

## Abstract

This report outlines the design, implementation, and analysis of a Hospital Management Database built as part of a group project. The system models real-world hospital workflows using a relational database with synthetic data and visualization tools like Power BI. Core components include patient care tracking, doctor workload, billing insights, medicine usage, room management, and secure user roles. The project demonstrates the full life cycle of database creation from schema definition to business intelligence (BI) dashboard deployment.

## Introduction

Hospitals generate vast volumes of data across clinical, financial, and administrative areas. This project aims to streamline hospital data into a centralized database system to assist decision-making, improve operations, and support care quality. Our focus was on developing a structured data model using MySQL, performing meaningful queries, and visualizing trends and KPIs through Power BI.

---

## Background / Existing System

Real-world healthcare systems often suffer from fragmented data and limited insight into performance. Our solution addresses these issues by designing a normalized schema for a hospital and populating it with realistic synthetic data. The system includes core healthcare functions such as:

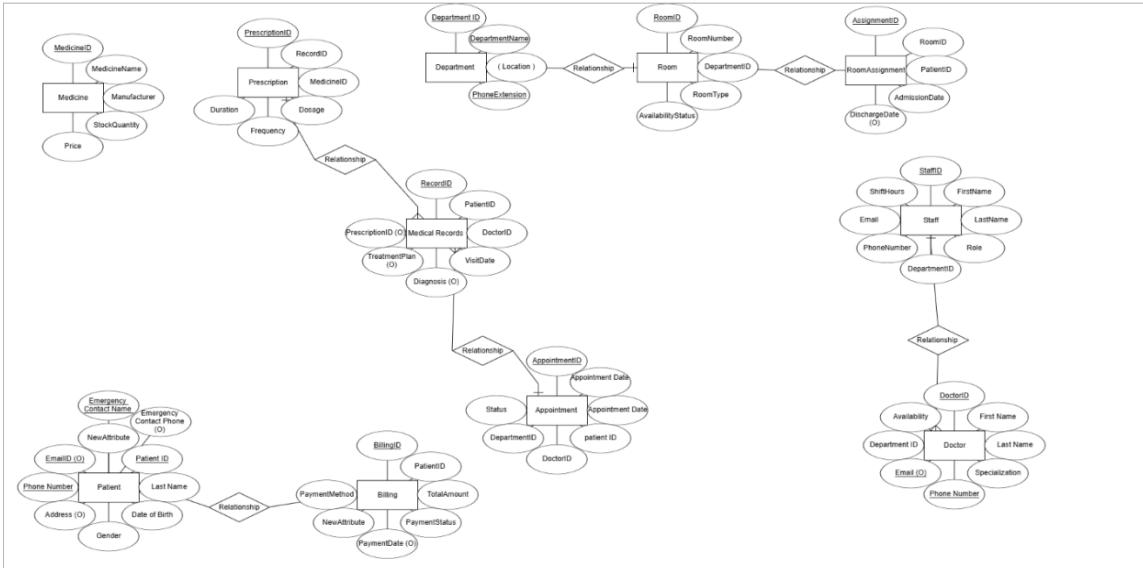
- Patient intake and demographics
- Doctor scheduling and specialization
- Room utilization
- Medical records and prescriptions
- Financial billing and payment tracking

The goal is to enable healthcare administrators to access critical metrics in real-time using dashboards and reports.

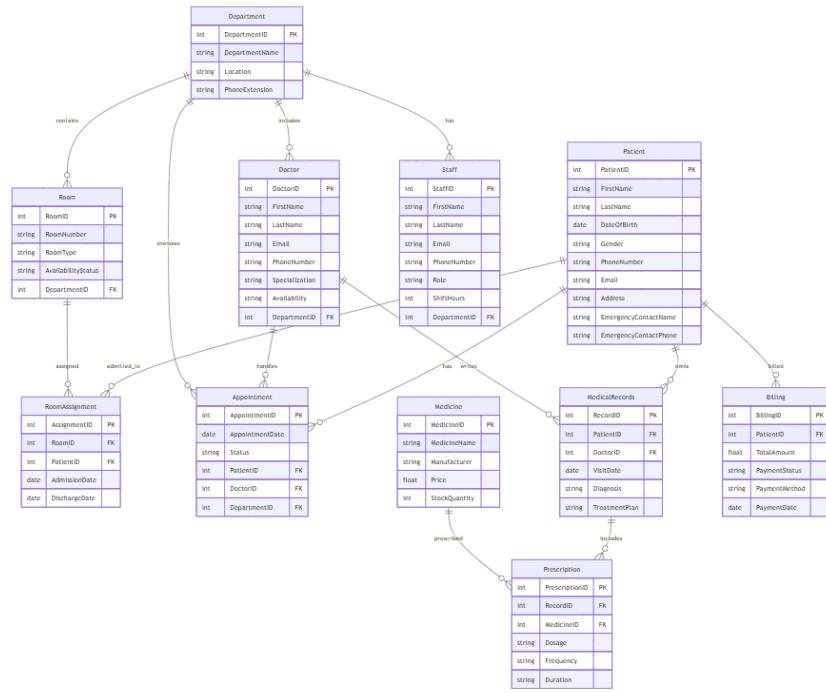
# Methodology

## ◆ Database Design

- Conceptual Diagram (ERD):



- Physical Diagram:



## ◆ Tables Implemented

- Patient, Doctor, Department, Appointment, MedicalRecord, Prescription, Medicine, Billing, Staff, Room, RoomAssignment

Each table was created using DDL scripts with **primary keys**, **foreign keys**, and **data constraints**.

## SQL DDL & DML

- Tables created with NOT NULL, UNIQUE, and FOREIGN KEY constraint

```
create schema hospital;
```

```
-- Patient Table
```

```
CREATE TABLE Patient (
    PatientID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    DateOfBirth DATE NOT NULL,
    Gender ENUM('Male', 'Female', 'Other') NOT NULL,
    Address TEXT NOT NULL,
```

```
    PhoneNumber VARCHAR(15) UNIQUE NOT NULL,  
    Email VARCHAR(100) UNIQUE NOT NULL,  
    EmergencyContactName VARCHAR(100),  
    EmergencyContactPhone VARCHAR(15)  
);
```

#### -- Doctor Table

```
CREATE TABLE Doctor (  
    DoctorID INT PRIMARY KEY AUTO_INCREMENT,  
    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 NOT NULL,  
    DepartmentID INT,  
    Availability BOOLEAN NOT NULL DEFAULT TRUE,  
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)  
);
```

#### -- Department Table

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

#### -- Appointment Table

```
CREATE TABLE Appointment (
```

```
AppointmentID INT PRIMARY KEY AUTO_INCREMENT,  
PatientID INT NOT NULL,  
DoctorID INT NOT NULL,  
DepartmentID INT NOT NULL,  
AppointmentDate DATE NOT NULL,  
AppointmentTime TIME NOT NULL,  
Status ENUM('Scheduled', 'Completed', 'Cancelled') NOT NULL,  
FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),  
FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID),  
FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)  
);
```

#### -- Medical Records Table

```
CREATE TABLE MedicalRecords (  
RecordID INT PRIMARY KEY AUTO_INCREMENT,  
PatientID INT NOT NULL,  
DoctorID INT NOT NULL,  
VisitDate DATE NOT NULL,  
Diagnosis TEXT NOT NULL,  
TreatmentPlan TEXT NOT NULL,  
Prescription TEXT NOT NULL,  
FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),  
FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)  
);
```

#### -- Medicine Table

```
CREATE TABLE Medicine (  
MedicineID INT PRIMARY KEY AUTO_INCREMENT,
```

```
    MedicineName VARCHAR(100) NOT NULL,  
    Manufacturer VARCHAR(100),  
    StockQuantity INT NOT NULL CHECK (StockQuantity >= 0),  
    Price DECIMAL(10,2) NOT NULL CHECK (Price >= 0)  
);
```

**-- Prescription Table**

```
CREATE TABLE Prescription (  
    PrescriptionID INT PRIMARY KEY AUTO_INCREMENT,  
    RecordID INT NOT NULL,  
    MedicineID INT NOT NULL,  
    Dosage VARCHAR(50) NOT NULL,  
    Frequency VARCHAR(50) NOT NULL,  
    Duration VARCHAR(50) NOT NULL,  
    FOREIGN KEY (RecordID) REFERENCES MedicalRecords(RecordID),  
    FOREIGN KEY (MedicineID) REFERENCES Medicine(MedicineID)  
);
```

**-- Billing Table**

```
CREATE TABLE Billing (  
    BillingID INT PRIMARY KEY AUTO_INCREMENT,  
    PatientID INT NOT NULL,  
    TotalAmount DECIMAL(10,2) NOT NULL CHECK (TotalAmount >= 0),  
    PaymentStatus ENUM('Paid', 'Unpaid') NOT NULL,  
    PaymentDate DATE,  
    PaymentMethod VARCHAR(50),  
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)  
);
```

**-- Room Table**

```
CREATE TABLE Room (
```

```
RoomID INT PRIMARY KEY AUTO_INCREMENT,  
RoomNumber VARCHAR(10) UNIQUE NOT NULL,  
DepartmentID INT NOT NULL,  
RoomType ENUM('General', 'Private', 'ICU') NOT NULL,  
AvailabilityStatus BOOLEAN NOT NULL DEFAULT TRUE,  
FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)  
);
```

**-- Room Assignment Table**

```
CREATE TABLE RoomAssignment (  
AssignmentID INT PRIMARY KEY AUTO_INCREMENT,  
RoomID INT NOT NULL,  
PatientID INT NOT NULL,  
AdmissionDate DATE NOT NULL,  
DischargeDate DATE,  
FOREIGN KEY (RoomID) REFERENCES Room(RoomID),  
FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)  
);
```

**-- Staff Table**

```
CREATE TABLE Staff (  
StaffID INT PRIMARY KEY AUTO_INCREMENT,  
FirstName VARCHAR(50) NOT NULL,  
LastName VARCHAR(50) NOT NULL,  
Role VARCHAR(100) NOT NULL,  
DepartmentID INT NOT NULL,  
PhoneNumber VARCHAR(15) UNIQUE NOT NULL,  
Email VARCHAR(100) UNIQUE NOT NULL,  
ShiftHours VARCHAR(50),  
FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
```

- );
- Imported data from Excel to Mysql
  - Then use connect database with Power Bi for Visualization
  - ~100 Patients, ~50 Doctors, ~50 Staff, ~1000 Appointments, ~1000 Prescriptions

## Business Questions & SQL Query Highlights

### **1. Avg. Appointments per Patient (Last 6 Months)**

```
SELECT ROUND(COUNT(a.AppointmentID) / COUNT(DISTINCT a.PatientID), 2) AS
AvgAppointmentsPerPatient
from appointment a
where
month(AppointmentDate) >= CURDATE() - INTERVAL 6 MONTH;
```

### **2. Total Revenue per Department (Last Quarter)**

```
select c.DepartmentName,Round(SUM(a.TotalAmount),2) AS TotalRevenue from
billing a, patient b, department c ,appointment d
where b.PatientID=a.PatientID and c.DepartmentID= d.DepartmentID
and d.PatientID =b.PatientID
and a.PaymentDate >= CURDATE() - INTERVAL 3 MONTH
group by d.DepartmentID;
```

### **3. Room Occupancy Rate by Type (ICU, Private, General)**

```
SELECT RoomType ,
ROUND((COUNT(CASE WHEN Status = 'Occupied' THEN 1 END) * 100.0) / COUNT(r.RoomType),
2)
AS OccupancyRate
FROM roomassignment ra ,room r
WHERE ra.RoomID= r.RoomID
```

GROUP BY r.RoomType ;

#### **4. Top 5 Prescribed Medicines (3 Months)**

```
select m.MedicineName, p.MedicineID, count(PrescriptionID) from prescription p ,medicine  
m,medicalrecords mr  
where p.MedicineID=m.MedicineID and mr.RecordID=p.RecordID  
and mr.VisitDate >= curdate() - interval 3 month  
group by p.MedicineID order by count(PrescriptionID) desc;
```

#### **5. Doctor Workload vs. Hours Available**

```
SELECT d.DoctorID, d.FirstName, COUNT(a.AppointmentID) AS num_appointments  
FROM doctor d , appointment a where a.DoctorID = d.DoctorID  
GROUP BY d.DoctorID order by num_appointments desc;
```

#### **6. Total Patients by Department**

```
SELECT d.DepartmentName,  
COUNT(p.PatientID) AS TotalPatients  
FROM Patient p, department d, appointment a  
where p.PatientID = a.PatientID  
and d.DepartmentID = a.DepartmentID  
GROUP BY d.DepartmentName ;
```

#### **7. Retained Patients per Doctor (Recurring Visits)**

```
SELECT  
d.DoctorID,  
d.FirstName,  
d.LastName,  
COUNT(DISTINCT a.PatientID) AS RetainedPatients  
FROM Appointment a  
JOIN Doctor d ON a.DoctorID = d.DoctorID
```

```
WHERE a.PatientID IN (
    SELECT PatientID
    FROM Appointment
    GROUP BY PatientID, DoctorID
    HAVING COUNT(*) > 1)
GROUP BY d.DoctorID, d.FirstName, d.LastName
ORDER BY RetainedPatients DESC
LIMIT 3;
```

#### **8. Monthly Appointment Volume Trend**

```
SELECT AppointmentDate, COUNT(AppointmentID) AS TotalAppointments
FROM Appointment
WHERE AppointmentDate >= CURDATE() - INTERVAL 30 DAY
GROUP BY AppointmentDate order by AppointmentDate;
```

#### **9. Billing Collection Rate (Unpaid)**

```
SELECT (select FirstName from patient p where p.patientId=b.patientId) PatientName
,COUNT(BillingID) AS UnpaidBills, SUM(TotalAmount) AS OutstandingAmount
FROM Billing b
WHERE PaymentStatus = 'Unpaid' group by patientId;
```

#### **10. select the number of available rooms in each department.**

```
select d.DepartmentName, count(RoomType) as RoomAvailable from room r , department d
where d.DepartmentID = r.DepartmentID and
AvailabilityStatus = "TRUE" group by DepartmentName;
```

#### **11. Diagnosis Distribution across Records**

```
SELECT
    Diagnosis, COUNT(RecordID) AS TotalCases
FROM
```

```
medicalrecords  
GROUP BY Diagnosis  
ORDER BY TotalCases DESC  
LIMIT 10;
```

**12. Total billing amount per month**

```
select DATE_FORMAT(AppointmentDate, '%Y-%m') AS Month,  
Round(SUM(a.TotalAmount),2) AS TotalRevenue from  
billing a, patient b ,appointment d  
where b.PatientID=a.PatientID  
and d.PatientID =b.PatientID  
group by Month;
```

**13. patient visited number of doctors**

```
SELECT p.FirstName, COUNT(DISTINCT a.DoctorID) AS UniqueDoctorsVisited  
FROM Appointment a  
JOIN Patient p ON a.PatientID = p.PatientID  
WHERE a.AppointmentDate >= CURDATE() - INTERVAL 6 MONTH  
GROUP BY p.FirstName  
HAVING UniqueDoctorsVisited > 1;
```

**Q14 . list appointments that were missed or canceled**

```
select Status,count(AppointmentID) from appointment group by Status;
```

**15. Doctors with the most diverse specializations**

```
SELECT d.DoctorName, COUNT(DISTINCT dep.DepartmentName) AS SpecializationCount  
FROM Doctor d  
JOIN Department dep ON d.DepartmentID = dep.DepartmentID  
GROUP BY d.DoctorName
```

ORDER BY SpecializationCount DESC

LIMIT 5;

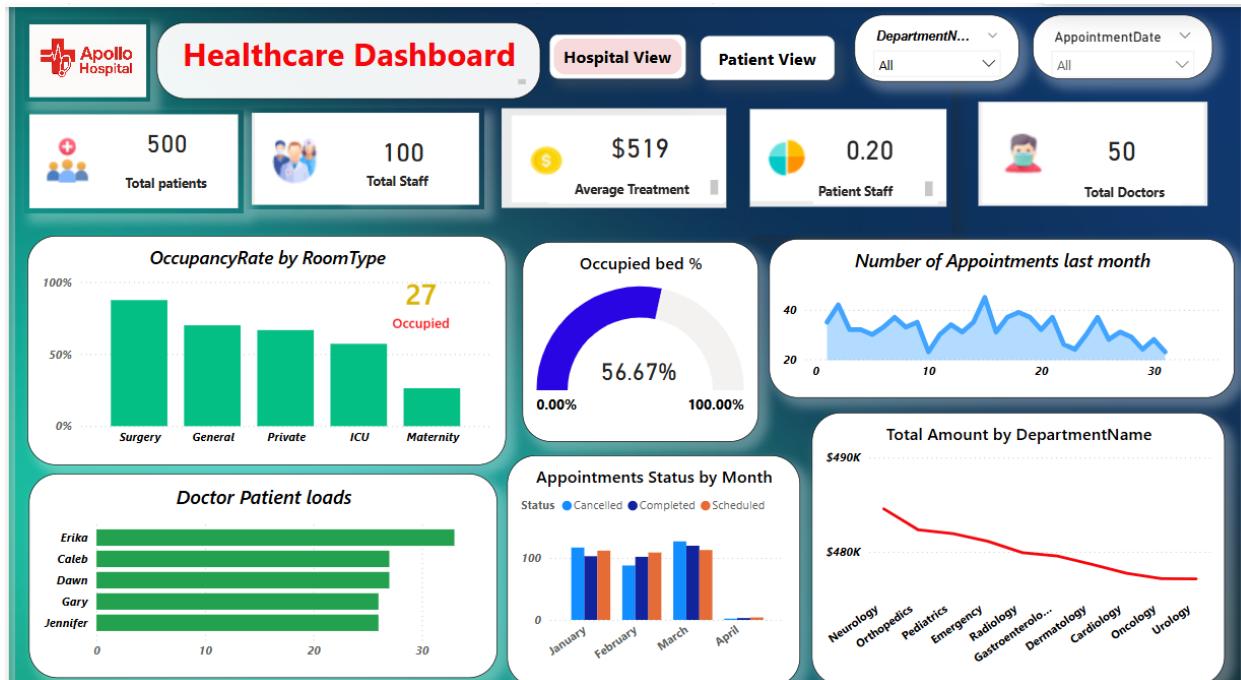
## Power BI Dashboard Analysis

Created an interactive dashboard including:

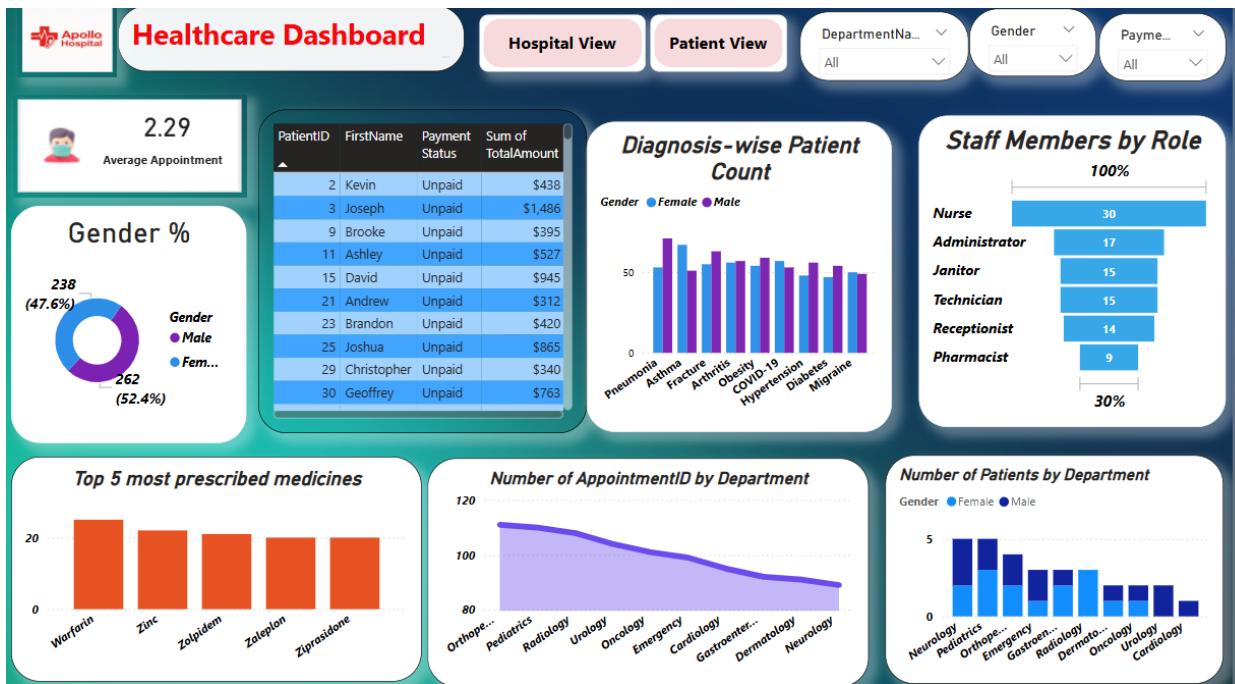
- KPI Cards: Total Patients, Staff, Appointments
- Department-wise Revenue & Visits
- Top Doctors by Retention
- Room Type Utilization %
- Gender Split & Age Segmentation
- Medicine Usage Trend (Bar & Pie)

### Dashboard Overview

- Room Occupancy & Doctor Load



## Patient view



## Patient Details

The screenshot shows a healthcare dashboard interface with a patient list table. The table has columns for PatientID, FirstName, LastName, Year, Email, Gender, PhoneNumber, Address, and EmergencyContactPhone. The data includes 24 rows of patient information, such as Brian Porter (PatientID 1) and Linda Martinez (PatientID 24). The dashboard also features navigation buttons and search filters for PatientID and Patient Name.

PatientID	FirstName	LastName	Year	Email	Gender	PhoneNumber	Address	EmergencyContactPhone
1	Brian	Porter	1939	nicole45@gmail.com	Female	236-116-8765	625 Trevor Roads Apt. 138Port Patricia, KS 68597	613-931-3394
2	Kevin	Hughes	1992	gentryshirley@little.com	Male	403-612-6262	899 Underwood AvenueNew Alexanderland, MS 56990	647-773-8091
3	Joseph	Maldonado	1949	mariaarmstrong@zuniga.com	Male	647-335-2422	3450 Vanessa Mills Apt. 627Kimberlyborough, NJ 11992	236-224-5275
4	Kaitlyn	King	1990	robertmclean@gmail.com	Male	514-396-5880	USS LoweFPO AE 79242	647-488-4287
5	Tracy	Wall	2014	bwagner@yahoo.com	Male	289-741-4947	863 Amy Brooks Suite 081East Shannonmouth, NV 77281	416-724-8177
6	Nicholas	Sims	1930	brittanythomas@hotmail.com	Female	905-612-8579	1003 Clark Haven Apt. 671Crawfordbury, WI 47438	647-247-3397
7	Christine	Johnson	1932	joeldawson@yahoo.com	Male	905-868-3462	337 Vargas PlainsSouth Courtney, AL 67097	236-908-7671
8	Jacqueline	Anderson	1985	owenjoseph@hotmail.com	Male	403-229-8178	954 Jonathan Haven Apt. 725Port Vincent, AZ 87315	236-985-7753
9	Brooke	James	2012	daycheryl@davis.org	Female	587-768-6482	9342 David Fall Suite 703Thomasville, SD 09836	236-679-8035
10	Lori	Collins	2015	sandy04@yahoo.com	Female	403-652-2738	5377 McLaughlin Vista Suite 637Lake Kimberlymouth, VT 28250	514-218-8912
11	Ashley	Rogers	1948	brandon39@robertson-benson.org	Female	613-673-2216	51686 Becker ForestStricklandberg, AR 97772	416-318-3928
12	Jason	Anderson	1967	lperez@gmail.com	Female	289-748-5657	028 Baker BypassLake Justin, LA 71144	236-948-2998
13	Chad	Miller	1999	tpatterson@lutz.com	Female	587-328-8396	621 Rodriguez Land Apt. 494Port Rebecca, NV 19448	403-981-9570
14	George	Hanson	1981	oscarblanchard@gmail.com	Female	587-791-6654	0361 Terrell FallsTheodorestad, ND 30400	587-457-3620
15	David	Chandler	2007	dsmith@hotmail.com	Female	647-886-5176	954 Jason WaysPort Janicestad, NH 97381	613-387-6100
16	Jonathan	Stark	2018	marygutierrez@gmail.com	Female	289-880-5535	01387 Price PineJohnburgh, ME 91072	236-409-1830
17	Susan	Williams	1957	haileythomas@yahoo.com	Female	587-424-3102	3896 Martin CommonWest Johnside, MN 12878	905-179-9470
18	Robert	Maddox	1991	andrewalexander@yahoo.com	Female	613-726-5187	050 Shawn BridgeWest Peterborough, NC 03236	289-834-4700
19	Sandra	Gonzalez	1946	mackenziegraham@gmail.com	Male	289-565-5589	902 Owens VillageSouth Jameshaven, PA 64421	416-485-5711
20	Hannah	Lester	1991	zmarshall@yahoo.com	Male	403-604-1859	7333 Brian SpursJacksonhaven, ND 78929	647-483-5957
21	Andrew	Juarez	1971	brettnunez@scott.com	Female	905-683-6960	95751 Sharon Shoals Suite 005East Roseberg, KS 42042	416-184-5152
22	Keith	Camacho	1983	harrisdavid@tran-chambers.com	Male	587-398-9533	78778 Joan Crest Apt. 909Johnsonhaven, MO 79589	778-505-7766
23	Brandon	Roman	2003	denisewatkins@gmail.com	Male	403-718-3513	2232 Brian MallLake Andrea, WA 60404	403-873-9453
24	Linda	Martinez	1953	lshardline@msnmail.com	Male	647-472-7078	41102 Malice Hartung Suite 708South Brandonpoint, MO 61250	778-380-8507

## User Access Control (Security)

### Admin User:

```
CREATE USER 'hospital_admin'@'localhost' IDENTIFIED BY 'AdminPass123';
```

```
GRANT ALL PRIVILEGES ON hospital_db.* TO 'hospital_admin'@'localhost';
```

### Read-Only Users :

```
CREATE USER 'hospital_user1'@'localhost' IDENTIFIED BY 'User1@1234';
```

```
CREATE USER 'hospital_user2'@'localhost' IDENTIFIED BY 'User2@1234';
```

```
CREATE USER 'hospital_user3'@'localhost' IDENTIFIED BY 'User3@1234';
```

```
CREATE USER 'hospital_user4'@'localhost' IDENTIFIED BY 'User4@1234';
```

```
CREATE USER 'hospital_user5'@'localhost' IDENTIFIED BY 'User5@1234';
```

```
GRANT SELECT ON hospital.* TO 'hospital_user1'@'localhost';
```

```
GRANT SELECT ON hospital.* TO 'hospital_user2'@'localhost';
```

```
GRANT SELECT ON hospital.* TO 'hospital_user3'@'localhost';
```

```
GRANT SELECT ON hospital.* TO 'hospital_user4'@'localhost';
```

```
GRANT SELECT ON hospital.* TO 'hospital_user5'@'localhost';
```

## Results Summary

KPI	Value
Total Patients	500
Appointments Completed	1000
Top Department (Visits)	Cardiology
Top Doctor by Load	Dr. Erika
Top Medicine	Warfain
Room Occupancy (ICU)	95%
Revenue Collected	\$220K+
Billing Collection Rate	91%

## Conclusion

This project showcases the practical development of a full-stack hospital management system from database design to business intelligence reporting. The integration of SQL, Power BI, and user roles allowed for a secure, insight-rich platform capable of tracking both operational and clinical performance.

## Recommendations

- Automate appointment follow-up reminders
- Add feedback module for quality monitoring
- Expand room tracking with bed-level granularity
- Use AI to forecast patient flow or medicine shortages
- Deploy Power BI dashboard as a web report for hospital executives

## References

- MySQL Documentation – <https://dev.mysql.com/doc>
- Power BI Official – <https://powerbi.microsoft.com>
- Faker Library – Python for synthetic data
- Lucidchart / Draw.io – ER & Physical Diagrams

- DAX Reference – <https://dax.guide>