UNIVERSITI TEKNOLOGI MALAYSIA FACULTY OF COMPUTING UTM Johor Bahru

Project : Phase 3

<PKU PRO MAX>

SECD2523 - DATABASE SEMESTER I, SESSION 2023/2024

Lecturer: Dr. Nur Hidayah binti Zakaria

Group Name: 404 Found

Name	**Matric No.**
Khoo Teong Lee	A22EC0174
Lim Bo Yuan	A22EC0181
Lim Yu An	A22EC0183
Loh Chee Huan	A22EC0186

Section : 08

Table of Contents

1. Introduction

The Pusat Kesihatan Universiti (PKU) of UTM currently relies on outdated manual systems and processes that create major inefficiencies in operations and delivery of care. For example, physical appointment books make it difficult for students to schedule timely appointments. Students may waste time walking into PKU only to find out slots are full for that day, resulting in long wait times and appointment scheduling issues.

Besides, PKU also informs quarantined patients about COVID-19 and influenza isolation protocols through WhatsApp. However, a centralized digital system would enable more systematic and convenient communication with quarantined patients. The system could allow patients to easily check-in, track quarantine progress, and check-out digitally. Currently, quarantined patients fill out separate Google Forms with their details and need to manually confirm status via WhatsApp, which is inconvenient.

Hence, our project aims to develop a management digital system to modernize hospital management systems in PKU. The new system will automate core functions like patient registration, appointment scheduling, billing and medical records using a centralized database accessible to all providers. Doctors can easily view test results and prescription history before meeting a patient, nurses can coordinate care and education more seamlessly, and administrators can track key performance metrics through the system.

When it comes to the patient's perspective, the new system will provide conveniences like online appointment booking, reminders and self-check-in upon arrival. It will also allow patients to securely access their medical information online at any time. By replacing fragmented paper trails with digitized processes, this hospital management system can improve quality of care, reduce delays and errors, lower costs, and provide better experiences for both patients and PKU authorities. With the new system, it will bring PKU to better serve the UTM community.

2. Overview of Project

PKU Pro Max is a system performed by our team to develop a comprehensive digital database system aimed at modernizing and improving operations at the Pusat Kesihatan Universiti (PKU) health clinic which currently serves the UTM university community. This is because the current systems at PKU face issues such as medicine being outdated as lack of checking from time to time, PKU staff need to manually include the paper appointment books which are not efficient, scattered paper medical records and others. Hence, we created this PKU Pro Max system for both UTM students and PKU staff so that they can benefit from self-service options like online appointment scheduling with reminders, paperless check-in/out, and mobile access to their health records. Besides, doctors can view integrated patient data including test results, health history, and immunization status from the system instantaneously when making treatment decisions for their patients.

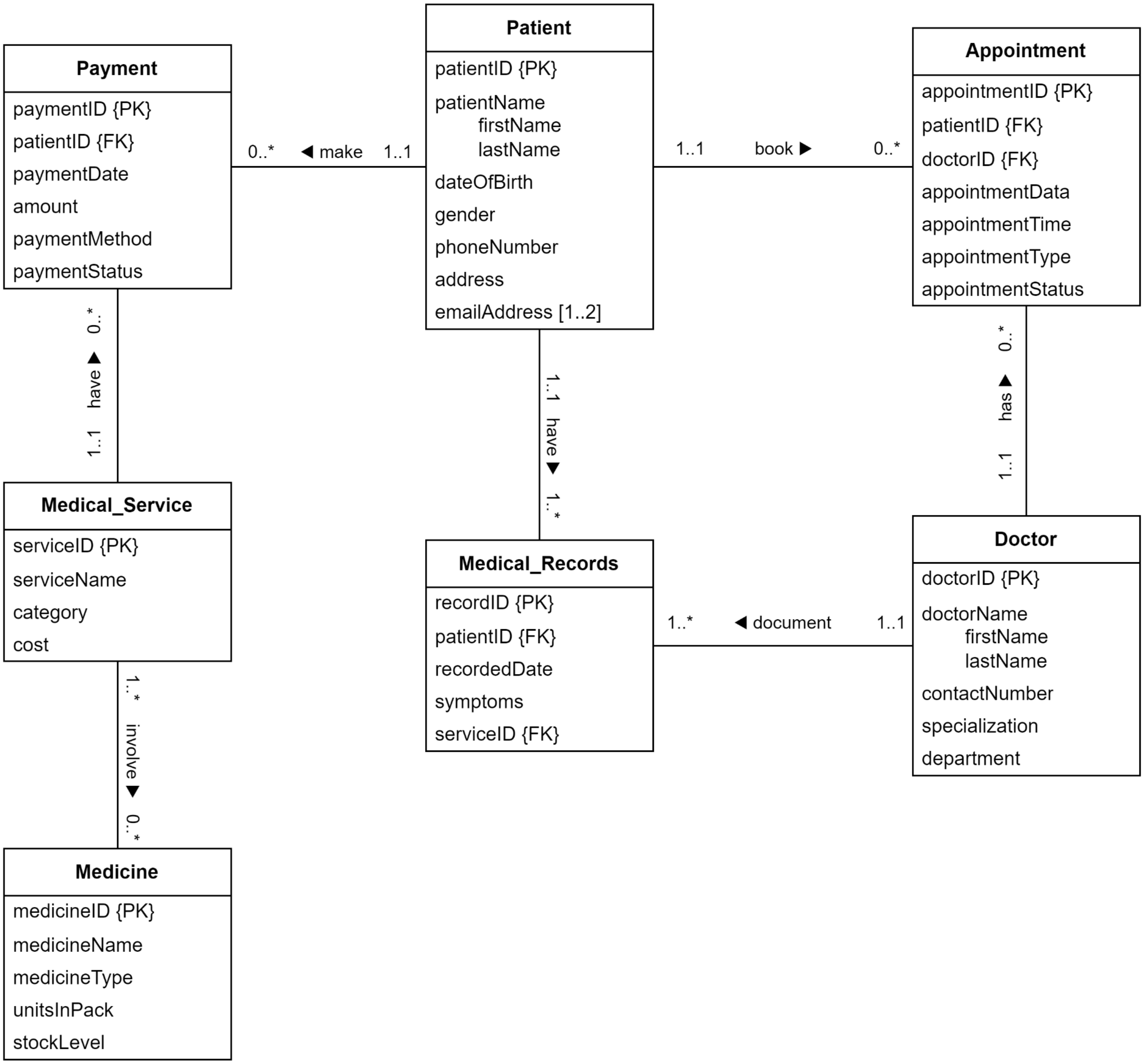
After completing phase 2 which is the database conceptual design phase, we proceeded to phase 3 to implement the database logical design and prepare the SQL statements which included both Data Definition Language (DDL) and Data Manipulation Language (DML) for our PKU Pro Max System. In this report, we will transform the conceptual entity relationship diagram (cERD) and the enhanced entity relationship diagram (EERD) in phase 2 into logical ERD. Then, we will update the data dictionary for the PKU Pro Max System by referring to the logical ERD. After that, we performed the normalization process to minimize the duplicated data so that the data are stored logically. Besides, normalization can ensure the minimal number of attributes necessary to support the data requirements of our PKU Pro Max System and minimal redundancy with each attribute represented only once with the important exception of attributes that form all or part of foreign keys, which are essential for the joining of related relations during the DML process. Next, we will get the Relational DB Schemas after the normalization process. Besides, the database for our PKU Pro Max System will be created using SQL statements which include both DDL and DML. Last but not least, we will perform some test queries that relate to the functions provided by our PKU Pro Max System such as displaying patient lists, displaying doctor lists, displaying medicine lists, checking medicine using medicine names, checking appointments for a doctor and others.

3. Database conceptual design

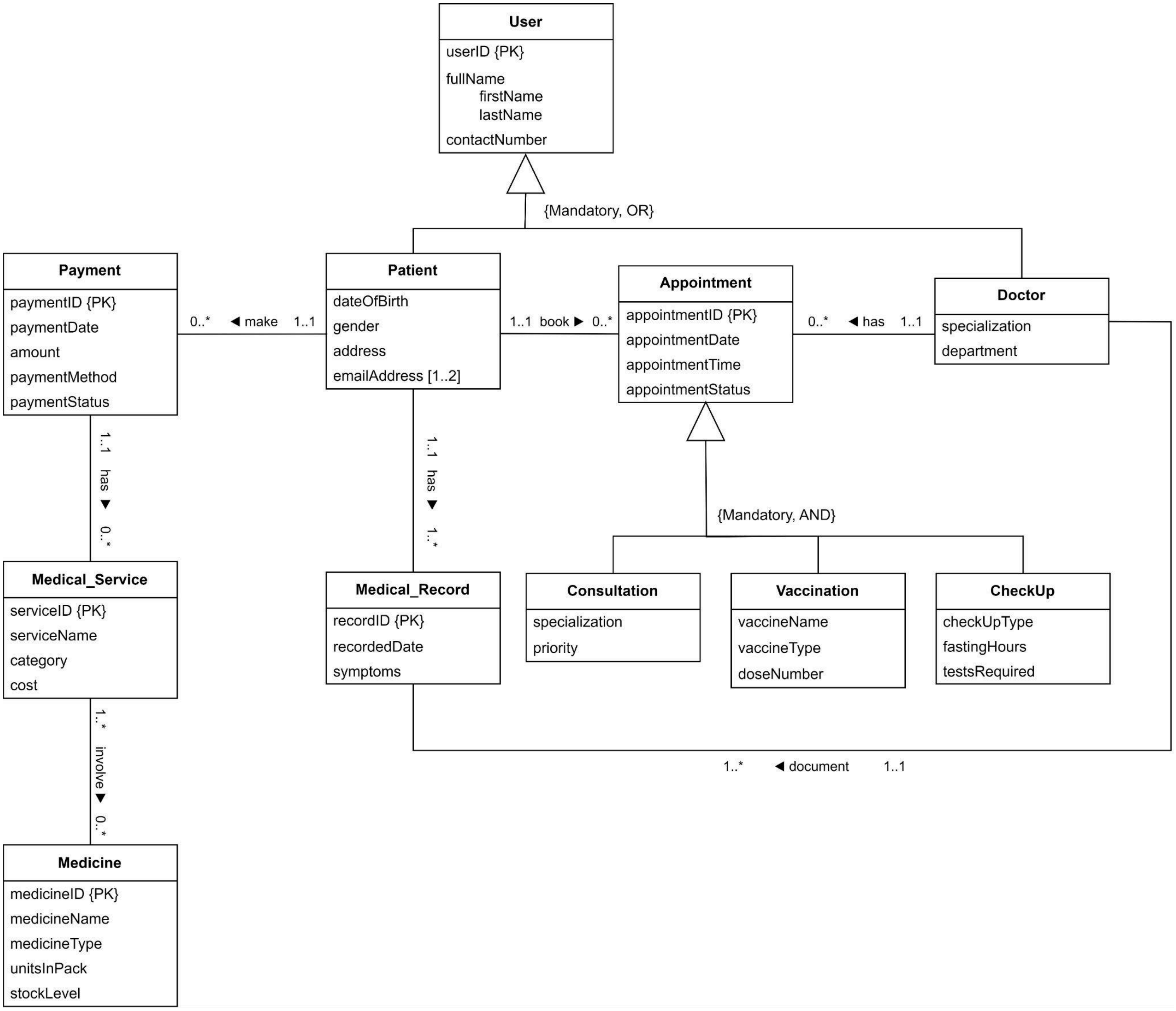
3.1. Updated business rule

- The system operates 20 hours a day, 7 days a week, with the flexibility to adjust operational hours based on demand.
- The system will be temporarily closed for service every day from 12:30 am until 1:30 am for system maintenance.
- Patients fill in their details (name, date of birth, email) before booking an appointment.
- Patients select their data, doctor and available time slot while booking appointments.
- Patients are allowed to cancel or reschedule the appointment online up to 24 hours before the scheduled time.
- Appointment reminder notifications will be sent to the patients 24 hours and 1 hour before the appointment time.
- The system will allow the selected non-medical staff to control access to relevant parts of the medical record.
- The system will set an auto save feature to periodically save the petients' medical record to prevent data loss.
- Patient details like date, symptoms and diagnosis notes are recorded in the medical report of each patient.
- The medical record can be updated once there are any changes made.
- The medical details are filled in by staff if there is any issuance or replenishment on the medicine stock.
- The system will automatically track the medicine stock level once the medicine details have been filled into the system.
- For payment, the fees calculation is automated based on the medical fees and medicine costs.
- The system will accept payment through cash, credit and debit card, Duit Now and online banking.
- The system will allow patients to access their payment receipts and medical records online through a patient portal.

3.2. Conceptual ERD

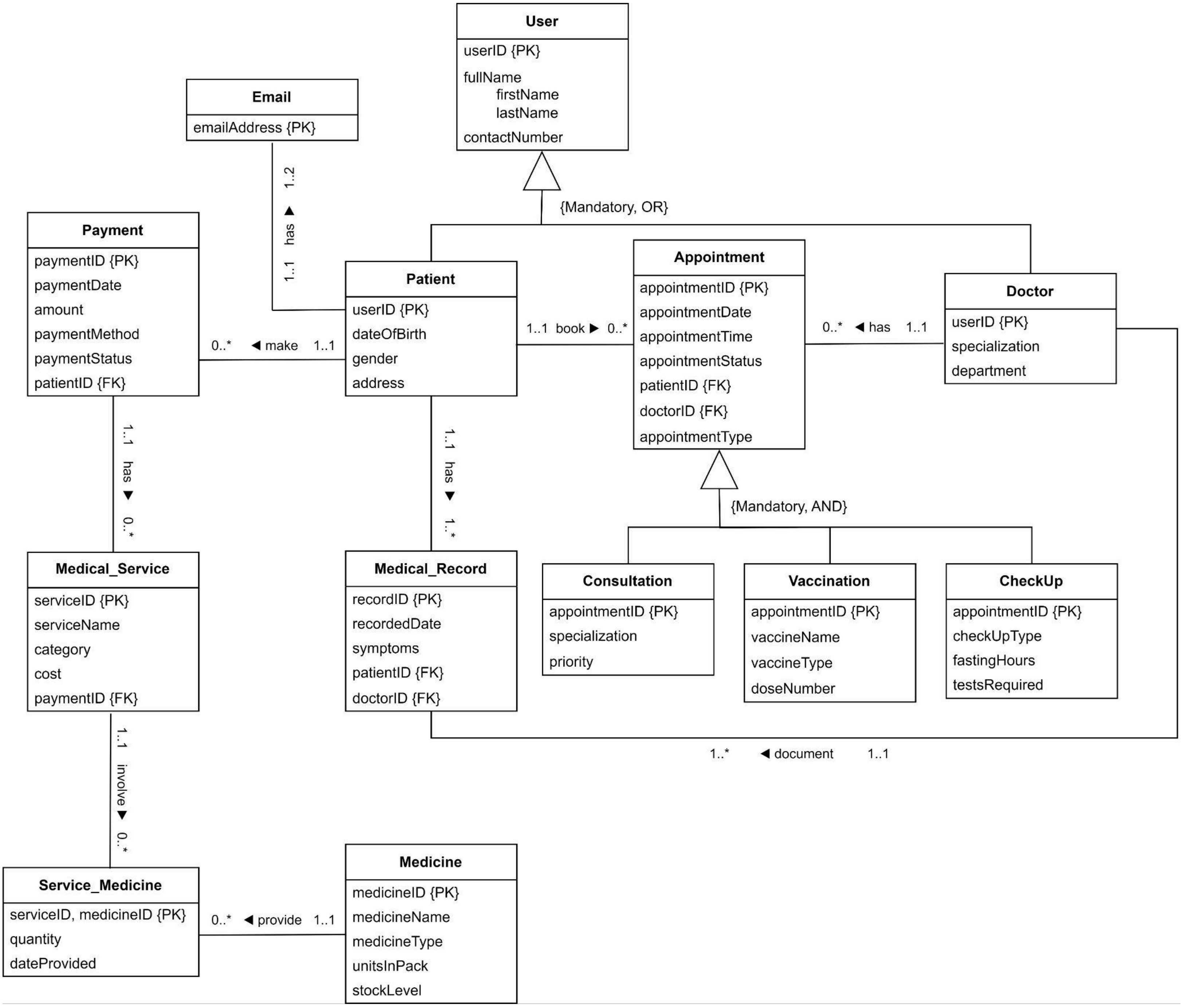


3.3 Conceptual EERD



4. DB logical design

4.1. Logical ERD



Step 1 : Strong Entity

1. User (**<u>userID</u>**, firstName, lastName, contactNumber)
2. Payment(**<u>paymentID</u>**, paymentDate, amount, paymentMethod, paymentStatus)
3. Medical_Service(**<u>serviceID</u>**, serviceName, category, cost)
4. Medicine(**<u>medicineID</u>**, medicineName, medicineType, unitsInPack, stockLevel)
5. Appointment(**<u>appointmentID</u>**, appointmentDate, appointmentTime, appointmentStatus)
6. Medical_Record(**<u>recordID</u>**, recordedDate, symptoms)

Step 2 : Weak Entity

1. Patient(dateOfBirth, gender, address, emailAddress)

   Primary Key : None (at present)

2. Doctor(specialization, department)

   Primary Key : None (at present)

3. Consultation(specialization, priority)

   Primary Key : None (at present)

4. Vaccination(vaccineName, vaccineType, doseNumber)

   Primary Key : None (at present)

5. CheckUp(checkUpType, fastingHours, testsRequired)

   Primary Key : None (at present)

Step 3 : Superclass/subclass relationship types

1. **Relationship 1 : Appointment {Mandatory, AND}**

   Appointment(**appointmentID**, appointmentDate, appointmentTime, appointmentStatus, *appointmentType*)

   Consultation(**appointmentID,** specialization, priority)

   Vaccination(**appointmentID,** vaccineName, vaccineType, doseNumber)

   CheckUp(**appointmentID,** checkUpType, fastingHours, testsRequired)

   ****Remark : Add one discriminator which is *appointmentType* to distinguish between the child entities.**

2. **Relationship 2 : User {Mandatory, OR}**

   Patient(**userID**, firstName, lastName, contactNumber, dateOfBirth, gender, address, emailAddress)

   Doctor(**userID**, firstName, lastName, contactNumber, specialization, department)

Step 4 : One-to-many (1:*) binary relationships types

1. **Relationship 1 : Patient makes payment**

   Parent → Patient

   Child → Payment

   Patient(**userID**, firstName, lastName, contactNumber, dateOfBirth, gender, address, emailAddress)

   Payment(**paymentID**, paymentDate, amount, paymentMethod, paymentStatus, *patientID*)

   FK : *patientID* references Patient(userID)

2. **Relationship 2 : Payment has medical_service**

   Parent → Patient

   Child → Medical_Service

   Payment(**paymentID**, paymentDate, amount, paymentMethod, paymentStatus, *userID*)

   Medical_Service(**serviceID**, serviceName, category, cost, *paymentID*)

3. **Relationship 3 : Patient books appointment**

   Parent → Patient

   Child → Appointment

   Patient(**userID**, firstName, lastName, contactNumber, dateOfBirth, gender, address, emailAddress)

   Appointment(**appointmentID**, appointmentDate, appointmentTime, appointmentStatus, *appointmentType, patientID*)

   FK : *patientID* references Patient(userID)

4. **Relationship 5 : Doctor has appointment**

   Parent → Doctor

   Child → Appointment

   Doctor(**userID**, firstName, lastName, contactNumber, specialization, department)

   Appointment(**appointmentID**, appointmentDate, appointmentTime, appointmentStatus, *appointmentType*, *patientID, doctorID*)

   FK : *patientID* references Patient(userID)

   doctorID references Doctor(userID)

5. **Relationship 4 : Patient has medical_record**

Parent → Patient

Child → Medical_Record

Patient(**userID**, firstName, lastName, contactNumber, dateOfBirth, gender, address, emailAddress)

Medical_Record(**recordID**, recordedDate, symptoms, *patientID*)

FK : *patientID* references Patient(userID)

6. **Relationship 6 : Doctor documents medical_record**

Parent → Doctor

Child → Medical_Record

Doctor(**userID**, firstName, lastName, contactNumber, specialization, department)

Medical_Record(**recordID**, recordedDate, symptoms, *patientID, doctorID*)

FK : *patientID* references Patient(userID)

doctorID references Doctor(userID)

Step 5 : Many-to-many (*:*) binary relationships types

1. **Relationship 1 : Medical Service involves medicine**

Medical_Service(**serviceID**, serviceName, category, cost, *userID*)

Medicine(**medicineID**, medicineName, medicineType, unitsInPack, stockLevel)

Service_Medicine(**serviceID, medicineID**)

Step 6 : Multi-valued attributes

1. **Relationship 1 : Patient has emailAddress [1..2]**

Patient(**userID**, firstName, lastName, contactNumber, dateOfBirth, gender, address)

Email(userID, **emailAddress**)

Finalization of Logical Relation Schema

1. User (**userID**, firstName, lastName, contactNumber)
2. Patient(**userID**, firstName, lastName, contactNumber, dateOfBirth, gender, address)
3. Doctor(**userID**, firstName, lastName, contactNumber, specialization, department)
4. Payment(**paymentID**, paymentDate, amount, paymentMethod, paymentStatus, *userID*)
5. Medical_Service(**serviceID**, serviceName, category, cost, *userID*)
6. Service_Medicine(**serviceID, medicineID**, quantity, dateProvided)
7. Medicine(**medicineID**, medicineName, medicineType, unitsInPack, stockLevel)
8. Appointment(**appointmentID**, appointmentDate, appointmentTime, appointmentStatus, *appointmentType, patientID, doctorID*)
9. Consultation(**appointmentID,** specialization, priority)
10. Vaccination(**appointmentID,** vaccineName, vaccineType, doseNumber)
11. CheckUp(**appointmentID,** checkUpType, fastingHours, testsRequired)
12. Medical_Record(**recordID**, recordedDate, symptoms, *patientID, doctorID*)
    * medical record got include service id mah (nope)
13. Email(userID, **emailAddress**)

Remark : **Bold underlined word is Primary Key

**Remark : *Italic* word is Foreign Key

4.2. Updated Data Dictionary

4.2.1 Description of Entity

Entity	Description	Occurrence
User	Hold user's information within the system.	User key in their name and contact number before entering the system.
Patient	Hold patient's information.	Patients fill in their details (name, date of birth, email) before booking an appointment.
Doctor	Hold doctor's information.	Doctor's information such as specialization and department is entered by the staff into the system.
Payment	Hold the payment details from the patient.	Patients make their payment after medical service.
Medical_Service	Hold information about medical service provided to the patient.	Medical service is provided to the patient who visits the doctor.
Service_Medicine	Hold information about which medicines are used in a variety of medical services.	-
Medicine	Hold details of medicine used in the hospital.	Staff enter medicines information in the stock.
Appointment	Hold appointment information from the patient.	Patients book an appointment before seeing the doctor.
Consultation	Hold consultation information.	-
Vaccination	Hold vaccination information	-
CheckUp	Hold checkup information	-
Medical_Record	Hold patient medical records.	The medical record is entered by the doctor to a specific patient based on their symptoms.
Email	Hold patient's email addresses.	The email address of patient is stored in the system once the patient is entered.

4.2.2 Description of Relationship

Entity	**Multiplicity**	**Relationship**	**Multiplicity**	**Entity**
Patient	1..1	has	0..*	Medical_Record
	1..1	book	0..*	Appointment
	1..1	make	0..*	Payment
	1..1	has	1..2	Email
Doctor	1..1	document	1..*	Medical_Record
	1..1	has	0..*	Appointment
Payment	1..1	has	0..*	Medical_Service
Medical_Service	1..1	involves	0..*	Service_Medicine
Medicine	1..1	provide	0..*	Service_Medicine

4.2.3 Description of Attributes

Entity	Attributes	Description	Data Type	Constraint	Multi-valued
User	userID	User id, auto generated	NUMBER(10)	PRIMARY KEY	No
	fullName firstName lastName	Full name of user that consists of first name and last name	VARCHAR2(20)	NOT NULL	No
	contactNumber	Contact number of the user	VARCHAR(15)	NOT NULL	No
Patient	userID	User id of patient, auto generated	NUMBER(10)	PRIMARY KEY	No
	dateOfBirth	Birth date of the patient	DATE	NOT NULL	No
	gender	Gender of the patient	VARCHAR2(10)	NOT NULL	No
	address	Address of the patient	VARCHAR2(60)	NOT NULL	No
Doctor	userID	User id of doctor, auto generated	NUMBER(10)	PRIMARY KEY	No
	specialization	Specialization of the doctor	VARCHAR2(50)	NOT NULL	No
	department	Department that the doctor belongs to	VARCHAR2(30)	NOT NULL	No
Payment	paymentID	Record id, auto generated	NUMBER(10)	PRIMARY KEY	No
	paymentDate	Date of making payment	DATE	NOT NULL	No
	amount	Amount of payment need to paid	VARCHAR2(15)	NOT NULL	No
	paymentMethod	Method used to make payment	VARCHAR2(20)	NOT NULL	No
	paymentStatus	Status of the payment such as	VARCHAR2(20)	NOT NULL	No

		successful or fail			
	userID	User id, auto generated	NUMBER(10)	FOREIGN KEY	No
Medical_Service	serviceID	Medical service id, auto generated	NUMBER(10)	PRIMARY KEY	No
	serviceName	Medical service given to the patient	VARCHAR2(50)	NOT NULL	No
	category	Category of the medical service such as diagnosis or treatment	VARCHAR2(15)	NOT NULL	No
	cost	Cost for the medical service	VARCHAR2(10)	NOT NULL	No
	userID	User id of patient, auto generated	NUMBER(10)	FOREIGN KEY	No
Service_Medicine	serviceID	Medical service id, auto generated	NUMBER(10)	PRIMARY KEY	No
	medicineID	Medicine id, auto generated	NUMBER(10)	PRIMARY KEY	No
Medicine	medicineID	Medicine id, auto generated	NUMBER(10)	PRIMARY KEY	No
	medicineName	Name of the medicine	VARCHAR2(50)	NOT NULL	No
	medicineType	Type of the medicine such as liquid, tablet or capsules	VARCHAR2(15)	NOT NULL	No
	unitsInPack	Number of medicine packed in the each dispensing bags	NUMBER(5)	NOT NULL	No
	stockLevel	The remaining quantity of the medicine	NUMBER(10)	NOT NULL	No
Appointment	appointmentID	Appointment id, auto generated	NUMBER(10)	PRIMARY KEY	No

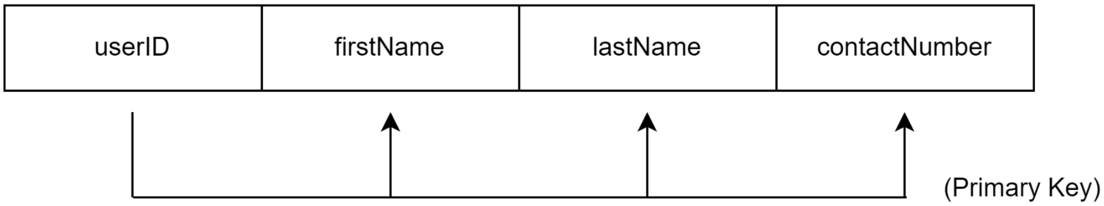
	appointmentDate	Appointment date	DATE	NOT NULL	No
	appointmentTime	Appointment time slot	TIMESTAMP	NOT NULL	No
	appointmentStatus	Status of appointment booking such as successful or fail	VARCHAR2(10)	NOT NULL	No
	appointmentType	The type of appointment	VARCHAR2(15)	NOT NULL	No
	patientID	User id of patient, foreign key from Patient relation	NUMBER(10)	FOREIGN KEY	No
	doctorID	User id of doctor, foreign key from Doctor relation	NUMBER(10)	FOREIGN KEY	No
Consultation	specialization	Specialization of the consultation such as general medicine, cardiology, dermatology or others	VARCHAR2(50)	NOT NULL	No
	priority	Priority of the consultation such as normal or urgent	VARCHAR2(25)	NOT NULL	No
	appointmentID	Appointment id, auto generated	NUMBER(10)	FOREIGN KEY	No
Vaccination	vaccineName	Vaccine name	VARCHAR2(50)	NOT NULL	No
	vaccineType	Type of vaccine for such as Influenza, Hepatitis B and others	VARCHAR2(25)	NOT NULL	No
	doseNumber	Number of dose taken	NUMBER(5)	NOT NULL	No
	appointmentID	Appointment id, auto generated	NUMBER(10)	FOREIGN KEY	No
CheckUp	checkUpType	Type of check up	VARCHAR2(50)	NOT NULL	No

		such as full body, diabetes or others			
	fastingHours	Recommended fasting time before check up	VARCHAR2(15)	NOT NULL	No
	testRequired	Tests included for the check up such as blood test, X-ray	VARCHAR2(10)	NOT NULL	No
	appointmentID	Appointment id, auto generated	NUMBER(10)	FOREIGN KEY	No
Medical Record	recordID	Record id, auto generated	NUMBER(10)	PRIMARY KEY	No
	recordedDate	Date for that specific medical service	DATE	NOT NULL	No
	symptoms	Symptoms for the patient	VARCHAR2(80)	NOT NULL	No
	doctorID	Medical service id, auto generated	NUMBER(10)	FOREIGN KEY	No
	patientID	Patient id, auto generated	NUMBER(10)	FOREIGN KEY	No
Email	userID	User id, auto generated	NUMBER(10)	NOT NULL	No
	emailAddress	Email address of the patient	VARCHAR2(50)	PRIMARY KEY	No

4.3. Normalization

1. User (userID, firstName, lastName, contactNumber)

   fd1 : userID → firstName, lastName, contactNumber (Primary Key)



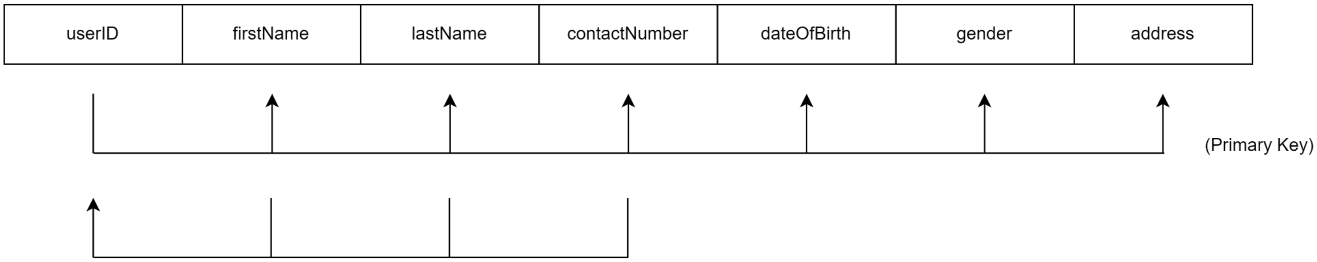
1NF&2NF&3NF&BCNF:

User (**<u>userID</u>**, firstName, lastName, contactNumber)

2. Patient(userID, firstName, lastName, contactNumber, dateOfBirth, gender, address)

   fd1 : userID → firstName, lastName, contactNumber, dateOfBirth, gender, address (Primary Key)

   fd2 : firstName, lastName, contactNumber → userID



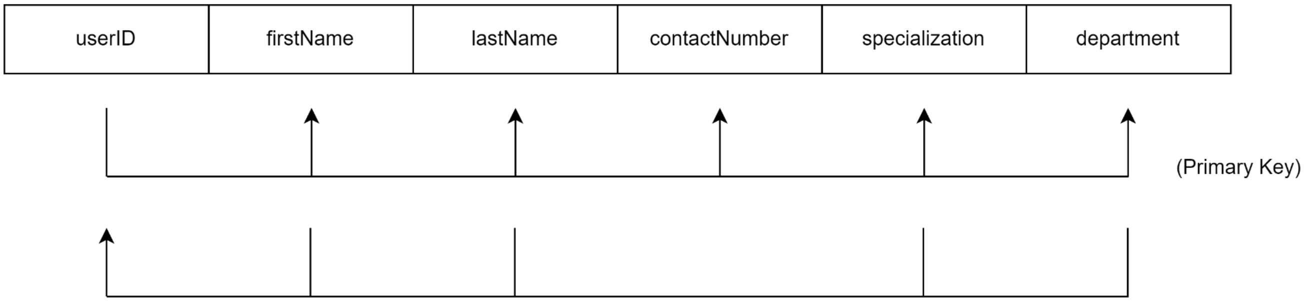
1NF&2NF&3NF&BCNF:

Patient (**<u>userID</u>**, firstName, lastName, contactNumber, dateOfBirth, gender, address)

3. Doctor(userID, firstName, lastName, contactNumber, specialization, department)

   fd1 : userID → firstName, lastName, contactNumber, specialization, department (Primary Key)

   fd2 : firstName, lastName, specialization, department → userID



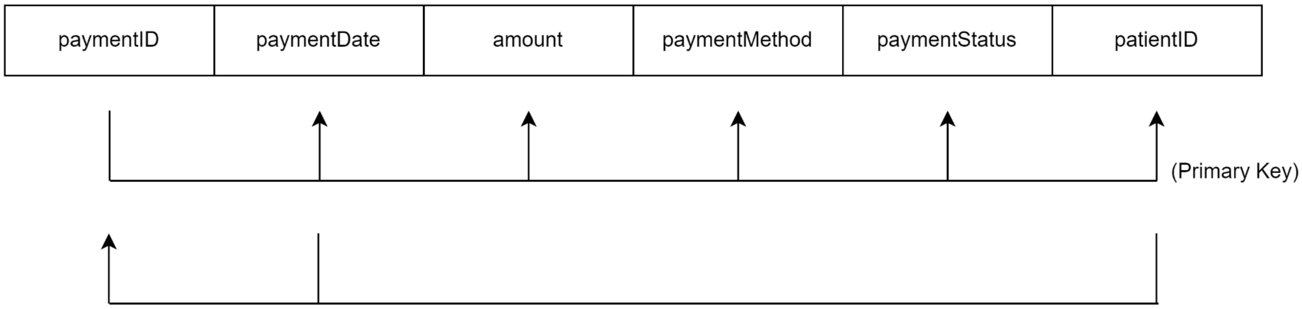
1NF&2NF&3NF&BCNF:

Doctor(**userID**, firstName, lastName, contactNumber, specialization, department)

4. Payment(paymentID, paymentDate, amount, paymentMethod, paymentStatus, *patientID*)

   fd1 : paymentID → paymentDate, amount, paymentMethod, paymentStatus, patientID (Primary Key)

   fd2 : patientID, paymentDate → paymentID



1NF&2NF&3NF&BCNF:

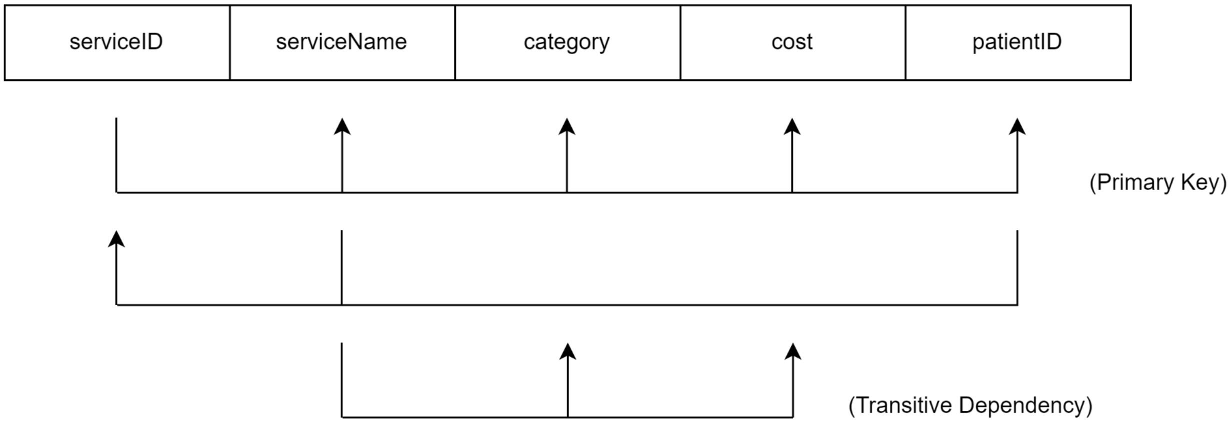
Payment(**paymentID**, paymentDate, amount, paymentMethod, paymentStatus, *patientID*)

5. Medical_Service(serviceID, serviceName, category, cost, *patientID*)

   fd1 : serviceID → serviceName, category, cost, patientID (Primary Key)

   fd2 : patientID, serviceName → serviceID

   fd3 : serviceName → category, cost (Transitive Dependency)



1NF&2NF:

Medical_Service(**serviceID**, serviceName, category, cost, *patientID*)3

3NF&BCNF :

Medical_Service(**serviceID**, serviceName, *patientID*)

Service_Detail(**serviceName**, category, cost)

6. Service_Medicine(**serviceID**, **medicineID**)

   fd1 : serviceID, medicineID → No other attributes (Primary Key)

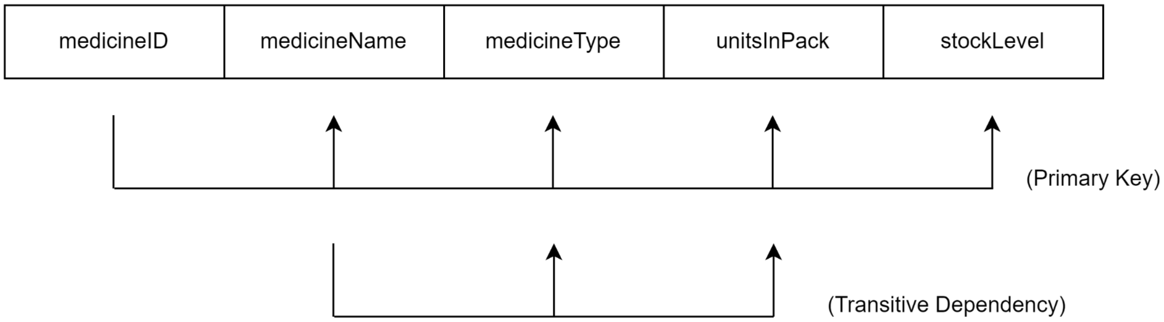
1NF&2NF&3NF&BCNF:

Service_Medicine(**serviceID**, **medicineID**)

7. Medicine(medicineID, medicineName, medicineType, unitsInPack, stockLevel)

   fd1 : medicineID → medicineName, medicineType, unitsInPack, stockLevel (Primary Key)

   fd2 : medicineName → medicineType, unitsInPack



1NF&2NF:

Medicine(**medicineID**, medicineName, medicineType, unitsInPack, stockLevel)

3NF&BCNF :

Medicine(**medicineID**, medicineName, stockLevel)

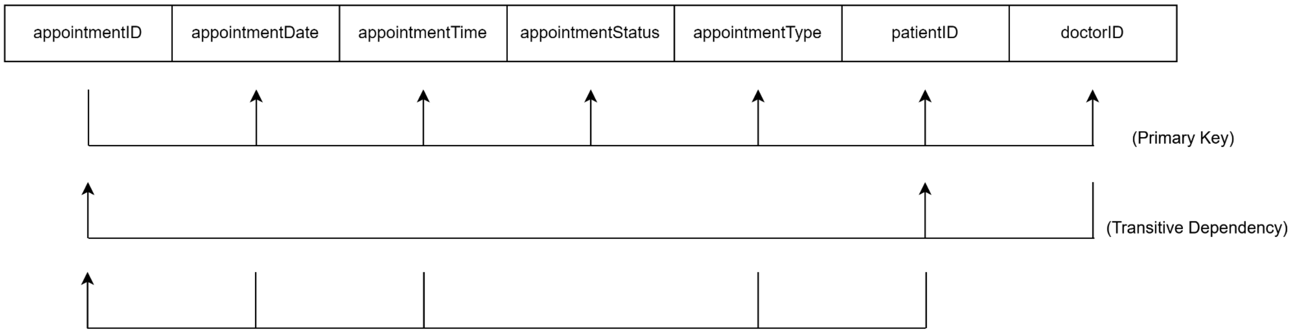
Medicine_Detail(**medicineName**, medicineType, unitsInPack)

8. Appointment(appointmentID, appointmentDate, appointmentTime, appointmentStatus, *appointmentType, patientID, doctorID*)

   fd1 : appointmentID → appointmentDate, appointmentTime, appointmentStatus, patientID, doctorID, appointmentType (Primary Key)

   fd2 : doctorID → appointmentID, patientID (Transitive Dependency)

   fd3 : appointmentDate, appointmentTime, appointmentType, patientID → appointmentID



1NF&2NF :

Appointment(**<u>appointmentID</u>**, appointmentDate, appointmentTime, appointmentStatus, *appointmentType, patientID*, *doctorID*)

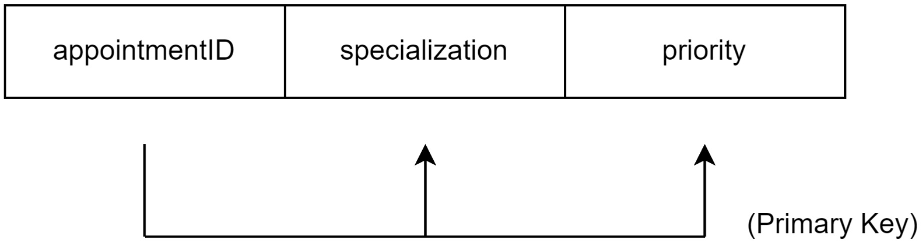
3NF&BCNF :

Appointment(**<u>appointmentID</u>**, appointmentDate, appointmentTime, appointmentStatus, *appointmentType, doctorID*)

Doctor_Appointment(**<u>doctorID</u>**, appointmentID, patientID)

9. Consultation(appointmentID, specialization, priority)

   fd1 : appointmentID→ specialization, priority (Primary Key)



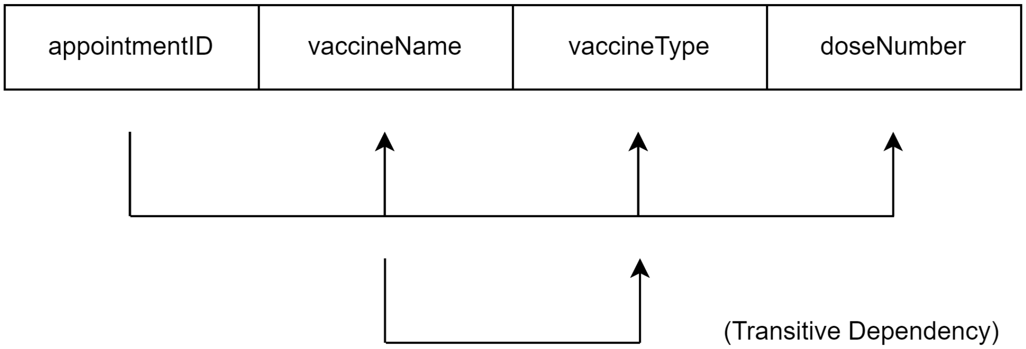
1NF&2NF&3NF&BCNF:

Consultation(**appointmentID**, specialization, priority)

10. Vaccination(appointmentID, vaccineName, vaccineType, doseNumber)

    fd1 : appointmentID → vaccineName, vaccineType, doseNumber (Primary Key)

    fd2 : vaccineName → vaccineType, doseNumber (Transitive Dependency)



1NF&2NF :

Vaccination(**appointmentID**, vaccineName, vaccineType, doseNumber)

&3NF&BCNF

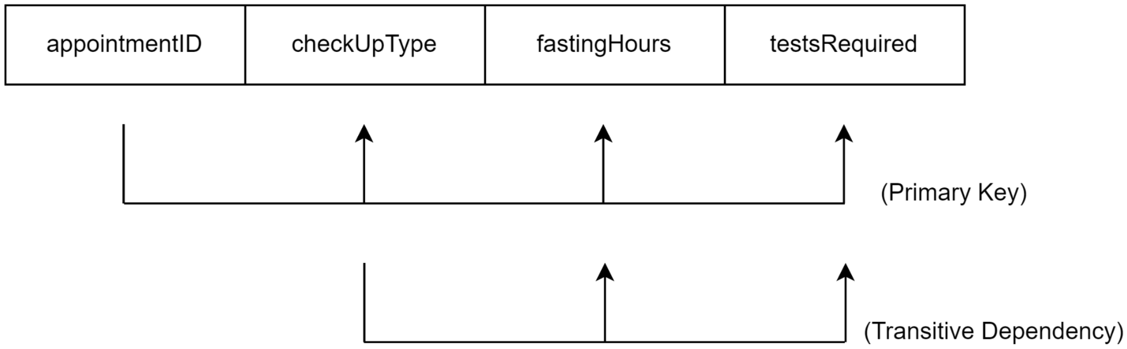
Vaccination(**appointmentID**, vaccineName, doseNumber)

Vaccine(**vaccineName**, vaccineType)

11. CheckUp(appointmentID, checkUpType, fastingHours, testsRequired)

fd1 : appointmentID → checkUpType, fastingHours, testsRequired (Primary Key)

fd2 : checkUpType → fastingHours, testsRequired (Transitive Dependency)



1NF&2NF :

CheckUp(**appointmentID**, checkUpType, fastingHours, testsRequired)

3NF&BCNF :

CheckUp(**appointmentID**, checkUpType)

CheckUpInfo(**checkUpType**, fastingHours, testsRequired)

12. Medical_Record(**recordID**, recordedDate, symptoms, *patientID, doctorID*)

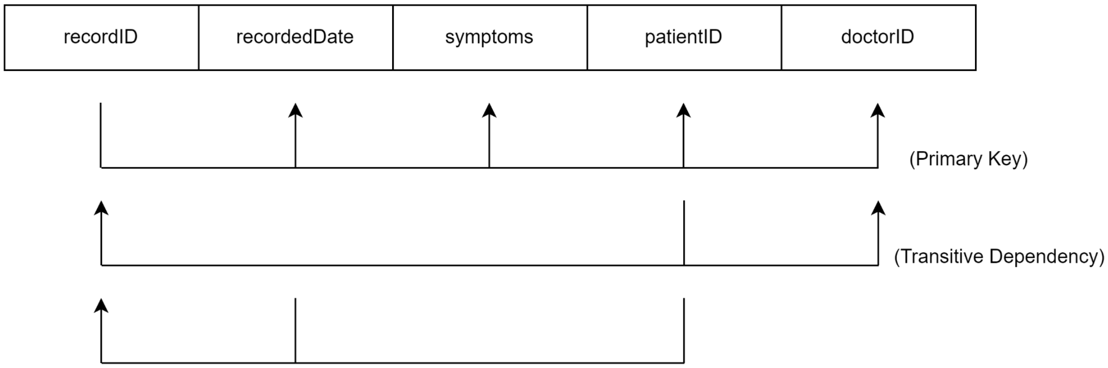
fd1 : recordID → recordedDate, symptoms, patientID, doctorID (Primary Key)

fd2 : patientID → recordID, doctorID (Transitive Dependency)

fd3 : patientID, recordedDate → recordID

1NF&2NF:

Medical_Record(**recordID**, recordedDate, symptoms, patientID*,* doctorID)



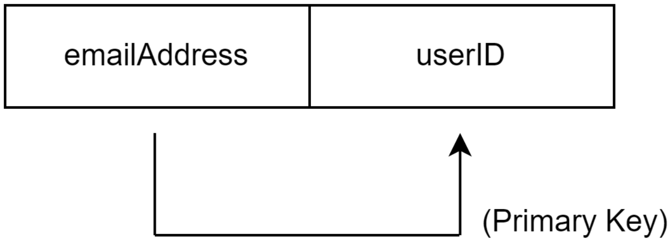
3NF&BCNF :

Medical_Record(**recordID**, recordedDate, symptoms, patientID)

Documented_Patient_Medical_Record(**patientID**, recordID, doctorID)

13. Email(userID, **emailAddress**)

fd1 : emailAddress → userID (Primary Key)



1NF&2NF&3NF&BCNF:

Email(**emailAddress,** userID)

5. Relational DB Schemas (after normalization)

The relational database schema for PKU Pro Max database after normalization is a set of relation schemas, namely :

User	(**userID**, firstName, lastName, contactNumber)
Patient	(**userID**, firstName, lastName, contactNumber, dateOfBirth, gender, address)
Doctor	(**userID**, firstName, lastName, contactNumber, specialization, department)
Payment	(**paymentID**, paymentDate, amount, paymentMethod, paymentStatus, *patientID*)
Medical_Service	(**serviceID**, serviceName, *patientID*)
Service_Detail	(**serviceName**, category, cost)
Service_Medicine	(**serviceID**, **medicineID**)
Medicine	(**medicineID**, medicineName, stockLevel)
Medicine_Detail	(**medicineName**, medicineType, unitsInPack)
Appointment	(**appointmentID**, appointmentDate, appointmentTime, appointmentStatus, *appointmentType, patientID, doctorID*)
Doctor_Appointment	(**doctorID**, appointmentID, patientID)
Consultation	(**appointmentID**, specialization, priority)
Vaccination	(**appointmentID**, vaccineName, doseNumber)
Vaccine	(**vaccineName**, vaccineType)
CheckUp	(**appointmentID**, checkUpType)

CheckUpInfo	(**checkUpType**, fastingHours, testsRequired)
Medical_Record	(**recordID**, recordedDate, symptoms, *patientID, doctorID*)
Documented_Patient_Medical_Record	(**patientID**, *recordID, doctorID*)
Email	(**emailAddress,** *userID*)

6. SQL Statements (DDL & DML)

6.1 Data Definition Language (DDL)

```
CREATE TABLE "USER" (
  userID              NUMBER(10) NOT NULL,
  firstName           VARCHAR2(20) NOT NULL,
  lastName            VARCHAR2(20) NOT NULL,
  contactNumber   VARCHAR(15) NOT NULL,
  CONSTRAINT user_pk PRIMARY KEY (userID)
);



CREATE TABLE Patient (
  userID          NUMBER(10) NOT NULL,
  dateOfBirth    DATE NOT NULL,
  gender          VARCHAR2(10) NOT NULL,
  address         VARCHAR2(60) NOT NULL,
  CONSTRAINT patient_fk FOREIGN KEY (userID)
     REFERENCES "USER" (userID),
  CONSTRAINT patient_pk PRIMARY KEY (userID)
);



CREATE TABLE Doctor (
  userID           NUMBER(10) NOT NULL,
  specialization  VARCHAR2(50) NOT NULL,
  department      VARCHAR2(30) NOT NULL,
  CONSTRAINT doctor_fk FOREIGN KEY (userID)
      REFERENCES "USER" (userID),
  CONSTRAINT doctor_pk PRIMARY KEY (userID)
);
```

```
);


CREATE TABLE Payment (
  paymentID          NUMBER(10) NOT NULL,
  paymentDate        DATE NOT NULL,
  amount             VARCHAR2(15) NOT NULL,
  paymentMethod      VARCHAR2(20) NOT NULL,
  paymentStatus      VARCHAR2(20) NOT NULL,
  patientID          NUMBER(10) NOT NULL,
  CONSTRAINT payment_fk FOREIGN KEY (patientID)
      REFERENCES Patient(userID),
  CONSTRAINT payment_pk PRIMARY KEY (paymentID)
);


CREATE TABLE Service_Detail (
serviceName    VARCHAR2(50) NOT NULL,
 category      VARCHAR2(20) NOT NULL,
 cost          VARCHAR2(10) NOT NULL,
 CONSTRAINT service_det_pk PRIMARY KEY (serviceName)
);


CREATE TABLE Medical_Service (
  serviceID      NUMBER(10) NOT NULL,
  serviceName    VARCHAR2(50) NOT NULL,
  patientID      NUMBER(10) NOT NULL,
  CONSTRAINT medical_service_fk1 FOREIGN KEY (patientID)
    REFERENCES Patient (userID),
  CONSTRAINT medical_service_fk2 FOREIGN KEY (serviceName)
      REFERENCES Service_Detail(serviceName),
  CONSTRAINT medical_service_pk PRIMARY KEY (serviceID)
);
```

```
CREATE TABLE Medicine_Detail (
medicineName   VARCHAR2(50) NOT NULL,
 medicineType    VARCHAR2(25) NOT NULL,
 unitsInPack        NUMBER(5) NOT NULL,
 CONSTRAINT medicine_det_pk PRIMARY KEY (medicineName)
);
```

```
CREATE TABLE Medicine (
  medicineID         NUMBER(10) NOT NULL,
  medicineName   VARCHAR2(50) NOT NULL,
  stockLevel          NUMBER(10) NOT NULL,
  CONSTRAINT medicine_fk FOREIGN KEY (medicineName)
      REFERENCES Medicine_Detail (medicineName),
  CONSTRAINT medicine_pk PRIMARY KEY (medicineID)
);
```

```
CREATE TABLE Service_Medicine (
  serviceID         NUMBER(10) NOT NULL,
  medicineID      NUMBER(10) NOT NULL,
  CONSTRAINT service_medicine_fk1 FOREIGN KEY (serviceID)
      REFERENCES Medical_Service (serviceID),
  CONSTRAINT service_medicine_fk2 FOREIGN KEY (medicineID)
      REFERENCES Medicine (medicineID),
  CONSTRAINT service_medicine_pk PRIMARY KEY (serviceID,medicineID)
);
```

```
CREATE TABLE Appointment (
  appointmentID        NUMBER(10) NOT NULL,
  appointmentDate      DATE NOT NULL,
  appointmentTime      TIMESTAMP NOT NULL,
  appointmentStatus    VARCHAR2(15) NOT NULL,
  appointmentType      VARCHAR2(20) NOT NULL,
  patientID            NUMBER(10) NOT NULL,
  doctorID             NUMBER(10) NOT NULL,
  CONSTRAINT appointment_fk1 FOREIGN KEY (patientID)
      REFERENCES Patient(userID),
  CONSTRAINT appointment_fk2 FOREIGN KEY (doctorID)
      REFERENCES Doctor(userID),
  CONSTRAINT appointment_pk PRIMARY KEY (appointmentID)
);
```

```
CREATE TABLE Doctor_Appointment (
  doctorID          NUMBER(10) NOT NULL,
  appointmentID     NUMBER(10) NOT NULL,
  patientID         NUMBER(10) NOT NULL,
  CONSTRAINT doctor_appointment_fk1 FOREIGN KEY (appointmentID)
     REFERENCES Appointment(appointmentID),
  CONSTRAINT doctor_appointment_fk2 FOREIGN KEY (patientID)
      REFERENCES Patient(userID),
  CONSTRAINT doctor_appointment_fk3 FOREIGN KEY (doctorID)
      REFERENCES Doctor(userID),
  CONSTRAINT doctor_appointment_pk PRIMARY KEY (doctorID)
);
```

```
CREATE TABLE Consultation (
  appointmentID    NUMBER(10) NOT NULL,
  specialization      VARCHAR2(50) NOT NULL,
  priority                VARCHAR2(25)  NOT NULL,
  CONSTRAINT consultant_fk FOREIGN KEY (appointmentID)
      REFERENCES Appointment (appointmentID),
  CONSTRAINT consultant_pk PRIMARY KEY (appointmentID)
);
```

```
CREATE TABLE Vaccine (
 vaccineName   VARCHAR2(50) NOT NULL,
 vaccineType     VARCHAR2(25) NOT NULL,
 CONSTRAINT vaccine_pk PRIMARY KEY (vaccineName)
);
```

```
CREATE TABLE Vaccination (
  appointmentID   NUMBER(10) NOT NULL,
  vaccineName     VARCHAR2(50) NOT NULL,
  doseNumber       NUMBER(5) NOT NULL,
  CONSTRAINT vaccination_fk FOREIGN KEY (appointmentID)
      REFERENCES Appointment (appointmentID),
  CONSTRAINT vaccine_fk FOREIGN KEY (vaccineName)
      REFERENCES Vaccine(vaccineName),
  CONSTRAINT vaccination_pk PRIMARY KEY (appointmentID)
);
```

```
CREATE TABLE CheckUpInfo (
 checkUpType      VARCHAR2(50) NOT NULL,
 fastingHours     VARCHAR2(15) NOT NULL,
 testRequired     VARCHAR2(10) NOT NULL,
 CONSTRAINT checkup_info_pk PRIMARY KEY (checkUpType)
);
```

```
CREATE TABLE CheckUp (
  appointmentID    NUMBER(10) NOT NULL,
  checkUpType      VARCHAR2(50) NOT NULL,
  CONSTRAINT checkup_fk FOREIGN KEY (appointmentID)
      REFERENCES Appointment (appointmentID),
 CONSTRAINT checkup_info_fk FOREIGN KEY (checkUpType )
      REFERENCES CheckUpInfo(checkUpType),
  CONSTRAINT checkup_pk PRIMARY KEY (appointmentID)
);
```

```
CREATE TABLE Medical_Record (
  recordID          NUMBER(10) NOT NULL,
  recordedDate      DATE NOT NULL,
  symptoms          VARCHAR2(80) NOT NULL,
  patientID         NUMBER(10) NOT NULL,
  doctorID          NUMBER(10) NOT NULL,
  CONSTRAINT medical_record_fk1 FOREIGN KEY (patientID)
      REFERENCES Patient (userID),
  CONSTRAINT medical_record_fk2 FOREIGN KEY (doctorID)
      REFERENCES Doctor (userID),
  CONSTRAINT medical_record_pk PRIMARY KEY (recordID)
);
```

```
CREATE TABLE Documented_Patient_Medical_Record(
  patientID     NUMBER(10) NOT NULL,
  recordID      NUMBER(10) NOT NULL,
  doctorID      NUMBER(10) NOT NULL,
  CONSTRAINT documented_patient_medical_record_fk1 FOREIGN KEY (patientID)
      REFERENCES Patient (userID),
  CONSTRAINT documented_patient_medical_record_fk2 FOREIGN KEY (recordID)
      REFERENCES Medical_Record (recordID),
  CONSTRAINT documented_patient_medical_record_fk3 FOREIGN KEY (doctorID)
      REFERENCES Doctor (userID),
  CONSTRAINT documented_patient_medical_record_pk PRIMARY KEY (patientID)
);


CREATE TABLE Email (
  userID              NUMBER(10) NOT NULL,
  emailAddress   VARCHAR2(50) NOT NULL,
  CONSTRAINT email_fk FOREIGN KEY (userID)
      REFERENCES Patient (userID),
  CONSTRAINT email_pk PRIMARY KEY (emailAddress)
);
```

6.2 Data Manipulation Language (DML)

```
------------------ Inserting 20 users (Patient) into the User table---------------------------------

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (1, 'John', 'Doe', '012-3456789');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (2, 'Jane', 'Smith', '017-8765432');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (3, 'Alice', 'Johnson', '014-1234567');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (4, 'Bob', 'Williams', '019-9876543');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (5, 'Eva', 'Miller', '016-2345678');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (6, 'David', 'Clark', '013-8765432');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (7, 'Sophie', 'Jones', '011-3456789');
```

```
INSERT INTO "USER" (userID, firstName, lastName, contactNumber)

VALUES (8, 'Michael', 'Taylor', '018-9876543');


INSERT INTO "USER" (userID, firstName, lastName, contactNumber)

VALUES (9, 'Olivia', 'Brown', '015-2345678');


INSERT INTO "USER" (userID, firstName, lastName, contactNumber)

VALUES (10, 'Daniel', 'Lee', '010-1234567');


INSERT INTO "USER" (userID, firstName, lastName, contactNumber)

VALUES (11, 'Grace', 'Anderson', '012-7890123');


INSERT INTO "USER" (userID, firstName, lastName, contactNumber)

VALUES (12, 'Henry', 'Moore', '017-8901234');


INSERT INTO "USER" (userID, firstName, lastName, contactNumber)

VALUES (13, 'Sophia', 'White', '014-5678901');


INSERT INTO "USER" (userID, firstName, lastName, contactNumber)

VALUES (14, 'William', 'Smith', '019-2345678');


INSERT INTO "USER" (userID, firstName, lastName, contactNumber)

VALUES (15, 'Ava', 'Martin', '016-7890123');


INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
```

```
VALUES (16, 'Liam', 'Johnson', '013-8901234');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (17, 'Emma', 'Davis', '011-5678901');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (18, 'Noah', 'Hall', '018-2345678');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (19, 'Isabella', 'Young', '015-7890123');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (20, 'Mason', 'Baker', '010-8901234');

-------------------Inserting 20 records(Doctor) into the User table---------------------------------------

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
 VALUES (101, 'Samantha', 'Turner', '011-1234567');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (102, 'Ethan', 'Reynolds', '011-2345678');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (103, 'Lily', 'Parker', '017-3456789');
```

```
INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (104, 'Owen', 'Mitchell', '018-4567890');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (105, 'Nora', 'Sullivan', '013-5678901');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (106, 'Sebastian', 'Foster', '010-6789012');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (107, 'Ruby', 'Fletcher', '017-7890123');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (108, 'Isaac', 'Bishop', '014-8901234');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (109, 'Hailey', 'Henderson', '012-9012345');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (110, 'Miles', 'Chandler', '018-0123456');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (111, 'Leah', 'Hammond', '019-1098765');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
```

```
VALUES (112, 'Sawyer', 'Daniels', '010-2109876');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (113, 'Stella', 'Hawkins', '013-3210987');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (114, 'Jaxon', 'Armstrong', '011-4321098');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (115, 'Peyton', 'Barrett', '016-5432109');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (116, 'Lincoln', 'Bowman', '017-6543210');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (117, 'Zoe', 'Holloway', '010-7654321');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (118, 'Ryder', 'Lawson', '014-8765432');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (119, 'Penelope', 'Robbins', '012-9876543');

INSERT INTO "USER" (userID, firstName, lastName, contactNumber)
VALUES (120, 'Gabriel', 'Thornton', '016-0987654');
```

```
--------------------------------- Inserting 20 records into the Patient table------------------------------------------

-- Using the existing User table for userID
-- Ensure that the User table already has at least 20 records with unique userID values

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (1, TO_DATE('1990-05-15', 'YYYY-MM-DD'), 'Male', '123 Jalan Ampang, Kuala Lumpur');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (2, TO_DATE('1985-08-22', 'YYYY-MM-DD'), 'Female', '456 Jalan Bukit Bintang, Kuala Lumpur');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (3, TO_DATE('1992-03-10', 'YYYY-MM-DD'), 'Male', '789 Jalan Pudu, Kuala Lumpur');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (4, TO_DATE('1988-11-05', 'YYYY-MM-DD'), 'Female', '101 Jalan Sultan Ismail, Penang');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (5, TO_DATE('1995-06-30', 'YYYY-MM-DD'), 'Male', '202 Jalan Tun Razak, Penang');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (6, TO_DATE('1983-09-17', 'YYYY-MM-DD'), 'Female', '303 Jalan Raja Chulan, Penang');
```

```
INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (7, TO_DATE('1998-02-25', 'YYYY-MM-DD'), 'Male', '404 Jalan Imbi, Kedah');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (8, TO_DATE('1980-12-08', 'YYYY-MM-DD'), 'Female', '505 Jalan Ampang Hilir, Kedah');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (9, TO_DATE('1993-07-12', 'YYYY-MM-DD'), 'Male', '606 Jalan Changkat, Sabah');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (10, TO_DATE('1987-04-05', 'YYYY-MM-DD'), 'Female', '707 Jalan Raja, Sabah');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (11, TO_DATE('1996-10-20', 'YYYY-MM-DD'), 'Male', '808 Jalan Bukit Nanas, Sarawak');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (12, TO_DATE('1982-01-13', 'YYYY-MM-DD'), 'Female', '909 Jalan Ampang Baru, Sarawak');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (13, TO_DATE('1991-08-28', 'YYYY-MM-DD'), 'Male', '111 Jalan Bukit Ceylon, Johor');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (14, TO_DATE('1986-05-03', 'YYYY-MM-DD'), 'Female', '222 Jalan Cochrane, Johor');
```

```
INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (15, TO_DATE('1994-12-18', 'YYYY-MM-DD'), 'Male', '333 Jalan Alor, Perak');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (16, TO_DATE('1989-07-07', 'YYYY-MM-DD'), 'Female', '444 Jalan Tuanku Abdul Rahman, Perak');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (17, TO_DATE('1997-02-15', 'YYYY-MM-DD'), 'Male', '555 Jalan Chow Kit, Terengganu');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (18, TO_DATE('1984-09-02', 'YYYY-MM-DD'), 'Female', '666 Jalan Dang Wangi, Terengganu');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (19, TO_DATE('1990-04-19', 'YYYY-MM-DD'), 'Male', '777 Jalan Sultan Hishamuddin, Kelantan');

INSERT INTO Patient (userID, dateOfBirth, gender, address)
VALUES (20, TO_DATE('1985-11-14', 'YYYY-MM-DD'), 'Female', '888 Jalan Ampang Kiri, Kelantan');

---------------------------------- Inserting 20 records into the Doctor table -----------------------------------------
-- Using user IDs starting from 101

INSERT INTO Doctor (userID, specialization, department)
VALUES (101, 'Allergy-Immunology', 'Allergies, Immunity');
```

```
INSERT INTO Doctor (userID, specialization, department)

VALUES (102, 'Orthopedics', 'Bones');


INSERT INTO Doctor (userID, specialization, department)

VALUES (103, 'Pediatrics', 'Heart');


INSERT INTO Doctor (userID, specialization, department)

VALUES (104, 'ENT', 'Ear, Nose, Throat');


INSERT INTO Doctor (userID, specialization, department)

VALUES (105,  'Infectious Diseases', 'Infectious Diseases');


INSERT INTO Doctor (userID, specialization, department)

VALUES (106, 'Dermatology', 'Skin');


INSERT INTO Doctor (userID, specialization, department)

VALUES (107, 'Orthopedics', 'Bones');


INSERT INTO Doctor (userID, specialization, department)

VALUES (108, 'Ophthalmology', 'Eyes');


INSERT INTO Doctor (userID, specialization, department)

VALUES (109, 'Cardiology', 'Heart');


INSERT INTO Doctor (userID, specialization, department)
```

```
VALUES (110, 'Urology', 'Urinary System');

INSERT INTO Doctor (userID, specialization, department)
VALUES (111, 'Orthopedics', 'Bones');

INSERT INTO Doctor (userID, specialization, department)
VALUES (112, 'Rheumatology', 'Joints');

INSERT INTO Doctor (userID, specialization, department)
VALUES (113, 'Psychiatry', 'Heart');

INSERT INTO Doctor (userID, specialization, department)
VALUES (114, 'ENT', 'Ear, Nose, Throat');

INSERT INTO Doctor (userID, specialization, department)
VALUES (115, 'Nephrology', 'Kidneys');

INSERT INTO Doctor (userID, specialization, department)
VALUES (116, 'General Surgery', 'Surgery');

INSERT INTO Doctor (userID, specialization, department)
VALUES (117, 'Dermatology', 'Skin');

INSERT INTO Doctor (userID, specialization, department)
VALUES (118, 'Allergy-Immunology', 'Allergies, Immunity');
```

```
INSERT INTO Doctor (userID, specialization, department)

VALUES (119, 'Infectious Diseases', 'Infectious Diseases');


INSERT INTO Doctor (userID, specialization, department)

VALUES (120, 'Urology', 'Urinary System');



----------------------------------- Inserting 20 records into the Payment table ----------------------------------------

-- Using user IDs from the Patient table


INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)

VALUES (1, TO_DATE('2023-05-15', 'YYYY-MM-DD'), 'RM 100.00', 'Credit Card', 'Paid', 1);


INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)

VALUES (2, TO_DATE('2023-06-20', 'YYYY-MM-DD'), 'RM 150.50', 'Cash', 'Paid', 2);


INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)

VALUES (3, TO_DATE('2023-07-10', 'YYYY-MM-DD'), 'RM 120.75', 'Debit Card', 'Paid', 3);


INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)

VALUES (4, TO_DATE('2023-08-05', 'YYYY-MM-DD'), 'RM 200.00', 'Credit Card', 'Pending', 4);


INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)

VALUES (5, TO_DATE('2023-09-22', 'YYYY-MM-DD'), 'RM 75.25', 'E-wallet', 'Paid', 5);
```

```
INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (6, TO_DATE('2023-10-18', 'YYYY-MM-DD'), 'RM 180.50', 'Credit Card', 'Paid', 6);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (7, TO_DATE('2023-11-10', 'YYYY-MM-DD'), 'RM 90.25', 'Debit Card', 'Paid', 7);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (8, TO_DATE('2023-12-15', 'YYYY-MM-DD'), 'RM 150.00', 'Cash', 'Pending', 8);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (9, TO_DATE('2023-01-05', 'YYYY-MM-DD'), 'RM 120.75', 'E-wallet', 'Paid', 9);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (10, TO_DATE('2023-02-20', 'YYYY-MM-DD'), 'RM 100.00', 'Cash', 'Paid', 10);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (11, TO_DATE('2023-03-18', 'YYYY-MM-DD'), 'RM 200.50', 'Debit Card', 'Pending', 11);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (12, TO_DATE('2023-04-10', 'YYYY-MM-DD'), 'RM 80.25', 'Credit Card', 'Paid', 12);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (13, TO_DATE('2023-05-15', 'YYYY-MM-DD'), 'RM 150.00', 'Cash', 'Paid', 13);
```

```
INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (14, TO_DATE('2023-06-18', 'YYYY-MM-DD'), 'RM 130.75', 'Debit Card', 'Paid', 14);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (15, TO_DATE('2023-07-05', 'YYYY-MM-DD'), 'RM 180.50', 'Credit Card', 'Pending', 15);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (16, TO_DATE('2023-08-22', 'YYYY-MM-DD'), 'RM 90.25', 'E-wallet', 'Paid', 16);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (17, TO_DATE('2023-09-18', 'YYYY-MM-DD'), 'RM 120.00', 'Debit Card', 'Paid', 17);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (18, TO_DATE('2023-10-10', 'YYYY-MM-DD'), 'RM 75.50', 'Credit Card', 'Paid', 18);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (19, TO_DATE('2023-11-15', 'YYYY-MM-DD'), 'RM 160.25', 'Cash', 'Pending', 19);

INSERT INTO Payment (paymentID, paymentDate, amount, paymentMethod, paymentStatus, patientID)
VALUES (20, TO_DATE('2023-11-21', 'YYYY-MM-DD'), 'RM 140.00', 'E-wallet', 'Paid', 20);


---------------------------- Inserting 10 records into the Service_Detail table --------------------------------

INSERT INTO Service_Detail (serviceName, category, cost)
```

```
VALUES ('Cardiac Consultation', 'Cardiology', 'RM 150.00');

INSERT INTO Service_Detail (serviceName, category, cost)
VALUES ('X-ray Imaging', 'Radiology', 'RM 75.50');

INSERT INTO Service_Detail (serviceName, category, cost)
VALUES ('Physical Therapy Session', 'Rehabilitation', 'RM 90.00');

INSERT INTO Service_Detail (serviceName, category, cost)
VALUES ('Blood Test', 'Laboratory', 'RM 45.25');

INSERT INTO Service_Detail (serviceName, category, cost)
VALUES ('Urinary Test', 'Laboratory', 'RM 55.00');

INSERT INTO Service_Detail (serviceName, category, cost)
VALUES ('Dental Checkup', 'Dentistry', 'RM 60.50');

INSERT INTO Service_Detail (serviceName, category, cost)
VALUES ('ENT Examination', 'Otolaryngology', 'RM 80.25');

INSERT INTO Service_Detail (serviceName, category, cost)
VALUES ('Eye Checkup', 'Ophthalmology', 'RM 70.00');

INSERT INTO Service_Detail (serviceName, category, cost)
VALUES ('Medical Checkup', 'General Medicine', 'RM 120.00');
```

```
--------------------- Inserting 10 records into the Medical_Service table -------------------------------

-- Using user IDs from the Patient table and service names from the Service_Detail table

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
VALUES (1, 'Cardiac Consultation', 1);

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
VALUES (2, 'X-ray Imaging', 2);

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
VALUES (3, 'Physical Therapy Session', 3);

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
VALUES (4, 'Blood Test', 4);

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
VALUES (5, 'Urinary Test', 5);

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
VALUES (6, 'Dental Checkup', 6);

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
```

```
VALUES (7, 'ENT Examination', 7);

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
VALUES (8, 'Eye Checkup', 8);

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
VALUES (9, 'Medical Checkup', 9);

INSERT INTO Medical_Service (serviceID, serviceName, patientID)
VALUES (10, 'Blood Test', 10);


--------------------------- Inserting 10 records into the Medicine_Detail table -------------------------

INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)
VALUES ('Aspirin', 'Pain Relief', 20);

INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)
VALUES ('Amoxicillin', 'Antibiotic', 30);

INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)
VALUES ('Ibuprofen', 'Pain Relief', 25);

INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)
VALUES ('Loratadine', 'Antihistamine', 15);
```

```
INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)

VALUES ('Omeprazole', 'Antacid', 10);


INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)

VALUES ('Simvastatin', 'Cholesterol Lowering', 30);


INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)

VALUES ('Metformin', 'Antidiabetic', 40);


INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)

VALUES ('Albuterol', 'Bronchodilator', 20);


INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)

VALUES ('Ciprofloxacin', 'Antibiotic', 15);


INSERT INTO Medicine_Detail (medicineName, medicineType, unitsInPack)

VALUES ('Paracetamol', 'Pain Relief', 25);



---------------------------- Inserting 10 records into the Medicine table ---------------------------------


INSERT INTO Medicine (medicineID, medicineName, stockLevel)

VALUES (1, 'Aspirin', 100);
```

```
INSERT INTO Medicine (medicineID, medicineName, stockLevel)
VALUES (2, 'Amoxicillin', 150);

INSERT INTO Medicine (medicineID, medicineName, stockLevel)
VALUES (3, 'Ibuprofen', 120);

INSERT INTO Medicine (medicineID, medicineName, stockLevel)
VALUES (4, 'Loratadine', 80);

INSERT INTO Medicine (medicineID, medicineName, stockLevel)
VALUES (5, 'Omeprazole', 50);

INSERT INTO Medicine (medicineID, medicineName, stockLevel)
VALUES (6, 'Simvastatin', 200);

INSERT INTO Medicine (medicineID, medicineName, stockLevel)
VALUES (7, 'Metformin', 180);

INSERT INTO Medicine (medicineID, medicineName, stockLevel)
VALUES (8, 'Albuterol', 90);

INSERT INTO Medicine (medicineID, medicineName, stockLevel)
VALUES (9, 'Ciprofloxacin', 70);

INSERT INTO Medicine (medicineID, medicineName, stockLevel)
```

```
VALUES (10, 'Paracetamol', 120);


------------------------ Inserting 10 records into the Service_Medicine table --------------------------

-- Using service and medicine IDs from their respective tables

INSERT INTO Service_Medicine (serviceID, medicineID)
VALUES (1, 1);

INSERT INTO Service_Medicine (serviceID, medicineID)
VALUES (2, 2);

INSERT INTO Service_Medicine (serviceID, medicineID)
VALUES (3, 3);

INSERT INTO Service_Medicine (serviceID, medicineID)
VALUES (4, 4);

INSERT INTO Service_Medicine (serviceID, medicineID)
VALUES (5, 5);

------------------------- Inserting 20 records into the Appointment table -----------------------------

-- Using user IDs from the Patient and Doctor tables
```

```
INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (1, TO_DATE('2023-01-10', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-01-10 09:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'Rescheduled', 'Check-up', 9, 101);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (2, TO_DATE('2023-01-15', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-01-15 10:30:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Consultation',12 , 111);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (3, TO_DATE('2023-02-20', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-02-20 14:15:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Vaccination',17 , 118);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (4, TO_DATE('2023-02-25', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-02-25 11:45:00',
'YYYY-MM-DD HH24:MI:SS'), 'Rescheduled', 'Check-up', 10, 104);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (5, TO_DATE('2023-03-30', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-03-30 13:30:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Consultation',15 , 114);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (6, TO_DATE('2023-04-05', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-04-05 08:45:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Vaccination',20 , 120);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (7, TO_DATE('2023-04-10', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-04-10 12:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'Rescheduled', 'Check-up',11 , 107);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (8, TO_DATE('2023-05-15', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-05-15 14:45:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Consultation',3 , 110 );

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (9, TO_DATE('2023-05-20', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-05-20 10:30:00',
'YYYY-MM-DD HH24:MI:SS'), 'Rescheduled', 'Vaccination', 5, 115 );

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (10, TO_DATE('2023-06-25', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-06-25 13:15:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Check-up', 4, 102);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (11, TO_DATE('2023-06-30', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-06-30 09:30:00',
'YYYY-MM-DD HH24:MI:SS'), 'Rescheduled', 'Consultation', 18, 112);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (12, TO_DATE('2023-07-05', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-07-05 11:45:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Vaccination',8 , 106);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (13, TO_DATE('2023-07-10', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-07-10 15:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Check-up',1 , 119);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (14, TO_DATE('2023-08-15', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-08-15 09:15:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Consultation', 19, 108);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (15, TO_DATE('2023-09-20', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-09-20 12:30:00',
'YYYY-MM-DD HH24:MI:SS'), 'Rescheduled', 'Vaccination', 13, 113);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (16, TO_DATE('2023-09-25', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-09-25 14:45:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Check-up',2 , 116);
```

```
INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (17, TO_DATE('2023-10-06', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-10-06 10:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Consultation',16 , 103);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (18, TO_DATE('2023-11-05', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-11-05 13:15:00',
'YYYY-MM-DD HH24:MI:SS'), 'Rescheduled', 'Vaccination',6 , 109);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (19, TO_DATE('2023-11-21', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-11-21 15:30:00',
'YYYY-MM-DD HH24:MI:SS'), 'Scheduled', 'Check-up', 7, 117);

INSERT INTO Appointment (appointmentID, appointmentDate, appointmentTime, appointmentStatus,
appointmentType, patientID, doctorID)
VALUES (20, TO_DATE('2023-12-15', 'YYYY-MM-DD'), TO_TIMESTAMP('2023-12-15 11:45:00',
'YYYY-MM-DD HH24:MI:SS'), 'Rescheduled', 'Consultation', 14, 105);


------------------------ Inserting 20 records into the Doctor_Appointment table ------------------------

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (101, 1, 9);
```

```
INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (111, 2, 12);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (118, 3, 17);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (104, 4, 10);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (114, 5, 15);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (120, 6, 20);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (107, 7, 11);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (110, 8, 3);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (115, 9, 5);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
```

```
VALUES (102, 10, 4);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (112, 11, 18);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (106, 12, 8);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (119, 13, 1);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (108, 14, 19);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (113, 15, 13);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (116, 16, 2);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (103, 17, 16);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (109, 18, 6);
```

```
INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (117, 19, 7);

INSERT INTO Doctor_Appointment (doctorID, appointmentID, patientID)
VALUES (105, 20, 14);


------------------------ Inserting records into the Consultation table --------------------------------

INSERT INTO Consultation (appointmentID, specialization, priority)
VALUES (2, 'Allergy-Immunology', 'High');

INSERT INTO Consultation (appointmentID, specialization, priority)
VALUES (5, 'ENT (Ear, Nose, Throat)', 'Low');

INSERT INTO Consultation (appointmentID, specialization, priority)
VALUES (8, 'Urology', 'Medium');

INSERT INTO Consultation (appointmentID, specialization, priority)
VALUES (11, 'Infectious Diseases', 'High');

INSERT INTO Consultation (appointmentID, specialization, priority)
VALUES (14, 'Allergy-Immunology', 'Low');
```

```sql
INSERT INTO Consultation (appointmentID, specialization, priority)

VALUES (17, 'Cardiology', 'Medium');


INSERT INTO Consultation (appointmentID, specialization, priority)

VALUES (20, 'ENT (Ear, Nose, Throat)', 'Medium');



-------------------- Inserting records into the Vaccine table ---------------------------------------


INSERT INTO Vaccine (vaccineName, vaccineType)

VALUES ('Pfizer-BioNTech', 'mRNA');


INSERT INTO Vaccine (vaccineName, vaccineType)

VALUES ('AstraZeneca-Oxford', 'Viral Vector');


INSERT INTO Vaccine (vaccineName, vaccineType)

VALUES ('Sinovac (CoronaVac)', 'Inactivated Virus');



--------------------- Inserting records into the Vaccination table ------------------------------------


INSERT INTO Vaccination (appointmentID, vaccineName, doseNumber)

VALUES (3, 'Pfizer-BioNTech', 1);


INSERT INTO Vaccination (appointmentID, vaccineName, doseNumber)
```

```
VALUES (6, 'AstraZeneca-Oxford', 1);


INSERT INTO Vaccination (appointmentID, vaccineName, doseNumber)

VALUES (9, 'Sinovac (CoronaVac)', 1);


INSERT INTO Vaccination (appointmentID, vaccineName, doseNumber)

VALUES (12, 'Pfizer-BioNTech', 2);


INSERT INTO Vaccination (appointmentID, vaccineName, doseNumber)

VALUES (15, 'AstraZeneca-Oxford', 2);


INSERT INTO Vaccination (appointmentID, vaccineName, doseNumber)

VALUES (18, 'Sinovac (CoronaVac)', 2);



---------------------- Inserting records into the CheckUpInfo table ----------------------------------


INSERT INTO CheckUpInfo (checkUpType, fastingHours, testRequired)

VALUES ('General Check-up', '8 hours', 'Yes');


INSERT INTO CheckUpInfo (checkUpType, fastingHours, testRequired)

VALUES ('Cardiac Check-up', '12 hours', 'Yes');


INSERT INTO CheckUpInfo (checkUpType, fastingHours, testRequired)

VALUES ('Blood Pressure Check-up', '6 hours', 'No');
```

```
-------------------------- Inserting records into the CheckUp table --------------------------------

INSERT INTO CheckUp (appointmentID, checkUpType)
VALUES (1, 'General Check-up');

INSERT INTO CheckUp (appointmentID, checkUpType)
VALUES (4, 'Cardiac Check-up');

INSERT INTO CheckUp (appointmentID, checkUpType)
VALUES (7, 'Blood Pressure Check-up');

INSERT INTO CheckUp (appointmentID, checkUpType)
VALUES (10, 'Cardiac Check-up');

INSERT INTO CheckUp (appointmentID, checkUpType)
VALUES (13, 'General Check-up');

INSERT INTO CheckUp (appointmentID, checkUpType)
VALUES (16, 'Blood Pressure Check-up');

INSERT INTO CheckUp (appointmentID, checkUpType)
VALUES (19, 'General Check-up');
```

```
---------------------- Inserting records into the Medical_Record table ---------------------------------


-- Record for Appointment ID: 1
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (1, TO_DATE('2023-01-10', 'YYYY-MM-DD'), 'Fever and cough', 9, 101);


-- Record for Appointment ID: 2
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (2, TO_DATE('2023-01-15', 'YYYY-MM-DD'), 'Headache and fatigue', 12, 111);


-- Record for Appointment ID: 3
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (3, TO_DATE('2023-02-20', 'YYYY-MM-DD'), 'Sore arm after vaccination', 17, 118);


-- Record for Appointment ID: 4
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (4, TO_DATE('2023-02-25', 'YYYY-MM-DD'), 'Routine check-up, no specific symptoms', 10, 104);


-- Record for Appointment ID: 5
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (5, TO_DATE('2023-03-30', 'YYYY-MM-DD'), 'Consultation for general health', 15, 114);


-- Record for Appointment ID: 6
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
```

```
VALUES (6, TO_DATE('2023-04-05', 'YYYY-MM-DD'), 'Slight fever after vaccination', 20, 120);

-- Record for Appointment ID: 7
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (7, TO_DATE('2023-04-10', 'YYYY-MM-DD'), 'Comprehensive check-up, no specific symptoms', 11, 107);

-- Record for Appointment ID: 8
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (8, TO_DATE('2023-05-15', 'YYYY-MM-DD'), 'Follow-up for chronic condition', 3, 110);

-- Record for Appointment ID: 9
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (9, TO_DATE('2023-05-20', 'YYYY-MM-DD'), 'No specific symptoms', 5, 115);

-- Record for Appointment ID: 10
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (10, TO_DATE('2023-06-25', 'YYYY-MM-DD'), 'Routine check-up, no specific symptoms', 4, 102);

-- Record for Appointment ID: 11
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (11, TO_DATE('2023-06-30', 'YYYY-MM-DD'), 'Migraine and nausea', 18, 112);

-- Record for Appointment ID: 12
```

```
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (12, TO_DATE('2023-07-05', 'YYYY-MM-DD'), 'No specific symptoms ', 8, 106);

-- Record for Appointment ID: 13
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (13, TO_DATE('2023-07-10', 'YYYY-MM-DD'), 'Routine check-up, no specific symptoms', 1, 119);

-- Record for Appointment ID: 14
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (14, TO_DATE('2023-08-15', 'YYYY-MM-DD'), 'Consultation for back pain', 19, 108);

-- Record for Appointment ID: 15
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (15, TO_DATE('2023-09-20', 'YYYY-MM-DD'), 'Fatigue and weakness', 13, 113);

-- Record for Appointment ID: 16
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (16, TO_DATE('2023-09-25', 'YYYY-MM-DD'), 'Routine check-up, no specific symptoms', 2, 116);


-- Record for Appointment ID: 17
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (17, TO_DATE('2023-10-06', 'YYYY-MM-DD'), 'Headache and dizziness', 16, 103);
```

```
-- Record for Appointment ID: 18
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (18, TO_DATE('2023-11-05', 'YYYY-MM-DD'), 'No specific symptoms', 6, 109);

-- Record for Appointment ID: 19
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (19, TO_DATE('2023-11-21', 'YYYY-MM-DD'), 'Routine check-up, no specific symptoms', 7, 117);

-- Record for Appointment ID: 20
INSERT INTO Medical_Record (recordID, recordedDate, symptoms, patientID, doctorID)
VALUES (20, TO_DATE('2023-12-15', 'YYYY-MM-DD'), 'Nasal congestion and sore throat', 14, 105);


-------------- Inserting records into the Documented_Patient_Medical_Record table ---------------------

-- Record for Patient ID: 9
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (9, 1, 101);

-- Record for Patient ID: 12
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (12, 2, 111);
```

```
-- Record for Patient ID: 17

INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)

VALUES (17, 3, 118);


-- Record for Patient ID: 10

INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)

VALUES (10, 4, 104);


-- Record for Patient ID: 15

INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)

VALUES (15, 5, 114);


-- Record for Patient ID: 20

INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)

VALUES (20, 6, 120);


-- Record for Patient ID: 11

INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)

VALUES (11, 7, 107);


-- Record for Patient ID: 3

INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)

VALUES (3, 8, 110);


-- Record for Patient ID: 5
```

```
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (5, 9, 115);

-- Record for Patient ID: 4
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (4, 10, 102);

-- Record for Patient ID: 18
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (18, 11, 112);

-- Record for Patient ID: 8
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (8, 12, 106);

-- Record for Patient ID: 1
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (1, 13, 119);

-- Record for Patient ID: 19
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (19, 14, 108);

-- Record for Patient ID: 13
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
```

```
VALUES (13, 15, 113);

-- Record for Patient ID: 2
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (2, 16, 116);

-- Record for Patient ID: 16
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (16, 17, 103);

-- Record for Patient ID: 6
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (6, 18, 109);

-- Record for Patient ID: 7
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (7, 19, 117);

-- Record for Patient ID: 14
INSERT INTO Documented_Patient_Medical_Record (patientID, recordID, doctorID)
VALUES (14, 20, 105);


------------------------------ Inserting records into the Email table -----------------------------------------
```

```
-- Record for User ID: 1
INSERT INTO Email (userID, emailAddress)
VALUES (1, 'john.doe@hotmail.com');

-- Record for User ID: 2
INSERT INTO Email (userID, emailAddress)
VALUES (2, 'jane.smith@gmail.com');

-- Record for User ID: 3
INSERT INTO Email (userID, emailAddress)
VALUES (3, 'alice.johnson@yahoo.com');

-- Record for User ID: 4
INSERT INTO Email (userID, emailAddress)
VALUES (4, 'bob.williams@hotmail.com');

-- Record for User ID: 5
INSERT INTO Email (userID, emailAddress)
VALUES (5, 'eva.miller@gmail.com');

-- Record for User ID: 6
INSERT INTO Email (userID, emailAddress)
VALUES (6, 'david.clark@yahoo.com');

-- Record for User ID: 7
```

```
INSERT INTO Email (userID, emailAddress)
VALUES (7, 'sophie.jones@hotmail.com');

-- Record for User ID: 8
INSERT INTO Email (userID, emailAddress)
VALUES (8, 'michael.taylor@gmail.com');

-- Record for User ID: 9
INSERT INTO Email (userID, emailAddress)
VALUES (9, 'olivia.brown@yahoo.com');

-- Record for User ID: 10
INSERT INTO Email (userID, emailAddress)
VALUES (10, 'daniel.lee@hotmail.com');

-- Record for User ID: 11
INSERT INTO Email (userID, emailAddress)
VALUES (11, 'grace.anderson@gmail.com');

-- Record for User ID: 12
INSERT INTO Email (userID, emailAddress)
VALUES (12, 'henry.moore@yahoo.com');

-- Record for User ID: 13
INSERT INTO Email (userID, emailAddress)
```

```sql
VALUES (13, 'sophia.white@hotmail.com');

-- Record for User ID: 14
INSERT INTO Email (userID, emailAddress)
VALUES (14, 'william.smith@gmail.com');

-- Record for User ID: 15
INSERT INTO Email (userID, emailAddress)
VALUES (15, 'ava.martin@yahoo.com');

-- Record for User ID: 16
INSERT INTO Email (userID, emailAddress)
VALUES (16, 'liam.johnson@hotmail.com');

-- Record for User ID: 17
INSERT INTO Email (userID, emailAddress)
VALUES (17, 'emma.davis@gmail.com');

-- Record for User ID: 18
INSERT INTO Email (userID, emailAddress)
VALUES (18, 'noah.hall@yahoo.com');

-- Record for User ID: 19
INSERT INTO Email (userID, emailAddress)
VALUES (19, 'isabella.young@hotmail.com');
```

```
-- Record for User ID: 20
INSERT INTO Email (userID, emailAddress)
VALUES (20, 'mason.baker@gmail.com');
```

```
--Update contactNumber for the existing userID 18 in the "USER" table
UPDATE "USER"
SET contactNumber = '011-4455114'
WHERE userID = 18;

--Update contactNumber for the existing userID 20 in the "USER" table
UPDATE "USER"
SET contactNumber = '012-3456777'
WHERE userID = 20;

--Update gender for the existing userID 11 in the Patient table
UPDATE Patient
SET gender = 'Male'
WHERE userID = 11;

--Update gender for the existing userID 14 in the Patient table
UPDATE Patient
SET gender = 'Female'
WHERE userID = 14;

--Update doctorID for the existing patientID 13 in the Appointment table
UPDATE Appointment
SET doctorID = 119
WHERE patientID = 13;
```

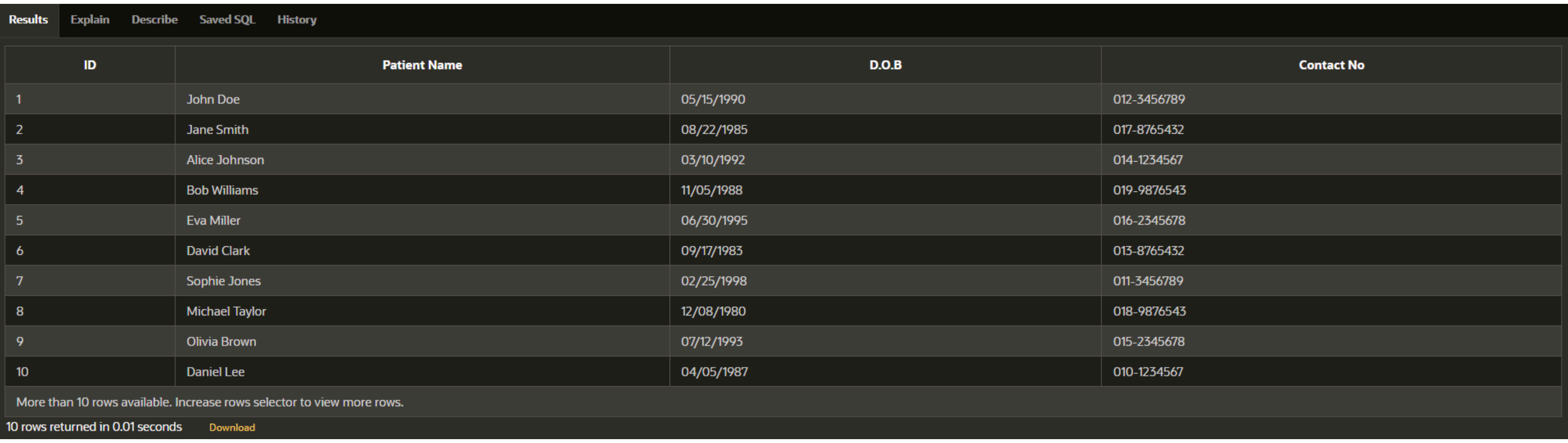
```
--Update doctorID for the existing patientID 15 in the Appointment table
UPDATE Appointment
SET doctorID = 119
WHERE patientID = 15;

--Update priority for the existing appointmentID 17 in the Consultation table
UPDATE Consultation
SET priority = 'Low'
WHERE appointmentID = 17;

--Update priority for the existing appointmentID 20 in the Consultation table
UPDATE Consultation
SET priority = 'High'
WHERE appointmentID = 20;

--Update emailAddress for the existing userID 12 in the Email table
UPDATE Email
SET emailAddress = 'kawaii.huan@hotmail.com'
WHERE userID = 12;

--Update emailAddress for the existing userID 14 in the Email table
UPDATE Email
SET emailAddress = 'xiangcai.huan@hotmail.com'
WHERE userID = 14;
```

```
--Remove the existing row from the Documented_Patient_Medical_Record that patientID is 6

DELETE Documented_Patient_Medical_Record

WHERE patientID = 6;


--Remove the existing row from the Documented_Patient_Medical_Record that patientID is 7

DELETE Documented_Patient_Medical_Record

WHERE patientID = 7;


--Remove the existing row from the Email that userID is 18

DELETE Email

WHERE userID = 18;


--Remove the existing row from the Email that userID is 19

DELETE Email

WHERE userID = 19;
```

6.3 Test Queries

1. Display patient list

```
SELECT userID "ID", firstName || ' ' || lastName AS "Patient Name", dateOfBirth "D.O.B", contactNumber "Contact No"

FROM Patient p JOIN "USER" u

USING (userID);
```

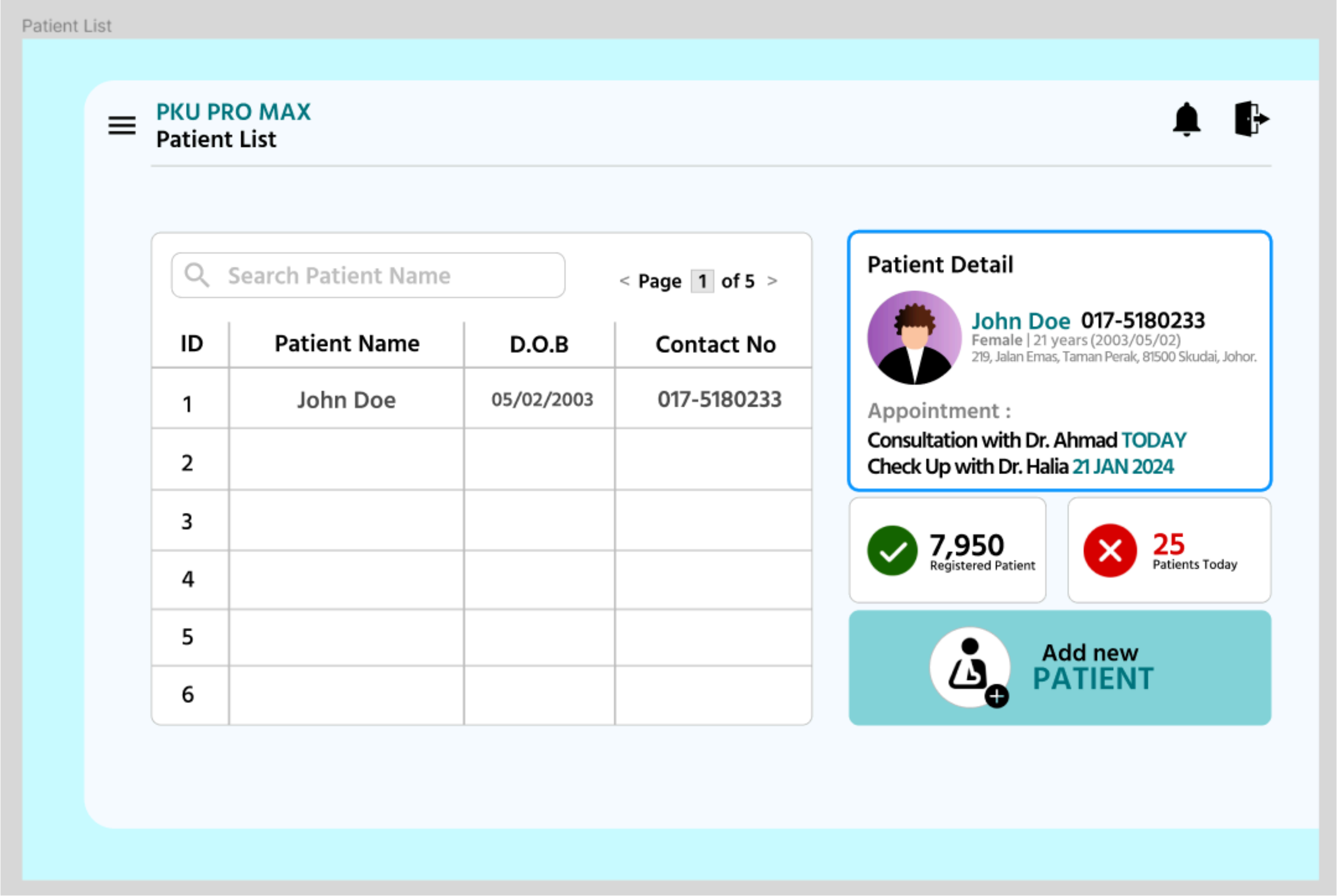


ID	Patient Name	D.O.B	Contact No
1	John Doe	05/15/1990	012-3456789
2	Jane Smith	08/22/1985	017-8765432
3	Alice Johnson	03/10/1992	014-1234567
4	Bob Williams	11/05/1988	019-9876543
5	Eva Miller	06/30/1995	016-2345678
6	David Clark	09/17/1983	013-8765432
7	Sophie Jones	02/25/1998	011-3456789
8	Michael Taylor	12/08/1980	018-9876543
9	Olivia Brown	07/12/1993	015-2345678
10	Daniel Lee	04/05/1987	010-1234567

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds

User Interface:



ID	Patient Name	D.O.B	Contact No
1	John Doe	05/02/2003	017-5180233
2			
3			
4			
5			
6			

Patient Detail

John Doe 017-5180233
Female | 21 years (2003/05/02)
219, Jalan Emas, Taman Perak, 81500 Skudai, Johor.

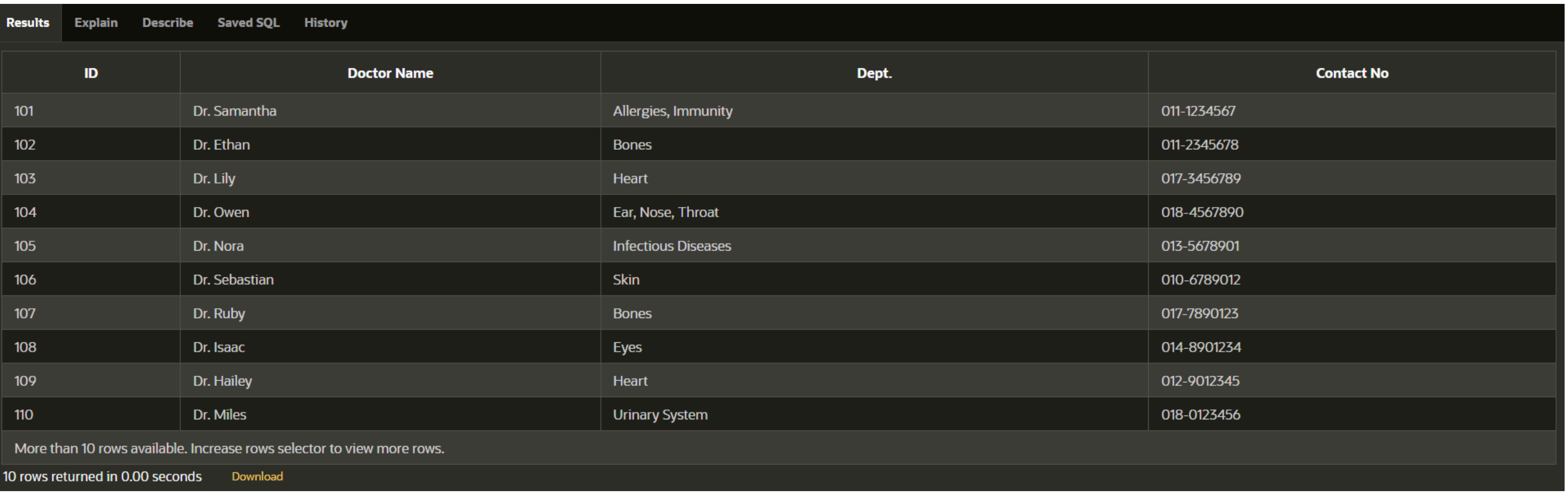
Appointment :
Consultation with Dr. Ahmad TODAY
Check Up with Dr. Halia 21 JAN 2024

✓ 7,950 Registered Patient

✗ 25 Patients Today

2. **Display doctor list**

```
SELECT userID "ID", 'Dr. ' || firstName AS "Doctor Name", department "Dept.", contactNumber "Contact No"

FROM Doctor d JOIN "USER" u

USING (userID);
```

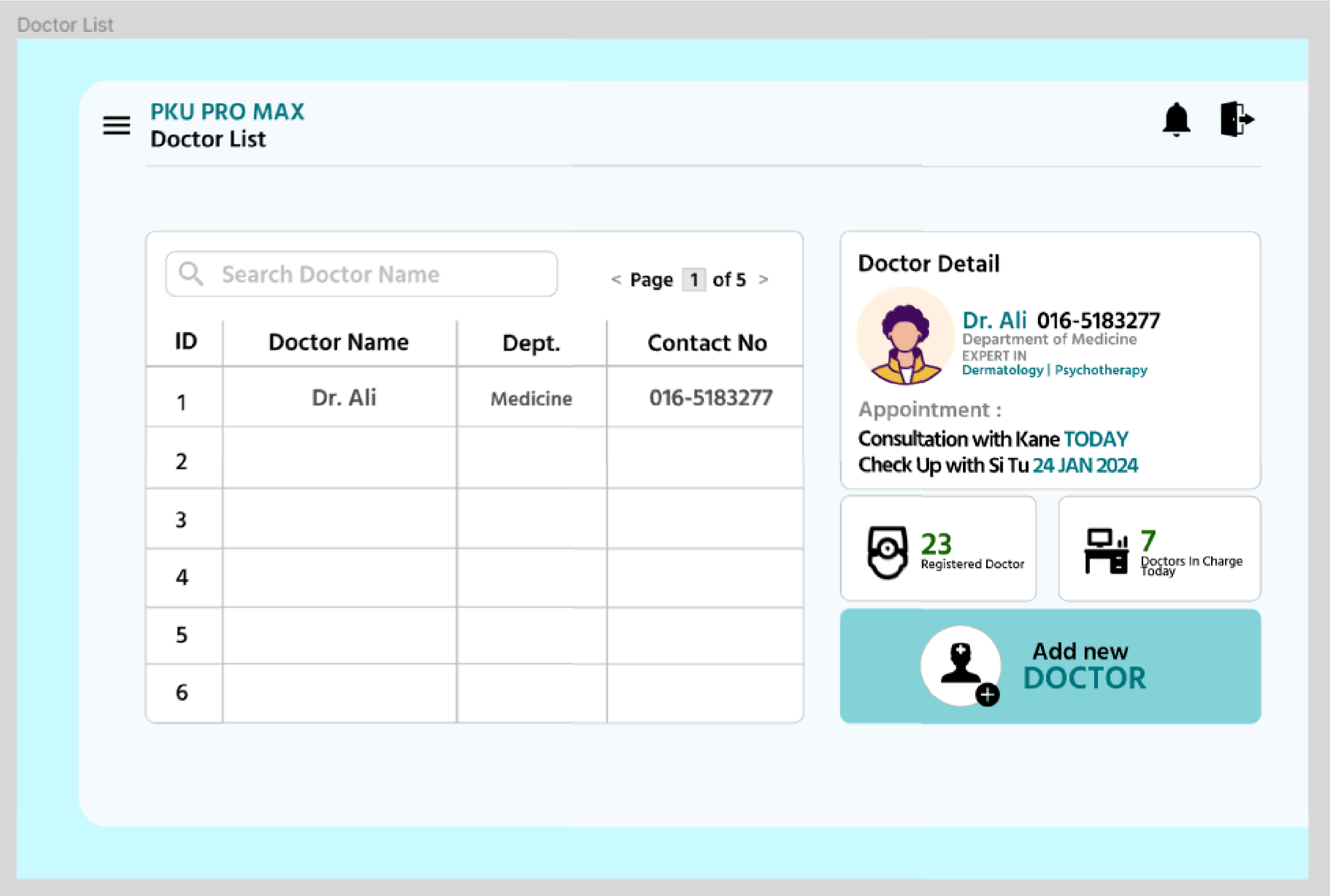


ID	Doctor Name	Dept.	Contact No
101	Dr. Samantha	Allergies, Immunity	011-1234567
102	Dr. Ethan	Bones	011-2345678
103	Dr. Lily	Heart	017-3456789
104	Dr. Owen	Ear, Nose, Throat	018-4567890
105	Dr. Nora	Infectious Diseases	013-5678901
106	Dr. Sebastian	Skin	010-6789012
107	Dr. Ruby	Bones	017-7890123
108	Dr. Isaac	Eyes	014-8901234
109	Dr. Hailey	Heart	012-9012345
110	Dr. Miles	Urinary System	018-0123456

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds

<u>User Interface:</u>



ID	Doctor Name	Dept.	Contact No
1	Dr. Ali	Medicine	016-5183277
2			
3			
4			
5			
6			

Doctor Detail

Dr. Ali 016-5183277
Department of Medicine
EXPERT IN
Dermatology | Psychotherapy

Appointment :
Consultation with Kane TODAY
Check Up with Si Tu 24 JAN 2024

23 Registered Doctor

7 Doctors In Charge Today

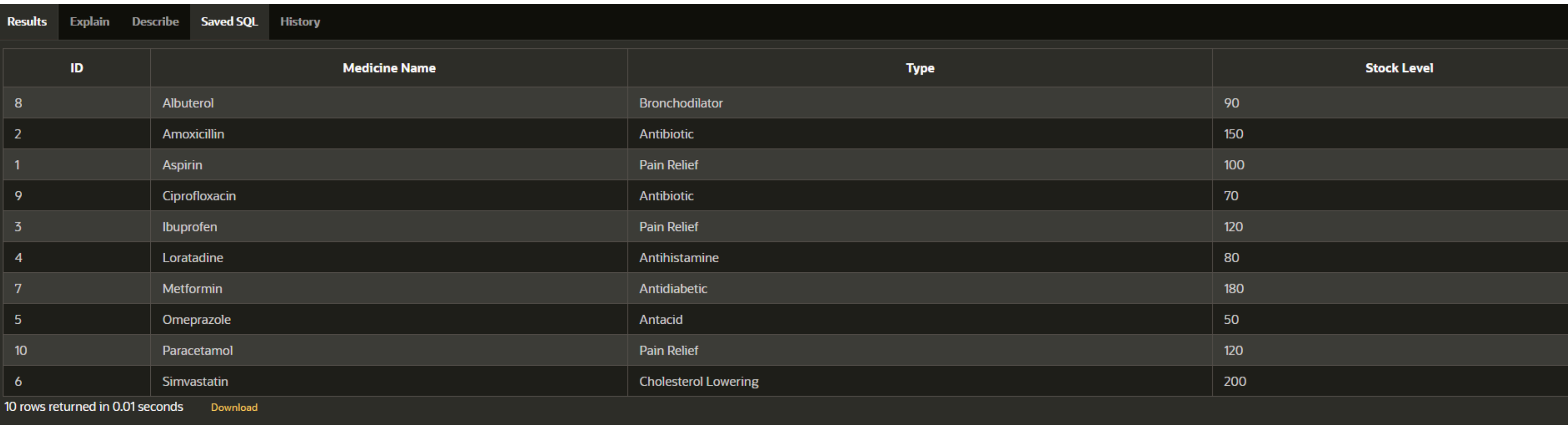
3. **View medicine list**

SELECT medicineID "ID", medicineName AS "Medicine Name", medicineType "Type", stockLevel "Stock Level"

FROM Medicine m JOIN Medicine_Detail md

USING (medicineName)

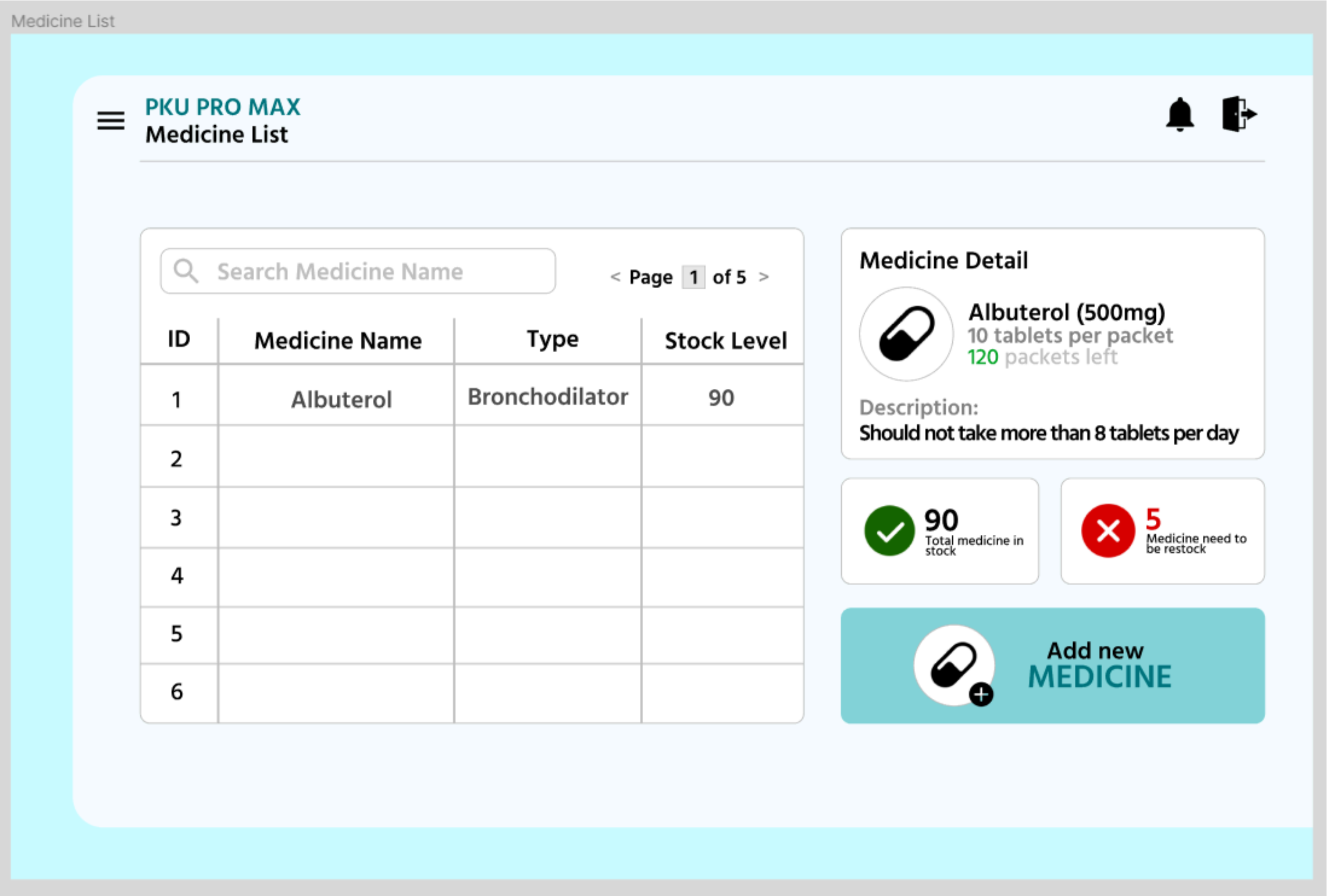
ORDER BY medicineName;



ID	Medicine Name	Type	Stock Level
8	Albuterol	Bronchodilator	90
2	Amoxicillin	Antibiotic	150
1	Aspirin	Pain Relief	100
9	Ciprofloxacin	Antibiotic	70
3	Ibuprofen	Pain Relief	120
4	Loratadine	Antihistamine	80
7	Metformin	Antidiabetic	180
5	Omeprazole	Antacid	50
10	Paracetamol	Pain Relief	120
6	Simvastatin	Cholesterol Lowering	200

10 rows returned in 0.01 seconds

User Interface:



ID	Medicine Name	Type	Stock Level
1	Albuterol	Bronchodilator	90
2			
3			
4			
5			
6			

Medicine Detail

Albuterol (500mg)
10 tablets per packet
120 packets left

Description:
Should not take more than 8 tablets per day

90 Total medicine in stock

5 Medicine need to be restock

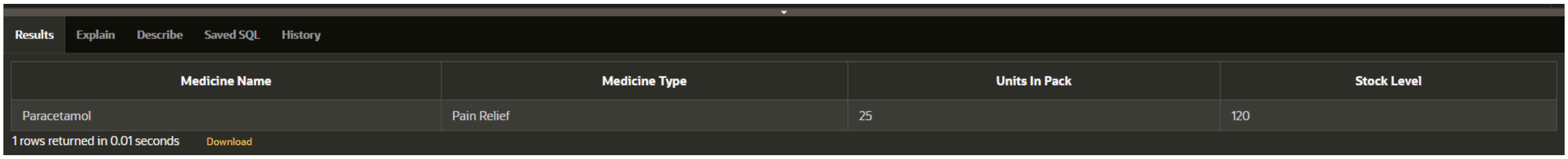
4. **Check medicine using medicine name**

SELECT medicineName "Medicine Name", medicineType "Medicine Type", unitsInPack "Units In Pack", stockLevel "Stock Level"

FROM Medicine m JOIN Medicine_Detail md

USING (medicineName)

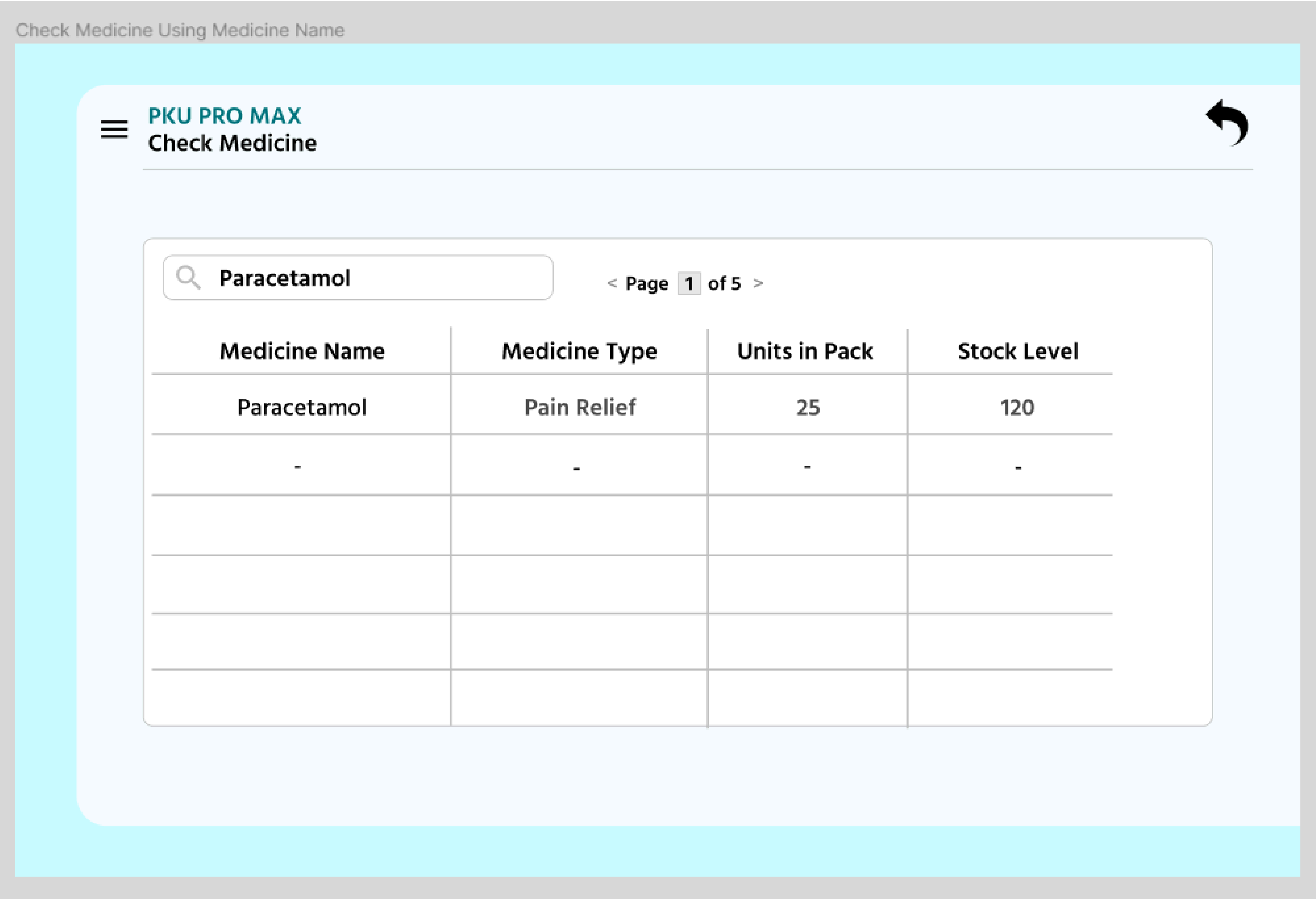
WHERE medicineName = 'Paracetamol';



Medicine Name	Medicine Type	Units In Pack	Stock Level
Paracetamol	Pain Relief	25	120

1 rows returned in 0.01 seconds

<u>User Interface:</u>

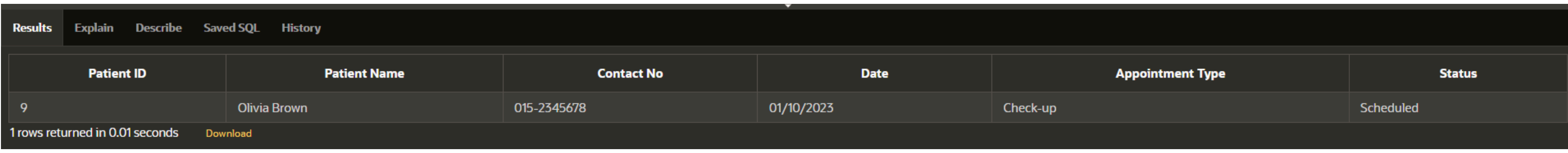


Check Medicine Using Medicine Name

Medicine Name	Medicine Type	Units in Pack	Stock Level
Paracetamol	Pain Relief	25	120
-	-	-	-

5. **A doctor checks his/her appointment (using doctor id)**

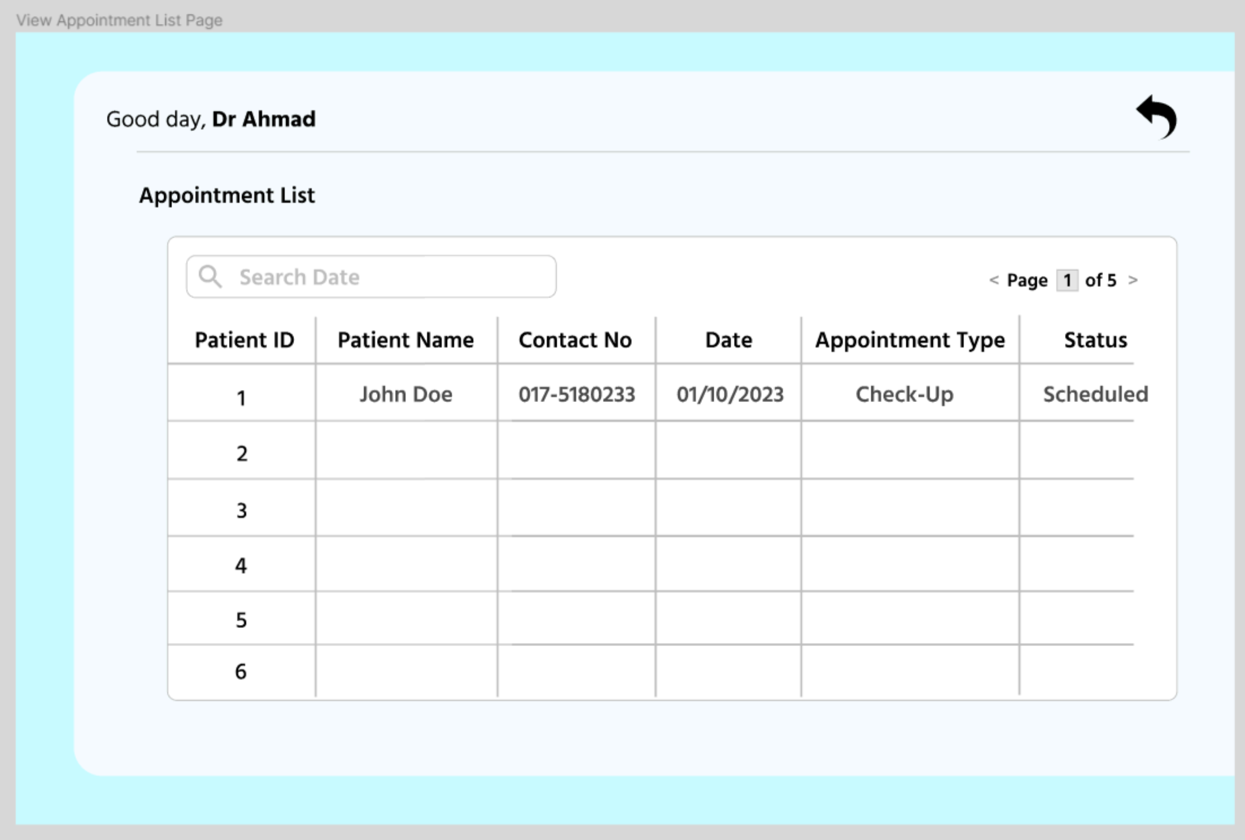
```
SELECT da.patientID "Patient ID", u.firstName ||' '||u.lastName "Patient Name", u.contactNumber "Contact No", appointmentDate  "Date", appointmentType "Appointment Type", appointmentStatus "Status"

FROM Doctor_Appointment da JOIN Appointment p

USING (doctorID)

JOIN  "USER" u

ON da.patientID = u.userID

WHERE doctorID = 101;
```



Patient ID	Patient Name	Contact No	Date	Appointment Type	Status
9	Olivia Brown	015-2345678	01/10/2023	Check-up	Scheduled

1 rows returned in 0.01 seconds

User Interface:



Good day, **Dr Ahmad**

Appointment List

Patient ID	Patient Name	Contact No	Date	Appointment Type	Status
1	John Doe	017-5180233	01/10/2023	Check-Up	Scheduled
2					
3					
4					
5					
6					

7. Summary

In this last phase, we had done the database logical design and the SQL implementation for the PKU Pro Max System. In this phase 3, we transformed the conceptual entity relationship diagram (cERD) and enhanced entity relationship diagram (EERD) in phase 2 into logical ERD by determining the functional dependencies between the relationships based on the updated business rules. We generated the updated data dictionary for the PKU Pro Max System by referring to the logical ERD. The updated data dictionary is necessary as the data dictionary provides detailed information about all the attributes, entities and relationships. After that, we performed the normalization process to minimize the duplicated data so that the data are stored logically. Then, we performed the normalization process from the first normal form (1NF) to the Boyce-Codd normal form (BCNF) to minimize the redundancy of the data and make the PKU Pro Max System easy to access and manipulate the PKU Pro Max System efficiently without compromising the data integrity which can benefit in the future work. After the normalization process, we will conclude the relational DB schemas to ease the SQL implementation process. Next, we created the tables by using the SQL statements through the Oracle Apex which included both Data Definition Language (DDL) and Data Manipulation Language (DML) for our PKU Pro Max System. Last but not least, we also performed some test queries that relate to the functions provided by our PKU Pro Max System such as displaying patient lists, displaying doctor lists, displaying medicine lists, checking medicine using medicine names, checking appointments for a doctor and others.

In conclusion, we hope that throughout this phase, we can produce a functional and user-friendly system to help the Pusat Kesihatan Universiti (PKU) health clinic to cover issues such as medicine being outdated as lack of checking from time to time, PKU staff need to manually include the paper appointment books which are not efficient, scattered paper medical records and others. We trust that the PKU Pro Max system can benefit both UTM students and PKU staff from self-service options like online appointment scheduling with reminders, paperless check-in/out, and mobile access to their health records.