Dictionary

DESCRIPTION OF ENTITIES

Entity	Description
Patient	**Individuals receiving medical care within the system.**
Appointment	**Scheduled sessions for patient consultations or treatments.**
Specialist	**Medical professionals with expertise in specific fields.**
FullTimeSpecialist	**Specialists employed on a full-time basis.**
TraineeSpecialist	**Specialists undergoing training or in a temporary role.**

Payment	**Financial transactions for healthcare services.**
InsurancePayment	**Payments processed through insurance coverage.**
OnlinePayment	**Payments made electronically through online channels.**
CheckPayment	**Payments made via traditional paper checks.**
CashPayment	**Payments made in physical currency.**
CreditCardPayment	**Payments made using credit card transactions.**
Treatment	**Medical procedures or interventions provided to patients.**
SubTreatment	**Subcategories or specific types of medical treatments.**
Invoice	**Document detailing services rendered and associated costs.**
InsuranceClaim	**Requests for reimbursement from insurance providers.**
MedicalRecord	**Comprehensive records of a patient's medical history.**
InventoryItem	**Items stocked within the healthcare facility.**
ReorderRequest	**Requests to replenish depleted inventory.**
InventoryReorder	**Systematic process for restocking medical inventory.**

Description of Attributes:

Patient:

Attribute	**Description**	**Data type**	**Constraint**
PatientName	**Patient's Name**	**VARCHAR2(40)**	**NOT NULL**
PatientID	**Patient's ID**	**VARCHAR2(10)**	**PRIMARY KEY**
ContactInfo	**Contact Info**	**VARCHAR2(15)**	**NOT NULL**
MedicalHistory	**Medical History**	**VARCHAR2(40)**	**NOT NULL**

Appointment:

Attribute	**Description**	**Data type**	**Constraint**
Date	**Appointment's date**	**DATE**	**NOT NULL**
Time	**Appointment's time**	**TIME**	**NOT NULL**
SpecialistID	**Specialis's ID**	**VARCHAR2(10)**	**FOREIGN KEY REFERENCE** **SPECIALIST**
PatientID	**Patient's ID**	**VARCHAR2(10)**	**FOREIGN KEY REFERENCE PATIENT**

Specialist:

Attribute	**Description**	**Data type**	**Constraint**
Name	**Specialist’s name**	**VARCHAR2(20)**	**NOT NULL**
SpecialistID	**Specialist’s ID**	**VARCHAR2(10)**	**PRIMARY KEY**
TelNum	**Specialist’s contact number**	**VARCHAR2(14)**	**NOT NULL**
Specialty	**Specialty**	**VARCHAR2(15)**	**NOT NULL**

FullTimeSpecialist:

Attribute	**Description**	**Data type**	**Constraint**
SpecialistID	**Specialist’s ID**	**VARCHAR2(10)**	**PRIMARY KEY.** **FOREIGN KEY REFERENCE** **SPECIALIST**
DegreeLevel	**Degree Level**	**VARCHAR2(15)**	**NOT NULL**

TraineeSpecialist:

Attribute	Description	Data type	Constraint
SpecialistID	Specialist’s ID	VARCHAR2(10)	PRIMARY KEY. FOREIGN KEY REFERENCE SPECIALIST
isPaid	isPaid	BOOLEAN	NOT NULL

Treatment:

Attribute	Description	Data type	Constraint
TreatmentDescription	Treatment’s Description	VARCHAR2(40)	NOT NULL
TreatmentID	Treatment’s ID	VARCHAR2(10)	PRIMARY KEY
MedicationsPrescribed	Medications Prescribed	VARCHAR2(30)	NOT NULL
SpecialistID	Specialist’s ID	VARCHAR2(10)	FOREIGN KEY REFERENCE SPECIALIST

SubTreatment:

Attribute	Description	Data type	Constraint

IsSurgery	**Surgery**	**BOOLEAN**	**NOT NULL**
TreatmentID	**Treatment’s ID**	**VARCHAR2(10)**	**PRIMARY KEY. FOREIGN KEY REFERENCE** **TREATMENT**
InvasiveDetails	**Invasive Details**	**VARCHAR2(50)**	**NOT NULL**
AnesthesiaType	**Anesthesia Type**	**VARCHAR2(10)**	**NOT NULL**
IsHomeTreatment	**Home Treatment**	**BOOLEAN**	**NOT NULL**
HomeInstructions	**Home Instructions**	**VARCHAR2(50)**	**NOT NULL**
CaregiverNotes	**Caregiver Notes**	**VARCHAR2(50)**	**NOT NULL**

Invoice:

Attribute	**Description**	**Data type**	**Constraint**
InvoiceID	**Invoice ID**	**VARCHAR2(10)**	**PRIMARY KEY**
Time	**Time of issue**	**VARCHAR2(35)**	**NOT NULL**

Date	Date of issue	DATE	NOT NULL
SpecialistID	Specialis’s ID	VARCHAR2(10)	FOREIGN KEY REFERENCE SPECIALIST
PatientID	Patient’s ID	VARCHAR2(10)	FOREIGN KEY REFERENCE PATIENT

InsuranceClaim:

Attribute	Description	Data type	Constraint
ClaimID	Claim ID	VARCHAR2(10)	PRIMARY KEY
Date	Date of issue	DATE	NOT NULL
AmountClaim	Amount Claim	VARCHAR2(35)	NOT NULL
PatientID	Patient’s ID	VARCHAR2(10)	FOREIGN KEY REFERENCE PATIENT

MedicalRecord:

Attribute	Description	Data type	Constraint
MedicalRecordID	Medical Record ID	VARCHAR2(10)	PRIMARY KEY

Date	**Date of issue**	**DATE**	**NOT NULL**
PatientID	**Patient’s ID**	**VARCHAR2(10)**	**FOREIGN KEY REFERENCE PATIENT**

Payment:

Attribute	**Description**	**Data type**	**Constraint**
PaymentID	**Payment ID**	**VARCHAR2(10)**	**PRIMARY KEY**
PaymentMethod	**Payment Method**	**VARCHAR2(15)**	**NOT NULL**
AmountPaid	**Amount Paid**	**VARCHAR2(15)**	**NOT NULL**
Date	**Date of payment**	**DATE**	**NOT NULL**
InvoiceID	**Invoice ID**	**VARCHAR2(10)**	**FOREIGN KEY REFERENCE** **INVOICE**
PatientID	**Patient’s ID**	**VARCHAR2(10)**	**FOREIGN KEY REFERENCE PATIENT**

InsurancePayment:

Attribute	**Description**	**Data type**	**Constraint**
PaymentID	**Payment ID**	**VARCHAR2(10)**	**PRIMARY KEY.** **FOREIGN KEY REFERENCE PAYMENT**

PolicyNumber	**Policy Number**	**VARCHAR2(15)**	**NOT NULL**
InsuranceProvider	**Insurance Provider**	**VARCHAR2(15)**	**NOT NULL**

OnlinePayment:

Attribute	**Description**	**Data type**	**Constraint**
PaymentID	**Payment ID**	**VARCHAR2(10)**	**PRIMARY KEY. FOREIGN KEY REFERENCE PAYMENT**
PaymentGateway	**Payment Gateway**	**VARCHAR2(15)**	**NOT NULL**
ConfirmationEmail	**Confirmation Email**	**VARCHAR2(35)**	**NOT NULL**

CheckPayment:

Attribute	**Description**	**Data type**	**Constraint**
PaymentID	**Payment ID**	**VARCHAR2(10)**	**PRIMARY KEY. FOREIGN KEY REFERENCE PAYMENT**
CheckNumber	**Check Number**	**VARCHAR2(15)**	**NOT NULL**
BankName	**Bank Name**	**VARCHAR2(35)**	**NOT NULL**

CashPayment:

Attribute	Description	Data type	Constraint
PaymentID	Payment ID	VARCHAR2(10)	PRIMARY KEY. FOREIGN KEY REFERENCE PAYMENT
ReceivedBy	Receiver's name	VARCHAR2(15)	NOT NULL

CashPayment:

Attribute	Description	Data type	Constraint
PaymentID	Payment ID	VARCHAR2(10)	PRIMARY KEY. FOREIGN KEY REFERENCE PAYMENT
CardNumber	Card Number	VARCHAR2(15)	NOT NULL
CardHolderName	Card Holder's Name	VARCHAR2(35)	NOT NULL

InventoryItem:

Attribute	Description	Data type	Constraint
ItemID	Item ID	VARCHAR2(10)	PRIMARY KEY
ItemName	Item Name	VARCHAR2(35)	NOT NULL
Quantity	Quantity	VARCHAR2(35)	NOT NULL
ReorderThreshold	Reorder Threshold	VARCHAR2(35)	NOT NULL

ReorderRequest:

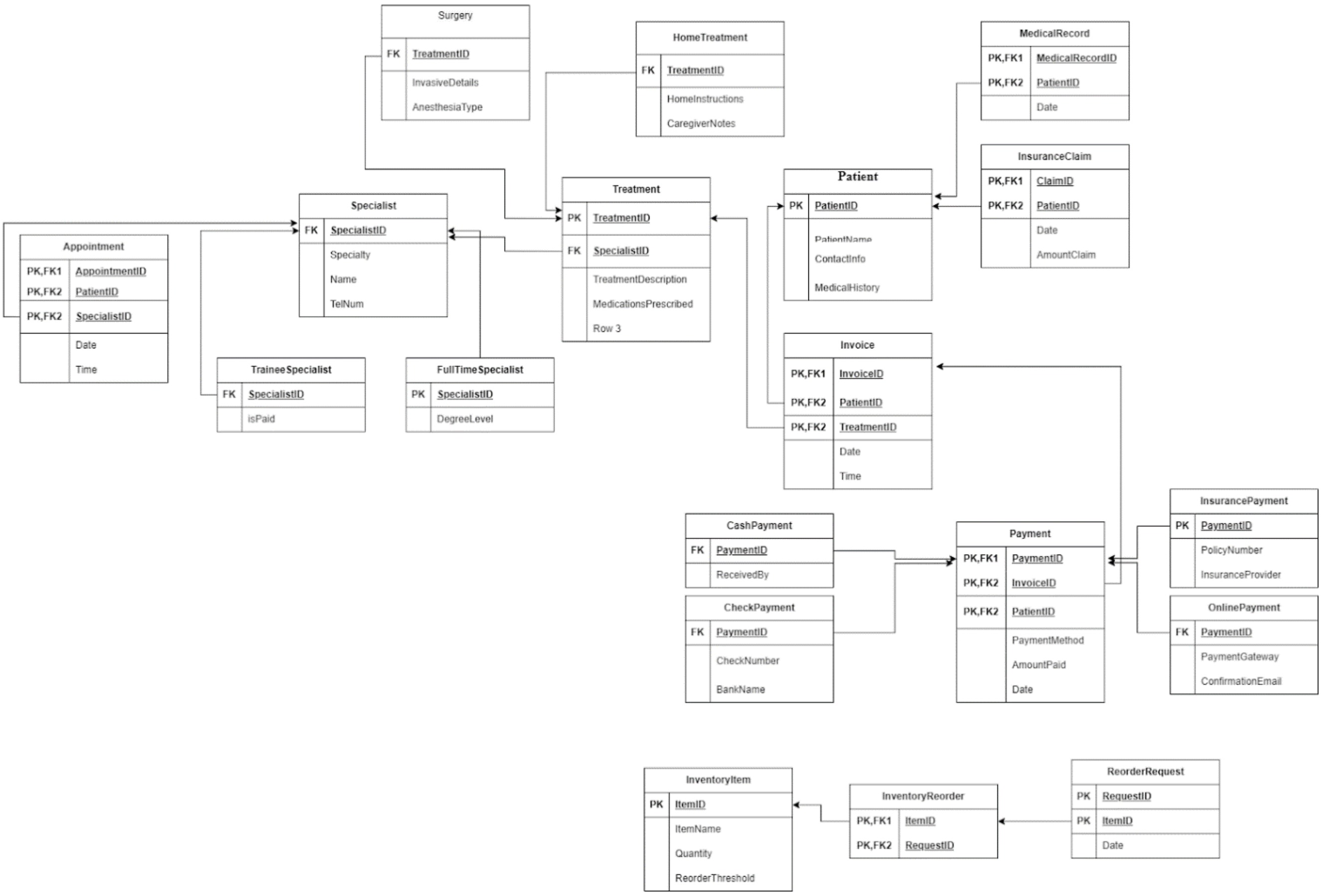
Attribute	Description	Data type	Constraint
RequestID	Request ID	VARCHAR2(10)	PRIMARY KEY
Date	Date of request	DATE	NOT NULL
Quantity	Quantity	VARCHAR2(35)	NOT NULL
ReorderThreshold	Reorder Threshold	VARCHAR2(35)	NOT NULL

InventoryReorder:

Attribute	Description	Data type	Constraint

RequestID	**Request ID**	**VARCHAR2(10)**	**FOREIGN KEY REFERENCE INVENTORYITEM**
RequestID	**Request ID**	**VARCHAR2(10)**	**FOREIGN KEY REFERENCE** **REORDERREQUEST**

Relational DB Schemas (after normalization)



Relation	Attributes and constrains
Patient	PatientID (Primary Key) PatientName ContactInfo MedicalHistory
Appointment	AppointmentID (Primary Key) Date Time PatientID (Foreign Key referencing Patient.PatientID) SpecialistID (Foreign Key referencing Specialist.SpecialistID)

Specialist	SpecialistID (Primary Key) Specialty Name TelNum
FullTimeSpecialist	SpecialistID (Primary Key, Foreign Key referencing Specialist.SpecialistID) DegreeLevel
TraineeSpecialist	SpecialistID (Primary Key, Foreign Key referencing Specialist.SpecialistID) isPaid
Treatment	TreatmentID (Primary Key) TreatmentDescription MedicationsPrescribed SpecialistID (Foreign Key referencing Specialist.SpecialistID)
SubTreatment	TreatmentID (Primary Key, Foreign Key referencing Treatment.TreatmentID) IsSurgery InvasiveDetails AnesthesiaType IsHomeTreatment HomeInstructions CaregiverNotes

Invoice	InvoiceID (Primary Key) Date Time PatientID (Foreign Key referencing Patient.PatientID) TreatmentID (Foreign Key referencing Treatment.TreatmentID)
InsuranceClaim	ClaimID (Primary Key) Date AmountClaim PatientID (Foreign Key referencing Patient.PatientID)
MedicalRecord	MedicalRecordID (Primary Key) Date PatientID (Foreign Key referencing Patient.PatientID)
Payment	PaymentID (Primary Key) PaymentMethod AmountPaid Date InvoiceID (Foreign Key referencing Invoice.InvoiceID) PatientID (Foreign Key referencing Patient.PatientID)
InsurancePayment	PaymentID (Primary Key, Foreign Key referencing Payment.PaymentID) PolicyNumber InsuranceProvider

OnlinePayment	PaymentID (Primary Key, Foreign Key referencing Payment.PaymentID) PaymentGateway ConfirmationEmail
CheckPayment	PaymentID (Primary Key, Foreign Key referencing Payment.PaymentID) CheckNumber BankName
CashPayment	PaymentID (Primary Key, Foreign Key referencing Payment.PaymentID) ReceivedBy
CreditCardPayment	PaymentID (Primary Key, Foreign Key referencing Payment.PaymentID) CardNumber CardHolderName
InventoryItem	ItemID (Primary Key) ItemName Quantity ReorderThreshold
ReorderRequest	RequestID (Primary Key) Date ItemID (Foreign Key referencing InventoryItem.ItemID)
InventoryReorder	ItemID (Foreign Key referencing InventoryItem.ItemID) RequestID (Foreign Key referencing ReorderRequest.RequestID)

SQL STATEMENTS (DML & DDL)

DDL (Table Creation)

```
-- Create Patient Table
CREATE TABLE Patient (
   PatientID INT PRIMARY KEY,
   PatientName VARCHAR2(255),
   ContactInfo VARCHAR2(255),
   MedicalHistory VARCHAR2(255)
);

-- Create Specialist Table
CREATE TABLE Specialist (
   SpecialistID INT PRIMARY KEY,
   Specialty VARCHAR2(255),
   Name VARCHAR2(255),
   TelNum VARCHAR2(20)
);

-- Create Appointment Table
CREATE TABLE Appointment (
   AppointmentID INT PRIMARY KEY,
   DateOfAppointment DATE,  -- Changed from "Date" to "DateOfAppointment"
   TimeOfAppointment TIMESTAMP,  -- Changed from "Time" to "TimeOfAppointment"
   PatientID INT REFERENCES Patient(PatientID),
   SpecialistID INT REFERENCES Specialist(SpecialistID)
);

-- Create FullTimeSpecialist Table
```

```
CREATE TABLE FullTimeSpecialist (
    SpecialistID INT PRIMARY KEY REFERENCES Specialist(SpecialistID),
    DegreeLevel VARCHAR2(50)
);

-- Create TraineeSpecialist Table
CREATE TABLE TraineeSpecialist (
    SpecialistID INT PRIMARY KEY REFERENCES Specialist(SpecialistID),
    isPaid VARCHAR2(5)
);

-- Create Treatment Table
CREATE TABLE Treatment (
    TreatmentID INT PRIMARY KEY,
    TreatmentDescription VARCHAR2(255),
    MedicationsPrescribed VARCHAR2(255),
    SpecialistID INT REFERENCES Specialist(SpecialistID)
);

-- Create Invoice Table
CREATE TABLE Invoice (
    InvoiceID INT PRIMARY KEY,
    DateOfInvoice DATE,  -- Changed from "Date" to "DateOfInvoice"
    TimeOfInvoice TIMESTAMP,  -- Changed from "Time" to "TimeOfInvoice"
    PatientID INT REFERENCES Patient(PatientID),
    TreatmentID INT REFERENCES Treatment(TreatmentID)
);

-- Create Payment Table
```

```
CREATE TABLE Payment (
   PaymentID INT PRIMARY KEY,
   PaymentMethod VARCHAR2(50),
   AmountPaid NUMBER(10, 2),
   DateOfPayment DATE,  -- Changed from "Date" to "DateOfPayment"
   InvoiceID INT REFERENCES Invoice(InvoiceID),
   PatientID INT REFERENCES Patient(PatientID)
);

-- Create InsurancePayment Table
CREATE TABLE InsurancePayment (
   PaymentID INT PRIMARY KEY REFERENCES Payment(PaymentID),
   PolicyNumber VARCHAR2(50),
   InsuranceProvider VARCHAR2(255)
);

-- Create OnlinePayment Table
CREATE TABLE OnlinePayment (
   PaymentID INT PRIMARY KEY REFERENCES Payment(PaymentID),
   PaymentGateway VARCHAR2(50),
   ConfirmationEmail VARCHAR2(255)
);

-- Create CheckPayment Table
CREATE TABLE CheckPayment (
   PaymentID INT PRIMARY KEY REFERENCES Payment(PaymentID),
   CheckNumber VARCHAR2(20),
   BankName VARCHAR2(255)
);
```

```
-- Create CashPayment Table
CREATE TABLE CashPayment (
   PaymentID INT PRIMARY KEY REFERENCES Payment(PaymentID),
   ReceivedBy VARCHAR2(255)
);

-- Create CreditCardPayment Table
CREATE TABLE CreditCardPayment (
   PaymentID INT PRIMARY KEY REFERENCES Payment(PaymentID),
   CardNumber VARCHAR2(20),
   CardHolderName VARCHAR2(255)
);

-- Create SubTreatment Table
CREATE TABLE SubTreatment (
   TreatmentID INT PRIMARY KEY REFERENCES Treatment(TreatmentID),
   IsSurgery VARCHAR2(5),
   InvasiveDetails VARCHAR2(255),
   AnesthesiaType VARCHAR2(255),
   IsHomeTreatment VARCHAR2(5),
   HomeInstructions VARCHAR2(255),
   CaregiverNotes VARCHAR2(255)
);

-- Create InsuranceClaim Table
CREATE TABLE InsuranceClaim (
   ClaimID INT PRIMARY KEY,
   DateOfClaim DATE,  -- Changed from "Date" to "DateOfClaim"
```

```
    AmountClaim NUMBER(10, 2),
    PatientID INT REFERENCES Patient(PatientID)
);

-- Create MedicalRecord Table
CREATE TABLE MedicalRecord (
    MedicalRecordID INT PRIMARY KEY,
    DateOfRecord DATE,  -- Changed from "Date" to "DateOfRecord"
    PatientID INT REFERENCES Patient(PatientID)
);

-- Create InventoryItem Table
CREATE TABLE InventoryItem (
    ItemID INT PRIMARY KEY,
    ItemName VARCHAR2(255),
    Quantity INT,
    ReorderThreshold INT
);

-- Create ReorderRequest Table
CREATE TABLE ReorderRequest (
    RequestID INT PRIMARY KEY,
    DateOfRequest DATE,  -- Changed from "Date" to "DateOfRequest"
    ItemID INT REFERENCES InventoryItem(ItemID)
);

-- Create InventoryReorder Table
CREATE TABLE InventoryReorder (
    ItemID INT REFERENCES InventoryItem(ItemID),
    RequestID INT REFERENCES ReorderRequest(RequestID),
    PRIMARY KEY (ItemID, RequestID)
);
```

DML1 (Insertion of the Rows)

```
-- Insert into Patient table
INSERT INTO Patient VALUES (1, 'John Doe', '123-456-7890', 'Previous surgeries');
INSERT INTO Patient VALUES (2, 'Jane Smith', '987-654-3210', 'Allergies: Penicillin');
INSERT INTO Patient VALUES (3, 'Alice Johnson', '555-123-4567', 'No special notes');
INSERT INTO Patient VALUES (4, 'Bob Anderson', '444-789-0123', 'Medication: Aspirin');
INSERT INTO Patient VALUES (5, 'Emily Davis', '777-888-9999', 'Diet restrictions: Gluten-free');
INSERT INTO Patient VALUES (6, 'David Miller', '333-222-1111', 'Family history: Diabetes');
INSERT INTO Patient VALUES (7, 'Sophia White', '888-777-6666', 'No known issues');
INSERT INTO Patient VALUES (8, 'Michael Brown', '111-222-3333', 'Allergies: Shellfish');
INSERT INTO Patient VALUES (9, 'Olivia Taylor', '666-555-4444', 'Previous surgeries');
INSERT INTO Patient VALUES (10, 'Matthew Harris', '222-333-4444', 'Medication: Blood pressure medication');
INSERT INTO Patient VALUES (11, 'Ava Martin', '444-555-6666', 'No special notes');
INSERT INTO Patient VALUES (12, 'Daniel Martinez', '999-888-7777', 'Allergies: Sulfa drugs');
INSERT INTO Patient VALUES (13, 'Grace Robinson', '777-666-5555', 'Diet restrictions: Lactose-free');
INSERT INTO Patient VALUES (14, 'Elijah Thompson', '555-666-7777', 'No known issues');
INSERT INTO Patient VALUES (15, 'Chloe Hall', '111-999-8888', 'Previous surgeries');
INSERT INTO Patient VALUES (16, 'Christopher Turner', '666-444-3333', 'Medication: Insulin');
INSERT INTO Patient VALUES (17, 'Lily Clark', '333-555-7777', 'Allergies: Peanuts');
INSERT INTO Patient VALUES (18, 'James Ward', '888-111-4444', 'No special notes');
INSERT INTO Patient VALUES (19, 'Nora King', '222-888-3333', 'Diet restrictions: Vegetarian');
INSERT INTO Patient VALUES (20, 'William Hill', '444-222-5555', 'Family history: Heart disease');
```

```
-- Insert into Specialist table
INSERT INTO Specialist VALUES (1, 'Cardiologist', 'Dr. Smith', '555-1234');
INSERT INTO Specialist VALUES (2, 'Dermatologist', 'Dr. Johnson', '555-5678');
INSERT INTO Specialist VALUES (3, 'Orthopedic Surgeon', 'Dr. Williams', '555-9876');
INSERT INTO Specialist VALUES (4, 'Neurologist', 'Dr. Davis', '555-4321');
INSERT INTO Specialist VALUES (5, 'Ophthalmologist', 'Dr. Taylor', '555-8765');
INSERT INTO Specialist VALUES (6, 'Gastroenterologist', 'Dr. Brown', '555-2345');
INSERT INTO Specialist VALUES (7, 'Pulmonologist', 'Dr. Anderson', '555-6789');
INSERT INTO Specialist VALUES (8, 'Endocrinologist', 'Dr. White', '555-3456');
INSERT INTO Specialist VALUES (9, 'Rheumatologist', 'Dr. Miller', '555-7890');
INSERT INTO Specialist VALUES (10, 'Urologist', 'Dr. Martin', '555-2109');
INSERT INTO Specialist VALUES (11, 'Nephrologist', 'Dr. Martinez', '555-5432');
INSERT INTO Specialist VALUES (12, 'Otolaryngologist', 'Dr. Robinson', '555-8765');
INSERT INTO Specialist VALUES (13, 'Psychiatrist', 'Dr. Thompson', '555-9876');
INSERT INTO Specialist VALUES (14, 'Allergist', 'Dr. Hall', '555-1234');
INSERT INTO Specialist VALUES (15, 'Hematologist', 'Dr. Turner', '555-5678');
INSERT INTO Specialist VALUES (16, 'Podiatrist', 'Dr. Clark', '555-4321');
INSERT INTO Specialist VALUES (17, 'Gynecologist', 'Dr. Ward', '555-8765');
INSERT INTO Specialist VALUES (18, 'Pediatrician', 'Dr. King', '555-2109');
INSERT INTO Specialist VALUES (19, 'Oncologist', 'Dr. Hill', '555-5432');
INSERT INTO Specialist VALUES (20, 'Dentist', 'Dr. Adams', '555-7890');
```

```
-- Insert into Appointment table
INSERT INTO Appointment VALUES (101, TO_DATE('2024-01-15', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-15 10:30:00', 'YYYY-MM-DD HH24:MI:SS'), 1, 1);
```

```
INSERT INTO Appointment VALUES (102, TO_DATE('2024-01-16', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-16 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), 2, 2);

INSERT INTO Appointment VALUES (103, TO_DATE('2024-01-17', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-17 11:45:00', 'YYYY-MM-DD HH24:MI:SS'), 3, 3);

INSERT INTO Appointment VALUES (104, TO_DATE('2024-01-18', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-18 09:15:00', 'YYYY-MM-DD HH24:MI:SS'), 4, 4);

INSERT INTO Appointment VALUES (105, TO_DATE('2024-01-19', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-19 16:30:00', 'YYYY-MM-DD HH24:MI:SS'), 5, 5);

INSERT INTO Appointment VALUES (106, TO_DATE('2024-01-20', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-20 13:00:00', 'YYYY-MM-DD HH24:MI:SS'), 6, 6);

INSERT INTO Appointment VALUES (107, TO_DATE('2024-01-21', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-21 15:45:00', 'YYYY-MM-DD HH24:MI:SS'), 7, 7);

INSERT INTO Appointment VALUES (108, TO_DATE('2024-01-22', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-22 08:30:00', 'YYYY-MM-DD HH24:MI:SS'), 8, 8);

INSERT INTO Appointment VALUES (109, TO_DATE('2024-01-23', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-23 12:00:00', 'YYYY-MM-DD HH24:MI:SS'), 9, 9);

INSERT INTO Appointment VALUES (110, TO_DATE('2024-01-24', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-24 17:15:00', 'YYYY-MM-DD HH24:MI:SS'), 10, 10);

INSERT INTO Appointment VALUES (111, TO_DATE('2024-01-25', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-25 10:45:00', 'YYYY-MM-DD HH24:MI:SS'), 11, 11);

INSERT INTO Appointment VALUES (112, TO_DATE('2024-01-26', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-26 14:30:00', 'YYYY-MM-DD HH24:MI:SS'), 12, 12);

INSERT INTO Appointment VALUES (113, TO_DATE('2024-01-27', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-27 09:00:00', 'YYYY-MM-DD HH24:MI:SS'), 13, 13);

INSERT INTO Appointment VALUES (114, TO_DATE('2024-01-28', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-28 16:45:00', 'YYYY-MM-DD HH24:MI:SS'), 14, 14);

INSERT INTO Appointment VALUES (115, TO_DATE('2024-01-29', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-29 11:30:00', 'YYYY-MM-DD HH24:MI:SS'), 15, 15);

INSERT INTO Appointment VALUES (116, TO_DATE('2024-01-30', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-30 15:00:00', 'YYYY-MM-DD HH24:MI:SS'), 16, 16);

INSERT INTO Appointment VALUES (117, TO_DATE('2024-01-31', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-31 12:15:00', 'YYYY-MM-DD HH24:MI:SS'), 17, 17);

INSERT INTO Appointment VALUES (118, TO_DATE('2024-02-01', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-02-01 08:45:00', 'YYYY-MM-DD HH24:MI:SS'), 18, 18);
```

```sql
INSERT INTO Appointment VALUES (119, TO_DATE('2024-02-02', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-02-02 13:30:00', 'YYYY-MM-DD HH24:MI:SS'), 19, 19);

INSERT INTO Appointment VALUES (120, TO_DATE('2024-02-03', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-02-03 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 20, 20);



-- Insert into FullTimeSpecialist table

INSERT INTO FullTimeSpecialist VALUES (1, 'MD');

INSERT INTO FullTimeSpecialist VALUES (2, 'DO');

INSERT INTO FullTimeSpecialist VALUES (3, 'MD');

INSERT INTO FullTimeSpecialist VALUES (4, 'DO');

INSERT INTO FullTimeSpecialist VALUES (5, 'MD');

INSERT INTO FullTimeSpecialist VALUES (6, 'DO');

INSERT INTO FullTimeSpecialist VALUES (7, 'MD');

INSERT INTO FullTimeSpecialist VALUES (8, 'DO');

INSERT INTO FullTimeSpecialist VALUES (9, 'MD');

INSERT INTO FullTimeSpecialist VALUES (10, 'DO');

INSERT INTO FullTimeSpecialist VALUES (11, 'MD');

INSERT INTO FullTimeSpecialist VALUES (12, 'DO');

INSERT INTO FullTimeSpecialist VALUES (13, 'MD');

INSERT INTO FullTimeSpecialist VALUES (14, 'DO');

INSERT INTO FullTimeSpecialist VALUES (15, 'MD');

INSERT INTO FullTimeSpecialist VALUES (16, 'DO');

INSERT INTO FullTimeSpecialist VALUES (17, 'MD');

INSERT INTO FullTimeSpecialist VALUES (18, 'DO');

INSERT INTO FullTimeSpecialist VALUES (19, 'MD');

INSERT INTO FullTimeSpecialist VALUES (20, 'DO');


-- Insert into TraineeSpecialist table

INSERT INTO TraineeSpecialist VALUES (1, 'Yes');
INSERT INTO TraineeSpecialist VALUES (2, 'No');

INSERT INTO TraineeSpecialist VALUES (3, 'Yes');

INSERT INTO TraineeSpecialist VALUES (4, 'No');

INSERT INTO TraineeSpecialist VALUES (5, 'Yes');

INSERT INTO TraineeSpecialist VALUES (6, 'No');

INSERT INTO TraineeSpecialist VALUES (7, 'Yes');

INSERT INTO TraineeSpecialist VALUES (8, 'No');

INSERT INTO TraineeSpecialist VALUES (9, 'Yes');

INSERT INTO TraineeSpecialist VALUES (10, 'No');

INSERT INTO TraineeSpecialist VALUES (11, 'Yes');

INSERT INTO TraineeSpecialist VALUES (12, 'No');

INSERT INTO TraineeSpecialist VALUES (13, 'Yes');

INSERT INTO TraineeSpecialist VALUES (14, 'No');

INSERT INTO TraineeSpecialist VALUES (15, 'Yes');

INSERT INTO TraineeSpecialist VALUES (16, 'No');

INSERT INTO TraineeSpecialist VALUES (17, 'Yes');

INSERT INTO TraineeSpecialist VALUES (18, 'No');

INSERT INTO TraineeSpecialist VALUES (19, 'Yes');

INSERT INTO TraineeSpecialist VALUES (20, 'No');



-- Insert into Payment table

INSERT INTO Payment VALUES (1001, 'Credit Card', 150.00, TO_DATE('2024-01-15', 'YYYY-MM-DD'), NULL, 1);

INSERT INTO Payment VALUES (1002, 'Cash', 75.00, TO_DATE('2024-01-16', 'YYYY-MM-DD'), NULL, 2);

INSERT INTO Payment VALUES (1003, 'Credit Card', 120.00, TO_DATE('2024-01-17', 'YYYY-MM-DD'), NULL, 3);

INSERT INTO Payment VALUES (1004, 'Cash', 90.00, TO_DATE('2024-01-18', 'YYYY-MM-DD'), NULL, 4);
INSERT INTO Payment VALUES (1005, 'Credit Card', 200.00, TO_DATE('2024-01-19', 'YYYY-MM-DD'), NULL, 5);

INSERT INTO Payment VALUES (1006, 'Cash', 50.00, TO_DATE('2024-01-20', 'YYYY-MM-DD'), NULL, 6);

INSERT INTO Payment VALUES (1007, 'Credit Card', 180.00, TO_DATE('2024-01-21', 'YYYY-MM-DD'), NULL, 7);

INSERT INTO Payment VALUES (1008, 'Cash', 100.00, TO_DATE('2024-01-22', 'YYYY-MM-DD'), NULL, 8);

INSERT INTO Payment VALUES (1009, 'Credit Card', 130.00, TO_DATE('2024-01-23', 'YYYY-MM-DD'), NULL, 9);

INSERT INTO Payment VALUES (1010, 'Cash', 60.00, TO_DATE('2024-01-24', 'YYYY-MM-DD'), NULL, 10);

INSERT INTO Payment VALUES (1011, 'Credit Card', 170.00, TO_DATE('2024-01-25', 'YYYY-MM-DD'), NULL, 11);

INSERT INTO Payment VALUES (1012, 'Cash', 85.00, TO_DATE('2024-01-26', 'YYYY-MM-DD'), NULL, 12);

INSERT INTO Payment VALUES (1013, 'Credit Card', 110.00, TO_DATE('2024-01-27', 'YYYY-MM-DD'), NULL, 13);

INSERT INTO Payment VALUES (1014, 'Cash', 95.00, TO_DATE('2024-01-28', 'YYYY-MM-DD'), NULL, 14);

INSERT INTO Payment VALUES (1015, 'Credit Card', 160.00, TO_DATE('2024-01-29', 'YYYY-MM-DD'), NULL, 15);

INSERT INTO Payment VALUES (1016, 'Cash', 120.00, TO_DATE('2024-01-30', 'YYYY-MM-DD'), NULL, 16);

INSERT INTO Payment VALUES (1017, 'Credit Card', 140.00, TO_DATE('2024-01-31', 'YYYY-MM-DD'), NULL, 17);

INSERT INTO Payment VALUES (1018, 'Cash', 80.00, TO_DATE('2024-02-01', 'YYYY-MM-DD'), NULL, 18);

INSERT INTO Payment VALUES (1019, 'Credit Card', 190.00, TO_DATE('2024-02-02', 'YYYY-MM-DD'), NULL, 19);

INSERT INTO Payment VALUES (1020, 'Cash', 70.00, TO_DATE('2024-02-03', 'YYYY-MM-DD'), NULL, 20);


-- Insert into InsurancePayment table

INSERT INTO InsurancePayment VALUES (1001, 'P12345', 'ABC Insurance');
INSERT INTO InsurancePayment VALUES (1002, 'P67890', 'XYZ Insurance');
INSERT INTO InsurancePayment VALUES (1003, 'P54321', 'DEF Insurance');
INSERT INTO InsurancePayment VALUES (1004, 'P09876', 'LMN Insurance');
INSERT INTO InsurancePayment VALUES (1005, 'P24680', 'GHI Insurance');
INSERT INTO InsurancePayment VALUES (1006, 'P13579', 'JKL Insurance');
INSERT INTO InsurancePayment VALUES (1007, 'P77777', 'MNO Insurance');
INSERT INTO InsurancePayment VALUES (1008, 'P88888', 'QRS Insurance');
INSERT INTO InsurancePayment VALUES (1009, 'P99999', 'TUV Insurance');
INSERT INTO InsurancePayment VALUES (1010, 'P11111', 'XYZ Insurance');
INSERT INTO InsurancePayment VALUES (1011, 'P22222', 'ABC Insurance');
INSERT INTO InsurancePayment VALUES (1012, 'P33333', 'DEF Insurance');
INSERT INTO InsurancePayment VALUES (1013, 'P44444', 'GHI Insurance');
INSERT INTO InsurancePayment VALUES (1014, 'P55555', 'JKL Insurance');
INSERT INTO InsurancePayment VALUES (1015, 'P66666', 'LMN Insurance');
INSERT INTO InsurancePayment VALUES (1016, 'P77777', 'MNO Insurance');
INSERT INTO InsurancePayment VALUES (1017, 'P88888', 'QRS Insurance');
INSERT INTO InsurancePayment VALUES (1018, 'P99999', 'TUV Insurance');
INSERT INTO InsurancePayment VALUES (1019, 'P12321', 'ABC Insurance');
INSERT INTO InsurancePayment VALUES (1020, 'P67876', 'XYZ Insurance');


-- Insert into OnlinePayment table
INSERT INTO OnlinePayment VALUES (1001, 'PayPal', 'john.doe@example.com');
INSERT INTO OnlinePayment VALUES (1002, 'Stripe', 'jane.smith@example.com');
INSERT INTO OnlinePayment VALUES (1003, 'PayPal', 'alice.johnson@example.com');
INSERT INTO OnlinePayment VALUES (1004, 'Stripe', 'bob.anderson@example.com');
INSERT INTO OnlinePayment VALUES (1005, 'PayPal', 'emily.davis@example.com');
INSERT INTO OnlinePayment VALUES (1006, 'Stripe', 'david.miller@example.com');
INSERT INTO OnlinePayment VALUES (1007, 'PayPal', 'sophia.white@example.com');
INSERT INTO OnlinePayment VALUES (1008, 'Stripe', 'michael.brown@example.com');
INSERT INTO OnlinePayment VALUES (1009, 'PayPal', 'olivia.taylor@example.com');
INSERT INTO OnlinePayment VALUES (1010, 'Stripe', 'matthew.harris@example.com');
INSERT INTO OnlinePayment VALUES (1011, 'PayPal', 'ava.martin@example.com');
INSERT INTO OnlinePayment VALUES (1012, 'Stripe', 'daniel.martinez@example.com');
INSERT INTO OnlinePayment VALUES (1013, 'PayPal', 'grace.robinson@example.com');
INSERT INTO OnlinePayment VALUES (1014, 'Stripe', 'elijah.thompson@example.com');
INSERT INTO OnlinePayment VALUES (1015, 'PayPal', 'chloe.hall@example.com');
INSERT INTO OnlinePayment VALUES (1016, 'Stripe', 'christopher.turner@example.com');
INSERT INTO OnlinePayment VALUES (1017, 'PayPal', 'lily.clark@example.com');
INSERT INTO OnlinePayment VALUES (1018, 'Stripe', 'james.ward@example.com');
INSERT INTO OnlinePayment VALUES (1019, 'PayPal', 'nora.king@example.com');
INSERT INTO OnlinePayment VALUES (1020, 'Stripe', 'william.hill@example.com');

-- Insert into CheckPayment table
INSERT INTO CheckPayment VALUES (1001, '123456', 'Bank of America');
INSERT INTO CheckPayment VALUES (1002, '789012', 'Chase');
INSERT INTO CheckPayment VALUES (1003, '345678', 'Wells Fargo');
INSERT INTO CheckPayment VALUES (1004, '901234', 'Citibank');
INSERT INTO CheckPayment VALUES (1005, '567890', 'US Bank');
INSERT INTO CheckPayment VALUES (1006, '234567', 'PNC Bank');
INSERT INTO CheckPayment VALUES (1007, '890123', 'TD Bank');
INSERT INTO CheckPayment VALUES (1008, '456789', 'SunTrust');
INSERT INTO CheckPayment VALUES (1009, '012345', 'Capital One');
INSERT INTO CheckPayment VALUES (1010, '678901', 'HSBC');
INSERT INTO CheckPayment VALUES (1011, '234567', 'BB&T');
INSERT INTO CheckPayment VALUES (1012, '890123', 'KeyBank');
INSERT INTO CheckPayment VALUES (1013, '345678', 'Regions Bank');
INSERT INTO CheckPayment VALUES (1014, '901234', 'Santander');
INSERT INTO CheckPayment VALUES (1015, '567890', 'Fifth Third Bank');
INSERT INTO CheckPayment VALUES (1016, '234567', 'Ally Bank');
INSERT INTO CheckPayment VALUES (1017, '890123', 'M&T Bank');
INSERT INTO CheckPayment VALUES (1018, '456789', 'Comerica');
INSERT INTO CheckPayment VALUES (1019, '012345', 'BBVA');
INSERT INTO CheckPayment VALUES (1020, '678901', 'Union Bank');


-- Insert into CashPayment table
INSERT INTO CashPayment VALUES (1001, 'Reception');
INSERT INTO CashPayment VALUES (1002, 'Front Desk');
INSERT INTO CashPayment VALUES (1003, 'Cashier');
INSERT INTO CashPayment VALUES (1004, 'Payment Counter');
INSERT INTO CashPayment VALUES (1005, 'Customer Service');
INSERT INTO CashPayment VALUES (1006, 'Service Desk');
INSERT INTO CashPayment VALUES (1007, 'Cash Register');
INSERT INTO CashPayment VALUES (1008, 'Point of Sale');
INSERT INTO CashPayment VALUES (1009, 'Cash Office');
INSERT INTO CashPayment VALUES (1010, 'Billing Department');
INSERT INTO CashPayment VALUES (1011, 'Teller');
INSERT INTO CashPayment VALUES (1012, 'Payment Booth');
INSERT INTO CashPayment VALUES (1013, 'Accounting Desk');
INSERT INTO CashPayment VALUES (1014, 'Financial Center');
INSERT INTO CashPayment VALUES (1015, 'Money Counter');
INSERT INTO CashPayment VALUES (1016, 'Cash Vault');
INSERT INTO CashPayment VALUES (1017, 'Treasury');
INSERT INTO CashPayment VALUES (1018, 'Finance Office');
INSERT INTO CashPayment VALUES (1019, 'Payment Kiosk');
INSERT INTO CashPayment VALUES (1020, 'Cash Management');
```

```
-- Insert into CreditCardPayment table
INSERT INTO CreditCardPayment VALUES (1001, '1234-5678-9012-3456', 'John Doe');
INSERT INTO CreditCardPayment VALUES (1002, '9876-5432-1098-7654', 'Jane Smith');
INSERT INTO CreditCardPayment VALUES (1003, '5678-9012-3456-7890', 'Alice Johnson');
INSERT INTO CreditCardPayment VALUES (1004, '4321-0987-6543-2109', 'Bob Anderson');
INSERT INTO CreditCardPayment VALUES (1005, '8765-4321-0987-6543', 'Emily Davis');
INSERT INTO CreditCardPayment VALUES (1006, '1111-2222-3333-4444', 'David Miller');
INSERT INTO CreditCardPayment VALUES (1007, '4444-5555-6666-7777', 'Sophia White');
INSERT INTO CreditCardPayment VALUES (1008, '3333-4444-5555-6666', 'Michael Brown');
INSERT INTO CreditCardPayment VALUES (1009, '6666-7777-8888-9999', 'Olivia Taylor');
INSERT INTO CreditCardPayment VALUES (1010, '2222-3333-4444-5555', 'Matthew Harris');
INSERT INTO CreditCardPayment VALUES (1011, '5555-6666-7777-8888', 'Ava Martin');
INSERT INTO CreditCardPayment VALUES (1012, '9999-8888-7777-6666', 'Daniel Martinez');
INSERT INTO CreditCardPayment VALUES (1013, '7777-6666-5555-4444', 'Grace Robinson');
INSERT INTO CreditCardPayment VALUES (1014, '5555-6666-7777-8888', 'Elijah Thompson');
INSERT INTO CreditCardPayment VALUES (1015, '1111-9999-8888-7777', 'Chloe Hall');
INSERT INTO CreditCardPayment VALUES (1016, '6666-4444-3333-2222', 'Christopher Turner');
INSERT INTO CreditCardPayment VALUES (1017, '3333-5555-7777-8888', 'Lily Clark');
INSERT INTO CreditCardPayment VALUES (1018, '8888-1111-4444-7777', 'James Ward');
INSERT INTO CreditCardPayment VALUES (1019, '2222-8888-3333-7777', 'Nora King');
INSERT INTO CreditCardPayment VALUES (1020, '4444-2222-5555-9999', 'William Hill');


-- Insert into Treatment table
INSERT INTO Treatment VALUES (201, 'Cardiac Checkup', 'Prescription: Aspirin', 1);
INSERT INTO Treatment VALUES (202, 'Skin Exam', 'Prescription: Cortisone Cream', 2);
INSERT INTO Treatment VALUES (203, 'Orthopedic Consultation', 'Prescription: Physical Therapy', 3);
INSERT INTO Treatment VALUES (204, 'Neurological Evaluation', 'Prescription: Pain Medication', 4);

INSERT INTO Treatment VALUES (205, 'Eye Examination', 'Prescription: Glasses', 5);

INSERT INTO Treatment VALUES (206, 'Gastrointestinal Checkup', 'Prescription: Antacids', 6);

INSERT INTO Treatment VALUES (207, 'Pulmonary Function Test', 'Prescription: Inhaler', 7);

INSERT INTO Treatment VALUES (208, 'Endocrine Consultation', 'Prescription: Hormone Replacement', 8);

INSERT INTO Treatment VALUES (209, 'Rheumatology Assessment', 'Prescription: Anti-inflammatory', 9);

INSERT INTO Treatment VALUES (210, 'Urological Evaluation', 'Prescription: Antibiotics', 10);

INSERT INTO Treatment VALUES (211, 'Nephrology Consultation', 'Prescription: Diuretics', 11);

INSERT INTO Treatment VALUES (212, 'Ear, Nose, and Throat Exam', 'Prescription: Antibiotics', 12);

INSERT INTO Treatment VALUES (213, 'Psychiatric Evaluation', 'Prescription: Antidepressants', 13);

INSERT INTO Treatment VALUES (214, 'Allergy Testing', 'Prescription: Antihistamines', 14);

INSERT INTO Treatment VALUES (215, 'Hematology Consultation', 'Prescription: Blood Thinners', 15);

INSERT INTO Treatment VALUES (216, 'Podiatry Visit', 'Prescription: Orthopedic Shoes', 16);

INSERT INTO Treatment VALUES (217, 'Gynecological Exam', 'Prescription: Birth Control Pills', 17);

INSERT INTO Treatment VALUES (218, 'Pediatric Checkup', 'Prescription: Childrens Vitamins', 18);

INSERT INTO Treatment VALUES (219, 'Oncology Consultation', 'Prescription: Chemotherapy', 19);

INSERT INTO Treatment VALUES (220, 'Dental Cleaning', 'Prescription: Fluoride Toothpaste', 20);


-- Insert into SubTreatment table

INSERT INTO SubTreatment VALUES (201, 'No', NULL, NULL, 'No', 'N/A', 'Follow up in 2 weeks');

INSERT INTO SubTreatment VALUES (202, 'No', NULL, NULL, 'Yes', 'Apply cream twice a day', 'None');

INSERT INTO SubTreatment VALUES (203, 'Yes', 'Physical Therapy sessions', 'Every Monday and Wednesday', 'Yes', 'Continue prescribed medication', 'Follow up in 4 weeks');

INSERT INTO SubTreatment VALUES (204, 'No', NULL, NULL, 'Yes', 'Take medication as needed for pain', 'None');
```

```sql
INSERT INTO SubTreatment VALUES (205, 'No', NULL, NULL, 'No', 'N/A', 'Annual checkup next year');

INSERT INTO SubTreatment VALUES (206, 'Yes', 'Dietary changes', 'Avoid spicy and acidic foods', 'Yes', 'Take antacids as prescribed', 'Follow up in 3 months');

INSERT INTO SubTreatment VALUES (207, 'Yes', 'Lung function exercises', 'Daily', 'Yes', 'Use inhaler as needed', 'Follow up in 6 weeks');

INSERT INTO SubTreatment VALUES (208, 'Yes', 'Hormone replacement therapy', 'Follow prescribed regimen', 'Yes', 'Regular blood tests', 'Follow up in 3 months');

INSERT INTO SubTreatment VALUES (209, 'Yes', 'Physical therapy sessions', 'Twice a week', 'Yes', 'Continue anti-inflammatory medication', 'Follow up in 4 weeks');

INSERT INTO SubTreatment VALUES (210, 'No', NULL, NULL, 'Yes', 'Take antibiotics as prescribed', 'None');

INSERT INTO SubTreatment VALUES (211, 'Yes', 'Prescription for diuretics', 'Follow prescribed regimen', 'Yes', 'Regular kidney function tests', 'Follow up in 6 weeks');

INSERT INTO SubTreatment VALUES (212, 'Yes', 'Prescription for antibiotics', 'Take full course', 'Yes', 'Follow prescribed medication', 'Follow up in 1 week');

INSERT INTO SubTreatment VALUES (213, 'Yes', 'Therapy sessions', 'Weekly', 'Yes', 'Continue antidepressant medication', 'Follow up in 2 weeks');

INSERT INTO SubTreatment VALUES (214, 'Yes', 'Allergy desensitization', 'As recommended', 'Yes', 'Take antihistamines as needed', 'Follow up in 6 months');

INSERT INTO SubTreatment VALUES (215, 'Yes', 'Blood thinner medication', 'Follow prescribed regimen', 'Yes', 'Regular blood tests', 'Follow up in 3 months');

INSERT INTO SubTreatment VALUES (216, 'Yes', 'Orthopedic shoe fittings', 'Follow recommended schedule', 'Yes', 'Foot exercises as prescribed', 'Follow up in 2 months');

INSERT INTO SubTreatment VALUES (217, 'No', NULL, NULL, 'Yes', 'Follow prescribed medication', 'None');

INSERT INTO SubTreatment VALUES (218, 'No', NULL, NULL, 'Yes', 'Follow prescribed regimen', 'None');

INSERT INTO SubTreatment VALUES (219, 'Yes', 'Chemotherapy sessions', 'As scheduled', 'Yes', 'Follow prescribed medication', 'Follow up in 2 weeks');

INSERT INTO SubTreatment VALUES (220, 'No', NULL, NULL, 'Yes', 'Maintain regular dental hygiene practices', 'None');



-- Insert into Invoice table
```

```
INSERT INTO Invoice VALUES (301, TO_DATE('2024-01-15', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-15 11:30:00', 'YYYY-MM-DD HH24:MI:SS'), 1, 201);

INSERT INTO Invoice VALUES (302, TO_DATE('2024-01-16', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-16 15:30:00', 'YYYY-MM-DD HH24:MI:SS'), 2, 202);

INSERT INTO Invoice VALUES (303, TO_DATE('2024-01-17', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-17 12:45:00', 'YYYY-MM-DD HH24:MI:SS'), 3, 203);

INSERT INTO Invoice VALUES (304, TO_DATE('2024-01-18', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-18 10:15:00', 'YYYY-MM-DD HH24:MI:SS'), 4, 204);

INSERT INTO Invoice VALUES (305, TO_DATE('2024-01-19', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-19 17:30:00', 'YYYY-MM-DD HH24:MI:SS'), 5, 205);

INSERT INTO Invoice VALUES (306, TO_DATE('2024-01-20', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-20 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), 6, 206);

INSERT INTO Invoice VALUES (307, TO_DATE('2024-01-21', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-21 16:45:00', 'YYYY-MM-DD HH24:MI:SS'), 7, 207);

INSERT INTO Invoice VALUES (308, TO_DATE('2024-01-22', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-22 09:30:00', 'YYYY-MM-DD HH24:MI:SS'), 8, 208);

INSERT INTO Invoice VALUES (309, TO_DATE('2024-01-23', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-23 13:00:00', 'YYYY-MM-DD HH24:MI:SS'), 9, 209);

INSERT INTO Invoice VALUES (310, TO_DATE('2024-01-24', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-24 18:15:00', 'YYYY-MM-DD HH24:MI:SS'), 10, 210);

INSERT INTO Invoice VALUES (311, TO_DATE('2024-01-25', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-25 11:45:00', 'YYYY-MM-DD HH24:MI:SS'), 11, 211);

INSERT INTO Invoice VALUES (312, TO_DATE('2024-01-26', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-26 15:30:00', 'YYYY-MM-DD HH24:MI:SS'), 12, 212);

INSERT INTO Invoice VALUES (313, TO_DATE('2024-01-27', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-27 10:00:00', 'YYYY-MM-DD HH24:MI:SS'), 13, 213);

INSERT INTO Invoice VALUES (314, TO_DATE('2024-01-28', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-28 17:45:00', 'YYYY-MM-DD HH24:MI:SS'), 14, 214);

INSERT INTO Invoice VALUES (315, TO_DATE('2024-01-29', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-29 12:30:00', 'YYYY-MM-DD HH24:MI:SS'), 15, 215);

INSERT INTO Invoice VALUES (316, TO_DATE('2024-01-30', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-30 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 16, 216);

INSERT INTO Invoice VALUES (317, TO_DATE('2024-01-31', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-31 13:15:00', 'YYYY-MM-DD HH24:MI:SS'), 17, 217);
```

```
INSERT INTO Invoice VALUES (318, TO_DATE('2024-02-01', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-02-01 09:45:00', 'YYYY-MM-DD HH24:MI:SS'), 18, 218);

INSERT INTO Invoice VALUES (319, TO_DATE('2024-02-02', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-02-02 14:30:00', 'YYYY-MM-DD HH24:MI:SS'), 19, 219);

INSERT INTO Invoice VALUES (320, TO_DATE('2024-02-03', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-02-03 17:00:00', 'YYYY-MM-DD HH24:MI:SS'), 20, 220);




-- Insert into InsuranceClaim table

INSERT INTO InsuranceClaim VALUES (401, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 100.00,
1);

INSERT INTO InsuranceClaim VALUES (402, TO_DATE('2024-01-16', 'YYYY-MM-DD'), 50.00, 2);

INSERT INTO InsuranceClaim VALUES (403, TO_DATE('2024-01-17', 'YYYY-MM-DD'), 120.00,
3);

INSERT INTO InsuranceClaim VALUES (404, TO_DATE('2024-01-18', 'YYYY-MM-DD'), 90.00, 4);

INSERT INTO InsuranceClaim VALUES (405, TO_DATE('2024-01-19', 'YYYY-MM-DD'), 200.00,
5);

INSERT INTO InsuranceClaim VALUES (406, TO_DATE('2024-01-20', 'YYYY-MM-DD'), 50.00, 6);

INSERT INTO InsuranceClaim VALUES (407, TO_DATE('2024-01-21', 'YYYY-MM-DD'), 180.00,
7);

INSERT INTO InsuranceClaim VALUES (408, TO_DATE('2024-01-22', 'YYYY-MM-DD'), 100.00,
8);

INSERT INTO InsuranceClaim VALUES (409, TO_DATE('2024-01-23', 'YYYY-MM-DD'), 130.00,
9);

INSERT INTO InsuranceClaim VALUES (410, TO_DATE('2024-01-24', 'YYYY-MM-DD'), 60.00,
10);

INSERT INTO InsuranceClaim VALUES (411, TO_DATE('2024-01-25', 'YYYY-MM-DD'), 170.00,
11);

INSERT INTO InsuranceClaim VALUES (412, TO_DATE('2024-01-26', 'YYYY-MM-DD'), 85.00,
12);

INSERT INTO InsuranceClaim VALUES (413, TO_DATE('2024-01-27', 'YYYY-MM-DD'), 110.00,
13);
```

```
INSERT INTO InsuranceClaim VALUES (414, TO_DATE('2024-01-28', 'YYYY-MM-DD'), 95.00, 14);

INSERT INTO InsuranceClaim VALUES (415, TO_DATE('2024-01-29', 'YYYY-MM-DD'), 160.00, 15);

INSERT INTO InsuranceClaim VALUES (416, TO_DATE('2024-01-30', 'YYYY-MM-DD'), 120.00, 16);

INSERT INTO InsuranceClaim VALUES (417, TO_DATE('2024-01-31', 'YYYY-MM-DD'), 140.00, 17);

INSERT INTO InsuranceClaim VALUES (418, TO_DATE('2024-02-01', 'YYYY-MM-DD'), 80.00, 18);

INSERT INTO InsuranceClaim VALUES (419, TO_DATE('2024-02-02', 'YYYY-MM-DD'), 190.00, 19);

INSERT INTO InsuranceClaim VALUES (420, TO_DATE('2024-02-03', 'YYYY-MM-DD'), 70.00, 20);



-- Insert into MedicalRecord table

INSERT INTO MedicalRecord VALUES (501, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 1);

INSERT INTO MedicalRecord VALUES (502, TO_DATE('2024-01-16', 'YYYY-MM-DD'), 2);

INSERT INTO MedicalRecord VALUES (503, TO_DATE('2024-01-17', 'YYYY-MM-DD'), 3);

INSERT INTO MedicalRecord VALUES (504, TO_DATE('2024-01-18', 'YYYY-MM-DD'), 4);

INSERT INTO MedicalRecord VALUES (505, TO_DATE('2024-01-19', 'YYYY-MM-DD'), 5);

INSERT INTO MedicalRecord VALUES (506, TO_DATE('2024-01-20', 'YYYY-MM-DD'), 6);

INSERT INTO MedicalRecord VALUES (507, TO_DATE('2024-01-21', 'YYYY-MM-DD'), 7);

INSERT INTO MedicalRecord VALUES (508, TO_DATE('2024-01-22', 'YYYY-MM-DD'), 8);

INSERT INTO MedicalRecord VALUES (509, TO_DATE('2024-01-23', 'YYYY-MM-DD'), 9);

INSERT INTO MedicalRecord VALUES (510, TO_DATE('2024-01-24', 'YYYY-MM-DD'), 10);

INSERT INTO MedicalRecord VALUES (511, TO_DATE('2024-01-25', 'YYYY-MM-DD'), 11);

INSERT INTO MedicalRecord VALUES (512, TO_DATE('2024-01-26', 'YYYY-MM-DD'), 12);

INSERT INTO MedicalRecord VALUES (513, TO_DATE('2024-01-27', 'YYYY-MM-DD'), 13);

INSERT INTO MedicalRecord VALUES (514, TO_DATE('2024-01-28', 'YYYY-MM-DD'), 14);

INSERT INTO MedicalRecord VALUES (515, TO_DATE('2024-01-29', 'YYYY-MM-DD'), 15);
INSERT INTO MedicalRecord VALUES (516, TO_DATE('2024-01-30', 'YYYY-MM-DD'), 16);
INSERT INTO MedicalRecord VALUES (517, TO_DATE('2024-01-31', 'YYYY-MM-DD'), 17);
INSERT INTO MedicalRecord VALUES (518, TO_DATE('2024-02-01', 'YYYY-MM-DD'), 18);
INSERT INTO MedicalRecord VALUES (519, TO_DATE('2024-02-02', 'YYYY-MM-DD'), 19);
INSERT INTO MedicalRecord VALUES (520, TO_DATE('2024-02-03', 'YYYY-MM-DD'), 20);


-- Insert into InventoryItem table
INSERT INTO InventoryItem VALUES (601, 'Bandages', 100, 20);
INSERT INTO InventoryItem VALUES (602, 'Painkillers', 50, 10);
INSERT INTO InventoryItem VALUES (603, 'Antibiotics', 30, 15);
INSERT INTO InventoryItem VALUES (604, 'Thermometers', 40, 25);
INSERT INTO InventoryItem VALUES (605, 'Gauze', 80, 30);
INSERT INTO InventoryItem VALUES (606, 'Cold Medicine', 60, 18);
INSERT INTO InventoryItem VALUES (607, 'Hand Sanitizer', 70, 22);
INSERT INTO InventoryItem VALUES (608, 'First Aid Kits', 25, 8);
INSERT INTO InventoryItem VALUES (609, 'Ace Bandages', 35, 12);
INSERT INTO InventoryItem VALUES (610, 'Cough Drops', 45, 14);
INSERT INTO InventoryItem VALUES (611, 'Adhesive Tape', 65, 28);
INSERT INTO InventoryItem VALUES (612, 'Antiseptic Wipes', 55, 26);
INSERT INTO InventoryItem VALUES (613, 'Peroxide', 20, 6);
INSERT INTO InventoryItem VALUES (614, 'Allergy Medication', 75, 24);
INSERT INTO InventoryItem VALUES (615, 'Splints', 15, 5);
INSERT INTO InventoryItem VALUES (616, 'Ibuprofen', 90, 32);
INSERT INTO InventoryItem VALUES (617, 'Sterile Dressings', 85, 29);
INSERT INTO InventoryItem VALUES (618, 'Burn Ointment', 33, 11);
INSERT INTO InventoryItem VALUES (619, 'Insect Repellent', 27, 9);
INSERT INTO InventoryItem VALUES (620, 'Hot/Cold Packs', 38, 16);
```

```
-- Insert into ReorderRequest table
INSERT INTO ReorderRequest VALUES (701, TO_DATE('2024-01-17', 'YYYY-MM-DD'), 601);
INSERT INTO ReorderRequest VALUES (702, TO_DATE('2024-01-18', 'YYYY-MM-DD'), 602);
INSERT INTO ReorderRequest VALUES (703, TO_DATE('2024-01-19', 'YYYY-MM-DD'), 603);
INSERT INTO ReorderRequest VALUES (704, TO_DATE('2024-01-20', 'YYYY-MM-DD'), 604);
INSERT INTO ReorderRequest VALUES (705, TO_DATE('2024-01-21', 'YYYY-MM-DD'), 605);
INSERT INTO ReorderRequest VALUES (706, TO_DATE('2024-01-22', 'YYYY-MM-DD'), 606);
INSERT INTO ReorderRequest VALUES (707, TO_DATE('2024-01-23', 'YYYY-MM-DD'), 607);
INSERT INTO ReorderRequest VALUES (708, TO_DATE('2024-01-24', 'YYYY-MM-DD'), 608);
INSERT INTO ReorderRequest VALUES (709, TO_DATE('2024-01-25', 'YYYY-MM-DD'), 609);
INSERT INTO ReorderRequest VALUES (710, TO_DATE('2024-01-26', 'YYYY-MM-DD'), 610);
INSERT INTO ReorderRequest VALUES (711, TO_DATE('2024-01-27', 'YYYY-MM-DD'), 611);
INSERT INTO ReorderRequest VALUES (712, TO_DATE('2024-01-28', 'YYYY-MM-DD'), 612);
INSERT INTO ReorderRequest VALUES (713, TO_DATE('2024-01-29', 'YYYY-MM-DD'), 613);
INSERT INTO ReorderRequest VALUES (714, TO_DATE('2024-01-30', 'YYYY-MM-DD'), 614);
INSERT INTO ReorderRequest VALUES (715, TO_DATE('2024-01-31', 'YYYY-MM-DD'), 615);
INSERT INTO ReorderRequest VALUES (716, TO_DATE('2024-02-01', 'YYYY-MM-DD'), 616);
INSERT INTO ReorderRequest VALUES (717, TO_DATE('2024-02-02', 'YYYY-MM-DD'), 617);
INSERT INTO ReorderRequest VALUES (718, TO_DATE('2024-02-03', 'YYYY-MM-DD'), 618);
INSERT INTO ReorderRequest VALUES (719, TO_DATE('2024-02-04', 'YYYY-MM-DD'), 619);
INSERT INTO ReorderRequest VALUES (720, TO_DATE('2024-02-05', 'YYYY-MM-DD'), 620);

-- Insert into InventoryReorder table
INSERT INTO InventoryReorder VALUES (601, 701);
INSERT INTO InventoryReorder VALUES (602, 702);
INSERT INTO InventoryReorder VALUES (603, 703);
INSERT INTO InventoryReorder VALUES (604, 704);
INSERT INTO InventoryReorder VALUES (605, 705);
INSERT INTO InventoryReorder VALUES (606, 706);
INSERT INTO InventoryReorder VALUES (607, 707);
INSERT INTO InventoryReorder VALUES (608, 708);
INSERT INTO InventoryReorder VALUES (609, 709);
INSERT INTO InventoryReorder VALUES (610, 710);
INSERT INTO InventoryReorder VALUES (611, 711);
INSERT INTO InventoryReorder VALUES (612, 712);
INSERT INTO InventoryReorder VALUES (613, 713);
INSERT INTO InventoryReorder VALUES (614, 714);
INSERT INTO InventoryReorder VALUES (615, 715);
INSERT INTO InventoryReorder VALUES (616, 716);
INSERT INTO InventoryReorder VALUES (617, 717);
INSERT INTO InventoryReorder VALUES (618, 718);
INSERT INTO InventoryReorder VALUES (619, 719);
INSERT INTO InventoryReorder VALUES (620, 720);
```

DML2 (Other DML Skills)

```
-- Update the contact information for a patient
UPDATE Patient SET ContactInfo = '999-8888' WHERE PatientID = 2;



-- Delete a specific appointment
DELETE FROM Appointment WHERE AppointmentID = 101;


-- Select patient and payment information
SELECT p.PatientID, p.PatientName, p.ContactInfo, pa.PaymentID, pa.PaymentMethod,
pa.AmountPaid
FROM Patient p
JOIN Payment pa ON p.PatientID = pa.PatientID;
```

```
-- Truncate the MedicalRecord table
TRUNCATE TABLE MedicalRecord;
```

```
-- Add a new column to the Patient table
ALTER TABLE Patient
ADD Gender VARCHAR2(10);
```

DML3 (JOIN TABLES)

```
-- JOINING TABLES APPOINTMENT, PATIENT, SPECIALIST
SELECT *
FROM Appointment
INNER JOIN Patient ON Appointment.PatientID = Patient.PatientID
INNER JOIN Specialist ON Appointment.SpecialistID = Specialist.SpecialistID;
```

```
-- JOINING TABLES TREATMENT, INOVICE, PAYMENT
SELECT *
FROM Treatment
INNER JOIN Invoice ON Treatment.TreatmentID = Invoice.TreatmentID
INNER JOIN Payment ON Invoice.InvoiceID = Payment.InvoiceID;
```

SUMMARY:

In summary, our team has successfully executed the comprehensive development of the database conceptual and logical designs tailored for the Ajman University Private Dental Clinic. Throughout this process, we meticulously refined our business rules, meticulously crafted the conceptual Entity-Relationship Diagram (ERD), intricately designed the logical ERD, conducted a thorough Normalization process, meticulously revised our Data Dictionary, presented the Relational Database Schemas post-normalization, and meticulously formulated the SQL statements to bring our database to fruition.