

Healthcare Management System Database Documentation

Table of Contents

Introduction

Purpose of the Document

Scope and Objectives

Database Design Overview

Entity-Relationship Diagram (ERD)

Database Schema Description

Data Dictionary

Table Creation Queries

3.1 Insurance Table

3.2 Department Table

3.3 Patient Table

3.4 Doctor Table

3.5 MedicalRecord Table

3.6 Prescription Table

3.7 Billing Table

Detailed Explanations and Sample Values for Each Table

1. Introduction

The Healthcare Management System database aims to streamline the management of healthcare-related information, including patient data, medical records, insurance details, and billing. This document serves as a comprehensive guide to understanding the database structure, table design, and data relationships.

1.1 Purpose of the Document

The purpose of this document is to provide developers, stakeholders, and system users with a clear understanding of the Healthcare Management System database. It explains the tables' purpose, the data they store, the relationships between tables, and sample queries to retrieve valuable information.

1.2 Scope and Objectives

The scope of this document encompasses the database design and structure for the Healthcare Management System. It includes the creation queries for each table and sample data to populate the tables for demonstration purposes. The document aims to:

Provide an overview of the database schema and its components.

Explain the purpose and usage of each table in the system.

Describe the relationships between tables to highlight data dependencies.

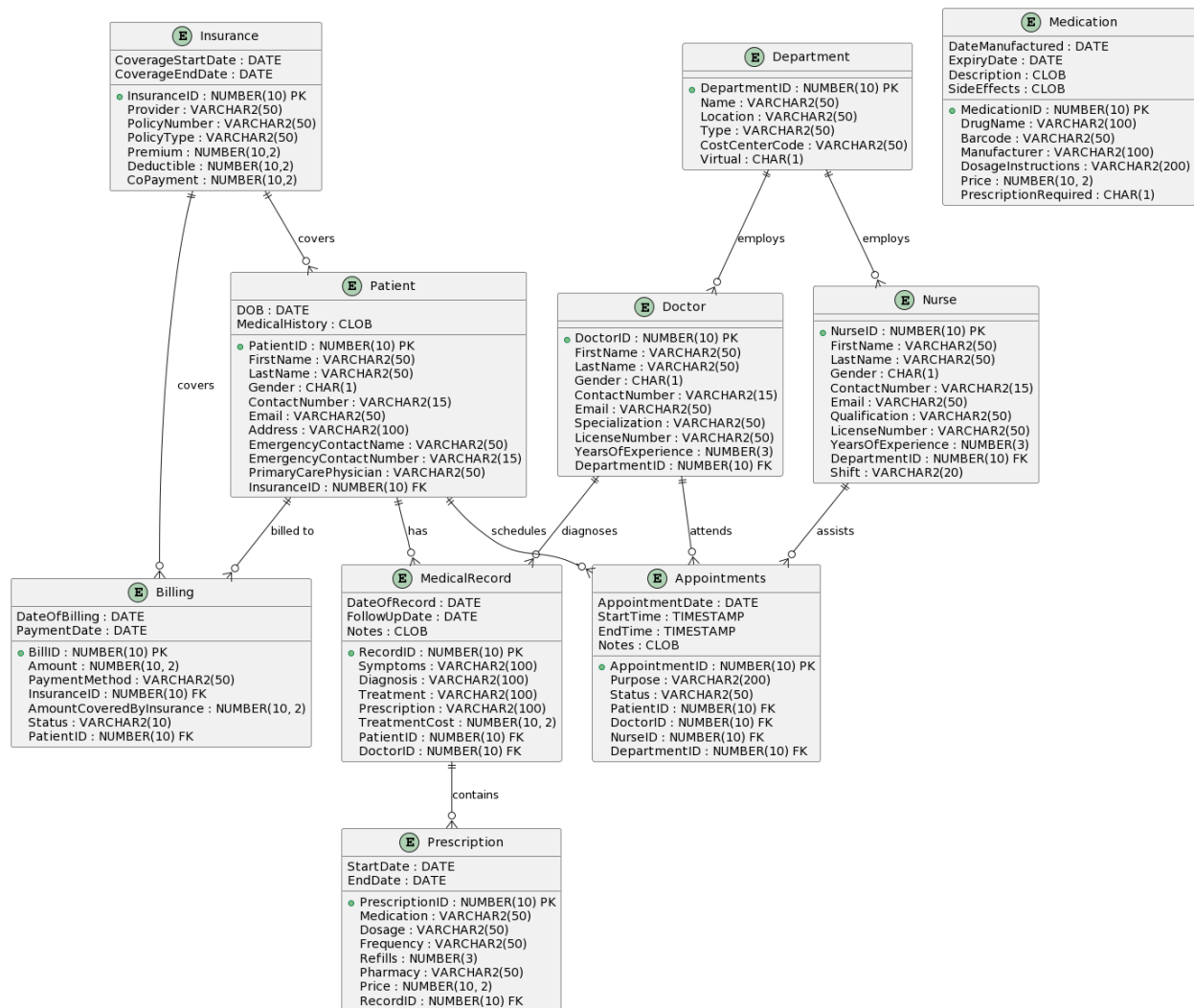
Offer sample data to illustrate how the tables store real-world information.
 Showcase sample queries to retrieve relevant information from the database.

2. Database Design Overview

2.1 Entity-Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) visually represents the database schema and the relationships between different entities (tables). The ERD for the Healthcare Management System database is as follows:

[Insert ERD Image]



Entities:

Insurance: Represents insurance policies that patients can have.

Department: Represents different departments within the healthcare facility.

Patient: Represents individuals receiving medical care.

Doctor: Represents medical professionals providing diagnoses and treatments.

Nurse: Represents nursing staff assisting doctors and caring for patients.

MedicalRecord: Represents records of medical consultations, including symptoms, diagnosis, and treatment.

Prescription: Represents medication prescriptions given to patients.

Billing: Represents billing information related to medical services.

Appointments: Represents scheduled meetings between patients and medical staff.

Medication: Represents different medications that can be prescribed.

Relationships:

Insurance - Patient:

Type: One-to-Many

Description: One insurance policy can cover multiple patients, but each patient has only one insurance policy. This is represented by the InsuranceID foreign key in the Patient table.

Department - Doctor:

Type: One-to-Many

Description: One department can employ multiple doctors, but each doctor belongs to only one department. This relationship is represented by the DepartmentID foreign key in the Doctor table.

Department - Nurse:

Type: One-to-Many

Description: One department can employ multiple nurses, but each nurse belongs to only one department. This is represented by the DepartmentID foreign key in the Nurse table.

Patient - MedicalRecord:

Type: One-to-Many

Description: One patient can have multiple medical records (from multiple visits or consultations), but each medical record pertains to only one patient. This relationship is represented by the PatientID foreign key in the MedicalRecord table.

Doctor - MedicalRecord:

Type: One-to-Many

Description: One doctor can diagnose and create multiple medical records for different patients, but each medical record is associated with only one doctor. This is represented by the DoctorID foreign key in the MedicalRecord table.

MedicalRecord - Prescription:

Type: One-to-Many

Description: One medical record can have multiple prescriptions (if multiple medications are prescribed during a consultation), but each prescription is linked to only one medical record. This relationship is represented by the RecordID foreign key in the Prescription table.

Patient - Billing:

Type: One-to-Many

Description: One patient can have multiple bills (from multiple services or visits), but each bill is associated with only one patient. This is represented by the PatientID foreign key in the Billing table.

Insurance - Billing:

Type: One-to-Many

Description: One insurance policy can cover multiple bills, but each bill is associated with only one insurance policy. This relationship is represented by the InsuranceID foreign key in the Billing table.

Patient - Appointments:

Type: One-to-Many

Description: One patient can have multiple appointments, but each appointment is for one patient. This relationship is represented by the PatientID foreign key in the Appointments table.

Doctor - Appointments:

Type: One-to-Many

Description: One doctor can have multiple appointments with different patients, but each appointment is with one specific doctor. This is represented by the DoctorID foreign key in the Appointments table.

Nurse - Appointments:

Type: One-to-Many (optional)

Description: One nurse can assist in multiple appointments, but each appointment might be assisted by one specific nurse. This is represented by the NurseID foreign key in the Appointments table.

2.2 Database Schema Description

The Healthcare Management System database consists of seven interconnected tables:

2.2.1 Insurance Table

The Insurance table stores information about various insurance policies provided by different insurance providers. It includes the following columns:

Insurance Provider Information

InsuranceID	Provider	PolicyNumber	PolicyType	CoverageStartDate	CoverageEndDate	Premium	Deductible	CoPayment
1111	Aetna	AET123456	Health	2023-01-01	2023-12-31	150.00	500.00	20.00
2378	Cigna	CGN987654	Health	2023-03-15	2023-12-31	180.00	300.00	25.00
3770	UnitedHealthcare	UHC543210	Health	2023-02-10	2023-12-31	200.00	400.00	15.00
4230	Anthem	ANT777777	Health	2023-04-01	2023-12-31	130.00	350.00	10.00
5872	BlueCross BlueShield	BCBS555555	Health	2023-02-25	2023-12-31	160.00	250.00	30.00
6651	Humana	HUM111111	Health	2023-01-15	2023-12-31	190.00	450.00	18.00
7000	Kaiser Permanente	KP888888	Health	2023-03-01	2023-12-31	175.00	320.00	22.00
8433	Aflac	AFL222222	Health	2023-01-10	2023-12-31	140.00	400.00	15.00
9700	MetLife	MET666666	Health	2023-04-15	2023-12-31	170.00	380.00	20.00
1000	Cobra Insurance	CBI444444	Health	2023-02-01	2023-12-31	155.00	300.00	25.00

InsuranceID (Primary Key): Unique identifier for each insurance policy.

Provider: The name of the insurance provider.

PolicyNumber: The policy number associated with the insurance policy.

PolicyType: The type of insurance policy (e.g., Health, Life, etc.).

CoverageStartDate: The start date of the insurance coverage.

CoverageEndDate: The end date of the insurance coverage.

Premium: The premium amount paid for the insurance policy.

Deductible: The deductible amount for the policy.

CoPayment: The copayment amount for medical services.

2.2.2 Department Table

The Department table stores information about different medical departments within the healthcare facility. It includes the following columns:

Inserting values into the Department table

DepartmentID	Name	Location	Type	CostCenterCode	Virtual
1	Pediatrics	Block A	Medical	CC123	N
2	Psychiatry	Block B	Medical	CC456	N
3	Orthopedics	Block C	Medical	CC789	N
4	Emergency Medicine	Block D	Medical	CC246	N
5	Urology	Block E	Medical	CC135	N
6	Neurology	Block F	Medical	CC680	N
7	Family Medicine	Block G	Medical	CC791	N
8	Anesthesiology	Block H	Medical	CC234	N
9	Pathology	Block I	Medical	CC567	N
10	Gynaecology	Block J	Medical	CC908	N

DepartmentID (Primary Key): Unique identifier for each department.

Name: The name of the medical department.

Location: The physical location of the department.

Type: The type of department (e.g., Medical, Surgical, etc.).

CostCenterCode: The cost center code associated with the department.

Virtual: A flag indicating whether the department is virtual (Y/N).

2.2.3 Patient Table

The Patient table stores data related to individual patients availing healthcare services. It includes the following columns:

Inserting values into the Patient table

PatientID	FirstName	LastName	DOB	Gender	ContactNumber	Email	Address	EmergencyContactName	EmergencyContactNumber	PrimaryCarePhysician	MedicalHistory	InsuranceID
1789	John	Doe	1980-01-01	M	1234567890	john.doe@gmail.com	123 Main St, City	Jane Doe	9876543210	Dr. Max Miller	No significant history.	1111
2657	Jane	Doe	1985-02-02	F	2345678901	jane.doe@outlook.com	456 Elm St, Town	John Smith	8765432109	Dr. Max Miller	Allergies to pollen.	2378
3790	Jim	Smith	1990-03-03	M	3456789012	jim.smith@gmail.com	789 Oak St, Village	Sarah Johnson	7654321098	Dr. Mike Davis	None.	3770
4112	Jill	Smith	1995-04-04	F	4567890123	jill.smith@yahoo.com	890 Maple St, Hamlet	Jake Johnson	6543210987	Dr. Mike Davis	Previous surgeries: Appendectomy.	4230
5654	Joe	Johnson	2000-05-05	M	5678901234	joe.johnson@hotmail.com	901 Pine St, City	Emily Johnson	5432109876	Dr. Daisy Davis	Asthma.	5872
6667	Jenny	Johnson	2005-06-06	F	6789012345	jenny.johnson@gmail.com	234 Cedar St, Town	Daniel Johnson	4321098765	Dr. Daisy Davis	None.	6651
7811	Jack	Williams	2010-07-07	M	7890123456	jack.williams@yahoo.com	567 Birch St, Village	Olivia Williams	3210987654	Dr. Daniel Wilson	Allergies: Peanuts.	7000
8000	Jessica	Williams	2015-08-08	F	8901234567	jessica.williams@outlook.com	678 Walnut St, Hamlet	William Williams	2109876543	Dr. Daniel Wilson	None.	8433
9754	James	Brown	2020-09-09	M	9012345678	james.brown@bing.com	789 Oak St, City	Sophia Brown	1098765432	Dr. Winnie Wilson	None.	9700
1056	Julia	Brown	2021-10-10	F	0123456789	julia.brown@yahoo.com	123 Elm St, Town	Ethan Brown	9876543210	Dr. Winnie Wilson	Allergies to penicillin.	1000

PatientID (Primary Key): Unique identifier for each patient.

FirstName: The first name of the patient.

LastName: The last name of the patient.

DOB: The date of birth of the patient.

Gender: The gender of the patient (M/F).

ContactNumber: The contact number of the patient.

Email: The email address of the patient.

Address: The address of the patient.

EmergencyContactName: The name of the patient's emergency contact person.

EmergencyContactNumber: The contact number of the patient's emergency contact person.

PrimaryCarePhysician: The name of the patient's primary care physician.

MedicalHistory: A CLOB field to store the patient's medical history.

InsuranceID: Foreign key referencing the Insurance table, representing the patient's insurance coverage.

2.2.4 Doctor Table

The Doctor table stores data related to healthcare providers (doctors) in the healthcare facility. It includes the following columns:

Inserting values into the Doctor table

DoctorID	FirstName	LastName	Gender	ContactNumber	Email	Specialization	LicenseNumber	YearsOfExperience	DepartmentID
1111	Dr. Max	Miller	M	1234567890	max.miller@example.com	Pediatrics	LIC123	15	1
2546	Dr. Mia	Miller	F	2345678901	mia.miller@example.com	Psychiatry	LIC456	10	2
3000	Dr. Mike	Davis	M	3456789012	mike.davis@example.com	Orthopedics	LIC789	8	3
4789	Dr. Daisy	Davis	F	4567890123	daisy.davis@example.com	Emergency Medicine	LIC246	12	4
5322	Dr. Daniel	Wilson	M	5678901234	daniel.wilson@example.com	Urology	LIC135	7	5
6111	Dr. Winnie	Wilson	F	6789012345	winnie.wilson@example.com	Neurology	LIC680	9	6
7000	Dr. William	Moore	M	7890123456	william.moore@example.com	Family Medicine	LIC791	6	7
8324	Dr. Monica	Moore	F	8901234567	monica.moore@example.com	Anesthesiology	LIC234	11	8
9333	Dr. Martin	Taylor	M	9012345678	martin.taylor@example.com	Pathology	LIC567	14	9
1011	Dr. Tina	Taylor	F	0123456789	tina.taylor@example.com	Gynaecology	LIC908	5	10

DoctorID (Primary Key): Unique identifier for each doctor.

FirstName: The first name of the doctor.

LastName: The last name of the doctor.

Gender: The gender of the doctor (M/F).

ContactNumber: The contact number of the doctor.

Email: The email address of the doctor.

Specialization: The medical specialization of the doctor (e.g., Cardiology, Pediatrics, etc.).

LicenseNumber: The license number of the doctor.

YearsOfExperience: The number of years of experience of the doctor.

DepartmentID: Foreign key referencing the Department table, representing the department to which the doctor belongs.

2.2.5 MedicalRecord Table

The MedicalRecord table stores medical records related to patient visits and consultations. It includes the following columns:

Inserting values into the MedicalRecord table

RecordID	PatientID	DoctorID	Symptoms	Diagnosis	Treatment	Prescription	TreatmentCost	DateOfRecord	FollowUpDate	Notes
1	1789	1111	Fever, Cough	Common Cold	Rest, Fluids	Cough Syrup	25.00	2023-08-06	2023-08-10	Recheck if symptoms persist.
2	2657	2546	Anxiety, Sleeplessness	Generalized Anxiety Disorder	Therapy, Meditation	Anti-anxiety Medication	40.00	2023-09-06	2023-09-15	Follow up on progress.
3	3790	3000	Knee Pain	Sprained Knee	Rest, Ice, Compression	Pain Reliever	30.00	2023-08-07	2023-08-14	Physical therapy recommended.
4	4112	4789	Fatigue, Weight Gain	Hypothyroidism	Hormone Replacement	Levothyroxine	15.00	2023-10-08	2023-10-15	Monitor thyroid levels.
5	5654	5322	Heartburn	Gastric Reflux	Dietary Changes	Antacid	10.00	2023-09-09	2023-09-16	Avoid spicy foods.
6	6667	6111	Depression	Major Depressive Disorder	Therapy, Medication	Antidepressant	35.00	2023-08-10	2023-08-20	Monitor mood changes.
7	7811	7000	High Blood Pressure	Hypertension	Lifestyle Changes	Blood Pressure Medication	20.00	2023-08-11	2023-08-18	Diet and exercise recommendations.
8	8000	8324	Cholesterol Check	Hypercholesterolemia	Dietary Changes	Statins	25.00	2023-08-12	2023-08-19	Follow up on cholesterol levels.
9	9754	9333	Asthma Exacerbation	Asthma	Inhaler, Rest	Albuterol	15.00	2023-08-13	2023-08-20	Monitor lung function.
10	1056	1011	Headache	Tension Headache	Rest, Hydration	Pain Reliever	5.00	2023-08-14	2023-08-21	Stress-related headache.

RecordID (Primary Key): Unique identifier for each medical record.

PatientID: Foreign key referencing the Patient table, representing the patient associated with the medical record.

DoctorID: Foreign key referencing the Doctor table, representing the doctor associated with the medical record.

Symptoms: A brief description of the patient's symptoms.

Diagnosis: The doctor's diagnosis based on the examination.

Treatment: The treatment plan prescribed by the doctor.

Prescription: Details of the medications prescribed to the patient.

TreatmentCost: The cost of the treatment provided.

DateOfRecord: The date of the medical record entry.

FollowUpDate: The date of the follow-up visit (if applicable).

Notes: Additional notes or comments regarding the medical record.

2.2.6 Prescription Table

The Prescription table stores data related to prescriptions issued to patients during medical visits. It includes the following columns:

Inserting values into the Prescription table

PrescriptionID	RecordID	Medication	Dosage	Frequency	StartDate	EndDate	Refills	Pharmacy	Price
1	1	Cough Syrup	2 tsp	Every 6 hours	2023-08-06	2023-08-15	2	PharmaCare	10.00
2	2	Anti-anxiety Medication	1 tablet	Once daily	2023-09-06	2023-09-20	1	MediCure	20.00
3	3	Pain Reliever	1 tablet	As needed	2023-08-07	2023-08-14	3	HealthRx	5.00
4	4	Levothyroxine	1 tablet	Once daily	2023-10-08	2023-10-20	2	PharmaNow	10.00
5	5	Antacid	1 tablet	Twice daily	2023-09-09	2023-09-20	1	MediCure	8.00
6	6	Antidepressant	1 tablet	Once daily	2023-08-10	2023-08-25	2	HealthRx	15.00
7	7	Blood Pressure Medication	1 tablet	Three times daily	2023-08-11	2023-08-22	3	PharmaNow	12.00
8	8	Statins	1 tablet	Once daily	2023-08-12	2023-08-25	1	MediCure	10.00
9	9	Albuterol	2 puffs	Once daily	2023-08-13	2023-08-21	2	HealthRx	10.00
10	10	Pain Reliever	1 tablet	As needed	2023-08-14	2023-08-24	3	PharmaNow	5.00

PrescriptionID (Primary Key): Unique identifier for each prescription.

RecordID: Foreign key referencing the MedicalRecord table, representing the medical record associated with the prescription.

Medication: The name of the medication prescribed.

Dosage: The recommended dosage of the medication.

Frequency: The frequency of medication intake.

StartDate: The start date of the prescription.

EndDate: The end date of the prescription.

Refills: The number of refills allowed for the prescription.

Pharmacy: The pharmacy where the prescription can be filled.

Price: The price of the medication.

2.2.7 Billing Table

The Billing table stores data related to patient billing and payments. It includes the following columns:

Inserting values into the Billing table

BillID	PatientID	Amount	DateOfBilling	PaymentMethod	PaymentDate	InsuranceID	AmountCoveredByInsurance	Status
1	1789	150.00	2023-08-06	Credit Card	2023-08-07	1111	100.00	Paid
2	2657	80.50	2023-09-06	Cash	2023-09-07	2378	50.00	Paid
3	3790	200.75	2023-08-07	Debit Card	2023-08-08	3770	180.00	Paid
4	4112	45.25	2023-10-08	Cash	2023-10-09	4230	0.00	Pending
5	5654	75.90	2023-09-09	Credit Card	2023-09-10	5872	70.00	Paid
6	6667	30.00	2023-08-10	Debit Card	2023-08-11	6651	20.00	Paid
7	7811	120.25	2023-08-11	Cash	2023-08-12	7000	100.00	Paid
8	8000	50.75	2023-08-12	Credit Card	2023-08-13	8433	30.00	Paid
9	9754	15.50	2023-08-13	Cash	2023-08-14	9700	0.00	Pending
10	1056	25.00	2023-08-14	Debit Card	2023-08-15	1000	20.00	Paid

BillID (Primary Key): Unique identifier for each billing record.

PatientID: Foreign key referencing the Patient table, representing the patient associated with the billing record.

Amount: The total amount to be paid for services.

DateOfBilling: The date of the billing record.

PaymentMethod: The method used for payment (e.g., Cash, Credit Card, etc.).

PaymentDate: The date of payment.

InsuranceID: Foreign key referencing the Insurance table, representing the insurance associated with the billing record.

AmountCoveredByInsurance: The amount covered by insurance for the services.

Status: The status of the billing record (e.g., Paid, Pending, etc.).

2.2.5 Nurse Table

The Nurse table stores data related to the nursing staff in the healthcare facility. It includes the following columns:

Inserting values into the Nurse table

NurseID	FirstName	LastName	Gender	ContactNumber	Email	Qualification	LicenseNumber	YearsOfExperience	DepartmentID	Shift
5001	Nancy	Smith	F	1234567891	nancy.smith@example.com	RN	NUR5001	5	1	Day
5002	Robert	Johnson	M	1234567892	robert.johnson@example.com	LPN	NUR5002	3	2	Night
5003	Linda	Williams	F	1234567893	linda.williams@example.com	RN	NUR5003	7	3	Day
5004	Michael	Jones	M	1234567894	michael.jones@example.com	LPN	NUR5004	2	4	Night
5005	Karen	Brown	F	1234567895	karen.brown@example.com	RN	NUR5005	6	5	Day
5006	James	Davis	M	1234567896	james.davis@example.com	LPN	NUR5006	4	6	Night
5007	Patricia	Miller	F	1234567897	patricia.miller@example.com	RN	NUR5007	8	7	Day
5008	John	Wilson	M	1234567898	john.wilson@example.com	LPN	NUR5008	3	8	Night
5009	Jennifer	Moore	F	1234567899	jennifer.moore@example.com	RN	NUR5009	9	9	Day
5010	William	Taylor	M	1234567800	william.taylor@example.com	LPN	NUR5010	5	10	Night

NurseID (Primary Key)*: Unique identifier for each nurse.

FirstName*: The first name of the nurse.

LastName*: The last name of the nurse.

Gender*: The gender of the nurse (M/F).

ContactNumber*: The contact number of the nurse.

Email*: The email address of the nurse.

Qualification*: The educational or professional qualification of the nurse.

LicenseNumber*: The license number of the nurse.

YearsOfExperience*: The number of years of experience of the nurse.

DepartmentID*: Foreign key referencing the Department table, representing the department to which the nurse belongs.

Shift*: The shift timings or period during which the nurse works (e.g., Morning, Evening, Night).

2.2.6 Medication Table

The Medication table stores data related to the medicines available in the healthcare facility. It includes the following columns:

Inserting values into the Medication table

MedicationID	DrugName	Barcode	Manufacturer	DateManufactured	ExpiryDate	Description	DosageInstructions	SideEffects	Price	PrescriptionRequired
2001	Paracetamol	123456789001	PharmaCo	2023-01-01	2025-01-01	Pain and fever reducer.	Take 1 tablet every 4-6 hours.	Nausea, upset stomach.	5.99	N
2002	Ibuprofen	123456789002	MediCare	2023-02-01	2025-02-01	Anti-inflammatory and pain reliever.	Take 1 tablet every 6-8 hours.	Dizziness, rash.	7.49	N
2003	Amoxicillin	123456789003	BioHealth	2023-03-01	2024-03-01	Antibiotic.	Take 1 capsule twice a day.	Diarrhea, stomach pain.	12.99	Y
2004	Lisinopril	123456789004	PharmaCo	2023-04-01	2025-04-01	Treats high blood pressure.	Take 1 tablet daily.	Dizziness, headache.	9.99	Y
2005	Metformin	123456789005	MediCare	2023-05-01	2025-05-01	Treats type 2 diabetes.	Take 1 tablet twice a day.	Nausea, upset stomach.	8.49	Y
2006	Atorvastatin	123456789006	BioHealth	2023-06-01	2025-06-01	Lowers cholesterol.	Take 1 tablet daily.	Muscle pain, liver problems.	11.99	Y
2007	Albuterol	123456789007	PharmaCo	2023-07-01	2024-07-01	Treats asthma and COPD.	Use as needed.	Shakiness, nervousness.	15.99	Y
2008	Omeprazole	123456789008	MediCare	2023-08-01	2025-08-01	Treats acid reflux.	Take 1 capsule daily.	Headache, stomach pain.	10.49	N
2009	Losartan	123456789009	BioHealth	2023-09-01	2025-09-01	Treats high blood pressure.	Take 1 tablet daily.	Dizziness, fatigue.	9.49	Y
2010	Hydrochlorothiazide	123456789010	PharmaCo	2023-10-01	2025-10-01	Diuretic.	Take 1 tablet daily.	Thirst, dry mouth.	7.99	Y

MedicationID (Primary Key)*: Unique identifier for each medication.

DrugName*: The name of the medication or drug.

Barcode*: A unique barcode associated with the medication.

Manufacturer*: The company or entity that manufactures the medication.

DateManufactured*: The date when the medication was manufactured.

ExpiryDate*: The date when the medication will expire.

Description*: A detailed description of the medication.

DosageInstructions*: Instructions on how the medication should be taken.

SideEffects*: Information about any potential side effects of the medication.

Price*: The price of the medication.

PrescriptionRequired*: Indicates whether a prescription is required to obtain the medication (Y/N).

2.2.7 Appointments Table*

The Appointments table stores data related to the appointments scheduled in the healthcare facility. It includes the following columns:

Inserting values into the Appointments table

AppointmentID	PatientID	DoctorID	NurseID	DepartmentID	AppointmentDate	StartTime	EndTime	Purpose	Status	Notes
1001	101	201	5001	1	2023-08-10	09:00:00	09:30:00	Routine Checkup	Scheduled	Patient has a history of hypertension.
1002	102	202	5002	2	2023-08-11	10:00:00	10:45:00	Follow-up after surgery	Scheduled	Ensure stitches are healing properly.
1003	103	203		3	2023-08-12	11:00:00	11:30:00	Dental Cleaning	Scheduled	Patient reported mild toothache.
1004	104	204	5004	4	2023-08-13	12:00:00	12:45:00	Eye Examination	Scheduled	Patient needs new prescription for glasses.
1005	105	205	5005	5	2023-08-14	13:00:00	13:30:00	Cardiology Consultation	Scheduled	Patient has irregular heartbeat.
1006	106	206		6	2023-08-15	14:00:00	14:45:00	Orthopedic Consultation	Scheduled	Patient has knee pain.
1007	107	207	5007	7	2023-08-16	15:00:00	15:30:00	Neurology Checkup	Scheduled	Patient has frequent headaches.
1008	108	208		8	2023-08-17	16:00:00	16:45:00	Dermatology Consultation	Scheduled	Patient has skin rash.
1009	109	209	5009	9	2023-08-18	17:00:00	17:30:00	ENT Checkup	Scheduled	Patient has ear infection.
1010	110	210		10	2023-08-19	18:00:00	18:45:00	Gynecology Consultation	Scheduled	Routine annual checkup.

AppointmentID (Primary Key)*: Unique identifier for each appointment.

PatientID*: Foreign key referencing the Patient table, representing the patient for whom the appointment is scheduled.

DoctorID*: Foreign key referencing the Doctor table, representing the doctor with whom the appointment is scheduled.

NurseID*: Optional foreign key referencing the Nurse table, representing the nurse assigned for the appointment, if any.

DepartmentID*: Foreign key referencing the Department table, representing the department where the appointment will take place.

AppointmentDate*: The date of the appointment.

StartTime*: The starting time of the appointment.

EndTime*: The ending time of the appointment.

Purpose*: The reason or purpose of the appointment. Status*: The current status of the appointment (e.g., 'Scheduled', 'Completed', 'Cancelled').

3. Table Creation Queries

The following are the CREATE TABLE queries for each table in the Healthcare Management System database:

```

CREATE TABLE Appointments (
    AppointmentID NUMBER(10) PRIMARY KEY,
    PatientID NUMBER(10),
    DoctorID NUMBER(10),
    NurseID NUMBER(10) NULL, -- Optional: In case a nurse is also assigned for the
appointment
    DepartmentID NUMBER(10),
    AppointmentDate DATE,
    StartTime TIMESTAMP,
    EndTime TIMESTAMP,
    Purpose VARCHAR2(200),
    Status VARCHAR2(50) DEFAULT 'Scheduled', -- e.g., 'Scheduled', 'Completed', 'Cancelled'
    Notes CLOB,
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID),
    FOREIGN KEY (NurseID) REFERENCES Nurse(NurseID),
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
);

```

Appointments Table Explanation:

AppointmentID (Primary Key, NUMBER(10)): This unique identifier ensures that each appointment has a distinct reference. Using a NUMBER data type allows for a large range of unique IDs, accommodating a vast number of appointments.

PatientID, DoctorID, NurseID, DepartmentID (NUMBER(10)): The NUMBER data type is used for these IDs to maintain consistency across tables and to allow for a large range of unique IDs. These foreign keys establish relationships with other tables, ensuring that the referenced entities (like patients or doctors) exist.

AppointmentDate (DATE): The DATE data type is chosen to store the specific day of the appointment, ensuring clarity and precision.

StartTime, EndTime (TIMESTAMP): TIMESTAMP is used to store both date and time, providing precision up to fractions of a second. This ensures accurate scheduling.

Purpose (VARCHAR2(200)): VARCHAR2 is used to store variable-length character strings, allowing for a detailed description of the appointment's purpose.

Status (VARCHAR2(50), DEFAULT 'Scheduled'): VARCHAR2 allows for variable-length status descriptions. The default constraint ensures that every new appointment is initially set to 'Scheduled'.

Notes (CLOB): CLOB (Character Large Object) is used to store large text values, allowing for extensive notes or comments about the appointment.

```
CREATE TABLE Nurse (  
    NurseID NUMBER(10) PRIMARY KEY,  
    FirstName VARCHAR2(50),  
    LastName VARCHAR2(50),  
    Gender CHAR(1),  
    ContactNumber VARCHAR2(15),  
    Email VARCHAR2(50),  
    Qualification VARCHAR2(50),  
    LicenseNumber VARCHAR2(50),  
    YearsOfExperience NUMBER(3),  
    DepartmentID NUMBER(10),  
    Shift VARCHAR2(20),  
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)  
);
```

Nurse Table Explanation:

NurseID (Primary Key, NUMBER(10)): This unique identifier ensures each nurse has a distinct reference. Using a NUMBER data type accommodates a vast number of unique nurse entries.

FirstName, LastName, Email, Qualification, Shift (VARCHAR2): VARCHAR2 is chosen for these fields to store variable-length character strings, providing flexibility in storing names, emails, and other textual data.

Gender (CHAR(1)): CHAR is used for fixed-length character strings. Since gender is represented by a single character (M/F), CHAR(1) is appropriate.

ContactNumber, LicenseNumber (VARCHAR2(15), VARCHAR2(50)): VARCHAR2 provides flexibility in storing variable-length phone numbers and license numbers.

YearsOfExperience (NUMBER(3)): The NUMBER data type allows for a range from 0 to 999, accommodating a wide range of experience years.

DepartmentID (NUMBER(10)): This foreign key establishes a relationship with the Department table, ensuring the referenced department exists.

```
CREATE TABLE Insurance (  
    InsuranceID NUMBER(10) PRIMARY KEY,
```

```
Provider VARCHAR2(50),
PolicyNumber VARCHAR2(50),
PolicyType VARCHAR2(50),
CoverageStartDate DATE,
CoverageEndDate DATE,
Premium NUMBER(10,2),
Deductible NUMBER(10,2),
CoPayment NUMBER(10,2)
);
```

Insurance Table Explanation:

InsuranceID (Primary Key, NUMBER(10)): This unique identifier ensures each insurance entry is distinct. The NUMBER data type allows for a vast range of unique insurance entries.

Provider, PolicyNumber, PolicyType (VARCHAR2): VARCHAR2 is chosen for its flexibility in storing variable-length character strings, suitable for various insurance-related textual data.

CoverageStartDate, CoverageEndDate (DATE): The DATE data type ensures precision in storing the specific start and end dates of insurance coverage.

Premium, Deductible, CoPayment (NUMBER(10,2)): The NUMBER data type with two decimal places is used to accurately represent monetary values, ensuring precise billing and financial calculations.

```
CREATE TABLE Department (
  DepartmentID NUMBER(10) PRIMARY KEY,
  Name VARCHAR2(50),
  Location VARCHAR2(50),
  Type VARCHAR2(50),
  CostCenterCode VARCHAR2(50),
  Virtual CHAR(1) CHECK (Virtual IN ('Y', 'N'))
);
```

Department Table Explanation:

DepartmentID (Primary Key, NUMBER(10)): This unique identifier ensures each department has a distinct reference. Using a NUMBER data type allows for a large range of unique department IDs.

Name, Location, Type, CostCenterCode (VARCHAR2): VARCHAR2 is chosen for these fields to store variable-length character strings, providing flexibility in storing department-related textual data.

Virtual (CHAR(1), CHECK (Virtual IN ('Y', 'N'))): CHAR is used for fixed-length character strings. The CHECK constraint ensures that only 'Y' or 'N' values are entered, indicating whether the department offers virtual consultations.

```
CREATE TABLE Patient (  
    PatientID NUMBER(10) PRIMARY KEY,  
    FirstName VARCHAR2(50),  
    LastName VARCHAR2(50),  
    DOB DATE,  
    Gender CHAR(1),  
    ContactNumber VARCHAR2(15),  
    Email VARCHAR2(50),  
    Address VARCHAR2(100),  
    EmergencyContactName VARCHAR2(50),  
    EmergencyContactNumber VARCHAR2(15),  
    PrimaryCarePhysician VARCHAR2(50),  
    MedicalHistory CLOB,  
    InsuranceID NUMBER(10),  
    FOREIGN KEY (InsuranceID) REFERENCES Insurance(InsuranceID)  
);
```

Patient Table Explanation:

PatientID (Primary Key, NUMBER(10)): This unique identifier ensures each patient has a distinct reference. Using a NUMBER data type allows for a large range of unique patient IDs.

FirstName, LastName, Email, Address, EmergencyContactName, PrimaryCarePhysician (VARCHAR2): VARCHAR2 is chosen for these fields to store variable-length character strings, providing flexibility in storing patient-related textual data.

DOB (DATE): The DATE data type ensures precision in storing the specific birth date of patients.

Gender (CHAR(1)): CHAR is used for fixed-length character strings. Since gender is represented by a single character (M/F), CHAR(1) is appropriate.

ContactNumber, EmergencyContactNumber (VARCHAR2(15)): VARCHAR2 provides flexibility in storing variable-length phone numbers.

MedicalHistory (CLOB): CLOB (Character Large Object) is used to store large text values, allowing for a detailed medical history.

InsuranceID (NUMBER(10)): This foreign key establishes a relationship with the Insurance table, ensuring the referenced insurance policy exists.

```
CREATE TABLE Doctor (  
    DoctorID NUMBER(10) PRIMARY KEY,  
    FirstName VARCHAR2(50),  
    LastName VARCHAR2(50),  
    Gender CHAR(1),  
    ContactNumber VARCHAR2(15),  
    Email VARCHAR2(50) NOT NULL,  
    Specialization VARCHAR2(50),  
    LicenseNumber VARCHAR2(50),  
    YearsOfExperience NUMBER(3),  
    DepartmentID NUMBER(10),  
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)  
);
```

Doctor Table Explanation:

DoctorID (Primary Key, NUMBER(10)): This unique identifier ensures each doctor has a distinct reference. Using a NUMBER data type allows for a large range of unique doctor IDs.

FirstName, LastName, Email, Specialization (VARCHAR2): VARCHAR2 is chosen for these fields to store variable-length character strings, providing flexibility in storing doctor-related textual data.

Gender (CHAR(1)): CHAR is used for fixed-length character strings. Since gender is represented by a single character (M/F), CHAR(1) is appropriate.

ContactNumber, LicenseNumber (VARCHAR2): VARCHAR2 provides flexibility in storing variable-length phone numbers and license numbers.

YearsOfExperience (NUMBER(3)): The NUMBER data type allows for a range from 0 to 999, accommodating a wide range of experience years.

DepartmentID (NUMBER(10)): This foreign key establishes a relationship with the Department table, ensuring the referenced department exists.

```
CREATE TABLE MedicalRecord (  
    RecordID NUMBER(10) PRIMARY KEY,  
    PatientID NUMBER(10),  
    DoctorID NUMBER(10),
```

```
Symptoms VARCHAR2(100),
Diagnosis VARCHAR2(100),
Treatment VARCHAR2(100),
PrescriptionID NUMBER(10),
TreatmentCost NUMBER(10, 2),
DateOfRecord DATE,
FollowUpDate DATE,
Notes CLOB,
FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)
);
```

MedicalRecord Table Explanation:

RecordID (Primary Key, NUMBER(10)): This unique identifier ensures each medical record has a distinct reference. Using a NUMBER data type allows for a large range of unique record IDs.

PatientID, DoctorID (NUMBER(10)): These foreign keys establish relationships with the Patient and Doctor tables, ensuring the referenced entities exist.

Symptoms, Diagnosis, Treatment (VARCHAR2): VARCHAR2 is chosen for these fields to store variable-length character strings, providing flexibility in storing medical record-related textual data.

PrescriptionID (NUMBER(10)): This foreign key establishes a relationship with the Prescription table, ensuring the referenced prescription exists.

TreatmentCost (NUMBER(10, 2)): The NUMBER data type with two decimal places is used to accurately represent monetary values related to treatment.

DateOfRecord, FollowUpDate (DATE): The DATE data type ensures precision in storing specific dates related to the medical record.

Notes (CLOB): CLOB (Character Large Object) is used to store large text values, allowing for extensive notes about the medical record.

```
CREATE TABLE Medication (
  MedicationID NUMBER(10) PRIMARY KEY,
  DrugName VARCHAR2(100) NOT NULL,
  Barcode VARCHAR2(50) UNIQUE NOT NULL,
  Manufacturer VARCHAR2(100),
  DateManufactured DATE,
  ExpiryDate DATE,
  Description CLOB,
```

```
DosageInstructions VARCHAR2(200),
SideEffects CLOB,
Price NUMBER(10, 2),
PrescriptionRequired CHAR(1) CHECK (PrescriptionRequired IN ('Y', 'N'))
);
```

Medication Table Explanation:

MedicationID (Primary Key, NUMBER(10)): This unique identifier ensures each medication entry is distinct. Using a NUMBER data type allows for a large range of unique medication entries.

DrugName, Barcode, Manufacturer, DosageInstructions (VARCHAR2): VARCHAR2 is chosen for these fields to store variable-length character strings, providing flexibility in storing medication-related textual data.

DateManufactured, ExpiryDate (DATE): The DATE data type ensures precision in storing specific manufacturing and expiry dates of medications.

Description, SideEffects (CLOB): CLOB is used to store large text values, allowing for detailed descriptions and side effects of the medication.

Price (NUMBER(10, 2)): The NUMBER data type with two decimal places is used to accurately represent the price of the medication.

PrescriptionRequired (CHAR(1), CHECK (PrescriptionRequired IN ('Y', 'N'))): CHAR is used for fixed-length character strings. The CHECK constraint ensures that only 'Y' or 'N' values are entered, indicating whether a prescription is required.

```
CREATE TABLE Prescription (
  PrescriptionID NUMBER(10) PRIMARY KEY,
  RecordID NUMBER(10),
  MedicationID NUMBER(10),
  Dosage VARCHAR2(50),
  Frequency VARCHAR2(50),
  StartDate DATE,
  EndDate DATE,
  Refills NUMBER(3),
  Pharmacy VARCHAR2(50),
  Price NUMBER(10, 2),
  FOREIGN KEY (RecordID) REFERENCES MedicalRecord(RecordID),
  FOREIGN KEY (MedicationID) REFERENCES Medication(MedicationID)
);
```

Prescription Table Explanation:

PrescriptionID (Primary Key, NUMBER(10)): This unique identifier ensures each prescription has a distinct reference. Using a NUMBER data type allows for a large range of unique prescription IDs.

RecordID, MedicationID (NUMBER(10)): These foreign keys establish relationships with the MedicalRecord and Medication tables, ensuring the referenced entities exist.

Dosage, Frequency, Pharmacy (VARCHAR2): VARCHAR2 is chosen for these fields to store variable-length character strings, providing flexibility in storing prescription-related textual data.

StartDate, EndDate (DATE): The DATE data type ensures precision in storing specific start and end dates of the prescription.

Refills (NUMBER(3)): The NUMBER data type allows for a range from 0 to 999, accommodating a wide range of refill counts.

Price (NUMBER(10, 2)): The NUMBER data type with two decimal places is used to accurately represent the price of the prescription.

```
CREATE TABLE Billing (  
    BillID NUMBER(10) PRIMARY KEY,  
    PatientID NUMBER(10),  
    Amount NUMBER(10, 2),  
    DateOfBilling DATE,  
    PaymentMethod VARCHAR2(50),  
    PaymentDate DATE,  
    InsuranceID NUMBER(10),  
    AmountCoveredByInsurance NUMBER(10, 2),  
    Status VARCHAR2(10),  
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),  
    FOREIGN KEY (InsuranceID) REFERENCES Insurance(InsuranceID)  
);
```

Billing Table Explanation:

BillID (Primary Key, NUMBER(10)): This unique identifier ensures each bill has a distinct reference. Using a NUMBER data type allows for a large range of unique bill IDs.

PatientID, InsuranceID (NUMBER(10)): These foreign keys establish relationships with the Patient and Insurance tables, ensuring the referenced entities exist.

Amount, AmountCoveredByInsurance (NUMBER(10, 2)): The NUMBER data type with two decimal places is used to accurately represent monetary values related to billing.

DateOfBilling, PaymentDate (DATE): The DATE data type ensures precision in storing specific billing and payment dates.

PaymentMethod, Status (VARCHAR2): VARCHAR2 is chosen for these fields to store variable-length character strings, providing flexibility in storing billing-related textual data.

Example Queries

Retrieve Patients and Doctors with Their Latest Medical Record:

```
SELECT
    p.PatientID,
    p.FirstName || ' ' || p.LastName AS PatientName,
    d.FirstName || ' ' || d.LastName AS DoctorName,
    mr.Symptoms,
    mr.Diagnosis,
    mr.Treatment
FROM Patient p
JOIN (
    SELECT
        PatientID,
        MAX(DateOfRecord) AS LatestRecordDate
    FROM MedicalRecord
    GROUP BY PatientID
) latest_mr ON p.PatientID = latest_mr.PatientID
JOIN MedicalRecord mr ON p.PatientID = mr.PatientID AND latest_mr.LatestRecordDate =
mr.DateOfRecord
JOIN Doctor d ON mr.DoctorID = d.DoctorID;
```

The screenshot shows a SQL IDE window titled 'healthproject.sql'. The query editor contains the following SQL code:

```
SELECT
    p.PatientID,
    p.FirstName || ' ' || p.LastName AS PatientName,
    d.FirstName || ' ' || d.LastName AS DoctorName,
    mr.Symptoms,
    mr.Diagnosis,
    mr.Treatment
FROM Patient p
JOIN (
    SELECT
        PatientID,
        MAX(DateOfRecord) AS LatestRecordDate
    FROM MedicalRecord
    GROUP BY PatientID
) latest_mr ON p.PatientID = latest_mr.PatientID
JOIN MedicalRecord mr ON p.PatientID = mr.PatientID AND latest_mr.LatestRecordDate = mr.DateOfRecord
JOIN Doctor d ON mr.DoctorID = d.DoctorID;
```

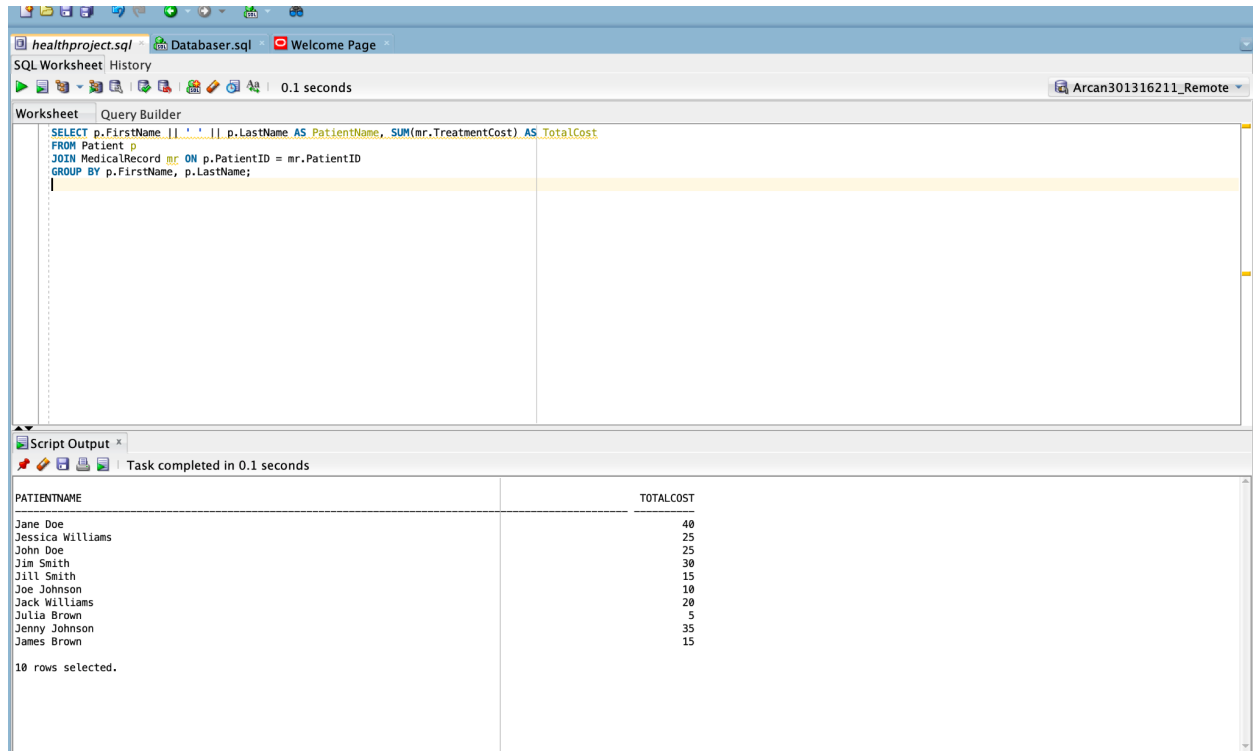
The Script Output pane shows the results of the query, indicating that 10 rows were selected. The results are displayed in a table with two columns: PATIENTID and DOCTORNAME.

PATIENTID	PATIENTNAME	DOCTORNAME
1789	John Doe	Dr. Max Miller
2657	Jane Doe	Dr. Mia Miller
3790	Jim Smith	Dr. Mike Davis
4112	Jill Smith	Dr. Daisy Davis
5654	Joe Johnson	Dr. Daniel Wilson
6667	Jenny Johnson	Dr. Winnie Wilson
7811	Jack Williams	Dr. William Moore
8000	Jessica Williams	Dr. Monica Moore
9754	James Brown	Dr. Martin Taylor
1056	Julia Brown	Dr. Tina Taylor

10 rows selected.

Calculate Total Treatment Cost for Each Patient:

```
SELECT p.FirstName || ' ' || p.LastName AS PatientName, SUM(mr.TreatmentCost) AS
TotalCost
FROM Patient p
JOIN MedicalRecord mr ON p.PatientID = mr.PatientID
GROUP BY p.FirstName, p.LastName;
```



The screenshot shows a SQL Worksheet interface with a query editor and a results pane. The query editor contains the following SQL code:

```
SELECT p.FirstName || ' ' || p.LastName AS PatientName, SUM(mr.TreatmentCost) AS TotalCost
FROM Patient p
JOIN MedicalRecord mr ON p.PatientID = mr.PatientID
GROUP BY p.FirstName, p.LastName;
```

The results pane shows the output of the query, which is a table with two columns: PATIENTNAME and TOTALCOST. The table contains 10 rows of data, and the status bar indicates that 10 rows were selected.

PATIENTNAME	TOTALCOST
Jane Doe	40
Jessica Williams	25
John Doe	25
Jim Smith	30
Jill Smith	15
Joe Johnson	10
Jack Williams	20
Julia Brown	5
Jenny Johnson	35
James Brown	15

10 rows selected.

List Patients with No Medical Records:

```
SELECT p.FirstName || ' ' || p.LastName AS PatientName
FROM Patient p
LEFT JOIN MedicalRecord mr ON p.PatientID = mr.PatientID
WHERE mr.RecordID IS NULL;
```

Conclusion

