

**SVKM's NMIMS**

**School of Technology Management & Engineering (Indore Campus)**

**Computer Engineering Department (B Tech CE Sem IV)**

**Database Management System**

**Project Report**

Program	BTech CE (B)	
Semester	IV	
Name of the Project:	SKIN DOCTOR APPOINTMENT	
Details of Project Members		
Batch	Roll No.	Name
02	D058	Pal Rathore
02	D066	Priyal Rathore
02	D073	Rishabh Ahuja
Date of Submission:		

**Contribution of each project Members:**

Roll No.	Name	Contribution
D058	Pal Rathore	ER diagram, Relational Schema, Normalization, SQL Queries
D066	Priyal Rathore	ER diagram, Relational Schema, Connectivity
D073	Rishabh Ahuja	ER diagram, Connectivity, SQL Queries



**GitHub Repository:**

[https://github.com/rishabhahuja12/DBMS\\_Project\\_D073\\_D066\\_D058.git](https://github.com/rishabhahuja12/DBMS_Project_D073_D066_D058.git)

# **Project Report**

## **Skin Doctor Appointment**

**by**

**Pal Rathore, Roll number: D058**

**Priyal Rathore, Roll number: D066**

**Rishabh Ahuja, Roll number: D073**

**Course: DBMS**

**AY: 2024-25**

## **Table of Contents**

<b>Sr no.</b>	<b>Topic</b>	<b>Page no.</b>
1	Storyline	
2	Components of Database Design	
3	Entity Relationship Diagram	
4	Relational Model	
5	Normalization	
6	SQL Queries	
7	Learning from the Project	
8	Project Demonstration	
9	Self-learning beyond classroom	
10	Learning from the project	
8	Challenges faced	
9	Conclusion	

# I. Storyline

## **Context & Motivation**

Skinfluence is conceived as an end-to-end online solution designed to transform the way patients interact with dermatology services. In today's fast-paced world, long waiting times and fragmented health records are common challenges. SkinFluence addresses these issues by integrating appointment scheduling, medical record management, test reporting, prescription handling, and even medicine ordering into one seamless platform.

## **Narrative Journey**

Consider the journey of Maya, a young professional dealing with recurring skin issues. Frustrated by the inconvenience of traditional clinic visits, she discovers SkinFluence—a user-friendly website that promises a streamlined experience.

### 1. Patient Registration:

Upon signing up, User's details (such as Name, Gender, DateOfBirth, ContactNumber, and Address) are stored in the Patient table, creating a unique PatientID for her. This ensures user's information is kept securely and can be referenced throughout the system.

### 2. Doctor & Clinic Affiliation:

User chooses Doctor —a top-rated dermatologist listed in the Doctor table. Dr. Sharma's information, including DoctorID, Specialization, and Credentials, is maintained in the system. Doctor is affiliated with a particular Clinic/Facility (e.g., "Skinfluence Care Center"), where details like FacilityName and Address are recorded. The system also keeps track of Doctor Schedule entries, ensuring that Doctor is available AppointmentDate, StartTime, and EndTime match user's preferred slot.

### 3. Booking an Appointment:

User logs in to the platform and books an Appointment—linking PatientID to DoctorID. The Appointment table captures key data such as AppointmentDate, AppointmentTime, Status (e.g., Scheduled).

**4. Medical\_Record Creation:**

On the day of user's appointment, all details of user's consultation are captured in a Medical\_Record, which references the AppointmentID. This record stores the Diagnosis, Treatment, and relevant Notes. Meanwhile, User's overall Patient Medical History is updated to reflect the new condition and treatment status.

**5. Lab Tests & Reports:**

If Doctor recommends lab tests, the system generates one or more Test Reports linked to the Medical Record via the RecordID. Each Test Report includes TestType, TestResults and TestDate. This ensures that all diagnostic data is consolidated in one place.

**6. Prescription & Medicines:**

Doctor prescribes medication, which is recorded in the Prescription table. The PrescriptionID references the Medical Record, detailing the MedicineDetails, DosageInstructions, and IssueDate. User can view this prescription online, where each Medicine is listed with its MedicineName, Description, Price, and Stock availability.

**7. Order & Payment:**

User decides to purchase her prescribed medicines through the SkinFluence platform. She places an Order that references her PatientID and includes the OrderDate, TotalAmount, and OrderStatus. Once user proceeds to checkout, a Payment record is generated—linking to the Order—to capture the PaymentMethod, Amount, and PaymentDate. The TransactionID is stored for external reference if needed.

## **II. Components of Database Design**

### **Entities & Attributes**

1. Patient (Strong Entity)
  - PatientID(Primary Key, INT)
  - Name (Composite attribute consisting of FirstName, LastName;VARCHAR)
  - DateOfBirth (DATE)
  - Age (Derived attribute, calculated from DateOfBirth,INT)
  - Gender (VARCHAR)
  - ContactNumber (VARCHAR)
  - Email (VARCHAR)
  - Address (Composite attribute: Street, City;VARCHAR)
2. Doctor (Strong Entity)
  - DoctorID (Primary Key, INT)
  - Name (Composite attribute consisting of FirstName, LastName;VARCHAR)
  - Specialization (VARCHAR)
  - ContactNumber (VARCHAR)
  - Email (VARCHAR)
  - Credentials (Multi-valued attribute, VARCHAR)
  - YearsOfExperience (INT)
3. Appointment (Strong Entity)
  - AppointmentID (Primary Key, INT)
  - Date (DATE)
  - Time (TIME)
  - Status (VARCHAR-Scheduled/Completed/Cancelled)
4. Medical\_Record (Strong Entity)
  - RecordID (Primary Key, INT)
  - Diagnosis (VARCHAR)
  - Treatment (VARCHAR)
  - VisitDate (DATE)

5. Test\_Report (Weak Entity)

- RecordID (INT)
- TestType (VARCHAR)
- TestResult (VARCHAR)
- TestDate (DATE)

6. Prescription (Strong Entity)

- PrescriptionID (Primary Key, INT)
- IssueDate (DATE)
- MedicationDetails (VARCHAR)
- Dosage (VARCHAR)

7. Medicine (Strong Entity)

- MedicineID (Primary Key, INT)
- MedicineName (VARCHAR)
- Description (VARCHAR)
- Price (FLOAT)
- Stock (INT)

8. Order (Strong Entity)

- OrderID (Primary Key, INT)
- OrderDate (DATE)
- TotalAmount (FLOAT)
- OrderStatus (VARCHAR- Pending/Shipped/Delivered/Cancelled)

9. Payment (Strong Entity)

- PaymentID (Primary Key, INT)
- PaymentDate (DATE)
- Amount (FLOAT)
- PaymentMethod (VARCHAR-Credit Card/Cash/Online)
- TransactionID (VARCHAR)

10. Clinic (Strong Entity)

- ClinicID (Primary Key, INT)
- ClinicName (VARCHAR)
- ClinicAddress (Composite attribute: Street, City; VARCHAR)
- PhoneNumber (VARCHAR)
- ClinicEmail (VARCHAR)

11. Doctor\_Schedule (Weak Entity)

- DoctorID (INT)
- AvailDate (DATE)
- StartTime (TIME)
- EndTime (TIME)
- SlotDuration (INT)

12. Patient\_Medical\_History (Weak Entity)

- PatientID (INT)
- Condition (VARCHAR)
- DiagnosedDate (DATE)
- CurrentStatus (VARCHAR- Active/Resolved/Chronic)

## **Relationships & Cardinalities**

### **1. Patient → books → Appointment**

- Entities: Patient (1) → Appointment (N)
- Linking Attribute: PatientID in Appointment
- Cardinality: 1: N
- Description: A Patient can book multiple Appointments.

### **2. Doctor → assigned → Appointment**

- Entities: Doctor (1) → Appointment (N)
- Linking Attribute: DoctorID in Appointment
- Cardinality: 1: N
- Description: A Doctor can handle many Appointments.

### **3. Appointment → generates → Medical\_Record (Identifying)**

- Entities: Appointment (1) → Medical Record (1)
- Linking Attribute: AppointmentID in Medical Record
- Cardinality: 1:1
- Description: Each Appointment generates one Medical Record.

### **4. Medical\_Record → has → Test\_Report (Identifying)**

- Entities: Medical Record (1) → Test Report (N)
- Linking Attribute: RecordID in Test Report
- Cardinality: 1: N
- Description: A single Medical Record may have multiple Test Reports.

### **5. Medical\_Record → creates → Prescription (Identifying)**

- Entities: Medical Record (1) → Prescription (N)
- Linking Attribute: RecordID in Prescription
- Cardinality: 1: N
- Description: One Medical Record can produce multiple Prescriptions.

## **6. Patient → places → Order**

- Entities: Patient (1) → Order (N)
- Linking Attribute: PatientID in Order
- Cardinality: 1: N
- Description: A Patient may place multiple Orders for Medicines.

## **7. Order → have → Payment**

- Entities: Order (1) → Payment (1)
- Linking Attribute: OrderID in Payment
- Cardinality: 1:1
- Description: Each Order is linked to one Payment transaction.

## **8. Doctor → affiliated\_with → Clinic**

- Entities: Doctor (N) → Clinic (1)
- Linking Attribute: ClinicID in Doctor
- Cardinality: 1:1
- Description: Each Doctor is affiliated with one Clinic.

## **9. Doctor → is\_alloted → Schedule (Identifying)**

- Entities: Doctor (1) → Doctor Schedule (N)
- Linking Attribute: DoctorID in Doctor Schedule
- Cardinality: 1: N
- Description: A Doctor's availability is recorded in multiple schedule entries.

## **10. Patient → maintain → Patient\_Medical\_History**

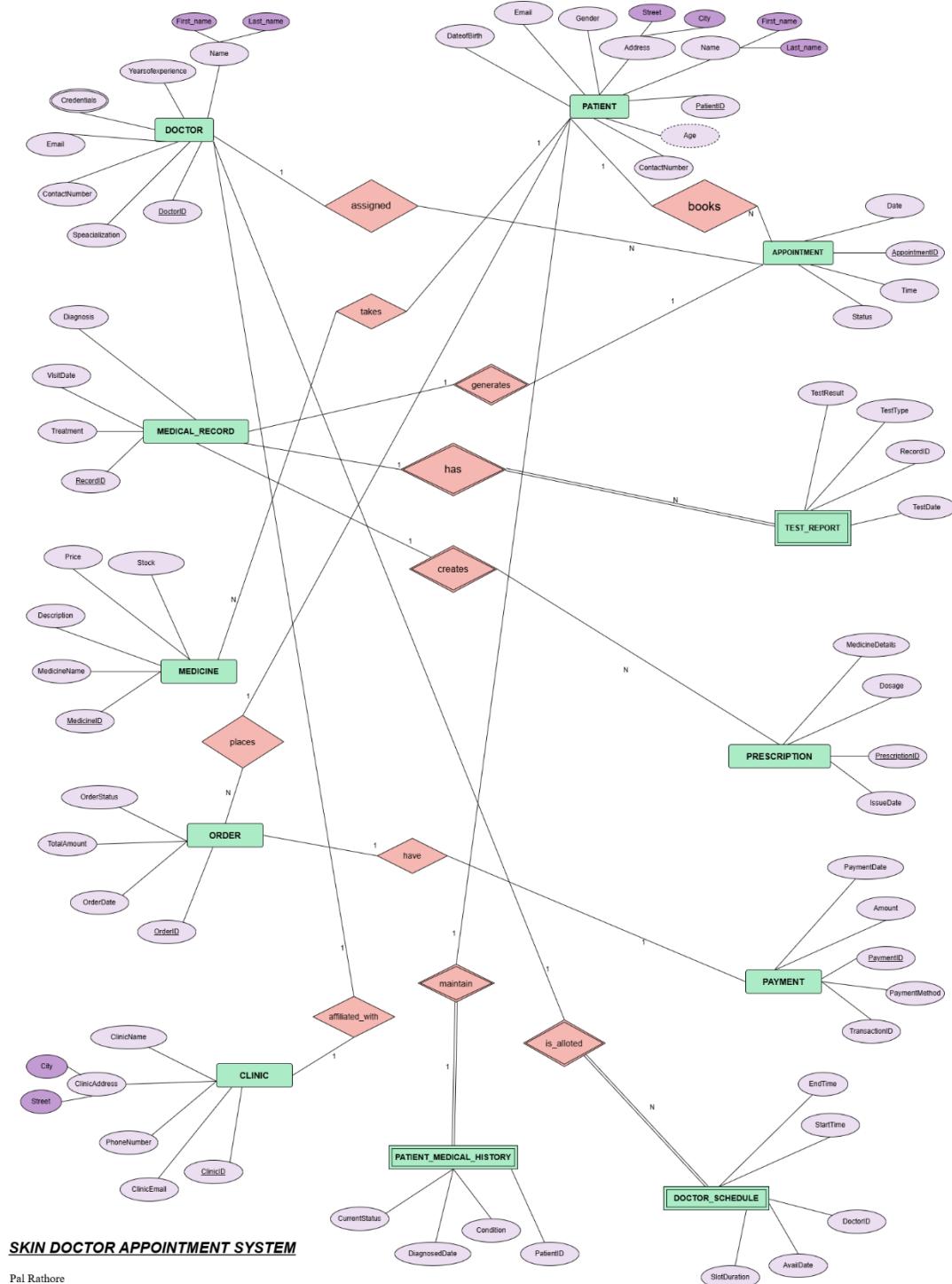
- Entities: Patient (1) → Patient Medical History (1)
- Linking Attribute: PatientID in Patient Medical History
- Cardinality: 1: 1
- Description: A Patient have one historical record reflecting its conditions.

## **11. Patient → takes → Medicine**

- Entities: Patient (1) → Medicine (N)
- Linking Attribute: MedicineID in Medicine
- Cardinality: 1: N
- Description: A Patient can take one or more medicine.

### III. Entity Relationship Diagram

Following is the ER Diagram for Skin Doctor Appointment System:



## IV. Relational Model

ER diagram as relational model:

### **TABLES FOR ENTITIES:**

#### **1.DOCTOR**

DoctorID	First_Name	Last_Name	Email	ContactNumber	Specialisation	YearofExperience	Credentials

***Schema:***

Doctor(DoctorID,First\_Name,Last\_Name,Email,ContactNumber,Specialisation, YearofExperience,Credentials)

#### **2.PATIENT**

PatientID	Name	Age	Street	City	First_name	Last_name	Email	DateofBirth	ContactNumber	Gender

***Schema:*** Patient (PatientID, Name, Age, Street,City,First\_name,Last\_name, Email, DateofBirth,ContactNumber,Gender)

#### **3.MEDICINE**

MedicineID	MedicineName	Description	Price	Stock

***Schema:*** Medicine (MedicineID,MedicineName,Description,Price,Stock)

#### 4.ORDER

<u>OrderID</u>	TotalAmount	OrderDate	OrderStatus
----------------	-------------	-----------	-------------

**Schema:** Order (OrderID,TotalAmount,OrderData, OrderStatus)

#### 5.CLINIC

<u>ClinicID</u>	ClinicEmail	ClinicName	City	Street	PhoneNumber
-----------------	-------------	------------	------	--------	-------------

**Schema:** Clinic (ClinicID,ClinicEmail,ClinicName,City,Street,PhoneNumber)

#### 6.APPOINTMENT

<u>AppointmentID</u>	Date	Time	Status
----------------------	------	------	--------

**Schema:** Appointment (AppointmentID,Date,Time,Status)

#### 7.PAYMENT

<u>PaymentID</u>	Amount	PaymentDate	TransactionID	OrderID	PaymentMethod
------------------	--------	-------------	---------------	---------	---------------

**Schema:**

Payment (PayemntID, Amount, PaymentData, TransactionID,OrderID,PaymentMethod)

## **8.MEDICAL\_RECORD**

<u>RecordID</u>	Diagonisis	VisitDate	Treatment
[REDACTED]			

### ***Schema:***

Medical\_Record (RecordID,Diagonisis,VisitDate,Treatment)

## **9.TEST\_REPORT**

<u>RecordID</u>	TestType	TestResult	TestDate
[REDACTED]			

### ***Schema:*** Test\_Report (RecordID,TestType,TestResult,TestDate)

## **10.PRESCRIPTION**

<u>PrescriptionID</u>	Dosage	IssueDate	MedicineDetails
[REDACTED]			

### ***Schema:*** Prescription (PrescriptionID,Dosage,IssueDate,MedicineDetails)

## **11.PATIENT\_MEDICAL\_HISTORY**

PatientID	Condition	DiagnosedDate	CurrentStatus
[REDACTED]			

### ***Schema:***

Patient\_Medical\_History (PatientID,Condition,DiagnosedData,CurrentStatus)

## **12.DOCTOR\_SCHEDULE**

<b>DoctorID</b>	<b>EndTime</b>	<b>StartTime</b>	<b>AvailDate</b>	<b>SlotDuration</b>
-----------------	----------------	------------------	------------------	---------------------

### ***Schema:***

Doctor\_Schedule (DoctorID,EndTime,StartTime,AvailDate,SlotDuration)

## **TABLES FOR RELATIONSHIP SETS:**

### 1. books

Patient\_books\_Appointment:

- PATIENT (PatientID) → APPOINTMENT (PatientID)
- Cardinality: 1: N
- Entity Types: Both Patient and Appointment are strong entities.

Here, two tables will be there-

1. A – **SAME AS PATIENT TABLE**

2. BR (Combined Table)

**TABLE-**

PatientID	AppointmentID	Date	Time	Status

### 2. assigned

Doctor\_assigned\_Appointment:

- DOCTOR (DoctorID) → APPOINTMENT (DoctorID)
- Cardinality: 1: N
- Entity Types: Both Doctor and Appointment are strong entities.

Here, two tables will be there-

1. A – **SAME AS DOCTOR TABLE**

2. BR (Combined Table)

**TABLE-**

DoctorID	AppointmentID	Date	Time	Status

### 3. places

Patient\_places\_Order:

- PATIENT (PatientID) → ORDER (PatientID)
- Cardinality: 1: N
- Entity Types: Both Patient and Order are strong entities.

Here, two tables will be there-

1. A – **SAME AS PATIENT TABLE**

2. BR (Combined Table)

**TABLE-**

PatientID	OrderID	OrderDate	TotalAmount	OrderStatus

### 4. takes

Patient\_takes\_Medicine:

- PATIENT (PatientID) → MEDICINE (MedicineID)
- Cardinality: 1: N
- Entity Types: Both Patient and Medicine are strong entities.

Here, two tables will be there-

1. A – **SAME AS PATIENT TABLE**

2. BR (Combined Table)

PatientID	MedicineID	MedicineName	Description	Price	Stock

### 5. have

Order\_have\_Payment:

- ORDER (OrderID) → PAYMENT (OrderID)
- Cardinality: 1:1
- Entity Types: Both Order and Payment are strong entities.

Here, two tables will be there-

1. A – **SAME AS ORDER TABLE**

2. BR (Combined Table)

**TABLE-**

OrderID	PaymentID	PaymentDate	Amount	PaymentMethod	TransactionID

6. affiliated\_with

Doctor\_affiliated\_With\_Clinic:

- DOCTOR (DoctorID) → CLINIC(ClinicID)
- Cardinality: 1:1
- Entity Types: Both Doctor and Clinic are strong entities.

Here, two tables will be there-

1. A – **SAME AS DOCTOR TABLE**

2. BR (Combined Table)

**TABLE-**

DoctorID	ClinicID	ClinicEmail	PhoneNumber	ClinicAddress	ClinicName

7. generates

Appointment\_generates\_Medical\_Record:

- APPOINTMENT (AppointmentID) → MEDICAL\_RECORD (AppointmentID)
- Cardinality: 1:1
- Entity Types: Medical\_Record is a weak entity and Appointment is the identifying relationship.

Weak entity sets are always used in association with identifying relationship.

Here, two tables will be there-

1. A – **SAME AS APPOINTMENT TABLE**

2. BR (Combined Table)

AppointmentID	Diagnosis	VisitDate	Treatment	RecordID

8. has

Medical\_Record\_has\_Test\_Report:

- MEDICAL\_RECORD (AppointmentID, RecordID) → TEST\_REPORT (AppointmentID, RecordID)
- Cardinality: 1: N
- Entity Types: Test\_Report is a weak entity and Medical\_Record is the identifying relationship.

Weak entity sets are always used in association with identifying relationship.

Here, two tables will be there-

1. A – **SAME AS MEDICAL\_RECORD TABLE**

2. BR (Combined Table)

RecordID	TestType	TestResult	TestDate
[REDACTED]			

9. creates

Medical\_RecordCreates\_Prescription:

- MEDICAL\_RECORD (AppointmentID, RecordID) → PRESCRIPTION(AppointmentID, RecordID)
- Cardinality: 1: N
- Entity Types: Prescription is a weak entity and Medical\_Record is the identifying relationship.

Weak entity sets are always used in association with identifying relationship.

Here, two tables will be there-

1. A – **SAME AS MEDICAL\_RECORD TABLE**

2. BR (Combined Table)

RecordID	MedicineDetails	Dosage	PrescriptionID	IssueDate
[REDACTED]				

## 10. maintain

Patient\_maintain\_Patient\_Medical\_History:

- PATIENT (PatientID) → PATIENT\_MEDICAL\_HISTORY (PatientID)
- Cardinality: 1:1
- Entity Types: Patient\_Medical\_History is a weak entity and Patient is the identifying relationship.

Weak entity sets are always used in association with identifying relationship.

Here, two tables will be there-

1. A – **SAME AS PATIENT TABLE**
2. BR (Combined Table)

PatientID	CurrentStatus	DiagnosedDate	Condition

## 11. is\_alloted

Doctor\_is\_alloted\_Doctor\_Schedule:

- DOCTOR (DoctorID) → DOCTOR\_SCHEDULE (DoctorID)
- Cardinality: 1: N
- Entity Types: Doctor\_Schedule is a weak entity and Doctor is the identifying relationship.

Weak entity sets are always used in association with identifying relationship.

Here, two tables will be there-

1. A – **SAME AS DOCTOR TABLE**
2. BR (Combined Table)

DoctorID	EndTime	StartTime	AvailDate	SlotDuration

# V. Normalization

Normalization in DBMS (Database Management System) is the process of organizing data in a database to reduce redundancy (duplicate data) and improve data integrity.

It involves dividing large tables into smaller ones and defining relationships between them using foreign keys. The goal is to ensure that each piece of data is stored only once, and updates, deletions, and insertions are done efficiently without anomalies.

**The normal forms are:**

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)

## RULES FOR FIRST NORMAL FORM (1NF):

- Each table cell must contain only a single (atomic) value.
- Each column must have a unique name.
- All values in a column must be of the same data type
- The table must not contain any repeating groups or columns.

**Goal:** Eliminate multi-valued or composite attributes.

## RULES FOR SECOND NORMAL FORM (2NF):

- The table must already be in First Normal Form (1NF).
- There should be no partial dependency—non-key attributes must depend on the entire primary key.

**Goal:** Eliminate partial dependency.

## RULES FOR THIRD NORMAL FORM (3NF):

- The table must already be in Second Normal Form (2NF).
- There should be no transitive dependency—non-key attributes must not depend on other non-key attributes.

**Goal:** Eliminate transitive dependency and ensure that every non-key attribute depends only on the primary key.

## NORMALIZATION OF TABLES:

### 1. Doctor

#### **Schema:**

Doctor (DoctorID, First\_Name, Last\_Name, Email, ContactNumber, Specialisation, YearofExperience, Credentials)

**Primary Key:** DoctorID

**1NF:** It has atomic values.

**2NF:** All non-key attributes depend on DoctorID.

**3NF:** No transitive dependencies (e.g., First\_Name doesn't determine Email).

Therefore, the table is already in **3NF**.

Attribute	Type	Key
DoctorID	INT	Primary Key
First_Name	VARCHAR	
Last_Name	VARCHAR	
Email	VARCHAR	
ContactNumber	VARCHAR	
Specialisation	VARCHAR	
YearofExperience	INT	
Credentials	VARCHAR	

## 2. Patient

### Schema:

Patient (PatientID, Name, Age, Street, City, First\_Name, Last\_Name, Email, DOB, ContactNumber, Gender)

**Primary Key:** PatientID

- **1NF:** It has atomic values.
- **2NF:** All non-key attributes fully depend on PatientID.
- **3NF:** Not in 3NF

Following points violates 3NF

- Redundancy: Name is derived from First\_Name + Last\_Name
- Repeating address fields in the table
- **Transitive Dependency:** First\_Name, Last\_Name → Name (derived dependency)

To make this in 3NF we remove name and separate address; split into two tables:

❖ Patient(PatientID, First\_Name, Last\_Name, Age, DOB, ContactNumber, Email, Gender)

Attribute	Type	Key
PatientID	INT	Primary Key
First_Name	VARCHAR	
Last_Name	VARCHAR	
DateOfBirth	DATE	
Age	INT	
ContactNumber	VARCHAR	
Email	VARCHAR	
Gender	VARCHAR	

❖ Patient\_Address(PatientID, Street, City)

Attribute	Type	Key
PatientID	INT	Foreign Key
Street	VARCHAR	
City	VARCHAR	

### 3. Medicine

#### Schema:

Medicine(MedicineID, MedicineName, Description, Price, Stock)

**Primary Key:** MedicineID

- **1NF:** It has atomic values.
- **2NF:** All non-key attributes depend on MedicineID.
- **3NF:** No transitive dependencies.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
MedicineID	INT	Primary Key
MedicineName	VARCHAR	
Description	VARCHAR	
Price	FLOAT	
Stock	INT	

## 4. Order

### Schema:

Order(OrderID, TotalAmount, OrderDate, OrderStatus)

### Primary Key: OrderID

- **1NF:** It has atomic values.
- **2NF:** All non-key attributes depend on OrderID.
- **3NF:** No transitive dependencies.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
OrderID	INT	Primary Key
TotalAmount	FLOAT	
OrderDate	DATE	
OrderStatus	VARCHAR	

## 5. Clinic

### Schema:

Clinic(ClinicID, ClinicEmail, ClinicName, City, Street, PhoneNumber)

### Primary Key: ClinicID

- **1NF:** It has atomic values — no multivalued or repeating groups.
- **2NF:** All non-key attributes fully depend on the primary key ClinicID.
- **3NF:** There are no transitive dependencies between non-key attributes.  
Although City and Street could be grouped logically as an address, there is no functional dependency between them (e.g., Street → City), so it doesn't violate 3NF.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
ClinicID	INT	Primary Key
ClinicEmail	VARCHAR	
ClinicName	VARCHAR	
City	VARCHAR	
Street	VARCHAR	
PhoneNumber	VARCHAR	

## 6. Appointment

### Schema:

Appointment(AppointmentID, Date, Time, Status)

**Primary Key:** AppointmentID

- **1NF:** It has atomic values.
- **2NF:** All non-key attributes depend on AppointmentID.
- **3NF:** No transitive dependencies.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
AppointmentID	INT	Primary Key
Date	DATE	
Time	TIME	
Status	VARCHAR	

## 7. Payment

### Schema:

Payment(PaymentID, Amount, PaymentDate, TransactionID, OrderID, PaymentMethod)

**Primary Key:** PaymentID

- **1NF:** It has atomic values.
- **2NF:** All non-key attributes depend on primary key.
- **3NF:** No transitive dependencies.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
PaymentID	INT	Primary Key
Amount	FLOAT	
PaymentDate	DATE	
TransactionID	VARCHAR	
OrderID	INT	Foreign Key
PaymentMethod	VARCHAR	

## 8. Medical\_Record

### Schema:

Medical\_Record(RecordID, Diagnosis, VisitDate, Treatment)

**Primary Key:** RecordID

- **1NF:** It has atomic values.
- **2NF:** All non-key attributes depend on primary key.
- **3NF:** No transitive dependencies.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
RecordID	INT	Primary Key
Diagnosis	VARCHAR	
VisitDate	DATE	
Treatment	VARCHAR	

## 9. Test\_Report

### Schema:

Test\_Report(RecordID, TestType, TestResult, TestDate)

**Foreign Key:** RecordID → Medical\_Record

**Composite Primary Key (assumed):** RecordID, TestType

*(Since one medical record can have multiple tests, and each test is uniquely identified by its type for that record)*

- **1NF:** All attributes are atomic.
- **2NF:** All non-key attributes (TestResult, TestDate) fully depend on the whole composite key (RecordID, TestType).
- **3NF:** There are no transitive dependencies among non-key attributes.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
RecordID	INT	Foreign Key
TestType	VARCHAR	Primary Key
TestResult	VARCHAR	
TestDate	DATE	

## 10. Prescription

### Schema:

Prescription(PrescriptionID, Dosage, IssueDate, MedicineDetails)

- **1NF:** Violation of 1NF (atomicity rule) as MedicineDetails likely contains multiple medicines which makes it non-atomic.

To make this in 1NF split MedicineDetails into a separate relation to ensure atomic values.

The new tables formed:

### ❖ New Table 1: Prescription

#### Schema:

Prescription(PrescriptionID, Dosage, IssueDate)

**Primary Key:** PrescriptionID

- **1NF:** All values are atomic
- **2NF:** All attributes depend on the primary key
- **3NF:** No transitive dependencies

Therefore, the table is already in **3NF**.

Attribute	Type	Key
PrescriptionID	INT	Primary Key
Dosage	VARCHAR	
IssueDate	DATE	

### ❖ New Table 2: Prescription\_Medicine

#### Schema:

Prescription\_Medicine(PrescriptionID, MedicinID, Quantity)

**Composite Primary Key:** (PrescriptionID, MedicinID)

**Foreign Keys:**

PrescriptionID → Prescription

MedicinID → Medicine

- **1NF:** Atomic values.
- **2NF:** Quantity depends on the whole primary key.
- **3NF:** No transitive dependencies.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
PrescriptionID	INT	Composite PK
MedicineID	INT	Composite PK
Quantity	INT	

## 11. Patient\_Medical\_History

### Schema:

Patient\_Medical\_History(PatientID, Condition, DiagnosedDate, CurrentStatus)

### Composite Primary Key: (PatientID, Condition, DiagnosedDate)

- **1NF:** The table has atomic values.
- **2NF:** The non-key attribute CurrentStatus depends on the full composite key.
- **3NF:** There are no transitive dependencies among non-key attributes.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
PatientID	INT	Foreign Key
Condition	VARCHAR	Primary Key
DiagnosedDate	DATE	
CurrentStatus	VARCHAR	

## 12. Doctor\_Schedule

### Schema:

Doctor\_Schedule(DoctorID, EndTime, StartTime, AvailDate, SlotDuration)

**Composite Primary Key:** (DoctorID, AvailDate)

- **1NF:** The table has atomic values.
- **2NF:** All non-key attributes depend on the full composite key.
- **3NF:** There are no transitive dependencies among non-key attributes.

Therefore, the table is already in **3NF**.

Attribute	Type	Key
DoctorID	INT	Foreign Key
AvailDate	DATE	Primary Key
StartTime	TIME	
EndTime	TIME	
SlotDuration	INT	

## NORMALIZATION OF RELATIONSHIP SET:

### **1. Patient\_books\_Appointment**

#### **Schema:**

Patient\_books\_appointment(PatientID,AppointmentID,Date,Time,Status)

**Composite Primary Key:** (PatientID, AppointmentID)

**1NF:** The table has atomic values.

**2NF:** All non-key attributes depend on the full composite key.

**3NF:** There are no transitive dependencies among non-key attributes.

Therefore, the table is already in **3NF**.

### **2. Doctor\_assigned\_Appointment**

#### **Schema:**

Doctor\_assigned\_Appointment (DoctorID, AppointmentID, Date, Time, Status )

**Composite Primary Key:** (DoctorID, AppointmentID)

**1NF:** The table has atomic values.

**2NF:** All non-key attributes depend on the full composite key.

**3NF:** There are no transitive dependencies among non-key attributes.

Therefore, the table is already in **3NF**.

### **3. Patient\_places\_Order**

#### **Schema:**

Patient\_places\_Order (PatientID , OrderID , OrderDate , TotalAmount , OrderStatus)

**Composite Key:** (PatientID, OrderID)

**1NF:** The table has atomic values.

**2NF:** All non-key attributes depend on the full composite key.

**3NF:** Not in 3NF as,

OrderDate, TotalAmount, and OrderStatus depend only on OrderID → **transitive dependency** via OrderID.

For 3NF, split into two tables:

**TABLE 1:** Patient\_Order(PatientID, OrderID)

**TABLE 2:** Order(OrderID, OrderDate, TotalAmount, OrderStatus)

(Already exists separately)

Normalized to **3NF**

#### **4. Patient\_takes\_Medicine**

**Schema:**

Patient\_takes\_Medicine (PatientID , MedicinelD , MedicineName , Description , Price , Stock)

**1NF:** The table has atomic values.

**2NF:** All non-key attributes are independent of each other.

**3NF:** Not in 3NF as,

MedicineName, Description, Price, Stock depend only on MedicinelD → **transitive dependency**

For 3NF, split into two tables:

**TABLE 1:** Patient\_Medicine(PatientID, MedicinelD)

**TABLE 2:** Medicine(MedicinelD, MedicineName, Description, Price, Stock)

(Already exists separately)

Normalized to **3NF**

#### **5. Order\_have\_Payment**

**Schema:**

Order\_have\_Payment (OrderID , PaymentID , PaymentDate , Amount , PaymentMethod , TransactionID)

**1NF:** The table has atomic values.

**2NF:** All non-key attributes are independent of each other.

**3NF:** Not in 3NF as,

Attributes depend on PaymentID, not on OrderID → **transitive dependency**

For 3NF, split into two tables:

**TABLE 1:** Order\_Payment(OrderID, PaymentID)

**TABLE 2:** Payment(PaymentID, PaymentDate, Amount, PaymentMethod, TransactionID)  
(Already exists)

Normalized to **3NF**

## **6. Doctor\_affiliated\_with\_Clinic**

**Schema:**

Doctor\_affiliated\_with\_Clinic (DoctorID , ClinicID , ClinicEmail, PhoneNumber ,  
ClinicAddress , ClinicName)

**1NF:** The table has atomic values.

**2NF:** All non-key attributes are independent of each other.

**3NF:** Not in 3NF as,

Clinic attributes depend on ClinicID only → **transitive dependency**

For 3NF, split into two tables:

**TABLE 1:** Doctor\_Clinic(DoctorID, ClinicID)

**TABLE 2:** Clinic(ClinicID, ClinicEmail, PhoneNumber, ClinicAddress, ClinicName)  
(Already exists)

Normalized to **3NF**

## **7. Appointment\_generates\_Medical\_Record**

**Schema:**

Appointment\_generates\_Medical\_Record (AppointmentID, Diagnosis, VisitDate,  
Treatment, RecordID)

**Assumption:** RecordID uniquely identifies a medical record. All other fields depend on it.

**1NF:** The table has atomic values.

**2NF:** All non-key attributes depend on the full composite key.

**3NF:** There are no transitive dependencies among non-key attributes.

Therefore, the table is already in **3NF**.

## **8. Medical\_Record\_has\_Test\_Report**

### **Schema:**

Medical\_Record\_has\_Test\_Report(RecordID , TestType , TestResult , TestDate )

**1NF:** The table has atomic values.

**2NF:** All non-key attributes depend on the full assumed primary key(RecordID).

**3NF:** There are no transitive dependencies among non-key attributes.

Therefore, the table is already in **3NF**.

## **9. Medical\_RecordCreates\_Prescription**

### **Schema:**

Medical\_RecordCreates\_Prescription(RecordID , MedicineDetails , Dosage , PrescriptionID , IssueDate)

**1NF:** Not in 1NF as,

MedicineDetails likely non-atomic → **violates 1NF**

For 1NF, split into two tables:

**TABLE 1:** Prescription(PrescriptionID, RecordID, Dosage, IssueDate)

**TABLE 2:** Prescription\_Medicine(PrescriptionID, MedicinID, Quantity)

Normalized to **3NF**

## **10. Patient\_maintain\_Patient\_Medical\_History**

### **Schema:**

Patient\_maintain\_Patient\_Medical\_History (PatientID , CurrentStatus , DiagnosedDate , Condition)

**Composite Key:** (PatientID, Condition, DiagnosedDate)

**1NF:** The table has atomic values.

**2NF:** All non-key attributes depend on the full composite key.

**3NF:** There are no transitive dependencies among non-key attributes.

Therefore, the table is already in **3NF**.

## **11. Doctor\_is\_alloted\_Doctor\_Schedule**

### **Schema:**

Doctor\_is\_alloted\_Doctor\_Schedule(DoctorID , EndTime , StartTime , AvailDate , SlotDuration )

**Composite Key:** (DoctorID, AvailDate)

**1NF:** The table has atomic values.

**2NF:** All non-key attributes depend on the full composite key.

**3NF:** There are no transitive dependencies among non-key attributes.

Therefore, the table is already in **3NF**.

## **Final Normalized Tables (from Entities)**

### **1. Doctor**

Doctor (DoctorID, First\_Name, Last\_Name, Email, ContactNumber, Specialisation, YearofExperience, Credentials)

Primary Key: DoctorID

---

### **2. Patient**

Patient(PatientID, First\_Name, Last\_Name, Age, DOB, ContactNumber, Email, Gender)

Patient\_Address(PatientID, Street, City)

Primary Key: PatientID

---

### **3. Medicine**

Medicine(MedicineID, MedicineName, Description, Price, Stock)

Primary Key: MedicineID

---

### **4. Order**

Order(OrderID, TotalAmount, OrderDate, OrderStatus)

Primary Key: OrderID

---

### **5. Clinic**

Clinic(ClinicID, ClinicEmail, ClinicName, City, Street, PhoneNumber)

Primary Key: ClinicID

---

### **6. Appointment**

Appointment(AppointmentID, Date, Time, Status)

Primary Key: AppointmentID

---

### **7. Payment**

Payment(PaymentID, Amount, PaymentDate, TransactionID, OrderID, PaymentMethod)

Primary Key: PaymentID

---

### **8. Medical\_Record**

Medical\_Record(RecordID, Diagnosis, VisitDate, Treatment)

Primary Key: RecordID

---

## **9. Test\_Report**

Test\_Report(RecordID, TestType, TestResult, TestDate)

Foreign Key: RecordID → Medical\_Record

Composite Primary Key (assumed): RecordID, TestType

---

## **10. Prescription**

Prescription(PrescriptionID, Dosage, IssueDate)

Primary Key: PrescriptionID

---

## **11. Prescription\_Medicine**

Prescription\_Medicine(PrescriptionID, MedicineID, Quantity)

Composite Primary Key: (PrescriptionID, MedicineID)

Foreign Keys:

PrescriptionID → Prescription

MedicineID → Medicine

---

## **12. Patient\_Medical\_History**

Patient\_Medical\_History(PatientID, Condition, DiagnosedDate, CurrentStatus)

Composite Primary Key: (PatientID, Condition, DiagnosedDate)

---

## **13. Doctor\_Schedule**

Doctor\_Schedule(DoctorID, EndTime, StartTime, AvailDate, SlotDuration)

Composite Primary Key: (DoctorID, AvailDate)

---

## **Final Normalized Tables (from Relationship Sets)**

### **1. Patient\_books\_Appointment**

Patient\_books\_Appointment(PatientID, AppointmentID, Date, Time, Status)

---

### **2. Doctor\_assigned\_Appointment**

Doctor\_assigned\_Appointment(DoctorID, AppointmentID, Date, Time, Status)

---

### **3. Split from Patient\_places\_Order:**

- Patient\_Order(PatientID, OrderID)
  - Order(OrderID, OrderDate, TotalAmount, OrderStatus)  
\*already exists as a separate entity table
- 

### **4. Split from Patient\_takes\_Medicine:**

- Patient\_Medicine(PatientID, MedicinelD)
  - Medicine(MedicinelD, MedicineName, Description, Price, Stock)  
\*already defined
- 

### **5. Split from Order\_have\_Payment:**

- Order\_Payment(OrderID, PaymentID)
  - Payment(PaymentID, PaymentDate, Amount, PaymentMethod, TransactionID)  
\*already exists
- 

### **6. Split from Doctor\_affiliated\_with\_Clinic:**

- Doctor\_Clinic(DoctorID, ClinicID)
  - Clinic(ClinicID, ClinicEmail, PhoneNumber, ClinicAddress, ClinicName)  
\*already exists
- 

### **7. Appointment\_generates\_Medical\_Record**

Appointment\_generates\_Medical\_Record(AppointmentID, Diagnosis, VisitDate, Treatment, RecordID)

---

**8. Medical\_Record\_has\_Test\_Report**

Medical\_Record\_has\_Test\_Report(RecordID, TestType, TestResult, TestDate)

---

**9. Split from Medical\_RecordCreatesPrescription:**

- Prescription(PrescriptionID, RecordID, Dosage, IssueDate)
  - Prescription\_Medicine(PrescriptionID, MedicinelD, Quantity)
- 

**10. Patient\_maintain\_Patient\_Medical\_History**

Patient\_Medical\_History(PatientID, Condition, DiagnosedDate, CurrentStatus)

---

**11. Doctor\_is\_alloted\_Doctor\_Schedule**

Doctor\_Schedule(DoctorID, AvailDate, StartTime, EndTime, SlotDuration)

---

**ORIGINALLY, THERE WERE 23 TABLES**

**NOW, THERE ARE 24 TABLES: (*2 tables already created*)**

## VI. SQL Queries

### CREATING DATABASE

```
mysql> create database skinfluence;
Query OK, 1 row affected (0.15 sec)

mysql> use skinfluence;
Database changed
```

### CREATING TABLES

#### 1. DOCTOR

```
create table doctor(doctorid int primary key, first_name varchar(50), last_name
varchar(50), email varchar(100), contactnumber varchar(15), specialisation varchar(100),
yearofexperience int, credentials varchar(200));
```

```
mysql> create table doctor(doctorid int primary key, first_name varchar(50), last_name varchar(50), email varchar(100),
contactnumber varchar(15), specialisation varchar(100), yearofexperience int, credentials varchar(200));
Query OK, 0 rows affected (0.23 sec)

mysql> desc doctor;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| doctorid | int | NO | PRI | NULL | |
| first_name | varchar(50) | YES | | NULL | |
| last_name | varchar(50) | YES | | NULL | |
| email | varchar(100) | YES | | NULL | |
| contactnumber | varchar(15) | YES | | NULL | |
| specialisation | varchar(100) | YES | | NULL | |
| yearofexperience | int | YES | | NULL | |
| credentials | varchar(200) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.08 sec)
```

## 2. PATIENT

```
create table patient(patientid int primary key, first_name varchar(50), last_name  
varchar(50), age int, dob date, contactnumber varchar(15), email varchar(100), gender  
varchar(10));
```

```
mysql> create table patient(patientid int primary key, first_name varchar(50), last_name varchar(50), age int, dob date,  
contactnumber varchar(15), email varchar(100), gender varchar(10));  
Query OK, 0 rows affected (0.10 sec)  
mysql> desc patient;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| patientid | int | NO | PRI | NULL |  
| first_name | varchar(50) | YES | NULL |  
| last_name | varchar(50) | YES | NULL |  
| age | int | YES | NULL |  
| dob | date | YES | NULL |  
| contactnumber | varchar(15) | YES | NULL |  
| email | varchar(100) | YES | NULL |  
| gender | varchar(10) | YES | NULL |  
+-----+-----+-----+-----+-----+-----+  
8 rows in set (0.00 sec)
```

## 3. PATIENT\_ADDRESS

```
create table patient_address(patientid int, street varchar(100), city varchar(50), foreign  
key(patientid) references patient(patientid));
```

```
mysql> create table patient_address(patientid int, street varchar(100), city varchar(50),  
foreign key(patientid) references patient(patientid));  
Query OK, 0 rows affected (0.14 sec)  
mysql> desc patient_address;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| patientid | int | YES | MUL | NULL |  
| street | varchar(100) | YES | NULL |  
| city | varchar(50) | YES | NULL |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

#### 4. MEDICINE

```
create table medicine(medicineid int primary key, medicinename varchar(100),  
description text, price decimal(10,2), stock int);
```

```
mysql> create table medicine(medicineid int primary key, medicinename varchar(100),  
description text, price decimal(10,2), stock int);  
Query OK, 0 rows affected (0.10 sec)  
  
mysql> desc medicine;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| medicineid | int | NO | PRI | NULL | |  
| medicinename | varchar(100) | YES | | NULL | |  
| description | text | YES | | NULL | |  
| price | decimal(10,2) | YES | | NULL | |  
| stock | int | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

#### 5. ORDER

```
create table order_table(orderid int primary key, totalamount decimal(10,2),  
orderdate date, orderstatus varchar(50));
```

```
mysql> create table order_table(orderid int primary key, totalamount decimal(10,2),  
orderdate date, orderstatus varchar(50));  
Query OK, 0 rows affected (0.08 sec)  
  
mysql> desc order_table;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| orderid | int | NO | PRI | NULL | |  
| totalamount | decimal(10,2) | YES | | NULL | |  
| orderdate | date | YES | | NULL | |  
| orderstatus | varchar(50) | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

## 6. CLINIC

```
create table clinic(clinicid int primary key, clinicemail varchar(100), clinicname  
varchar(100), city varchar(50), street varchar(100), phonenumbr varchar(15));
```

```
mysql> create table clinic(clinicid int primary key, clinicemail varchar(100), clinicname  
varchar(100), city varchar(50), street varchar(100), phonenumbr varchar(15));  
Query OK, 0 rows affected (0.10 sec)

mysql> desc clinic;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| clinicid | int | NO | PRI | NULL |       |
| clinicemail | varchar(100) | YES |       | NULL |       |
| clinicname | varchar(100) | YES |       | NULL |       |
| city | varchar(50) | YES |       | NULL |       |
| street | varchar(100) | YES |       | NULL |       |
| phonenumbr | varchar(15) | YES |       | NULL |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

## 7. APPOINTMENT

```
create table appointment(appointmentid int primary key, date date, time time, status  
varchar(50));
```

```
mysql> create table appointment(appointmentid int primary key, date date,  
time time, status varchar(50));  
Query OK, 0 rows affected (0.09 sec)

mysql> desc appointment;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| appointmentid | int | NO | PRI | NULL |       |
| date | date | YES |       | NULL |       |
| time | time | YES |       | NULL |       |
| status | varchar(50) | YES |       | NULL |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## 8. PAYMENT

```
create table payment(paymentid int primary key, amount decimal(10,2), paymentdate date, transactionid varchar(100), orderid int, paymentmethod varchar(50), foreign key(orderid) references order_table(orderid));
```

```
mysql> create table payment(paymentid int primary key, amount decimal(10,2),  
    , paymentdate date, transactionid varchar(100), orderid int, paymentmethod  
    varchar(50), foreign key(orderid) references order_table(orderid));  
Query OK, 0 rows affected (0.12 sec)  
  
mysql> desc payment;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| paymentid | int | NO | PRI | NULL |  
| amount | decimal(10,2) | YES | | NULL |  
| paymentdate | date | YES | | NULL |  
| transactionid | varchar(100) | YES | | NULL |  
| orderid | int | YES | MUL | NULL |  
| paymentmethod | varchar(50) | YES | | NULL |  
+-----+-----+-----+-----+-----+  
6 rows in set (0.00 sec)
```

## 9. MEDICAL\_RECORD

```
create table medical_record(recordid int primary key, diagnosis varchar(255), visitdate date, treatment text);
```

```
mysql> create table medical_record(recordid int primary key, diagnosis varchar(255),  
    visitdate date, treatment text);  
Query OK, 0 rows affected (0.05 sec)  
  
mysql> desc medical_record;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| recordid | int | NO | PRI | NULL |  
| diagnosis | varchar(255) | YES | | NULL |  
| visitdate | date | YES | | NULL |  
| treatment | text | YES | | NULL |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

## **10. TEST\_REPORT**

```
create table test_report(recordid int, testtype varchar(100), testresult varchar(255),
testdate date, primary key(recordid, testtype), foreign key(recordid) references
medical_record(recordid);
```

```
mysql> create table test_report(recordid int, testtype varchar(100), testresult varchar
(255), testdate date, primary key(recordid, testtype), foreign key(recordid) references
medical_record(recordid));
Query OK, 0 rows affected (0.10 sec)

mysql> desc test_report;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| recordid | int | NO | PRI | NULL |       |
| testtype | varchar(100) | NO | PRI | NULL |       |
| testresult | varchar(255) | YES |     | NULL |       |
| testdate | date | YES |     | NULL |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## **11. PRESCRIPTION**

```
create table prescription(prescriptionid int primary key, dosage varchar(100), issuedate
date);
```

```
mysql> create table prescription(prescriptionid int primary key, dosage varchar(100),
issuedate date);
Query OK, 0 rows affected (0.10 sec)

mysql> desc prescription;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| prescriptionid | int | NO | PRI | NULL |       |
| dosage | varchar(100) | YES |     | NULL |       |
| issuedate | date | YES |     | NULL |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

## **12. PRESCRIPTION\_MEDICINE**

```
create table prescription_medicine(prescriptionid int, medicineid int, quantity int, primary
key(prescriptionid, medicineid), foreign key(prescriptionid) references
prescription(prescriptionid), foreign key(medicineid) references medicine(medicineid));
```

```

mysql> create table prescription_medicine(prescriptionid int, medicineid int, quantity int,
primary key(prescriptionid, medicineid), foreign key(prescriptionid) references prescription
(prescriptionid), foreign key(medicineid) references medicine(medicineid));
Query OK, 0 rows affected (0.08 sec)

mysql> desc prescription_medicine;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| prescriptionid | int | NO | PRI | NULL | 
| medicineid | int | NO | PRI | NULL | 
| quantity | int | YES | | NULL | 
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

### **13. PATIENT\_MEDICAL\_HISTORY**

```

create table patient_medical_history(patientid int, medical_condition varchar(100),
diagnoseddate date, currentstatus varchar(100), primary key(patientid,
medical_condition, diagnoseddate), foreign key(patientid) references patient(patientid));

```

```

mysql> create table patient_medical_history(patientid int, medical_condition varchar(100),
diagnoseddate date, currentstatus varchar(100), primary key(patientid, medical_condition,
diagnoseddate), foreign key(patientid) references patient(patientid));
Query OK, 0 rows affected (0.13 sec)

mysql> desc patient_medical_history;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| patientid | int | NO | PRI | NULL | 
| medical_condition | varchar(100) | NO | PRI | NULL | 
| diagnoseddate | date | NO | PRI | NULL | 
| currentstatus | varchar(100) | YES | | NULL | 
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

### **14. DOCTOR\_SCHEDULE**

```

create table doctor_schedule(doctorid int, endtime time, starttime time, availdate date,
slotduration int, primary key(doctorid, availdate), foreign key(doctorid) references
doctor(doctorid));

```

```

mysql> create table doctor_schedule(doctorid int, endtime time, starttime time, availdate
date, slotduration int, primary key(doctorid, availdate), foreign key(doctorid) references
doctor(doctorid));
Query OK, 0 rows affected (0.09 sec)

mysql> desc doctor_schedule;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| doctorid | int | NO | PRI | NULL | 
| endtime | time | YES | | NULL | 
| starttime | time | YES | | NULL | 
| availdate | date | NO | PRI | NULL | 
| slotduration | int | YES | | NULL | 
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

All the tables created:

```
mysql> show tables;
+-----+
| Tables_in_skinfluence |
+-----+
| appointment
| clinic
| doctor
| doctor_schedule
| medical_record
| medicine
| order_table
| patient
| patient_address
| patient_medical_history
| payment
| prescription
| prescription_medicine
| test_report
+-----+
14 rows in set (0.09 sec)
```

## CREATING TABLES FOR RELATIONSHIP SETS:

### 1. Patient\_books\_Appointment

```
CREATE TABLE Patient_books_Appointment (
    PatientID INT,
    AppointmentID INT,
    Date DATE,
    Time TIME,
    Status VARCHAR(50),
    PRIMARY KEY (PatientID, AppointmentID)
);
```

```
mysql> CREATE TABLE Patient_books_Appointment (
    ->     PatientID INT,
    ->     AppointmentID INT,
    ->     Date DATE,
    ->     Time TIME,
    ->     Status VARCHAR(50),
    ->     PRIMARY KEY (PatientID, AppointmentID)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc Patient_books_Appointment;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| PatientID | int    | NO   | PRI  | NULL    |        |
| AppointmentID | int    | NO   | PRI  | NULL    |        |
| Date | date   | YES  |      | NULL    |        |
| Time | time   | YES  |      | NULL    |        |
| Status | varchar(50) | YES  |      | NULL    |        |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

## **2. Doctor\_assigned\_Appointment**

```
CREATE TABLE Doctor_assigned_Appointment (
    DoctorID INT,
    AppointmentID INT,
    Date DATE,
    Time TIME,
    Status VARCHAR(50),
    PRIMARY KEY (DoctorID, AppointmentID)
);
```

```
mysql> CREATE TABLE Doctor_assigned_Appointment (
    ->     DoctorID INT,
    ->     AppointmentID INT,
    ->     Date DATE,
    ->     Time TIME,
    ->     Status VARCHAR(50),
    ->     PRIMARY KEY (DoctorID, AppointmentID)
    -> );
Query OK, 0 rows affected (0.14 sec)

mysql> desc Doctor_assigned_Appointment;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| DoctorID | int | NO | PRI | NULL | 
| AppointmentID | int | NO | PRI | NULL | 
| Date | date | YES | YES | NULL | 
| Time | time | YES | YES | NULL | 
| Status | varchar(50) | YES | YES | NULL | 
+-----+-----+-----+-----+-----+
5 rows in set (0.04 sec)
```

## **3. Patient\_Order (*from Patient\_places\_Order*)**

```
CREATE TABLE Patient_Order (
    PatientID INT,
    OrderID INT,
    PRIMARY KEY (PatientID, OrderID)
);
```

```
mysql> CREATE TABLE Patient_Order (
    ->     PatientID INT,
    ->     OrderID INT,
    ->     PRIMARY KEY (PatientID, OrderID)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> desc Patient_Order;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| PatientID | int | NO | PRI | NULL | 
| OrderID | int | NO | PRI | NULL | 
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

#### 4. Patient\_Medicine (*from Patient\_takes\_Medicine*)

```
CREATE TABLE Patient_Medicine (
    PatientID INT,
    MedicineID INT,
    PRIMARY KEY (PatientID, MedicineID)
);
```

```
mysql> CREATE TABLE Patient_Medicine (
    ->     PatientID INT,
    ->     MedicineID INT,
    ->     PRIMARY KEY (PatientID, MedicineID)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc Patient_Medicine;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| PatientID | int | NO | PRI | NULL | 
| MedicineID | int | NO | PRI | NULL | 
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

#### 5. Order\_Payment (*from Order\_have\_Payment*)

```
CREATE TABLE Order_Payment (
    OrderID INT,
    PaymentID INT,
    PRIMARY KEY (OrderID, PaymentID)
);
```

```
mysql> CREATE TABLE Order_Payment (
    ->     OrderID INT,
    ->     PaymentID INT,
    ->     PRIMARY KEY (OrderID, PaymentID)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc Order_Payment;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| OrderID | int | NO | PRI | NULL | 
| PaymentID | int | NO | PRI | NULL | 
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## **6. Doctor\_Clinic (*from Doctor\_affiliated\_with\_Clinic*)**

```
CREATE TABLE Doctor_Clinic (
    DoctorID INT,
    ClinicID INT,
    PRIMARY KEY (DoctorID, ClinicID)
);
```

```
mysql> CREATE TABLE Doctor_Clinic (
    ->     DoctorID INT,
    ->     ClinicID INT,
    ->     PRIMARY KEY (DoctorID, ClinicID)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc Doctor_Clinic;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| DoctorID | int | NO | PRI | NULL |
| ClinicID | int | NO | PRI | NULL |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## **7. Appointment\_generates\_Medical\_Record**

```
CREATE TABLE Appointment_generates_Medical_Record (
    AppointmentID INT PRIMARY KEY,
    Diagnosis VARCHAR(255),
    VisitDate DATE,
    Treatment VARCHAR(255),
    RecordID INT UNIQUE
);
```

```
mysql> CREATE TABLE Appointment_generates_Medical_Record (
    ->     AppointmentID INT PRIMARY KEY,
    ->     Diagnosis VARCHAR(255),
    ->     VisitDate DATE,
    ->     Treatment VARCHAR(255),
    ->     RecordID INT UNIQUE
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> desc Appointment_generates_Medical_Record;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| AppointmentID | int | NO | PRI | NULL |
| Diagnosis | varchar(255) | YES | NULL |
| VisitDate | date | YES | NULL |
| Treatment | varchar(255) | YES | NULL |
| RecordID | int | YES | UNI | NULL |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

## **8. Medical\_Record\_has\_Test\_Report**

```
CREATE TABLE Medical_Record_has_Test_Report (
    RecordID INT,
    TestType VARCHAR(100),
    TestResult VARCHAR(255),
    TestDate DATE,
    PRIMARY KEY (RecordID, TestType)
);
```

```
mysql> CREATE TABLE Medical_Record_has_Test_Report (
    ->     RecordID INT,
    ->     TestType VARCHAR(100),
    ->     TestResult VARCHAR(255),
    ->     TestDate DATE,
    ->     PRIMARY KEY (RecordID, TestType)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc Medical_Record_has_Test_Report;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| RecordID | int | NO | PRI | NULL | 
| TestType | varchar(100) | NO | PRI | NULL | 
| TestResult | varchar(255) | YES | | NULL | 
| TestDate | date | YES | | NULL | 
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## INSERTING VALUES IN EACH TABLE

### 1. DOCTOR

```
INSERT INTO doctor (doctorid, first_name, last_name, email, contactnumber, specialisation, yearofexperience, credentials) VALUES  
(1, 'Raj', 'Verma', 'raj.verma@example.com', '9123456780', 'Cardiologist', 12, 'MBBS, MD'),  
(2, 'Anjali', 'Patel', 'anjali.patel@example.com', '9876543210', 'Dermatologist', 8, 'MBBS, DDVL'),  
(3, 'Amit', 'Singh', 'amit.singh@example.com', '9012345678', 'Neurologist', 15, 'MBBS, DM'),  
(4, 'Priya', 'Sharma', 'priya.sharma@example.com', '9988776655', 'Gynecologist', 10, 'MBBS, MS'),  
(5, 'Karan', 'Joshi', 'karan.joshi@example.com', '9765432189', 'Pediatrician', 7, 'MBBS, DCH'),  
(6, 'Sneha', 'Rao', 'sneha.rao@example.com', '9345612789', 'Orthopedic', 9, 'MBBS, MS'),  
(7, 'Ravi', 'Desai', 'ravi.desai@example.com', '9123012456', 'ENT', 6, 'MBBS, DLO'),  
(8, 'Neha', 'Kapoor', 'neha.kapoor@example.com', '9234567812', 'Oncologist', 13, 'MBBS, DM'),  
(9, 'Arjun', 'Yadav', 'arjun.yadav@example.com', '9321456789', 'Psychiatrist', 11, 'MBBS, MD'),  
(10, 'Isha', 'Mehta', 'isha.mehta@example.com', '9182736450', 'General Physician', 5, 'MBBS'),  
(11, 'Sahil', 'Chopra', 'sahil.chopra@example.com', '9012876453', 'Nephrologist', 10, 'MBBS, DM'),  
(12, 'Divya', 'Reddy', 'divya.reddy@example.com', '9102938475', 'Urologist', 6, 'MBBS, MCh'),  
(13, 'Manoj', 'Gupta', 'manoj.gupta@example.com', '9923458712', 'Gastroenterologist', 9, 'MBBS, DM'),  
(14, 'Ritika', 'Bajaj', 'ritika.bajaj@example.com', '9812345671', 'Hematologist', 7, 'MBBS, MD'),  
(15, 'Nikhil', 'Nair', 'nikhil.nair@example.com', '9987612345', 'Rheumatologist', 4, 'MBBS, MD');
```

mysql> select * from doctor;								
doctorid	first_name	last_name	email	contactnumber	specialisation	yearofexperience	credentials	
1	Raj	Verma	raj.verma@example.com	9123456780	Cardiologist	12	MBBS, MD	
2	Anjali	Patel	anjali.patel@example.com	9876543210	Dermatologist	8	MBBS, DDVL	
3	Amit	Singh	amit.singh@example.com	9012345678	Neurologist	15	MBBS, DM	
4	Priya	Sharma	priya.sharma@example.com	9988776655	Gynecologist	10	MBBS, MS	
5	Karan	Joshi	karan.joshi@example.com	9765432189	Pediatrician	7	MBBS, DCH	
6	Sneha	Rao	sneha.rao@example.com	9345612789	Orthopedic	9	MBBS, MS	
7	Ravi	Desai	ravi.desai@example.com	9123012456	ENT	6	MBBS, DLO	
8	Neha	Kapoor	neha.kapoor@example.com	9234567812	Oncologist	13	MBBS, DM	
9	Arjun	Yadav	arjun.yadav@example.com	9321456789	Psychiatrist	11	MBBS, MD	
10	Isha	Mehta	isha.mehta@example.com	9182736450	General Physician	5	MBBS	
11	Sahil	Chopra	sahil.chopra@example.com	9012876453	Nephrologist	10	MBBS, DM	
12	Divya	Reddy	divya.reddy@example.com	9102938475	Urologist	6	MBBS, MCh	
13	Manoj	Gupta	manoj.gupta@example.com	9923458712	Gastroenterologist	9	MBBS, DM	
14	Ritika	Bajaj	ritika.bajaj@example.com	9812345671	Hematologist	7	MBBS, MD	
15	Nikhil	Nair	nikhil.nair@example.com	9987612345	Rheumatologist	4	MBBS, MD	

15 rows in set (0.01 sec)

## 2. PATIENT

```
INSERT INTO patient (patientid, first_name, last_name, age, dob, contactnumber, email, gender) VALUES  
(1, 'Rohan', 'Malik', 25, '1999-04-15', '9123456780', 'rohan.malik@example.com', 'Male'),  
(2, 'Simran', 'Kaur', 30, '1994-02-10', '9988776655', 'simran.kaur@example.com',  
'Female'),  
(3, 'Aarav', 'Sharma', 22, '2002-08-20', '9876543210', 'aarav.sharma@example.com',  
'Male'),  
(4, 'Meera', 'Iyer', 28, '1996-01-12', '9765432109', 'meera.iyer@example.com', 'Female'),  
(5, 'Kabir', 'Nair', 33, '1991-05-30', '9345612789', 'kabir.nair@example.com', 'Male'),  
(6, 'Ananya', 'Pillai', 27, '1997-09-09', '9234567891', 'ananya.pillai@example.com',  
'Female'),  
(7, 'Rahul', 'Verma', 29, '1995-12-01', '9356781234', 'rahul.verma@example.com', 'Male'),  
(8, 'Sneha', 'Desai', 24, '2000-03-18', '9456123789', 'sneha.desai@example.com',  
'Female'),  
(9, 'Yash', 'Gupta', 21, '2003-07-11', '9123012456', 'yash.gupta@example.com', 'Male'),  
(10, 'Neha', 'Kumar', 26, '1998-11-22', '9213456721', 'neha.kumar@example.com',  
'Female'),  
(11, 'Viraj', 'Reddy', 31, '1993-06-17', '9345678910', 'viraj.reddy@example.com', 'Male'),  
(12, 'Divya', 'Bansal', 23, '2001-02-25', '9812345671', 'divya.bansal@example.com',  
'Female'),  
(13, 'Kunal', 'Joshi', 34, '1990-10-10', '9023456789', 'kunal.joshi@example.com', 'Male'),  
(14, 'Ishita', 'Kapoor', 25, '1999-08-05', '9134567821', 'ishita.kapoor@example.com',  
'Female'),  
(15, 'Aditya', 'Singh', 27, '1997-01-01', '9102030405', 'aditya.singh@example.com', 'Male');
```

patientid	first_name	last_name	age	dob	contactnumber	email	gender
1	Rohan	Malik	25	1999-04-15	9123456780	rohan.malik@example.com	Male
2	Simran	Kaur	30	1994-02-10	9988776655	simran.kaur@example.com	Female
3	Aarav	Sharma	22	2002-08-20	9876543210	aarav.sharma@example.com	Male
4	Meera	Iyer	28	1996-01-12	9765432109	meera.iyer@example.com	Female
5	Kabir	Nair	33	1991-05-30	9345612789	kabir.nair@example.com	Male
6	Ananya	Pillai	27	1997-09-09	9234567891	ananya.pillai@example.com	Female
7	Rahul	Verma	29	1995-12-01	9356781234	rahul.verma@example.com	Male
8	Sneha	Desai	24	2000-03-18	9456123789	sneha.desai@example.com	Female
9	Yash	Gupta	21	2003-07-11	9123012456	yash.gupta@example.com	Male
10	Neha	Kumar	26	1998-11-22	9213456721	neha.kumar@example.com	Female
11	Viraj	Reddy	31	1993-06-17	9345678910	viraj.reddy@example.com	Male
12	Divya	Bansal	23	2001-02-25	9812345671	divya.bansal@example.com	Female
13	Kunal	Joshi	34	1990-10-10	9023456789	kunal.joshi@example.com	Male
14	Ishita	Kapoor	25	1999-08-05	9134567821	ishita.kapoor@example.com	Female
15	Aditya	Singh	27	1997-01-01	9102030405	aditya.singh@example.com	Male

### **3. PATIENT\_ADDRESS**

```
INSERT INTO patient_address (patientid, street, city) VALUES  
(1, '12 Rose Villa', 'Mumbai'),  
(2, '88 Green Street', 'Delhi'),  
(3, '57 Palm Drive', 'Bangalore'),  
(4, '3 Lotus Lane', 'Chennai'),  
(5, '104 Banyan Road', 'Hyderabad'),  
(6, '42 Sunrise Apartments', 'Pune'),  
(7, '76 West End', 'Ahmedabad'),  
(8, '29 Seaside Colony', 'Kochi'),  
(9, '91 Golden Park', 'Indore'),  
(10, '17 Maple Court', 'Chandigarh'),  
(11, '61 Royal Plaza', 'Jaipur'),  
(12, '37 Hilltop Residency', 'Bhopal'),  
(13, '50 Pearl Residency', 'Nagpur'),  
(14, '22 Metro Heights', 'Lucknow'),  
(15, '33 Green Fields', 'Noida');
```

```
mysql> select * from patient_address;  
+-----+-----+-----+  
| patientid | street | city |  
+-----+-----+-----+  
| 1 | 12 Rose Villa | Mumbai |  
| 2 | 88 Green Street | Delhi |  
| 3 | 57 Palm Drive | Bangalore |  
| 4 | 3 Lotus Lane | Chennai |  
| 5 | 104 Banyan Road | Hyderabad |  
| 6 | 42 Sunrise Apartments | Pune |  
| 7 | 76 West End | Ahmedabad |  
| 8 | 29 Seaside Colony | Kochi |  
| 9 | 91 Golden Park | Indore |  
| 10 | 17 Maple Court | Chandigarh |  
| 11 | 61 Royal Plaza | Jaipur |  
| 12 | 37 Hilltop Residency | Bhopal |  
| 13 | 50 Pearl Residency | Nagpur |  
| 14 | 22 Metro Heights | Lucknow |  
| 15 | 33 Green Fields | Noida |  
+-----+-----+-----+  
15 rows in set (0.00 sec)
```

## 4. MEDICINE

```
INSERT INTO medicine (medicineid, medicinename, description, price, stock) VALUES
(1, 'Paracetamol', 'Pain reliever and fever reducer', 20.00, 150),
(2, 'Amoxicillin', 'Antibiotic used for bacterial infections', 75.00, 100),
(3, 'Cetirizine', 'Used for allergy relief', 15.00, 200),
(4, 'Metformin', 'Used to treat type 2 diabetes', 60.00, 120),
(5, 'Aspirin', 'Reduces pain, fever, and inflammation', 25.00, 180),
(6, 'Atorvastatin', 'Used to lower cholesterol', 80.00, 90),
(7, 'Omeprazole', 'Treats acid reflux and ulcers', 45.00, 110),
(8, 'Azithromycin', 'Antibiotic for various infections', 100.00, 70),
(9, 'Loratadine', 'Allergy medication', 22.00, 140),
(10, 'Ibuprofen', 'Pain reliever and anti-inflammatory', 30.00, 130),
(11, 'Dolo 650', 'Used to reduce fever and relieve pain', 18.00, 160),
(12, 'Ciprofloxacin', 'Antibiotic for serious bacterial infections', 95.00, 85),
(13, 'Pantoprazole', 'Reduces stomach acid', 40.00, 125),
(14, 'Salbutamol', 'Bronchodilator for asthma', 50.00, 100),
(15, 'Zincovit', 'Nutritional supplement with vitamins and minerals', 35.00, 190);
```

medicineid	medicinename	description	price	stock
1	Paracetamol	Pain reliever and fever reducer	20.00	150
2	Amoxicillin	Antibiotic used for bacterial infections	75.00	100
3	Cetirizine	Used for allergy relief	15.00	200
4	Metformin	Used to treat type 2 diabetes	60.00	120
5	Aspirin	Reduces pain, fever, and inflammation	25.00	180
6	Atorvastatin	Used to lower cholesterol	80.00	90
7	Omeprazole	Treats acid reflux and ulcers	45.00	110
8	Azithromycin	Antibiotic for various infections	100.00	70
9	Loratadine	Allergy medication	22.00	140
10	Ibuprofen	Pain reliever and anti-inflammatory	30.00	130
11	Dolo 650	Used to reduce fever and relieve pain	18.00	160
12	Ciprofloxacin	Antibiotic for serious bacterial infections	95.00	85
13	Pantoprazole	Reduces stomach acid	40.00	125
14	Salbutamol	Bronchodilator for asthma	50.00	100
15	Zincovit	Nutritional supplement with vitamins and minerals	35.00	190

## 5. ORDER\_TABLE

```
INSERT INTO order_table (orderid, totalamount, orderdate, orderstatus) VALUES
(1, 120.00, '2025-03-01', 'Completed'),
(2, 250.00, '2025-03-02', 'Processing'),
(3, 180.00, '2025-03-03', 'Shipped'),
(4, 90.00, '2025-03-04', 'Cancelled'),
(5, 320.00, '2025-03-05', 'Completed'),
(6, 150.00, '2025-03-06', 'Processing'),
(7, 210.00, '2025-03-07', 'Completed'),
```

```
(8, 175.00, '2025-03-08', 'Shipped'),
(9, 60.00, '2025-03-09', 'Completed'),
(10, 300.00, '2025-03-10', 'Processing'),
(11, 140.00, '2025-03-11', 'Completed'),
(12, 230.00, '2025-03-12', 'Completed'),
(13, 195.00, '2025-03-13', 'Shipped'),
(14, 275.00, '2025-03-14', 'Completed'),
(15, 110.00, '2025-03-15', 'Cancelled');
```

```
mysql> select * from order_table;
+-----+-----+-----+-----+
| orderid | totalamount | orderdate | orderstatus |
+-----+-----+-----+-----+
|      1 |    120.00 | 2025-03-01 | Completed   |
|      2 |    250.00 | 2025-03-02 | Processing  |
|      3 |    180.00 | 2025-03-03 | Shipped     |
|      4 |     90.00 | 2025-03-04 | Cancelled   |
|      5 |    320.00 | 2025-03-05 | Completed   |
|      6 |    150.00 | 2025-03-06 | Processing  |
|      7 |    210.00 | 2025-03-07 | Completed   |
|      8 |    175.00 | 2025-03-08 | Shipped     |
|      9 |     60.00 | 2025-03-09 | Completed   |
|     10 |    300.00 | 2025-03-10 | Processing  |
|     11 |    140.00 | 2025-03-11 | Completed   |
|     12 |    230.00 | 2025-03-12 | Completed   |
|     13 |    195.00 | 2025-03-13 | Shipped     |
|     14 |    275.00 | 2025-03-14 | Completed   |
|     15 |    110.00 | 2025-03-15 | Cancelled   |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

## 6. CLINIC

```
INSERT INTO clinic (clinicid, clinicemail, clinicname, city, street, phonenumbers) VALUES
(1, 'info@lifecare.com', 'LifeCare Clinic', 'Mumbai', '12 Rose Street', '9123456780'),
(2, 'hello@healthplus.in', 'HealthPlus', 'Delhi', '88 Green Avenue', '9988776655'),
(3, 'contact@mediclinic.org', 'MediClinic', 'Bangalore', '57 Tech Park', '9876543210'),
(4, 'connect@healingtouch.com', 'Healing Touch', 'Chennai', '3 Lotus Lane', '9765432109'),
(5, 'support@familycare.in', 'FamilyCare', 'Hyderabad', '104 Main Road', '9345612789'),
(6, 'info@sunshinehealth.com', 'Sunshine Health', 'Pune', '42 Sunrise Blvd', '9234567891'),
(7, 'care@urbanmed.com', 'Urban Med', 'Ahmedabad', '76 Sector 17', '9356781234'),
```

```
(8, 'clinic@coastlinehealth.org', 'Coastline Health', 'Kochi', '29 Marine Drive',
'9456123789'),
(9, 'help@wellbeingclinic.in', 'WellBeing Clinic', 'Indore', '91 Liberty Square', '9123012456'),
(10, 'mail@metromedical.com', 'Metro Medicals', 'Chandigarh', '17 Maple Street',
'9213456721'),
(11, 'info@citymedcenter.in', 'City Med Center', 'Jaipur', '61 Royal Marg', '9345678910'),
(12, 'hello@healhub.com', 'HealHub', 'Bhopal', '37 Hilltop View', '9812345671'),
(13, 'care@healthzone.in', 'HealthZone', 'Nagpur', '50 Pearl Street', '9023456789'),
(14, 'contact@newlifeclinic.com', 'New Life Clinic', 'Lucknow', '22 Metro Lane',
'9134567821'),
(15, 'team@fastaidclinic.in', 'FastAid Clinic', 'Noida', '33 Green Belt', '9102030405');
```

clinicid	clinicemail	clinicname	city	street	phonenumer
1	info@lifecare.com	LifeCare Clinic	Mumbai	12 Rose Street	9123456780
2	hello@healthplus.in	HealthPlus	Delhi	88 Green Avenue	9988776655
3	contact@mediclinic.org	MediClinic	Bangalore	57 Tech Park	9876543210
4	connect@healingtouch.com	Healing Touch	Chennai	3 Lotus Lane	9765432109
5	support@familycare.in	FamilyCare	Hyderabad	104 Main Road	9345612789
6	info@sunshinehealth.com	Sunshine Health	Pune	42 Sunrise Blvd	9234567891
7	care@urbanmed.com	Urban Med	Ahmedabad	76 Sector 17	9356781234
8	clinic@coastlinehealth.org	Coastline Health	Kochi	29 Marine Drive	9456123789
9	help@wellbeingclinic.in	WellBeing Clinic	Indore	91 Liberty Square	9123012456
10	mail@metromedical.com	Metro Medicals	Chandigarh	17 Maple Street	9213456721
11	info@citymedcenter.in	City Med Center	Jaipur	61 Royal Marg	9345678910
12	hello@healhub.com	HealHub	Bhopal	37 Hilltop View	9812345671
13	care@healthzone.in	HealthZone	Nagpur	50 Pearl Street	9023456789
14	contact@newlifeclinic.com	New Life Clinic	Lucknow	22 Metro Lane	9134567821
15	team@fastaidclinic.in	FastAid Clinic	Noida	33 Green Belt	9102030405

15 rows in set (0.00 sec)

## 7. APPOINTMENT

```
INSERT INTO appointment (appointmentid, date, time, status) VALUES
(1, '2025-04-01', '10:00:00', 'Scheduled'),
(2, '2025-04-01', '11:00:00', 'Completed'),
(3, '2025-04-02', '09:30:00', 'Cancelled'),
(4, '2025-04-02', '14:00:00', 'Scheduled'),
(5, '2025-04-03', '15:30:00', 'Completed'),
(6, '2025-04-04', '16:00:00', 'Scheduled'),
(7, '2025-04-05', '10:30:00', 'Completed'),
(8, '2025-04-05', '13:00:00', 'Cancelled'),
(9, '2025-04-06', '09:00:00', 'Scheduled'),
(10, '2025-04-06', '11:45:00', 'Scheduled'),
(11, '2025-04-07', '12:30:00', 'Completed'),
(12, '2025-04-08', '15:00:00', 'Scheduled'),
(13, '2025-04-09', '10:00:00', 'Scheduled'),
(14, '2025-04-10', '14:30:00', 'Cancelled'),
(15, '2025-04-11', '13:15:00', 'Completed');
```

```

mysql> select * from appointment;
+-----+-----+-----+-----+
| appointmentid | date      | time    | status   |
+-----+-----+-----+-----+
| 1 | 2025-04-01 | 10:00:00 | Scheduled |
| 2 | 2025-04-01 | 11:00:00 | Completed |
| 3 | 2025-04-02 | 09:30:00 | Cancelled |
| 4 | 2025-04-02 | 14:00:00 | Scheduled |
| 5 | 2025-04-03 | 15:30:00 | Completed |
| 6 | 2025-04-04 | 16:00:00 | Scheduled |
| 7 | 2025-04-05 | 10:30:00 | Completed |
| 8 | 2025-04-05 | 13:00:00 | Cancelled |
| 9 | 2025-04-06 | 09:00:00 | Scheduled |
| 10 | 2025-04-06 | 11:45:00 | Scheduled |
| 11 | 2025-04-07 | 12:30:00 | Completed |
| 12 | 2025-04-08 | 15:00:00 | Scheduled |
| 13 | 2025-04-09 | 10:00:00 | Scheduled |
| 14 | 2025-04-10 | 14:30:00 | Cancelled |
| 15 | 2025-04-11 | 13:15:00 | Completed |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)

```

## 8. PAYMENT

INSERT INTO payment (paymentid, amount, paymentdate, transactionid, orderid, paymentmethod) VALUES

```

(1, 120.00, '2025-03-01', 'TXN1001', 1, 'Credit Card'),
(2, 250.00, '2025-03-02', 'TXN1002', 2, 'Debit Card'),
(3, 180.00, '2025-03-03', 'TXN1003', 3, 'UPI'),
(4, 90.00, '2025-03-04', 'TXN1004', 4, 'Cash'),
(5, 320.00, '2025-03-05', 'TXN1005', 5, 'Net Banking'),
(6, 150.00, '2025-03-06', 'TXN1006', 6, 'UPI'),
(7, 210.00, '2025-03-07', 'TXN1007', 7, 'Credit Card'),
(8, 175.00, '2025-03-08', 'TXN1008', 8, 'Debit Card'),
(9, 60.00, '2025-03-09', 'TXN1009', 9, 'Cash'),
(10, 300.00, '2025-03-10', 'TXN1010', 10, 'Credit Card'),
(11, 140.00, '2025-03-11', 'TXN1011', 11, 'Net Banking'),
(12, 230.00, '2025-03-12', 'TXN1012', 12, 'UPI'),
(13, 195.00, '2025-03-13', 'TXN1013', 13, 'Credit Card'),
(14, 275.00, '2025-03-14', 'TXN1014', 14, 'Debit Card'),
(15, 110.00, '2025-03-15', 'TXN1015', 15, 'Cash');

```

```

mysql> select * from payment;
+-----+-----+-----+-----+-----+-----+
| paymentid | amount | paymentdate | transactionid | orderid | paymentmethod |
+-----+-----+-----+-----+-----+-----+
| 1 | 120.00 | 2025-03-01 | TXN1001 | 1 | Credit Card |
| 2 | 250.00 | 2025-03-02 | TXN1002 | 2 | Debit Card |
| 3 | 180.00 | 2025-03-03 | TXN1003 | 3 | UPI |
| 4 | 90.00 | 2025-03-04 | TXN1004 | 4 | Cash |
| 5 | 320.00 | 2025-03-05 | TXN1005 | 5 | Net Banking |
| 6 | 150.00 | 2025-03-06 | TXN1006 | 6 | UPI |
| 7 | 210.00 | 2025-03-07 | TXN1007 | 7 | Credit Card |
| 8 | 175.00 | 2025-03-08 | TXN1008 | 8 | Debit Card |
| 9 | 60.00 | 2025-03-09 | TXN1009 | 9 | Cash |
| 10 | 300.00 | 2025-03-10 | TXN1010 | 10 | Credit Card |
| 11 | 140.00 | 2025-03-11 | TXN1011 | 11 | Net Banking |
| 12 | 230.00 | 2025-03-12 | TXN1012 | 12 | UPI |
| 13 | 195.00 | 2025-03-13 | TXN1013 | 13 | Credit Card |
| 14 | 275.00 | 2025-03-14 | TXN1014 | 14 | Debit Card |
| 15 | 110.00 | 2025-03-15 | TXN1015 | 15 | Cash |
+-----+-----+-----+-----+-----+

```

15 rows in set (0.00 sec)

## 9. MEDICAL\_RECORD

```

INSERT INTO medical_record (recordid, diagnosis, visitdate, treatment) VALUES
(1, 'Hypertension', '2025-03-01', 'Prescribed medication and advised lifestyle changes'),
(2, 'Diabetes Type 2', '2025-03-02', 'Started insulin therapy and recommended diet changes'),
(3, 'Asthma', '2025-03-03', 'Given inhaler and scheduled follow-up'),
(4, 'Allergic Rhinitis', '2025-03-04', 'Prescribed antihistamines and nasal spray'),
(5, 'Migraine', '2025-03-05', 'Pain management plan and trigger diary provided'),
(6, 'Arthritis', '2025-03-06', 'Started physiotherapy and NSAIDs'),
(7, 'Skin Rash', '2025-03-07', 'Topical creams prescribed'),
(8, 'Thyroid Disorder', '2025-03-08', 'Started thyroid hormone therapy'),
(9, 'Gastritis', '2025-03-09', 'Advised antacids and food habits change'),
(10, 'Depression', '2025-03-10', 'Started counseling and low-dose antidepressants'),
(11, 'Anemia', '2025-03-11', 'Iron supplements and diet chart given'),
(12, 'Back Pain', '2025-03-12', 'Muscle relaxants and rest advised'),
(13, 'Sinus Infection', '2025-03-13', 'Prescribed antibiotics'),
(14, 'Acne', '2025-03-14', 'Skin treatment routine initiated'),
(15, 'High Cholesterol', '2025-03-15', 'Statins prescribed and lifestyle modifications recommended');

```

```

mysql> select * from medical_record;
+-----+-----+-----+-----+
| recordid | diagnosis | visitdate | treatment |
+-----+-----+-----+-----+
| 1 | Hypertension | 2025-03-01 | Prescribed medication and advised lifestyle changes |
| 2 | Diabetes Type 2 | 2025-03-02 | Started insulin therapy and recommended diet changes |
| 3 | Asthma | 2025-03-03 | Given inhaler and scheduled follow-up |
| 4 | Allergic Rhinitis | 2025-03-04 | Prescribed antihistamines and nasal spray |
| 5 | Migraine | 2025-03-05 | Pain management plan and trigger diary provided |
| 6 | Arthritis | 2025-03-06 | Started physiotherapy and NSAIDs |
| 7 | Skin Rash | 2025-03-07 | Topical creams prescribed |
| 8 | Thyroid Disorder | 2025-03-08 | Started thyroid hormone therapy |
| 9 | Gastritis | 2025-03-09 | Advised antacids and food habits change |
| 10 | Depression | 2025-03-10 | Started counseling and low-dose antidepressants |
| 11 | Anemia | 2025-03-11 | Iron supplements and diet chart given |
| 12 | Back Pain | 2025-03-12 | Muscle relaxants and rest advised |
| 13 | Sinus Infection | 2025-03-13 | Prescribed antibiotics |
| 14 | Acne | 2025-03-14 | Skin treatment routine initiated |
| 15 | High Cholesterol | 2025-03-15 | Statins prescribed and lifestyle modifications recommended |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)

```

## 10. TEST\_REPORT

```

INSERT INTO test_report (recordid, testtype, testresult, testdate) VALUES
(1, 'Blood Pressure', '140/90 mmHg', '2025-03-01'),
(2, 'HbA1c', '7.2%', '2025-03-02'),
(3, 'Spirometry', 'Moderate obstruction', '2025-03-03'),
(4, 'Allergy Test', 'Positive to pollen', '2025-03-04'),
(5, 'MRI Brain', 'No abnormalities', '2025-03-05'),
(6, 'X-Ray Knee', 'Mild joint space narrowing', '2025-03-06'),
(7, 'Skin Biopsy', 'Contact dermatitis', '2025-03-07'),
(8, 'TSH', '8.5 µIU/mL', '2025-03-08'),
(9, 'Endoscopy', 'Inflammation in stomach lining', '2025-03-09'),
(10, 'Psych Eval', 'Mild depression symptoms', '2025-03-10'),
(11, 'CBC', 'Low hemoglobin - 9 g/dL', '2025-03-11'),
(12, 'Spine MRI', 'Lumbar disc bulge', '2025-03-12'),
(13, 'Sinus CT', 'Sinus fluid buildup', '2025-03-13'),
(14, 'Skin Culture', 'Propionibacterium acnes present', '2025-03-14'),
(15, 'Lipid Profile', 'LDL - 190 mg/dL', '2025-03-15');

```

```

mysql> select * from test_report;
+-----+-----+-----+-----+
| recordid | testtype | testresult | testdate |
+-----+-----+-----+-----+
| 1 | Blood Pressure | 140/90 mmHg | 2025-03-01 |
| 2 | HbA1c | 7.2% | 2025-03-02 |
| 3 | Spirometry | Moderate obstruction | 2025-03-03 |
| 4 | Allergy Test | Positive to pollen | 2025-03-04 |
| 5 | MRI Brain | No abnormalities | 2025-03-05 |
| 6 | X-Ray Knee | Mild joint space narrowing | 2025-03-06 |
| 7 | Skin Biopsy | Contact dermatitis | 2025-03-07 |
| 8 | TSH | 8.5 µIU/mL | 2025-03-08 |
| 9 | Endoscopy | Inflammation in stomach lining | 2025-03-09 |
| 10 | Psych Eval | Mild depression symptoms | 2025-03-10 |
| 11 | CBC | Low hemoglobin - 9 g/dL | 2025-03-11 |
| 12 | Spine MRI | Lumbar disc bulge | 2025-03-12 |
| 13 | Sinus CT | Sinus fluid buildup | 2025-03-13 |
| 14 | Skin Culture | Propionibacterium acnes present | 2025-03-14 |
| 15 | Lipid Profile | LDL - 190 mg/dL | 2025-03-15 |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)

```

## 11. PRESCRIPTION

```
INSERT INTO prescription (prescriptionid, dosage, issuedate) VALUES  
(1, '1 tablet twice daily after meals', '2025-03-01'),  
(2, '10 units before breakfast', '2025-03-02'),  
(3, '2 puffs every 6 hours', '2025-03-03'),  
(4, '1 spray in each nostril twice daily', '2025-03-04'),  
(5, '1 tablet at onset of migraine', '2025-03-05'),  
(6, 'Take 1 tablet in the morning', '2025-03-06'),  
(7, 'Apply cream twice daily', '2025-03-07'),  
(8, '1 tablet daily before breakfast', '2025-03-08'),  
(9, '1 antacid 30 mins before meal', '2025-03-09'),  
(10, 'Take 1 capsule daily with food', '2025-03-10'),  
(11, '1 iron tablet twice daily', '2025-03-11'),  
(12, '1 tablet as needed for pain', '2025-03-12'),  
(13, '1 capsule daily for 5 days', '2025-03-13'),  
(14, 'Apply gel at night', '2025-03-14'),  
(15, '1 tablet in the evening', '2025-03-15');
```

```
mysql> select * from prescription;  
+-----+-----+-----+  
| prescriptionid | dosage | issuedate |  
+-----+-----+-----+  
| 1 | 1 tablet twice daily after meals | 2025-03-01 |  
| 2 | 10 units before breakfast | 2025-03-02 |  
| 3 | 2 puffs every 6 hours | 2025-03-03 |  
| 4 | 1 spray in each nostril twice daily | 2025-03-04 |  
| 5 | 1 tablet at onset of migraine | 2025-03-05 |  
| 6 | Take 1 tablet in the morning | 2025-03-06 |  
| 7 | Apply cream twice daily | 2025-03-07 |  
| 8 | 1 tablet daily before breakfast | 2025-03-08 |  
| 9 | 1 antacid 30 mins before meal | 2025-03-09 |  
| 10 | Take 1 capsule daily with food | 2025-03-10 |  
| 11 | 1 iron tablet twice daily | 2025-03-11 |  
| 12 | 1 tablet as needed for pain | 2025-03-12 |  
| 13 | 1 capsule daily for 5 days | 2025-03-13 |  
| 14 | Apply gel at night | 2025-03-14 |  
| 15 | 1 tablet in the evening | 2025-03-15 |  
+-----+-----+-----+  
15 rows in set (0.00 sec)
```

## **12. PRESCRIPTION\_MEDICINE**

```
INSERT INTO prescription_medicine (prescriptionid, medicineid, quantity) VALUES  
(1, 1, 30),  
(2, 2, 15),  
(3, 3, 20),  
(4, 4, 25),  
(5, 5, 10),  
(6, 6, 30),  
(7, 7, 15),  
(8, 8, 20),  
(9, 9, 10),  
(10, 10, 30),  
(11, 11, 20),  
(12, 12, 25),  
(13, 13, 15),  
(14, 14, 10),  
(15, 15, 20);
```

```
mysql> select * from prescription_medicine;  
+-----+-----+-----+  
| prescriptionid | medicineid | quantity |  
+-----+-----+-----+  
| 1 | 1 | 30 |  
| 2 | 2 | 15 |  
| 3 | 3 | 20 |  
| 4 | 4 | 25 |  
| 5 | 5 | 10 |  
| 6 | 6 | 30 |  
| 7 | 7 | 15 |  
| 8 | 8 | 20 |  
| 9 | 9 | 10 |  
| 10 | 10 | 30 |  
| 11 | 11 | 20 |  
| 12 | 12 | 25 |  
| 13 | 13 | 15 |  
| 14 | 14 | 10 |  
| 15 | 15 | 20 |  
+-----+-----+-----+  
15 rows in set (0.00 sec)
```

### **13. PATIENT\_MEDICAL\_HISTORY**

```
INSERT INTO patient_medical_history (patientid, medical_condition, diagnoseddate, currentstatus) VALUES  
(1, 'Hypertension', '2022-05-01', 'Under control'),  
(2, 'Diabetes Type 2', '2021-03-15', 'Needs monitoring'),  
(3, 'Asthma', '2019-11-10', 'Controlled with medication'),  
(4, 'Allergic Rhinitis', '2023-02-22', 'Mild symptoms'),  
(5, 'Migraine', '2020-08-05', 'Occasional attacks'),  
(6, 'Arthritis', '2018-06-30', 'Chronic pain management ongoing'),  
(7, 'Eczema', '2023-09-18', 'Stable with treatment'),  
(8, 'Hypothyroidism', '2020-01-12', 'Stable on medication'),  
(9, 'Gastritis', '2024-04-10', 'Under dietary control'),  
(10, 'Depression', '2022-12-01', 'Improving'),  
(11, 'Iron Deficiency Anemia', '2023-07-20', 'In recovery'),  
(12, 'Lower Back Pain', '2023-01-05', 'Managed with therapy'),  
(13, 'Sinusitis', '2023-10-17', 'Occasional flare-ups'),  
(14, 'Acne Vulgaris', '2024-01-11', 'Mild and improving'),  
(15, 'Hyperlipidemia', '2022-03-27', 'On medication');
```

```
mysql> select * from patient_medical_history;  
+-----+-----+-----+-----+  
| patientid | medical_condition | diagnoseddate | currentstatus |  
+-----+-----+-----+-----+  
| 1 | Hypertension | 2022-05-01 | Under control |  
| 2 | Diabetes Type 2 | 2021-03-15 | Needs monitoring |  
| 3 | Asthma | 2019-11-10 | Controlled with medication |  
| 4 | Allergic Rhinitis | 2023-02-22 | Mild symptoms |  
| 5 | Migraine | 2020-08-05 | Occasional attacks |  
| 6 | Arthritis | 2018-06-30 | Chronic pain management ongoing |  
| 7 | Eczema | 2023-09-18 | Stable with treatment |  
| 8 | Hypothyroidism | 2020-01-12 | Stable on medication |  
| 9 | Gastritis | 2024-04-10 | Under dietary control |  
| 10 | Depression | 2022-12-01 | Improving |  
| 11 | Iron Deficiency Anemia | 2023-07-20 | In recovery |  
| 12 | Lower Back Pain | 2023-01-05 | Managed with therapy |  
| 13 | Sinusitis | 2023-10-17 | Occasional flare-ups |  
| 14 | Acne Vulgaris | 2024-01-11 | Mild and improving |  
| 15 | Hyperlipidemia | 2022-03-27 | On medication |  
+-----+-----+-----+-----+  
15 rows in set (0.00 sec)
```

## 14. DOCTOR\_SCHEDULE

```
INSERT INTO doctor_schedule (doctorid, endtime, starttime, availdate, slotduration)
VALUES
(1, '12:00:00', '09:00:00', '2025-04-11', 30),
(2, '16:00:00', '13:00:00', '2025-04-11', 20),
(3, '18:00:00', '15:00:00', '2025-04-11', 15),
(4, '13:00:00', '10:00:00', '2025-04-11', 30),
(5, '17:00:00', '14:00:00', '2025-04-11', 20),
(6, '11:00:00', '08:00:00', '2025-04-11', 10),
(7, '15:00:00', '12:00:00', '2025-04-11', 25),
(8, '14:00:00', '11:00:00', '2025-04-11', 20),
(9, '16:30:00', '13:30:00', '2025-04-11', 15),
(10, '12:30:00', '09:30:00', '2025-04-11', 30),
(11, '17:00:00', '14:00:00', '2025-04-11', 20),
(12, '11:30:00', '08:30:00', '2025-04-11', 30),
(13, '13:30:00', '10:30:00', '2025-04-11', 10),
(14, '15:30:00', '12:30:00', '2025-04-11', 15),
(15, '16:00:00', '13:00:00', '2025-04-11', 20);
```

```
mysql> select * from doctor_schedule;
+-----+-----+-----+-----+-----+
| doctorid | endtime | starttime | availdate | slotduration |
+-----+-----+-----+-----+-----+
| 1 | 12:00:00 | 09:00:00 | 2025-04-11 | 30 |
| 2 | 16:00:00 | 13:00:00 | 2025-04-11 | 20 |
| 3 | 18:00:00 | 15:00:00 | 2025-04-11 | 15 |
| 4 | 13:00:00 | 10:00:00 | 2025-04-11 | 30 |
| 5 | 17:00:00 | 14:00:00 | 2025-04-11 | 20 |
| 6 | 11:00:00 | 08:00:00 | 2025-04-11 | 10 |
| 7 | 15:00:00 | 12:00:00 | 2025-04-11 | 25 |
| 8 | 14:00:00 | 11:00:00 | 2025-04-11 | 20 |
| 9 | 16:30:00 | 13:30:00 | 2025-04-11 | 15 |
| 10 | 12:30:00 | 09:30:00 | 2025-04-11 | 30 |
| 11 | 17:00:00 | 14:00:00 | 2025-04-11 | 20 |
| 12 | 11:30:00 | 08:30:00 | 2025-04-11 | 30 |
| 13 | 13:30:00 | 10:30:00 | 2025-04-11 | 10 |
| 14 | 15:30:00 | 12:30:00 | 2025-04-11 | 15 |
| 15 | 16:00:00 | 13:00:00 | 2025-04-11 | 20 |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

## INSERTING VALUES INTO RELATIONSHIP-SET TABLE

### 1. Patient\_books\_Appointment

```
INSERT INTO Patient_books_Appointment (PatientID, AppointmentID, Date, Time, Status)
VALUES
(101, 201, '2025-04-01', '10:00:00', 'Scheduled'),
(102, 202, '2025-04-02', '11:00:00', 'Scheduled'),
(103, 203, '2025-04-03', '12:00:00', 'Completed'),
(104, 204, '2025-04-04', '09:30:00', 'Cancelled'),
(105, 205, '2025-04-05', '14:00:00', 'Scheduled'),
(106, 206, '2025-04-06', '10:30:00', 'Scheduled'),
(107, 207, '2025-04-07', '13:00:00', 'Completed'),
(108, 208, '2025-04-08', '15:00:00', 'Scheduled'),
(109, 209, '2025-04-09', '16:00:00', 'Cancelled'),
(110, 210, '2025-04-10', '08:30:00', 'Scheduled');
```

```
mysql> select * from Patient_books_Appointment;
+-----+-----+-----+-----+-----+
| PatientID | AppointmentID | Date       | Time     | Status   |
+-----+-----+-----+-----+-----+
|      101    |      201    | 2025-04-01 | 10:00:00 | Scheduled |
|      102    |      202    | 2025-04-02 | 11:00:00 | Scheduled |
|      103    |      203    | 2025-04-03 | 12:00:00 | Completed |
|      104    |      204    | 2025-04-04 | 09:30:00 | Cancelled |
|      105    |      205    | 2025-04-05 | 14:00:00 | Scheduled |
|      106    |      206    | 2025-04-06 | 10:30:00 | Scheduled |
|      107    |      207    | 2025-04-07 | 13:00:00 | Completed |
|      108    |      208    | 2025-04-08 | 15:00:00 | Scheduled |
|      109    |      209    | 2025-04-09 | 16:00:00 | Cancelled |
|      110    |      210    | 2025-04-10 | 08:30:00 | Scheduled |
+-----+-----+-----+-----+-----+
10 rows in set (0.06 sec)
```

### 2. Doctor\_assigned\_Appointment

```
INSERT INTO Doctor_assigned_Appointment (DoctorID, AppointmentID, Date, Time,
Status) VALUES
(501, 201, '2025-04-01', '10:00:00', 'Scheduled'),
(502, 202, '2025-04-02', '11:00:00', 'Scheduled'),
(503, 203, '2025-04-03', '12:00:00', 'Completed'),
(504, 204, '2025-04-04', '09:30:00', 'Cancelled'),
(505, 205, '2025-04-05', '14:00:00', 'Scheduled'),
(506, 206, '2025-04-06', '10:30:00', 'Scheduled'),
(507, 207, '2025-04-07', '13:00:00', 'Completed'),
(508, 208, '2025-04-08', '15:00:00', 'Scheduled'),
(509, 209, '2025-04-09', '16:00:00', 'Cancelled'),
(510, 210, '2025-04-10', '08:30:00', 'Scheduled');
```

```
mysql> select * from Doctor_assigned_Appointment;
+-----+-----+-----+-----+-----+
| DoctorID | AppointmentID | Date       | Time       | Status      |
+-----+-----+-----+-----+-----+
|      501 |         201 | 2025-04-01 | 10:00:00  | Scheduled   |
|      502 |         202 | 2025-04-02 | 11:00:00  | Scheduled   |
|      503 |         203 | 2025-04-03 | 12:00:00  | Completed   |
|      504 |         204 | 2025-04-04 | 09:30:00  | Cancelled   |
|      505 |         205 | 2025-04-05 | 14:00:00  | Scheduled   |
|      506 |         206 | 2025-04-06 | 10:30:00  | Scheduled   |
|      507 |         207 | 2025-04-07 | 13:00:00  | Completed   |
|      508 |         208 | 2025-04-08 | 15:00:00  | Scheduled   |
|      509 |         209 | 2025-04-09 | 16:00:00  | Cancelled   |
|     510 |        210 | 2025-04-10 | 08:30:00  | Scheduled   |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

### **3. Patient\_Order**

```
INSERT INTO Patient_Order (PatientID, OrderID) VALUES
(101, 301),
(102, 302),
(103, 303),
(104, 304),
(105, 305),
(106, 306),
(107, 307),
(108, 308),
(109, 309),
(110, 310);
```

```
mysql> select * from Patient_Order;
+-----+-----+
| PatientID | OrderID |
+-----+-----+
|      101 |     301 |
|      102 |     302 |
|      103 |     303 |
|      104 |     304 |
|      105 |     305 |
|      106 |     306 |
|      107 |     307 |
|      108 |     308 |
|      109 |     309 |
|     110 |    310 |
+-----+-----+
10 rows in set (0.00 sec)
```

#### **4. Patient\_Medicine**

```
INSERT INTO Patient_Medicine (PatientID, MedicineID) VALUES  
(101, 901),  
(102, 902),  
(103, 903),  
(104, 904),  
(105, 905),  
(106, 906),  
(107, 907),  
(108, 908),  
(109, 909),  
(110, 910);
```

```
mysql> select * from Patient_Medicine;  
+-----+-----+  
| PatientID | MedicineID |  
+-----+-----+  
|      101  |      901  |  
|      102  |      902  |  
|      103  |      903  |  
|      104  |      904  |  
|      105  |      905  |  
|      106  |      906  |  
|      107  |      907  |  
|      108  |      908  |  
|      109  |      909  |  
|      110  |      910  |  
+-----+-----+  
10 rows in set (0.00 sec)
```

## **5. Order\_Payment**

```
INSERT INTO Order_Payment (OrderID, PaymentID) VALUES  
(301, 701),  
(302, 702),  
(303, 703),  
(304, 704),  
(305, 705),  
(306, 706),  
(307, 707),  
(308, 708),  
(309, 709),  
(310, 710);
```

```
mysql> select * from Order_Payment;  
+-----+-----+  
| OrderID | PaymentID |  
+-----+-----+  
|      301 |        701 |  
|      302 |        702 |  
|      303 |        703 |  
|      304 |        704 |  
|      305 |        705 |  
|      306 |        706 |  
|      307 |        707 |  
|      308 |        708 |  
|      309 |        709 |  
|     310 |        710 |  
+-----+-----+  
10 rows in set (0.00 sec)
```

## **6. Doctor\_Clinic**

```
INSERT INTO Doctor_Clinic (DoctorID, ClinicID) VALUES  
(501, 801),  
(502, 802),  
(503, 803),  
(504, 804),  
(505, 805),  
(506, 806),  
(507, 807),  
(508, 808),  
(509, 809),  
(510, 810);
```

```

mysql> select * from Doctor_Clinic;
+-----+-----+
| DoctorID | ClinicID |
+-----+-----+
|      501 |     801 |
|      502 |     802 |
|      503 |     803 |
|      504 |     804 |
|      505 |     805 |
|      506 |     806 |
|      507 |     807 |
|      508 |     808 |
|      509 |     809 |
|      510 |     810 |
+-----+-----+
10 rows in set (0.00 sec)

```

## 7. Appointment\_generates\_Medical\_Record

```

INSERT INTO Appointment_generates_Medical_Record (AppointmentID, Diagnosis,
VisitDate, Treatment, RecordID) VALUES
(201, 'Acne', '2025-04-01', 'Topical Cream', 1001),
(202, 'Eczema', '2025-04-02', 'Steroid Lotion', 1002),
(203, 'Psoriasis', '2025-04-03', 'Phototherapy', 1003),
(204, 'Allergy', '2025-04-04', 'Antihistamines', 1004),
(205, 'Fungal Infection', '2025-04-05', 'Antifungal Spray', 1005),
(206, 'Rosacea', '2025-04-06', 'Oral Antibiotics', 1006),
(207, 'Dermatitis', '2025-04-07', 'Moisturizer & Cream', 1007),
(208, 'Scabies', '2025-04-08', 'Permethrin Lotion', 1008),
(209, 'Warts', '2025-04-09', 'Cryotherapy', 1009),
(210, 'Vitiligo', '2025-04-10', 'Light Therapy', 1010);

```

```

mysql> select * from Appointment_generates_Medical_Record;
+-----+-----+-----+-----+-----+
| AppointmentID | Diagnosis | VisitDate | Treatment | RecordID |
+-----+-----+-----+-----+-----+
|      201 | Acne     | 2025-04-01 | Topical Cream | 1001 |
|      202 | Eczema   | 2025-04-02 | Steroid Lotion | 1002 |
|      203 | Psoriasis | 2025-04-03 | Phototherapy | 1003 |
|      204 | Allergy   | 2025-04-04 | Antihistamines | 1004 |
|      205 | Fungal Infection | 2025-04-05 | Antifungal Spray | 1005 |
|      206 | Rosacea   | 2025-04-06 | Oral Antibiotics | 1006 |
|      207 | Dermatitis | 2025-04-07 | Moisturizer & Cream | 1007 |
|      208 | Scabies   | 2025-04-08 | Permethrin Lotion | 1008 |
|      209 | Warts     | 2025-04-09 | Cryotherapy | 1009 |
|      210 | Vitiligo  | 2025-04-10 | Light Therapy | 1010 |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

```

## **8. Medical\_Record\_has\_Test\_Report**

```
INSERT INTO Medical_Record_has_Test_Report (RecordID, TestType, TestResult, TestDate) VALUES
(1001, 'Skin Biopsy', 'Normal', '2025-04-01'),
(1002, 'Allergy Patch', 'Positive', '2025-04-02'),
(1003, 'Blood Test', 'Elevated IgE', '2025-04-03'),
(1004, 'Culture', 'Fungal Detected', '2025-04-04'),
(1005, 'Dermatoscopy', 'Benign', '2025-04-05'),
(1006, 'Skin Scraping', 'Mites Found', '2025-04-06'),
(1007, 'Wood's Lamp', 'Positive Fluorescence', '2025-04-07'),
(1008, 'Skin Biopsy', 'Inflammation', '2025-04-08'),
(1009, 'Blood Test', 'Normal', '2025-04-09'),
(1010, 'Patch Test', 'Negative', '2025-04-10');
```

```
mysql> select * from Medical_Record_has_Test_Report;
+-----+-----+-----+-----+
| RecordID | TestType | TestResult | TestDate |
+-----+-----+-----+-----+
| 1001 | Skin Biopsy | Normal | 2025-04-01 |
| 1002 | Allergy Patch | Positive | 2025-04-02 |
| 1003 | Blood Test | Elevated IgE | 2025-04-03 |
| 1004 | Culture | Fungal Detected | 2025-04-04 |
| 1005 | Dermatoscopy | Benign | 2025-04-05 |
| 1006 | Skin Scraping | Mites Found | 2025-04-06 |
| 1007 | Wood's Lamp | Positive Fluorescence | 2025-04-07 |
| 1008 | Skin Biopsy | Inflammation | 2025-04-08 |
| 1009 | Blood Test | Normal | 2025-04-09 |
| 1010 | Patch Test | Negative | 2025-04-10 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

## SQL QUERIES

**Q1.** Retrieve all doctor names who specialize in 'Cardiologist'.

```
SELECT first_name, last_name FROM doctor WHERE specialisation = 'Cardiologist';
```

```
mysql> SELECT first_name, last_name FROM doctor WHERE specialisation = 'Cardiologist';
+-----+-----+
| first_name | last_name |
+-----+-----+
| Raj        | Verma      |
+-----+-----+
1 row in set (0.00 sec)
```

**Q2.** Find all patients who are older than 20.

```
SELECT first_name, last_name, age FROM patient WHERE age > 20;
```

```
mysql> SELECT first_name, last_name, age FROM patient WHERE age > 20;
+-----+-----+-----+
| first_name | last_name | age   |
+-----+-----+-----+
| Rohan     | Malik     | 25    |
| Simran    | Kaur      | 30    |
| Aarav     | Sharma    | 22    |
| Meera     | Iyer      | 28    |
| Kabir     | Nair      | 33    |
| Ananya    | Pillai    | 27    |
| Rahul     | Verma    | 29    |
| Sneha     | Desai    | 24    |
| Yash      | Gupta    | 21    |
| Neha      | Kumar    | 26    |
| Viraj     | Reddy    | 31    |
| Divya     | Bansal    | 23    |
| Kunal     | Joshi    | 34    |
| Ishita    | Kapoor   | 25    |
| Aditya    | Singh    | 27    |
+-----+-----+-----+
15 rows in set (0.00 sec)
```

**Q3.** List medicines that cost between 100 and 500.

```
SELECT medicinename, price FROM medicine WHERE price BETWEEN 100 AND 500;
```

```
mysql> SELECT medicinename, price FROM medicine WHERE price BETWEEN 100 AND 500;
+-----+-----+
| medicinename | price  |
+-----+-----+
| Azithromycin | 100.00 |
+-----+-----+
1 row in set (0.00 sec)
```

**Q4.** Get all appointments that are still marked as 'Scheduled'.

```
SELECT appointmentid, date, time FROM appointment WHERE status = 'Scheduled';
```

```
mysql> SELECT appointmentid, date, time FROM appointment WHERE status = 'Scheduled';
+-----+-----+-----+
| appointmentid | date      | time      |
+-----+-----+-----+
|       1 | 2025-04-01 | 10:00:00 |
|       4 | 2025-04-02 | 14:00:00 |
|       6 | 2025-04-04 | 16:00:00 |
|       9 | 2025-04-06 | 09:00:00 |
|      10 | 2025-04-06 | 11:45:00 |
|      12 | 2025-04-08 | 15:00:00 |
|      13 | 2025-04-09 | 10:00:00 |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

**Q5.** Show patients whose names start with 'A'.

```
SELECT first_name, last_name FROM patient WHERE first_name LIKE 'A%';
```

```
mysql> SELECT first_name, last_name FROM patient WHERE first_name LIKE 'A%';
+-----+-----+
| first_name | last_name |
+-----+-----+
| Aarav      | Sharma    |
| Ananya     | Pillai    |
| Aditya     | Singh     |
+-----+-----+
3 rows in set (0.01 sec)
```

**Q6.** Show all patients from either 'Mumbai', 'Pune', or 'Nagpur'.

```
SELECT p.first_name, p.last_name, a.city FROM patient p JOIN patient_address a ON p.patientid = a.patientid WHERE a.city IN ('Mumbai', 'Pune', 'Nagpur');
```

```
mysql> SELECT p.first_name, p.last_name, a.city FROM patient p JOIN patient_address a ON p.patientid = a.patientid WHERE a.city IN ('Mumbai', 'Pune', 'Nagpur');
+-----+-----+-----+
| first_name | last_name | city   |
+-----+-----+-----+
| Rohan     | Malik     | Mumbai |
| Ananya    | Pillai    | Pune   |
| Kunal     | Joshi    | Nagpur |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

**Q7.** Retrieve all female patients aged under 30.

```
SELECT first_name, last_name FROM patient WHERE gender = 'Female' AND age < 30;
```

```
mysql> SELECT first_name, last_name FROM patient WHERE gender = 'Female' AND age < 30;
+-----+-----+
| first_name | last_name |
+-----+-----+
| Meera      | Iyer       |
| Ananya     | Pillai     |
| Sneha      | Desai      |
| Neha       | Kumar      |
| Divya      | Bansal     |
| Ishita     | Kapoor     |
+-----+-----+
6 rows in set (0.00 sec)
```

**Q8.** Show all prescriptions with medicine names and quantities.

```
SELECT pm.prescriptionid, m.medicinename, pm.quantity FROM prescription_medicine pm
JOIN medicine m ON pm.medicineid = m.medicineid;
```

```
mysql> SELECT pm.prescriptionid, m.medicinename, pm.quantity FROM prescription_medicine pm JOIN
medicine m ON pm.medicineid = m.medicineid;
+-----+-----+-----+
| prescriptionid | medicinename | quantity |
+-----+-----+-----+
| 1             | Paracetamol   | 30        |
| 2             | Amoxicillin  | 15        |
| 3             | Cetirizine    | 20        |
| 4             | Metformin    | 25        |
| 5             | Aspirin      | 10        |
| 6             | Atorvastatin | 30        |
| 7             | Omeprazole   | 15        |
| 8             | Azithromycin | 20        |
| 9             | Loratadine   | 10        |
| 10            | Ibuprofen    | 30        |
| 11            | Dolo 650     | 20        |
| 12            | Ciprofloxacin| 25        |
| 13            | Pantoprazole | 15        |
| 14            | Salbutamol   | 10        |
| 15            | Zincovit     | 20        |
+-----+-----+-----+
15 rows in set (0.00 sec)
```

**Q9.** List appointments along with patient names.

```
SELECT a.appointmentid, a.date, p.first_name, p.last_name FROM appointment a JOIN
patient p ON a.appointmentid = p.patientid;
```

```
mysql> SELECT a.appointmentid, a.date, p.first_name, p.last_name FROM appointment a JOIN patient
p ON a.appointmentid = p.patientid;
+-----+-----+-----+-----+
| appointmentid | date      | first_name | last_name |
+-----+-----+-----+-----+
| 1           | 2025-04-01 | Rohan     | Malik     |
| 2           | 2025-04-01 | Simran    | Kaur      |
| 3           | 2025-04-02 | Aarav     | Sharma    |
| 4           | 2025-04-02 | Meera     | Iyer      |
| 5           | 2025-04-03 | Kabir     | Nair      |
| 6           | 2025-04-04 | Ananya   | Pillai    |
| 7           | 2025-04-05 | Rahul     | Verma    |
| 8           | 2025-04-05 | Sneha     | Desai    |
| 9           | 2025-04-06 | Yash      | Gupta    |
| 10          | 2025-04-06 | Neha      | Kumar    |
| 11          | 2025-04-07 | Viraj     | Reddy    |
| 12          | 2025-04-08 | Divya     | Bansal   |
| 13          | 2025-04-09 | Kunal     | Joshi    |
| 14          | 2025-04-10 | Ishita    | Kapoor   |
| 15          | 2025-04-11 | Aditya   | Singh    |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

**Q10.** Show all test reports along with diagnosis from medical records.

```
SELECT tr.testtype, tr.testresult, mr.diagnosis FROM test_report tr JOIN medical_record mr  
ON tr.recordid = mr.recordid;
```

```
mysql> SELECT tr.testtype, tr.testresult, mr.diagnosis FROM test_report tr JOIN medical_record  
mr ON tr.recordid = mr.recordid;  
+-----+-----+-----+  
| testtype | testresult | diagnosis |  
+-----+-----+-----+  
| Blood Pressure | 140/90 mmHg | Hypertension |  
| HbA1c | 7.2% | Diabetes Type 2 |  
| Spirometry | Moderate obstruction | Asthma |  
| Allergy Test | Positive to pollen | Allergic Rhinitis |  
| MRI Brain | No abnormalities | Migraine |  
| X-Ray Knee | Mild joint space narrowing | Arthritis |  
| Skin Biopsy | Contact dermatitis | Skin Rash |  
| TSH | 8.5 µIU/mL | Thyroid Disorder |  
| Endoscopy | Inflammation in stomach lining | Gastritis |  
| Psych Eval | Mild depression symptoms | Depression |  
| CBC | Low hemoglobin - 9 g/dL | Anemia |  
| Spine MRI | Lumbar disc bulge | Back Pain |  
| Sinus CT | Sinus fluid buildup | Sinus Infection |  
| Skin Culture | Propionibacterium acnes present | Acne |  
| Lipid Profile | LDL - 190 mg/dL | High Cholesterol |  
+-----+-----+-----+  
15 rows in set (0.00 sec)
```

**Q11.** List all medicines ordered by price descending.

```
SELECT medicinename, price FROM medicine ORDER BY price DESC;
```

```
mysql> SELECT medicinename, price FROM medicine ORDER BY price DESC;  
+-----+-----+  
| medicinename | price |  
+-----+-----+  
| Azithromycin | 100.00 |  
| Ciprofloxacin | 95.00 |  
| Atorvastatin | 80.00 |  
| Amoxicillin | 75.00 |  
| Metformin | 60.00 |  
| Salbutamol | 50.00 |  
| Omeprazole | 45.00 |  
| Pantoprazole | 40.00 |  
| Zincovit | 35.00 |  
| Ibuprofen | 30.00 |  
| Aspirin | 25.00 |  
| Loratadine | 22.00 |  
| Paracetamol | 20.00 |  
| Dolo 650 | 18.00 |  
| Cetirizine | 15.00 |  
+-----+-----+  
15 rows in set (0.01 sec)
```

**Q12.** Count number of appointments for each status.

```
SELECT status, COUNT(*) AS total FROM appointment GROUP BY status;
```

```
mysql> SELECT status, COUNT(*) AS total FROM appointment GROUP BY status;
+-----+-----+
| status | total |
+-----+-----+
| Scheduled |    7 |
| Completed |    5 |
| Cancelled |    3 |
+-----+-----+
3 rows in set (0.05 sec)
```

**Q13.** Find total amount paid grouped by payment method.

```
SELECT paymentmethod, SUM(amount) AS total_paid FROM payment GROUP BY paymentmethod;
```

```
mysql> SELECT paymentmethod, SUM(amount) AS total_paid FROM payment GROUP BY paymentmethod;
+-----+-----+
| paymentmethod | total_paid |
+-----+-----+
| Credit Card | 825.00 |
| Debit Card | 700.00 |
| UPI | 560.00 |
| Cash | 260.00 |
| Net Banking | 460.00 |
+-----+-----+
5 rows in set (0.04 sec)
```

**Q14.** Display doctor names and their scheduled availability.

```
SELECT d.first_name, d.last_name, s.availdate, s.starttime, s.endtime FROM doctor d
JOIN doctor_schedule s ON d.doctorid = s.doctorid;
```

```
mysql> SELECT d.first_name, d.last_name, s.availdate, s.starttime, s.endtime FROM doctor d JOIN
doctor_schedule s ON d.doctorid = s.doctorid;
+-----+-----+-----+-----+
| first_name | last_name | availdate | starttime | endtime |
+-----+-----+-----+-----+
| Raj | Verma | 2025-04-11 | 09:00:00 | 12:00:00 |
| Anjali | Patel | 2025-04-11 | 13:00:00 | 16:00:00 |
| Amit | Singh | 2025-04-11 | 15:00:00 | 18:00:00 |
| Priya | Sharma | 2025-04-11 | 10:00:00 | 13:00:00 |
| Karan | Joshi | 2025-04-11 | 14:00:00 | 17:00:00 |
| Sneha | Rao | 2025-04-11 | 08:00:00 | 11:00:00 |
| Ravi | Desai | 2025-04-11 | 12:00:00 | 15:00:00 |
| Neha | Kapoor | 2025-04-11 | 11:00:00 | 14:00:00 |
| Arjun | Yadav | 2025-04-11 | 13:30:00 | 16:30:00 |
| Isha | Mehta | 2025-04-11 | 09:30:00 | 12:30:00 |
| Sahil | Chopra | 2025-04-11 | 14:00:00 | 17:00:00 |
| Divya | Reddy | 2025-04-11 | 08:30:00 | 11:30:00 |
| Manoj | Gupta | 2025-04-11 | 10:30:00 | 13:30:00 |
| Ritika | Bajaj | 2025-04-11 | 12:30:00 | 15:30:00 |
| Nikhil | Nair | 2025-04-11 | 13:00:00 | 16:00:00 |
+-----+-----+-----+-----+
15 rows in set (0.04 sec)
```

**Q15.** Find all patients and the medicines they've been prescribed.

```
SELECT p.first_name, p.last_name, m.medicinename FROM patient p JOIN prescription pr ON p.patientid = pr.prescriptionid JOIN prescription_medicine pm ON pr.prescriptionid = pm.prescriptionid JOIN medicine m ON pm.medicineid = m.medicineid;
```

```
mysql> SELECT p.first_name, p.last_name, m.medicinename FROM patient p JOIN prescription pr ON p.patientid = pr.prescriptionid JOIN prescription_medicine pm ON pr.prescriptionid = pm.prescriptionid JOIN medicine m ON pm.medicineid = m.medicineid;
+-----+-----+-----+
| first_name | last_name | medicinename |
+-----+-----+-----+
| Rohan     | Malik      | Paracetamol   |
| Simran    | Kaur       | Amoxicillin  |
| Aarav     | Sharma     | Cetirizine    |
| Meera     | Iyer       | Metformin    |
| Kabir     | Nair       | Aspirin      |
| Ananya    | Pillai     | Atorvastatin |
| Rahul     | Verma      | Omeprazole   |
| Sneha     | Desai      | Azithromycin |
| Yash      | Gupta      | Loratadine   |
| Neha      | Kumar      | Ibuprofen   |
| Viraj     | Reddy      | Dolo 650     |
| Divya     | Bansal     | Ciprofloxacin|
| Kunal     | Joshi      | Pantoprazole |
| Ishita    | Kapoor     | Salbutamol  |
| Aditya    | Singh      | Zincovit    |
+-----+-----+-----+
15 rows in set (0.00 sec)
```

**Q16.** List all cities with either a clinic or a patient.

```
SELECT city FROM clinic UNION SELECT city FROM patient_address;
```

```
mysql> SELECT city FROM clinic UNION SELECT city FROM patient_address;
+-----+
| city |
+-----+
| Mumbai |
| Delhi  |
| Bangalore |
| Chennai |
| Hyderabad |
| Pune   |
| Ahmedabad |
| Kochi  |
| Indore |
| Chandigarh |
| Jaipur |
| Bhopal |
| Nagpur |
| Lucknow |
| Noida  |
+-----+
15 rows in set (0.01 sec)
```

**Q17.** Show doctors grouped by specialization and count how many in each group.

```
SELECT specialisation, COUNT(*) AS doctor_count FROM doctor GROUP BY specialisation;
```

```
mysql> SELECT specialisation, COUNT(*) AS doctor_count FROM doctor GROUP BY specialisation;
+-----+-----+
| specialisation | doctor_count |
+-----+-----+
| Cardiologist    | 1
| Dermatologist   | 1
| Neurologist     | 1
| Gynecologist    | 1
| Pediatrician    | 1
| Orthopedic      | 1
| ENT             | 1
| Oncologist      | 1
| Psychiatrist    | 1
| General Physician | 1
| Nephrologist    | 1
| Urologist       | 1
| Gastroenterologist | 1
| Hematologist    | 1
| Rheumatologist   | 1
+-----+-----+
15 rows in set (0.00 sec)
```

**Q18.** List all cities with more than or equal to 1 patient.

```
SELECT city, COUNT(*) AS patient_count FROM patient_address GROUP BY city HAVING COUNT(*) >= 1;
```

```
mysql> SELECT city, COUNT(*) AS patient_count FROM patient_address GROUP BY city HAVING COUNT(*) >= 1;
+-----+-----+
| city | patient_count |
+-----+-----+
| Mumbai | 1
| Delhi | 1
| Bangalore | 1
| Chennai | 1
| Hyderabad | 1
| Pune | 1
| Ahmedabad | 1
| Kochi | 1
| Indore | 1
| Chandigarh | 1
| Jaipur | 1
| Bhopal | 1
| Nagpur | 1
| Lucknow | 1
| Noida | 1
+-----+-----+
15 rows in set (0.00 sec)
```

**Q19.** Retrieve appointments scheduled today.

```
SELECT * FROM appointment WHERE date = CURRENT_DATE;
```

```
mysql> SELECT * FROM appointment WHERE date = CURRENT_DATE;
+-----+-----+-----+-----+
| appointmentid | date      | time     | status   |
+-----+-----+-----+-----+
|          15 | 2025-04-11 | 13:15:00 | Completed |
+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

**Q20.** Get all medicines whose stock is below 110.

```
SELECT medicinename, stock FROM medicine WHERE stock < 110;
```

```
mysql> SELECT medicinename, stock FROM medicine WHERE stock < 110;
+-----+-----+
| medicinename | stock |
+-----+-----+
| Amoxicillin  | 100  |
| Atorvastatin | 90   |
| Azithromycin  | 70   |
| Ciprofloxacin | 85   |
| Salbutamol    | 100  |
+-----+-----+
5 rows in set (0.00 sec)
```

**Q21.** Display all records from test\_report where test result contains 'positive'.

```
SELECT * FROM test_report WHERE testresult LIKE '%positive%';
```

```
mysql> SELECT * FROM test_report WHERE testresult LIKE '%positive%';
+-----+-----+-----+-----+
| recordid | testtype      | testresult           | testdate   |
+-----+-----+-----+-----+
|        4 | Allergy Test | Positive to pollen | 2025-03-04 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

**Q22.** List the top 3 most expensive medicines.

```
SELECT medicinename, price FROM medicine ORDER BY price DESC LIMIT 3;
```

```
mysql> SELECT medicinename, price FROM medicine ORDER BY price DESC LIMIT 3;
+-----+-----+
| medicinename | price |
+-----+-----+
| Azithromycin | 100.00 |
| Ciprofloxacin | 95.00 |
| Atorvastatin | 80.00 |
+-----+
3 rows in set (0.00 sec)
```

**Q23.** Count how many prescriptions were issued on each date.

```
SELECT issuedate, COUNT(*) AS total_prescriptions FROM prescription GROUP BY issuedate;
```

```
mysql> SELECT issuedate, COUNT(*) AS total_prescriptions FROM prescription GROUP BY issuedate;
+-----+-----+
| issuedate | total_prescriptions |
+-----+-----+
| 2025-03-01 | 1 |
| 2025-03-02 | 1 |
| 2025-03-03 | 1 |
| 2025-03-04 | 1 |
| 2025-03-05 | 1 |
| 2025-03-06 | 1 |
| 2025-03-07 | 1 |
| 2025-03-08 | 1 |
| 2025-03-09 | 1 |
| 2025-03-10 | 1 |
| 2025-03-11 | 1 |
| 2025-03-12 | 1 |
| 2025-03-13 | 1 |
| 2025-03-14 | 1 |
| 2025-03-15 | 1 |
+-----+
15 rows in set (0.00 sec)
```

**Q24.** Find all clinics whose name starts with 'S'.

```
SELECT * FROM clinic WHERE clinicname LIKE 'S%';
```

```
mysql> SELECT * FROM clinic WHERE clinicname LIKE 'S%';
+-----+-----+-----+-----+-----+
| clinicid | clinicemail | clinicname | city | street | phonenumbers |
+-----+-----+-----+-----+-----+
| 6 | info@sunshinehealth.com | Sunshine Health | Pune | 42 Sunrise Blvd | 9234567891 |
+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

**Q25.** Get a list of cities that have clinics.

```
SELECT DISTINCT city FROM clinic;
```

```
mysql> SELECT DISTINCT city FROM clinic;
+-----+
| city |
+-----+
| Mumbai|
| Delhi |
| Bangalore|
| Chennai|
| Hyderabad|
| Pune   |
| Ahmedabad|
| Kochi  |
| Indore |
| Chandigarh|
| Jaipur |
| Bhopal |
| Nagpur |
| Lucknow|
| Noida  |
+-----+
15 rows in set (0.00 sec)
```

**Q26.** Modify the column orderstatus to increase its size to 100 characters.

```
ALTER TABLE order_table MODIFY orderstatus VARCHAR(100);
```

```
mysql> ALTER TABLE order_table MODIFY orderstatus VARCHAR(100);
Query OK, 15 rows affected (0.22 sec)
Records: 15  Duplicates: 0  Warnings: 0

mysql> desc order_table;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| orderid | int    | NO   | PRI | NULL    |       |
| totalamount | decimal(10,2) | YES  |     | NULL    |       |
| orderdate | date   | YES  |     | NULL    |       |
| orderstatus | varchar(100) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)
```

**Q27.** Rename the column totalamount to total\_price.

```
ALTER TABLE order_table RENAME COLUMN totalamount TO total_price;
```

```
mysql> ALTER TABLE order_table RENAME COLUMN totalamount TO total_price;
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc order_table;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| orderid | int    | NO   | PRI | NULL    |       |
| total_price | decimal(10,2) | YES  |     | NULL    |       |
| orderdate | date   | YES  |     | NULL    |       |
| orderstatus | varchar(100) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

**Q28.** Update the orderstatus to 'Delivered' where orderdate is before '2025-03-14':

```
UPDATE order_table SET orderstatus = 'Delivered' WHERE orderdate < '2025-03-14';
```

```
mysql> UPDATE order_table SET orderstatus = 'Delivered' WHERE orderdate < '2025-03-14';
Query OK, 13 rows affected (0.04 sec)
Rows matched: 13  Changed: 13  Warnings: 0

mysql> select * from order_table;
+-----+-----+-----+-----+
| orderid | total_price | orderdate | orderstatus |
+-----+-----+-----+-----+
| 1 | 120.00 | 2025-03-01 | Delivered |
| 2 | 250.00 | 2025-03-02 | Delivered |
| 3 | 180.00 | 2025-03-03 | Delivered |
| 4 | 90.00 | 2025-03-04 | Delivered |
| 5 | 320.00 | 2025-03-05 | Delivered |
| 6 | 150.00 | 2025-03-06 | Delivered |
| 7 | 210.00 | 2025-03-07 | Delivered |
| 8 | 175.00 | 2025-03-08 | Delivered |
| 9 | 60.00 | 2025-03-09 | Delivered |
| 10 | 300.00 | 2025-03-10 | Delivered |
| 11 | 140.00 | 2025-03-11 | Delivered |
| 12 | 230.00 | 2025-03-12 | Delivered |
| 13 | 195.00 | 2025-03-13 | Delivered |
| 14 | 275.00 | 2025-03-14 | Completed |
| 15 | 110.00 | 2025-03-15 | Cancelled |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

**Q29.** Update currentstatus to 'Recovered' in patient\_medical\_history for all patients diagnosed before '2022-01-01'.

```
UPDATE patient_medical_history SET currentstatus = 'Recovered' WHERE diagnoseddate < '2022-01-01';
```

```
mysql> UPDATE patient_medical_history SET currentstatus = 'Recovered' WHERE diagnoseddate < '2022-01-01';
Query OK, 5 rows affected (0.01 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> select * from patient_medical_history;
+-----+-----+-----+-----+
| patientid | medical_condition | diagnoseddate | currentstatus |
+-----+-----+-----+-----+
| 1 | Hypertension | 2022-05-01 | Under control |
| 2 | Diabetes Type 2 | 2021-03-15 | Recovered |
| 3 | Asthma | 2019-11-10 | Recovered |
| 4 | Allergic Rhinitis | 2023-02-22 | Mild symptoms |
| 5 | Migraine | 2020-08-05 | Recovered |
| 6 | Arthritis | 2018-06-30 | Recovered |
| 7 | Eczema | 2023-09-18 | Stable with treatment |
| 8 | Hypothyroidism | 2020-01-12 | Recovered |
| 9 | Gastritis | 2024-04-10 | Under dietary control |
| 10 | Depression | 2022-12-01 | Improving |
| 11 | Iron Deficiency Anemia | 2023-07-20 | In recovery |
| 12 | Lower Back Pain | 2023-01-05 | Managed with therapy |
| 13 | Sinusitis | 2023-10-17 | Occasional flare-ups |
| 14 | Acne Vulgaris | 2024-01-11 | Mild and improving |
| 15 | Hyperlipidemia | 2022-03-27 | On medication |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

**Q30.** Select all clinics located in Mumbai, Pune, or Nashik:

```
SELECT * FROM clinic WHERE city IN ('Mumbai', 'Pune', 'Nashik');
```

```
mysql> SELECT * FROM clinic WHERE city IN ('Mumbai', 'Pune', 'Nashik');
+-----+-----+-----+-----+-----+
| clinicid | clinicemail | clinicname | city | street | phonenumer |
+-----+-----+-----+-----+-----+
|      1 | info@lifecare.com | LifeCare Clinic | Mumbai | 12 Rose Street | 9123456780 |
|      6 | info@sunshinehealth.com | Sunshine Health | Pune | 42 Sunrise Blvd | 9234567891 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

**Q31.** Combine all cities from both clinics and patient addresses (no duplicates):

```
(SELECT city FROM clinic) UNION (SELECT city FROM patient_address);
```

```
mysql> (SELECT city FROM clinic) UNION (SELECT city FROM patient_address);
+-----+
| city |
+-----+
| Mumbai |
| Delhi |
| Bangalore |
| Chennai |
| Hyderabad |
| Pune |
| Ahmedabad |
| Kochi |
| Indore |
| Chandigarh |
| Jaipur |
| Bhopal |
| Nagpur |
| Lucknow |
| Noida |
+-----+
15 rows in set (0.00 sec)
```

**Q32.** Add a UNIQUE constraint to contactnumber in the doctor table.

```
ALTER TABLE doctor ADD CONSTRAINT unique_contact UNIQUE(contactnumber);
```

```
mysql> ALTER TABLE doctor ADD CONSTRAINT unique_contact UNIQUE(contactnumber);
Query OK, 0 rows affected (0.18 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc doctor;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| doctorid | int | NO | PRI | NULL |
| first_name | varchar(50) | YES | NULL |
| last_name | varchar(50) | YES | NULL |
| email | varchar(100) | YES | NULL |
| contactnumber | varchar(15) | YES | UNI | NULL |
| specialisation | varchar(100) | YES | NULL |
| yearofexperience | int | YES | NULL |
| credentials | varchar(200) | YES | NULL |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

**Q33.** Get all orders with orderid in a given list(1,3,10,16).

```
SELECT * FROM order_table WHERE orderid IN (1, 3, 10, 16);
```

```
mysql> SELECT * FROM order_table WHERE orderid IN (1, 3, 10, 16);
+-----+-----+-----+-----+
| orderid | total_price | orderdate | orderstatus |
+-----+-----+-----+-----+
|      1 |     120.00 | 2025-03-01 | Delivered |
|      3 |     180.00 | 2025-03-03 | Delivered |
|     10 |     300.00 | 2025-03-10 | Delivered |
+-----+-----+-----+-----+
3 rows in set (0.04 sec)
```

**Q34.** Get patient details along with their medical conditions:

```
SELECT pt.patientid, pt.first_name, pt.last_name, pmh.medical_condition,
pmh.currentstatus FROM patient pt INNER JOIN patient_medical_history pmh ON
pt.patientid = pmh.patientid;
```

```
mysql> SELECT pt.patientid, pt.first_name, pt.last_name, pmh.medical_condition, pmh.currentstatus FROM patient pt INNER
JOIN patient_medical_history pmh ON pt.patientid = pmh.patientid;
+-----+-----+-----+-----+-----+
| patientid | first_name | last_name | medical_condition | currentstatus |
+-----+-----+-----+-----+-----+
|      1 | Rohan     | Malik     | Hypertension       | Under control |
|      2 | Simran    | Kaur      | Diabetes Type 2   | Recovered      |
|      3 | Aarav     | Sharma    | Asthma            | Recovered      |
|      4 | Meera     | Iyer      | Allergic Rhinitis | Mild symptoms  |
|      5 | Kabir     | Nair      | Migraine          | Recovered      |
|      6 | Ananya    | Pillai    | Arthritis          | Recovered      |
|      7 | Rahul     | Verma     | Eczema             | Stable with treatment |
|      8 | Sneha     | Desai     | Hypothyroidism    | Recovered      |
|      9 | Yash      | Gupta     | Gastritis          | Under dietary control |
|     10 | Neha      | Kumar     | Depression         | Improving      |
|     11 | Viraj     | Reddy     | Iron Deficiency Anemia | In recovery   |
|     12 | Divya     | Bansal    | Lower Back Pain    | Managed with therapy |
|     13 | Kunal     | Joshi     | Sinusitis          | Occasional flare-ups |
|     14 | Ishita    | Kapoor    | Acne Vulgaris     | Mild and improving |
|     15 | Aditya    | Singh     | Hyperlipidemia     | On medication  |
+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

**Q35.** Find clinics with 'Care' in their name.

```
SELECT clinicid, clinicname, city FROM clinic WHERE clinicname LIKE '%Care%';
```

```
mysql> SELECT clinicid, clinicname, city FROM clinic WHERE clinicname LIKE '%Care%';
+-----+-----+-----+
| clinicid | clinicname | city   |
+-----+-----+-----+
|      1 | LifeCare Clinic | Mumbai |
|      5 | FamilyCare      | Hyderabad |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

**Q36.** Get clinic cities and check if any patient resides in the same city:

```
SELECT c.city AS clinic_city, pa.city AS patient_city FROM patient_address pa RIGHT JOIN
clinic c ON pa.city = c.city;
```

```
mysql> SELECT c.city AS clinic_city, pa.city AS patient_city FROM patient_address pa RIGHT JOIN
      clinic c ON pa.city = c.city;
      +-----+-----+
      | clinic_city | patient_city |
      +-----+-----+
      | Mumbai      | Mumbai
      | Delhi       | Delhi
      | Bangalore   | Bangalore
      | Chennai     | Chennai
      | Hyderabad   | Hyderabad
      | Pune        | Pune
      | Ahmedabad   | Ahmedabad
      | Kochi        | Kochi
      | Indore      | Indore
      | Chandigarh  | Chandigarh
      | Jaipur      | Jaipur
      | Bhopal      | Bhopal
      | Nagpur      | Nagpur
      | Lucknow     | Lucknow
      | Noida        | Noida
      +-----+-----+
15 rows in set (0.05 sec)
```

**Q37.** Update the status of orders with total price greater than 200 to 'Shipped'.

```
UPDATE order_table SET orderstatus = 'Shipped' WHERE total_price > 200;
```

```
mysql> UPDATE order_table SET orderstatus = 'Shipped' WHERE total_price > 200;
Query OK, 6 rows affected (0.02 sec)
Rows matched: 6  Changed: 6  Warnings: 0

mysql> select * from order_table;
+-----+-----+-----+-----+
| orderid | total_price | orderdate | orderstatus |
+-----+-----+-----+-----+
| 1       | 120.00    | 2025-03-01 | Delivered
| 2       | 250.00    | 2025-03-02 | Shipped
| 3       | 180.00    | 2025-03-03 | Delivered
| 4       | 90.00     | 2025-03-04 | Delivered
| 5       | 320.00    | 2025-03-05 | Shipped
| 6       | 150.00    | 2025-03-06 | Delivered
| 7       | 210.00    | 2025-03-07 | Shipped
| 8       | 175.00    | 2025-03-08 | Delivered
| 9       | 60.00     | 2025-03-09 | Delivered
| 10      | 300.00    | 2025-03-10 | Shipped
| 11      | 140.00    | 2025-03-11 | Delivered
| 12      | 230.00    | 2025-03-12 | Shipped
| 13      | 195.00    | 2025-03-13 | Delivered
| 14      | 275.00    | 2025-03-14 | Shipped
| 15      | 110.00    | 2025-03-15 | Cancelled
+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

**Q38.** List patients whose last name ends with 'son'.

```
SELECT patientid, first_name, last_name FROM patient WHERE last_name LIKE '%son';
```

```
mysql> SELECT patientid, first_name, last_name FROM patient WHERE last_name LIKE '%son';
Empty set (0.00 sec)
```

**Q39.** Create a view of all patients with their addresses.

```
CREATE VIEW patient_with_address AS SELECT p.patientid, p.first_name, p.last_name,
a.city, a.street FROM patient p JOIN patient_address a ON p.patientid = a.patientid;
```

```
mysql> CREATE VIEW patient_with_address AS SELECT p.patientid, p.first_name, p.last_name, a.city,
a.street FROM patient p JOIN patient_address a ON p.patientid = a.patientid;
Query OK, 0 rows affected (0.07 sec)

mysql> desc patient_with_address;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| patientid | int    | NO   | PRI | NULL    |       |
| first_name | varchar(50) | YES  |     | NULL    |       |
| last_name  | varchar(50) | YES  |     | NULL    |       |
| city        | varchar(50) | YES  |     | NULL    |       |
| street      | varchar(100) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.04 sec)
```

**Q40.** Query the view to show all patients in 'Mumbai'.

```
SELECT * FROM patient_with_address WHERE city = 'Mumbai';
```

```
mysql> SELECT * FROM patient_with_address WHERE city = 'Mumbai';
+-----+-----+-----+-----+
| patientid | first_name | last_name | city      | street    |
+-----+-----+-----+-----+
|         1 | Rohan     | Malik     | Mumbai    | 12 Rose Villa |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

**Q41.** Start a transaction to update stock after a prescription is issued.

```
START TRANSACTION;
```

```
UPDATE medicine SET stock = stock - 1 WHERE medicineid = 3;
```

```
COMMIT;
```

```
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE medicine SET stock = stock - 1 WHERE medicineid = 3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> COMMIT;
Query OK, 0 rows affected (0.01 sec)

mysql> desc medicine;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| medicineid | int    | NO   | PRI | NULL    |       |
| medicinename | varchar(100) | YES  |     | NULL    |       |
| description  | text    | YES  |     | NULL    |       |
| price        | decimal(10,2) | YES  |     | NULL    |       |
| stock        | int    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

**Q42.** Find the order with the highest payment amount.

```
SELECT p.* FROM payment p WHERE p.amount = (SELECT MAX(amount) FROM payment);
```

```
mysql> SELECT p.* FROM payment p WHERE p.amount = (SELECT MAX(amount) FROM payment);
+-----+-----+-----+-----+-----+
| paymentid | amount | paymentdate | transactionid | orderid | paymentmethod |
+-----+-----+-----+-----+-----+
|      5 | 320.00 | 2025-03-05 | TXN1005     |      5 | Net Banking   |
+-----+-----+-----+-----+-----+
1 row in set (0.05 sec)
```

**Q43.** Show orders where the total price is greater than the average of all orders.

```
SELECT * FROM order_table WHERE total_price > (SELECT AVG(total_price) FROM order_table);
```

```
mysql> SELECT * FROM order_table WHERE total_price > (SELECT AVG(total_price) FROM order_table);
+-----+-----+-----+-----+
| orderid | total_price | orderdate | orderstatus |
+-----+-----+-----+-----+
|      2 |    250.00 | 2025-03-02 | Shipped   |
|      5 |    320.00 | 2025-03-05 | Shipped   |
|      7 |    210.00 | 2025-03-07 | Shipped   |
|     10 |    300.00 | 2025-03-10 | Shipped   |
|     12 |    230.00 | 2025-03-12 | Shipped   |
|     13 |    195.00 | 2025-03-13 | Delivered |
|     14 |    275.00 | 2025-03-14 | Shipped   |
+-----+-----+-----+-----+
7 rows in set (0.04 sec)
```

**Q44.** Get all patient IDs who have both placed an order and made a payment

```
SELECT orderid FROM order_table INTERSECT SELECT orderid FROM payment;
```

```
mysql> SELECT orderid FROM order_table INTERSECT SELECT orderid FROM payment;
+-----+
| orderid |
+-----+
|      1 |
|      2 |
|      3 |
|      4 |
|      5 |
|      6 |
|      7 |
|      8 |
|      9 |
|     10 |
|     11 |
|     12 |
|     13 |
|     14 |
|     15 |
+-----+
15 rows in set (0.00 sec)
```

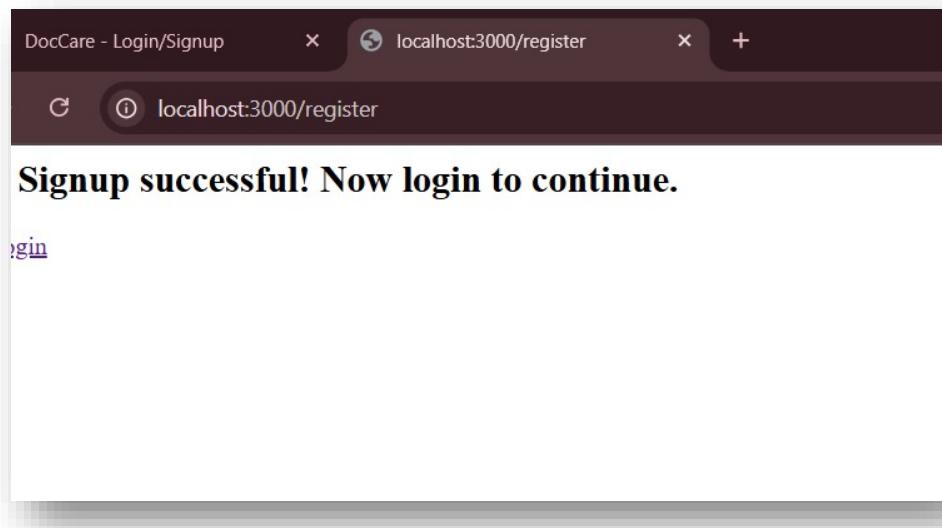
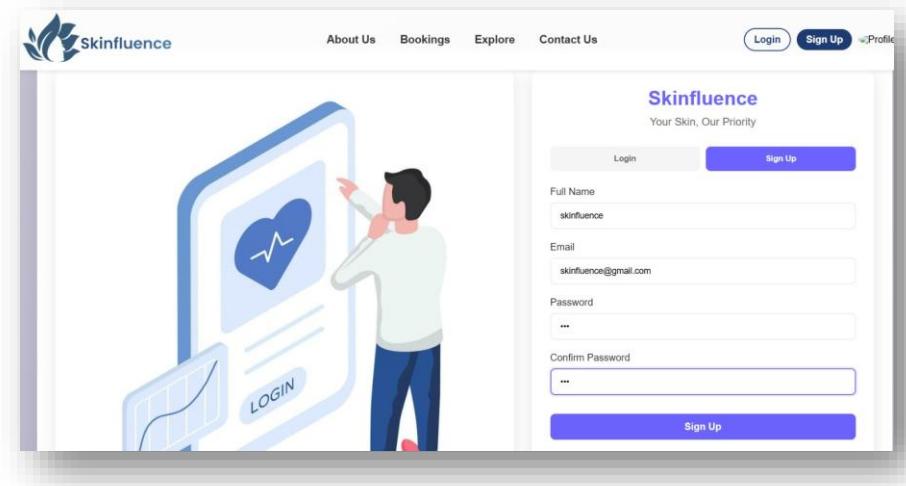
**Q45.** Get the top 10 most expensive orders sorted by total price in descending order.

```
SELECT orderid, total_price FROM order_table ORDER BY total_price DESC LIMIT 10;
```

```
mysql> SELECT orderid, total_price FROM order_table ORDER BY
total_price DESC LIMIT 10;
+-----+-----+
| orderid | total_price |
+-----+-----+
|      5  |    320.00  |
|     10  |    300.00  |
|     14  |    275.00  |
|      2  |    250.00  |
|     12  |    230.00  |
|      7  |    210.00  |
|     13  |    195.00  |
|      3  |    180.00  |
|      8  |    175.00  |
|      6  |    150.00  |
+-----+-----+
10 rows in set (0.00 sec)
```

## VI. Project demonstration

- Tools/software/ libraries used:  
VS CODE, MYSQL, NODEJS, JAVASCRIPT, NPM INSTALLED, EXPRESS JS, MYSQL2, JSON PACKAGES, HTTP SERVERS, HTML, CSS
- Screenshot and Description of the Demonstration of project (If GUI is made)





# Skinfluence

Your Skin, Our Priority

Book your appointment with top dermatologists.  
Experience personalized care for every skin type.

**About Us**

**Bookings**

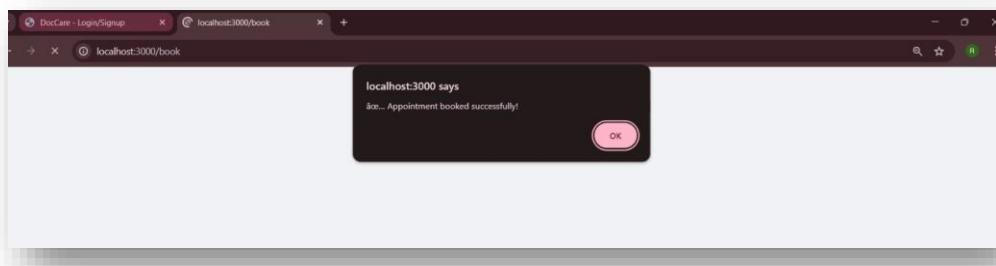
**Explore**

**Contact Us**

[Login](#) | [Sign Up](#) | [Profile icon](#)

### Appointment Registration

Full Name	Gender
Skinfluence	Male
Age	Diagnosis/Concern
28	Acne
Describe Your Condition	
Acne	
Blood Group	
A+	
Address	
Hobbies (Indicate)	
Email	Phone Number
skinfluence@gmail.com	9999999999
Appointment Date	Preferred Time
24-01-2025	19:30
<b>Book Appointment</b>	



```
mysql> select * from users;
+----+-----+-----+-----+-----+
| id | full_name | email           | password | created_at |
+----+-----+-----+-----+-----+
| 1  | rishabh   | R@GMAIL.COM     | h         | 2025-04-12 05:14:44 |
| 2  | skinfluence | skinfluence@gmail.com | abc      | 2025-04-12 14:26:50 |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## **VII. Self -Learning beyond classroom**

During this project, we explored several concepts and tools independently that were not covered in traditional lectures. This included learning how to normalize relationship sets and design weak entities in complex real-world systems like healthcare.

Additionally, we practiced generating realistic test data and using tools like DBMS diagram designers to visually validate our schema structure.

## **VIII. Learning from the Project**

- This project helped us understand how theoretical database concepts apply in real-world scenarios, especially in a domain as data-sensitive as healthcare.
- We gained hands-on experience in data modeling, normalization, and query building, which improved both our technical knowledge and problem-solving ability.
- Working through complex entity relationships strengthened our understanding of schema design and relational integrity.
- Overall, the project enhanced our confidence in SQL, prepared us for larger data-driven applications, and gave us a clear perspective on the role of databases in professional environments.

## **IX. Challenges Faced**

Throughout the project, we faced several challenges that enhanced our understanding of database systems:

- Mapping Complex Relationships: Designing accurate links between entities like patients, doctors, and appointments required careful planning.
- Normalization Issues: Ensuring all tables adhered to 3NF involved identifying and resolving transitive and partial dependencies.
- Handling Weak Entities: Implementing weak entities with composite keys added complexity to the schema and queries.
- Maintaining Data Integrity: Managing multiple foreign keys across related tables demanded attention to referential integrity.
- Optimizing Queries: Writing efficient SQL queries for multi-table operations required several iterations and performance tuning.

These challenges significantly improved our problem-solving skills and deepened our SQL knowledge.

## **X. Conclusion**

This project provided us with valuable hands-on experience in designing and implementing a comprehensive healthcare database management system using SQL. By modeling real-world entities such as patients, doctors, appointments, prescriptions, and test reports, we gained practical skills in data structuring, relationship mapping, and normalization up to 3NF.

Working with realistic healthcare scenarios—like tracking patient appointments, medical records, and prescriptions—helped us understand the importance of data consistency, integrity, and efficiency in critical systems. We applied advanced SQL techniques such as JOIN operations, subqueries, and transaction control to manage complex interactions across multiple tables.

Additionally, we explored important database administration tasks including the use of constraints, keys, and normalization practices to eliminate redundancy and ensure data integrity. We also dealt with technical challenges such as designing weak entity relationships and resolving transitive dependencies through decomposition, which sharpened our problem-solving and optimization skills.

This project has not only strengthened our understanding of relational databases and SQL but also prepared us for handling real-life database applications. Moving forward, we aim to enhance this system by integrating a user-friendly front-end interface and implementing features such as appointment notifications and prescription analytics. Overall, this project significantly boosted our confidence and proficiency in database design and management.