

*A Comprehensive Approach to Design,  
Implementation, and Insight*

# Hospital Database & Visualization Dashboard

By

Prajna Ganji, Shiva, Neetika Upadhyay and  
Maruta Zalane



# Project Overview



## Project Objective

- To design a normalized hospital database system capable of storing and managing healthcare data effectively.



## Scope

- Includes database schema design, synthetic data generation, SQL-based data analysis, and performance insights.



## Key Components

- Entity-Relationship Diagram (ERD)
- Table creation with constraints and relationships
- Data population using Python Faker
- SQL queries for operational analysis



## Use Case

- Designed to support hospital administration, improve decision-making, and optimize healthcare delivery.



## Project Deliverable

- A functional and efficient hospital database and Power BI Dashboard providing insights into hospital performance

# Goals & Objectives

## Goal 1: Design a normalized and relational hospital database

Create an ERD that defines core hospital entities and their relationships

Implement tables with appropriate data types, constraints, and relationships



## Goal 2: Simulate real-world hospital operations through synthetic data

Generate realistic synthetic data using Chat GPT(Python Faker Library)

Write and execute SQL queries to extract actionable insights



## Goal 3: Enable meaningful data-driven insights for hospital management

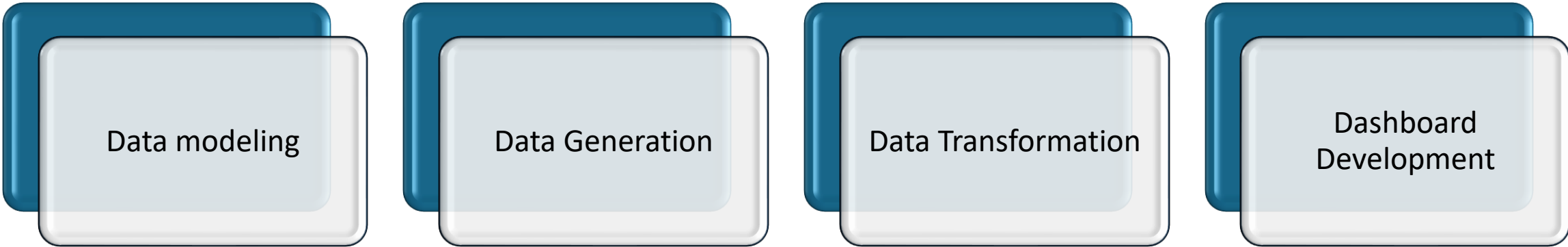
Designing the dashboard related to appointments, departments, Patients etc.

Create visuals and performance metrics using Power BI

# Tools & Technologies

Database	MS SQL Server
Data Generation	ChatGPT-4 that used Python, Faker Library
Analysis & Queries	Structured Query Language (SQL)
Visualization	Power BI
ERD tool	Mermaid

# Methodology



Data modeling

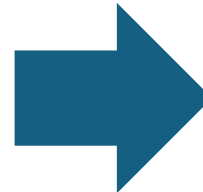
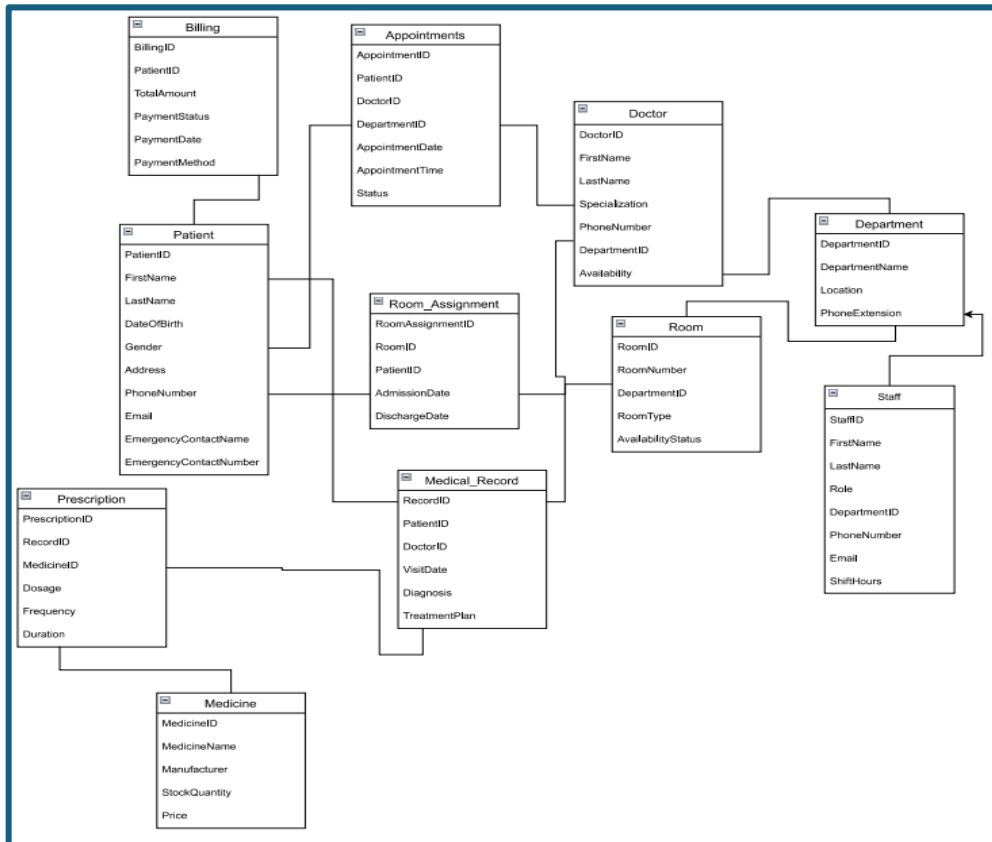
Data Generation

Data Transformation

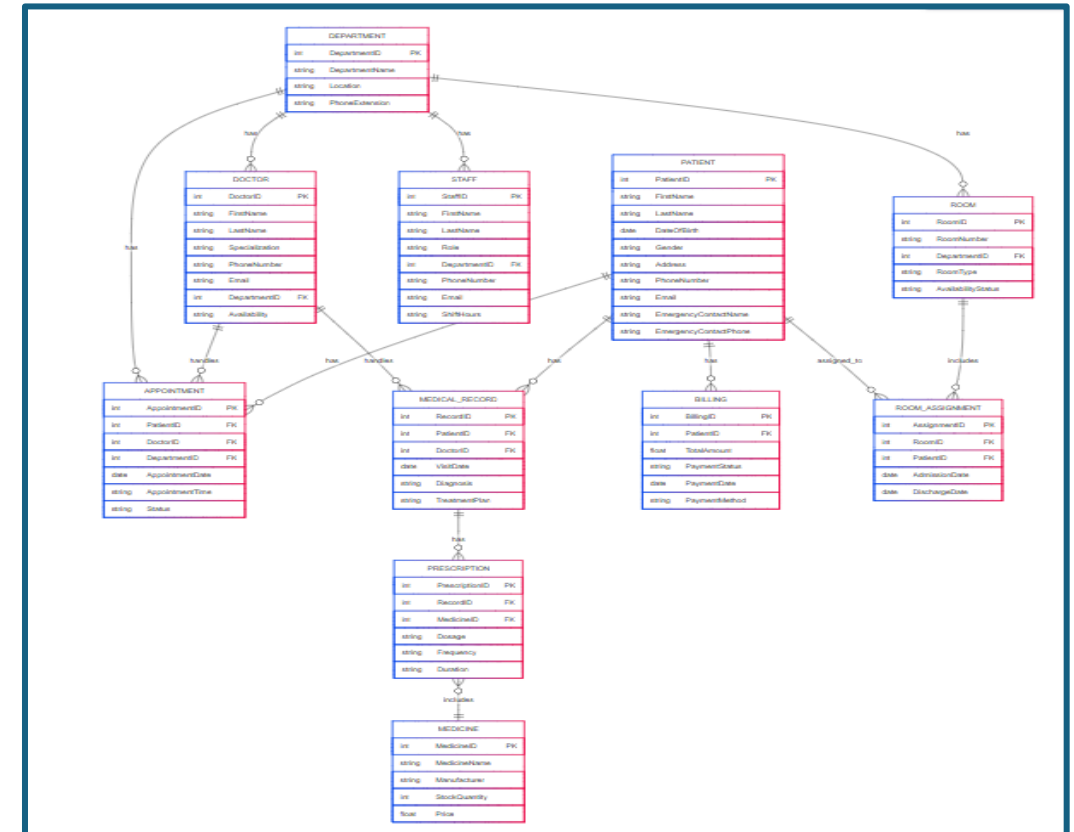
Dashboard  
Development

# Data Modeling (ER Diagram)

## Conceptual Diagram



## Physical Diagram



# Synthetic Data Generation

## Approach Using ChatGPT - 4

- Used GPT-4 to generate synthetic data for all tables in the hospital database.
- Designed structured prompts to simulate realistic records for Patients, Doctors, Appointments, etc.
- Initially received random, disconnected data without consistency across related entities.

## Prompt Engineering Process

- Refined and expanded prompts:
  - Started with 12 rules, increased to 20+ rules for better data control
  - Added constraints and formatting rules (e.g., valid dates, department-doctor mapping)
- Repeated prompt iterations across multiple tables to preserve referential integrity.

## Iterative Improvements

- Ran multiple iterations, learning from output mismatches.
- Gradually achieved datasets with valid primary-foreign key relationships (e.g., matching PatientID in Appointments, valid Room assignments).

## Data Validation

### Referential Integrity Check

- Ensured all foreign key relationships are valid.

### Data Type Validation

- Confirmed all columns use the correct types

### Logical Validations:

- Discharge date  $\geq$  Admission date
- Billing amounts  $\geq 0$

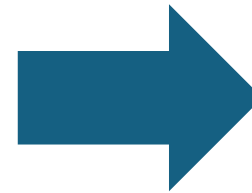
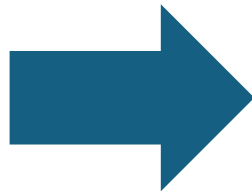
### Uniqueness Checks

- Verified no duplicates in primary keys

### Null Value Check

- Checked mandatory fields are not null.

# Data Transformation – SQL Server to Power BI



## SQL Server

- Created and populated normalized hospital tables with synthetic data.
- Created SQL queries & views using Joins to ensure clean, relational data for business questions.

## Power BI

- Connected using Power BI SQL Server connector
- Data imported with minimal transformation in Power BI

## Dashboard

- DAX measures added for insights
- Created visuals to explore hospital insights



# Dashboard Development

## Hospital Performance Dashboard

**Target User-** Hospital Executives (CEO, Leadership Team, Decision Makers)

**Purpose-** High-level KPIs, strategic insights, overall performance snapshot.

### Patient & Appointment Overview Dashboard

**Target User**

- Hospital Administrators

**Purpose**

- Patient volume, appointment trends, cancellations.

### Doctor Availability & Workload Dashboard

**Target User**

- Medical Directors & Doctors

**Purpose**

- Scheduling, workload distribution, availability tracking.

### Financial Overview Dashboard

**Target User**

- Finance Department

**Purpose**

- Revenue, expenses, top-earning department, billing efficiency.

### Prescriptions Overview

**Target User**

- Pharmacy Department

**Purpose**

- Medication trends, prescribing habits, potential overuse alerts

### Room Utilization Overview Dashboard

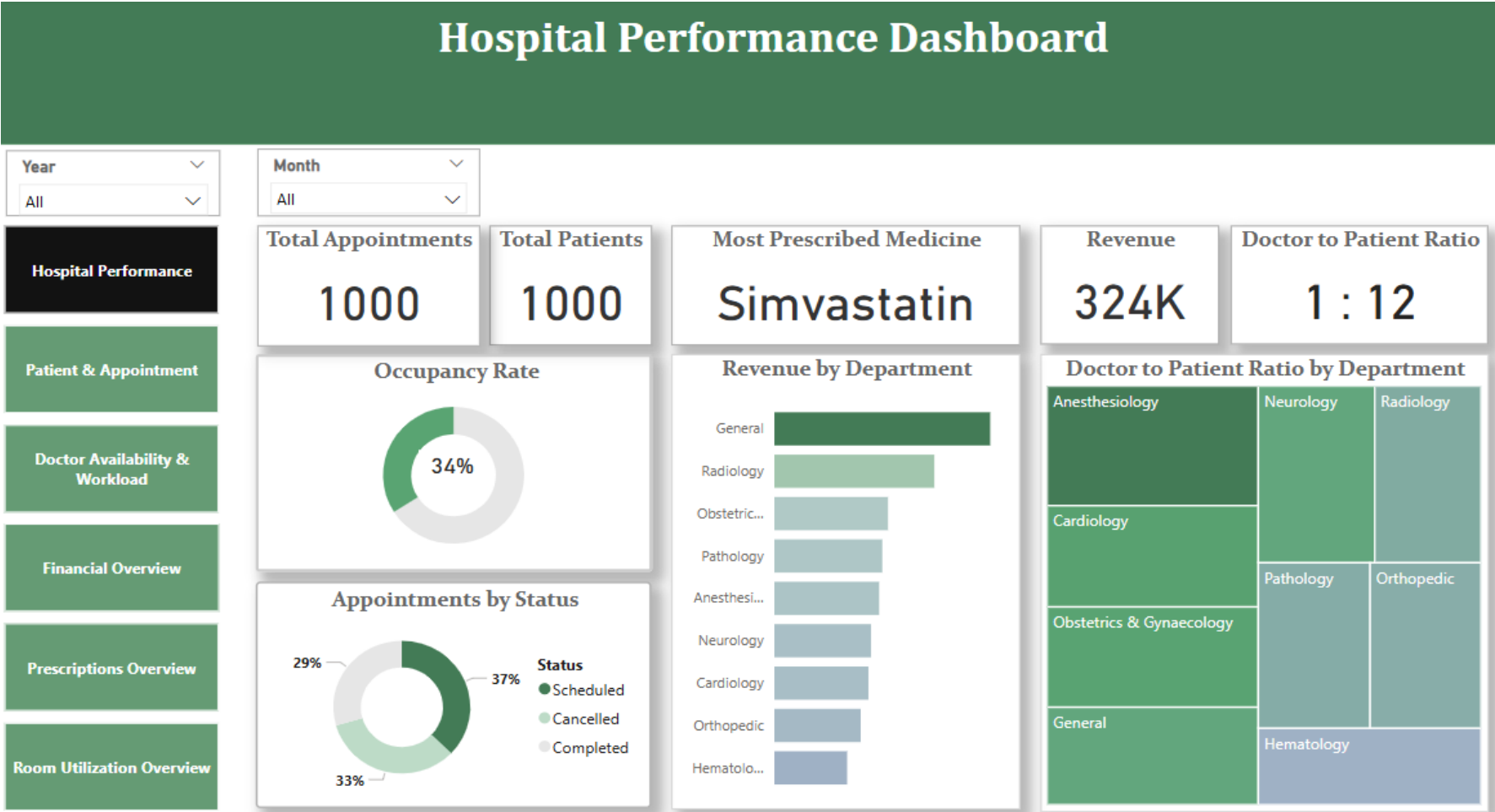
**Target User**

- Facility Management

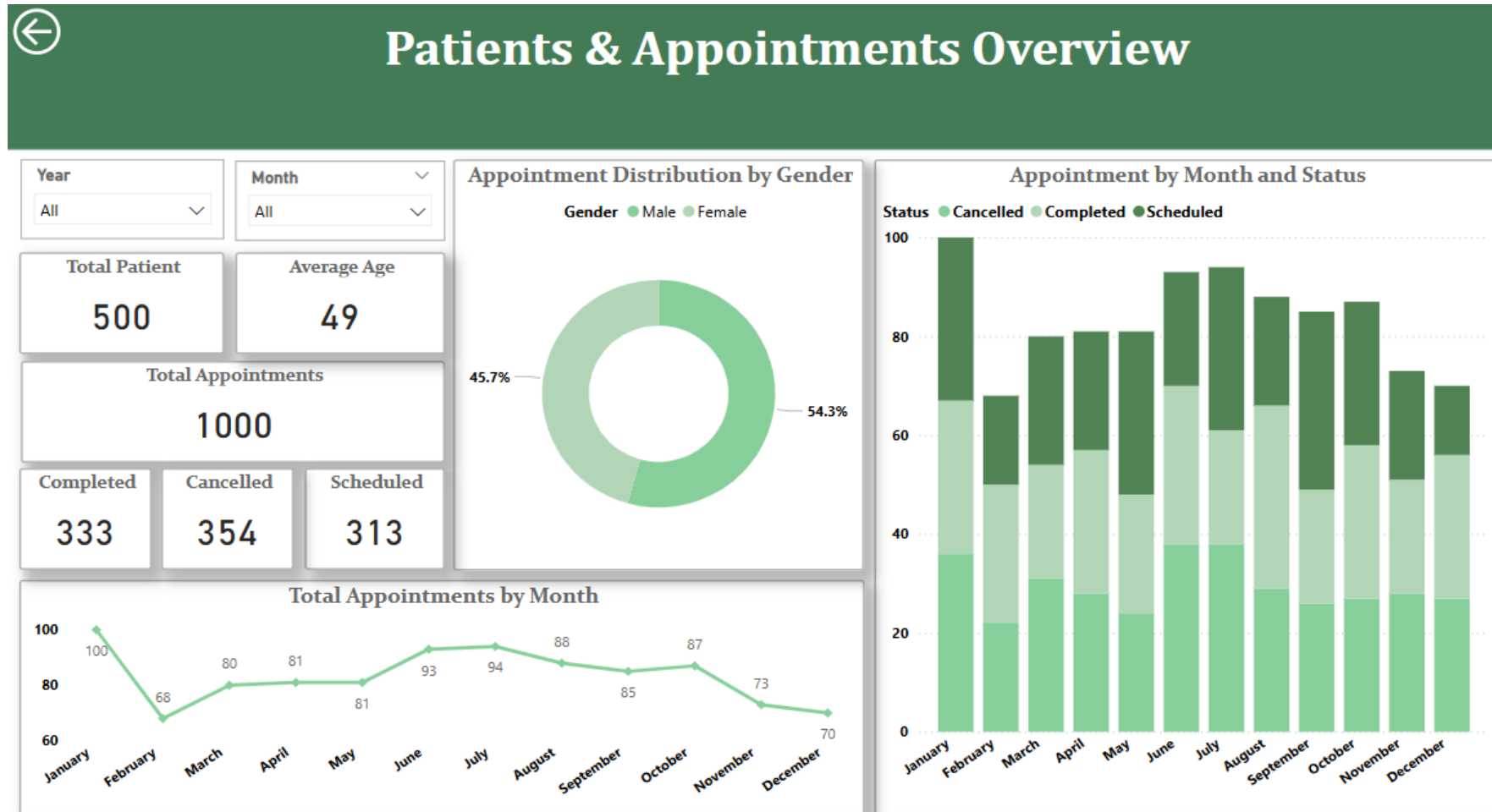
**Purpose**

- Room occupancy, usage efficiency, turnover rates.

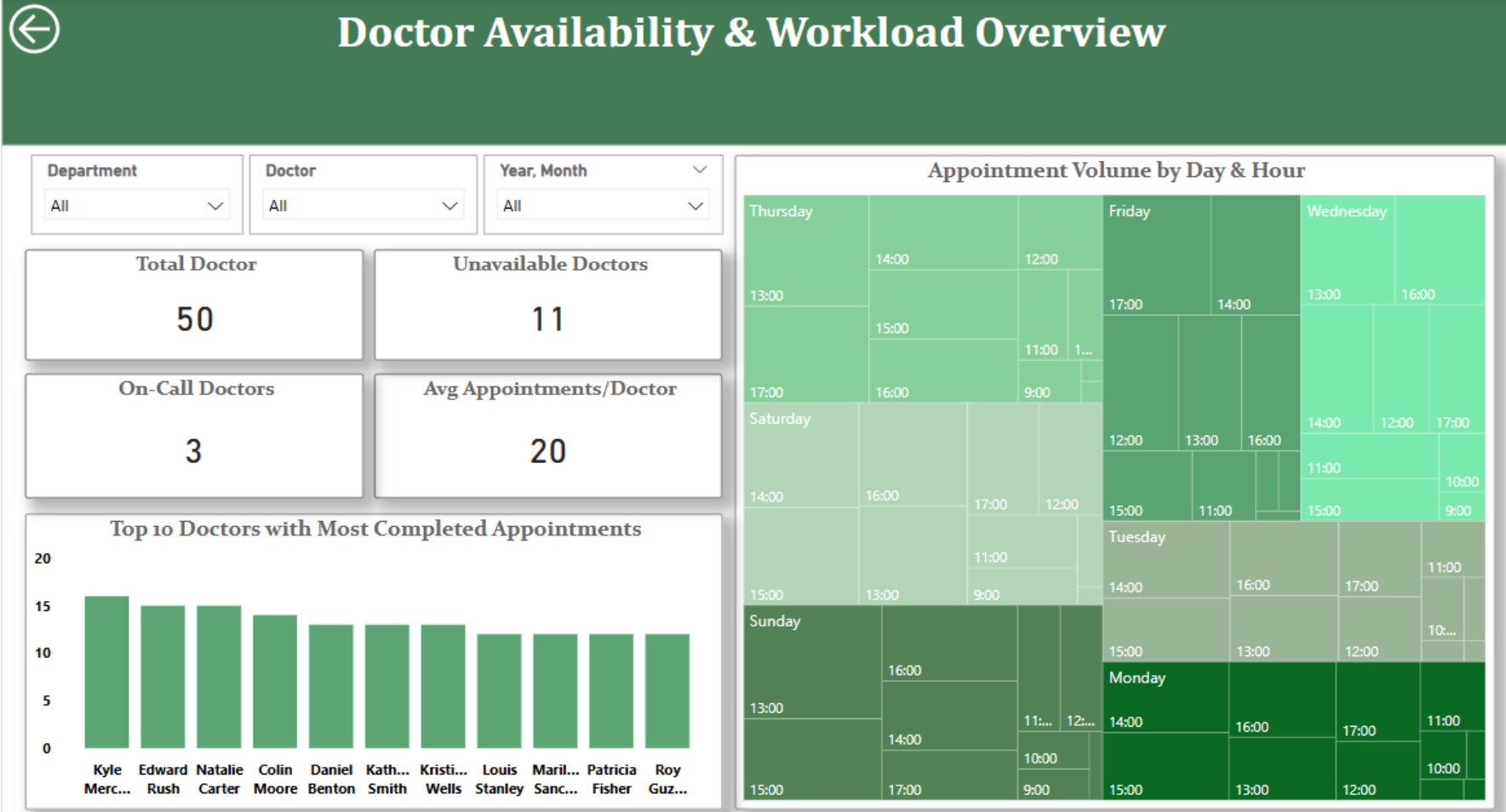
# Hospital Performance Overview Dashboard



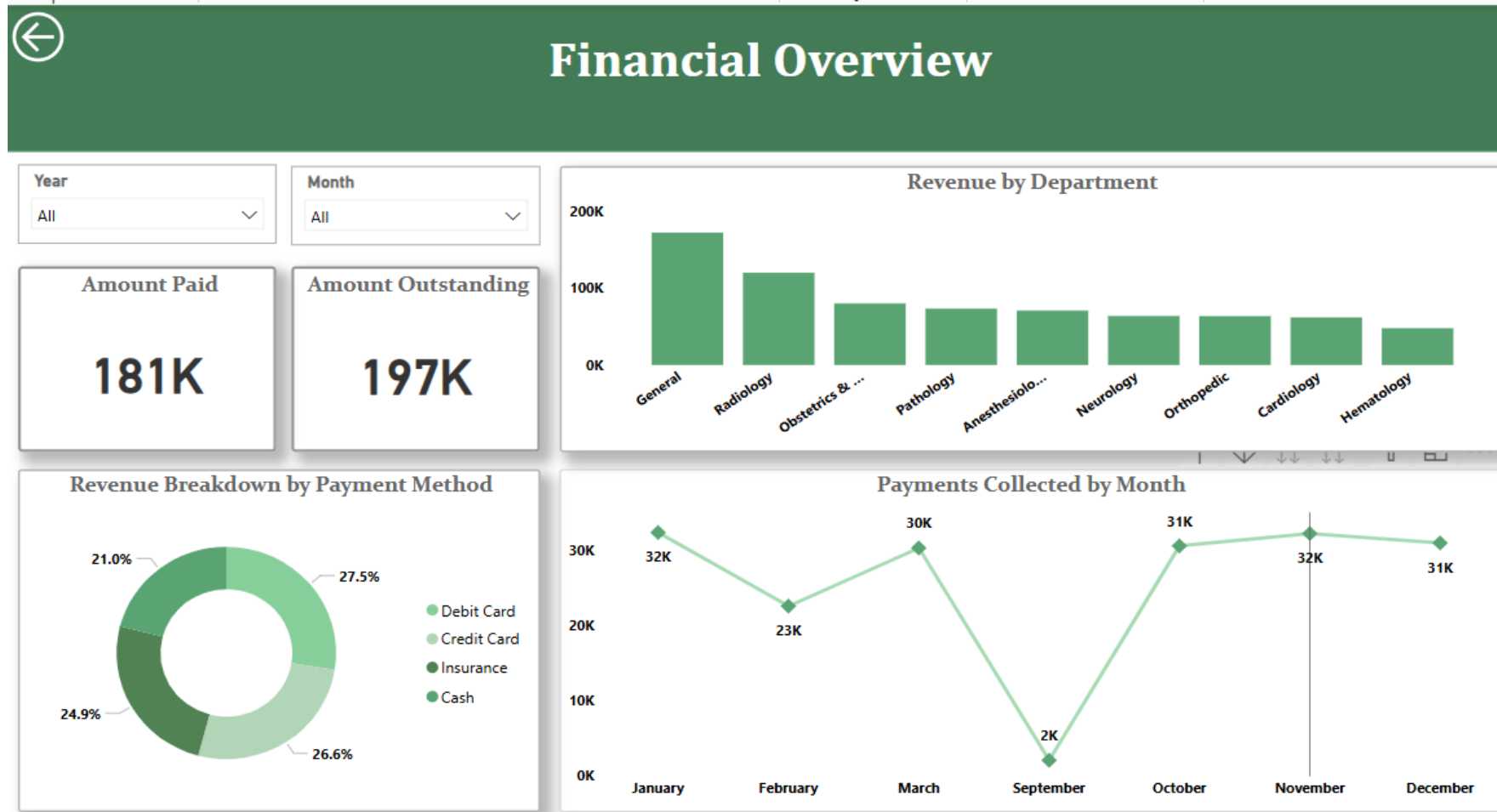
# Patient and Appointment Overview Dashboard



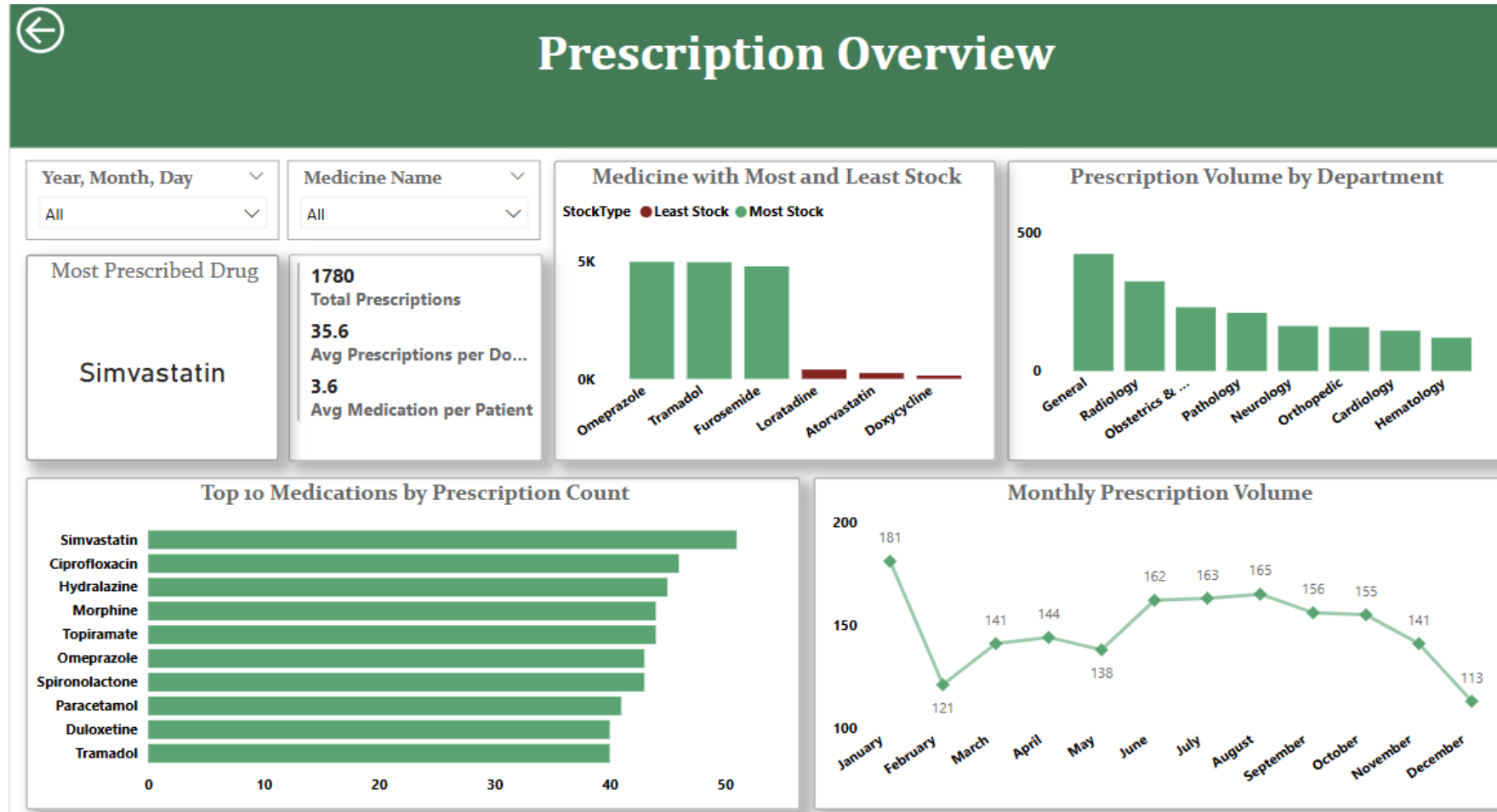
# Doctor Availability and Workload Dashboard



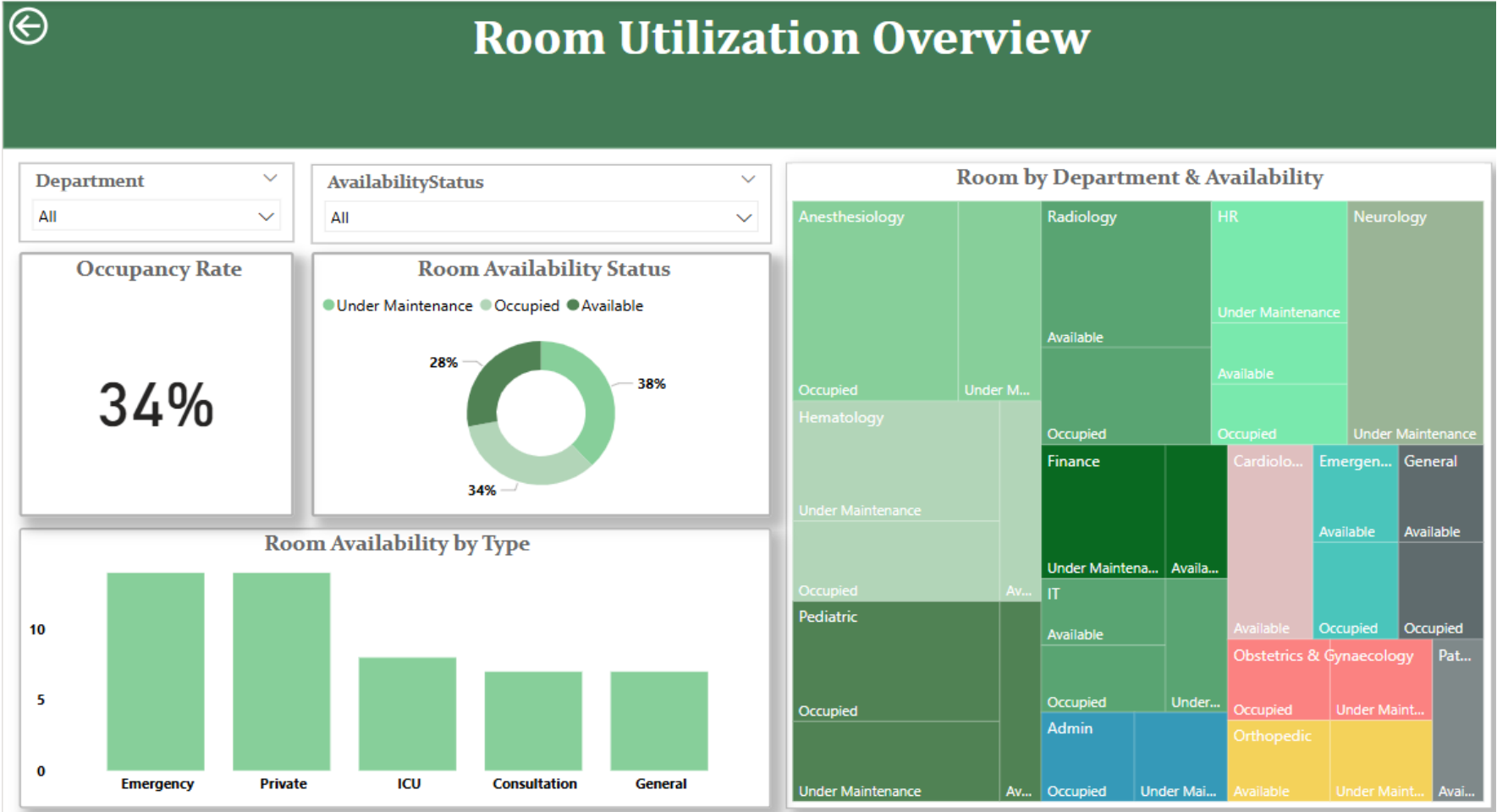
# Financial Overview Dashboard



# Prescriptions Overview Dashboard



# Room Utilization Overview Dashboard



# Challenges & Limitations

## Data Generation & Referential Integrity Challenges

- Generating large volumes of realistic yet relational synthetic data was time-consuming.
- Early GPT-4 outputs had random, disconnected values, breaking referential integrity.
- Required multiple prompt refinements and iterations to generate consistent, rule-based data.

## SQL-to-Power BI Integration

- Needed to write custom SQL queries and views to pre-aggregate and clean data before import.
- Power BI sometimes inferred incorrect relationships requiring manual adjustment.



# Learnings & Takeaways

This project strengthened our end-to-end understanding of data systems—from database design and synthetic data creation to real-time analytics and reporting.

## 1. Technical Learnings

- Gained hands-on experience in designing Entity-Relationship (ER) diagrams.
- Improved SQL skills for data generation, validation, and analytical querying.
- Learned to use prompt engineering with GPT-4 to generate high-quality synthetic data.
- Developed data transformation workflows connecting SQL Server with Power BI.

## 2. Analytical & Modeling Skills

- Built a relational data model with real-world hospital operations in mind.
- Understood the importance of referential integrity and data consistency for meaningful analysis.
- Created and optimized DAX measures and visualizations in Power BI.

## 3. Problem Solving & Iteration

- Discovered how iterative refinement (in prompts and design) leads to better outcomes.
- Adapted quickly to challenges in data inconsistencies and modeling logic.

# Conclusion

This project demonstrates how integrated data systems and thoughtful design can drive better healthcare management, resource planning, and patient experience analysis.

## Project Summary

- Successfully designed and implemented a relational hospital database with normalized tables and strong referential integrity.
- Used GPT-4 and prompt engineering to generate realistic synthetic data across multiple hospital operations.
- Connected the SQL Server database to Power BI, transforming and modeling data to build insightful reports.
- Extracted valuable insights using SQL queries and Power BI dashboards, helping simulate real-world healthcare decision-making.

# Appendix (1)

## Database Schema

Table Name	Primary Key	Foreign Keys	Purpose
Patients	PatientID		Holds patient demographic information
Doctors	DoctorID	DepartmentID	Contains doctor information and department link
Departments	DepartmentID		Stores hospital department details
Appointments	AppointmentID	PatientID, DoctorID, DepartmentID	Records patient-doctor appointments
MedicalRecords	RecordID	PatientID, DoctorID, AppointmentID	Stores diagnosis and treatment plans
Prescriptions	PrescriptionID	RecordID, MedicineID	Links medicines to medical records
Medicines	MedicineID		Contains medicine details and stock information
Billing	BillingID	PatientID	Manages billing and payment status
Staff	StaffID	DepartmentID	Stores non-doctor hospital staff details
Rooms	RoomID	DepartmentID	Manages room details and availability
Rooms Assignment	AssignmentID	PatientID, RoomID	Tracks patient room assignments and date

# Appendix (2)

## Sample Snippets of Data Generated

Patient Table

PatientID	FirstName	LastName	DateOfBirth	Gender	Address	PhoneNumber	Email	EmergencyContactName	EmergencyContactPhone
1	Benjamin	Ferguson	1965-04-02	Female	7619 Jeff Light, Josephland, BC P1A3G1	15749615235	jeremyholmes@hernandez.com	Miss Tina Roberts	13301267477
2	Douglas	Marsh	1989-11-01	Female	00272 Stanley Lodge, South Carlrosside, MB V4X6N3	13882060237	zvargas@phillips.com	Shannon Hall	13732294132

Staff Table

StaffID	FirstName	LastName	Role	DepartmentID	PhoneNumber	Email	ShiftHours
1	Tracy	Day	Technician	15	19415719327	jessicatan@harrison.com	07:00 - 15:00
2	Jennifer	Hanna	Lab Technician	14	12021968007	dgomez@gonzalez-hernandez.com	09:00 - 17:00

Doctor Table

DoctorID	FirstName	LastName	Specialization	PhoneNumber	Email	DepartmentID	Availability
1	Joseph	Vasquez	Neurology	13176361831	matajustin@hanson.net	10	13:00 - 21:00
2	Katherine	Smith	Radiology	12844467190	houstonsarah@nelson.com	14	11:00 - 19:00

Appointment Table

AppointmentID	PatientID	DoctorID	DepartmentID	AppointmentDate	AppointmentTime	Status
1	424	7	7	2024-04-16	15:00	Completed
2	326	31	6	2024-05-05	16:00	Scheduled

Department Table

DepartmentID	DepartmentName	Location	PhoneExtension
1	Admin	A-203	2000
2	Finance	B-259	2001

# Appendix (3)

## DDL

```
-- PATIENT TABLE
CREATE TABLE Patient (
    PatientID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    DateOfBirth DATE NOT NULL,
    Gender VARCHAR(10),
    Address VARCHAR(255),
    PhoneNumber VARCHAR(15) UNIQUE NOT NULL,
    Email VARCHAR(100) UNIQUE,
    EmergencyContactName VARCHAR(100),
    EmergencyContactPhone VARCHAR(15)
);

-- DOCTOR TABLE
CREATE TABLE Doctor (
    DoctorID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Specialization VARCHAR(100) NOT NULL,
    PhoneNumber VARCHAR(15) UNIQUE NOT NULL,
    Email VARCHAR(100) UNIQUE,
    DepartmentID INT NOT NULL,
    Availability VARCHAR(50),
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
);

-- DEPARTMENT TABLE
CREATE TABLE Department (
    DepartmentID INT PRIMARY KEY,
    DepartmentName VARCHAR(100) NOT NULL,
    Location VARCHAR(10) UNIQUE,
    PhoneExtension VARCHAR(10)
```

## DML

```
--Bulk CSV Import Using T-SQL
BULK INSERT appointment
FROM 'E:\Neeti\BISI\Data Analytics\Project\Appointment.csv'
with (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2,
    TABLOCK
);

BULK INSERT Billing
FROM 'E:\Neeti\BISI\Data Analytics\Project\Billing.csv'
with (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2,
    TABLOCK
);

BULK INSERT Department
FROM 'E:\Neeti\BISI\Data Analytics\Project\Departments.csv'
with (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2,
    TABLOCK
);
```

## User Roles and Access Control

```
--Admin Login
Create Login Admin_1
with Password = '123';

use hospital_database_fp;
Create user admin_1 for login admin_1;

--User Login
Create Login user_1
with Password = '1234';

use hospital_database_fp;
Create user user_1 for login user_1;

Create Login user_2
with Password = '1234';

use hospital_database_fp;
Create user user_2 for login user_2;

Create Login user_3
with Password = '1234';

use hospital_database_fp;
Create user user_3 for login user_3;

Create Login user_4
with Password = '1234';

use hospital_database_fp;
Create user user_4 for login user_4;

Create Login user_5
with Password = '1234';

use hospital_database_fp;
Create user user_5 for login user_5;
```

## Views for Business Question

```
-- 7. View: Room Availability Summary
CREATE VIEW vw_RoomAvailability AS
SELECT
    RoomType,
    COUNT(*) AS TotalRooms,
    SUM(CASE WHEN AvailabilityStatus = 'Available' THEN 1 ELSE 0 END) AS AvailableRooms,
    SUM(CASE WHEN AvailabilityStatus = 'Occupied' THEN 1 ELSE 0 END) AS OccupiedRooms
FROM Room
GROUP BY RoomType;

-- 8. View: Staff Count by Department
CREATE VIEW vw_StaffByDepartment AS
SELECT
    d.DepartmentName,
    COUNT(s.StaffID) AS TotalStaff
FROM Department d
JOIN Staff s ON d.DepartmentID = s.DepartmentID
GROUP BY d.DepartmentName;
```

Any questions?





Thank you for your  
attention!