Hospital Management System – SQL & Power BI Project

Project Title:

Hospital Management System – Data Analytics with SQL & Power BI

Project Type:

Portfolio Project for Data Analyst Role

Developed By:

Niraj Pandey

Aspiring Data Analyst

SQL • Power BI • Python

pniraj124@gmail.com

Project Summary

This project focuses on analyzing healthcare operational data through structured SQL queries and interactive Power BI dashboards. The analysis includes insights on patient demographics, doctor performance, appointment trends, treatment analysis, and hospital revenue metrics.

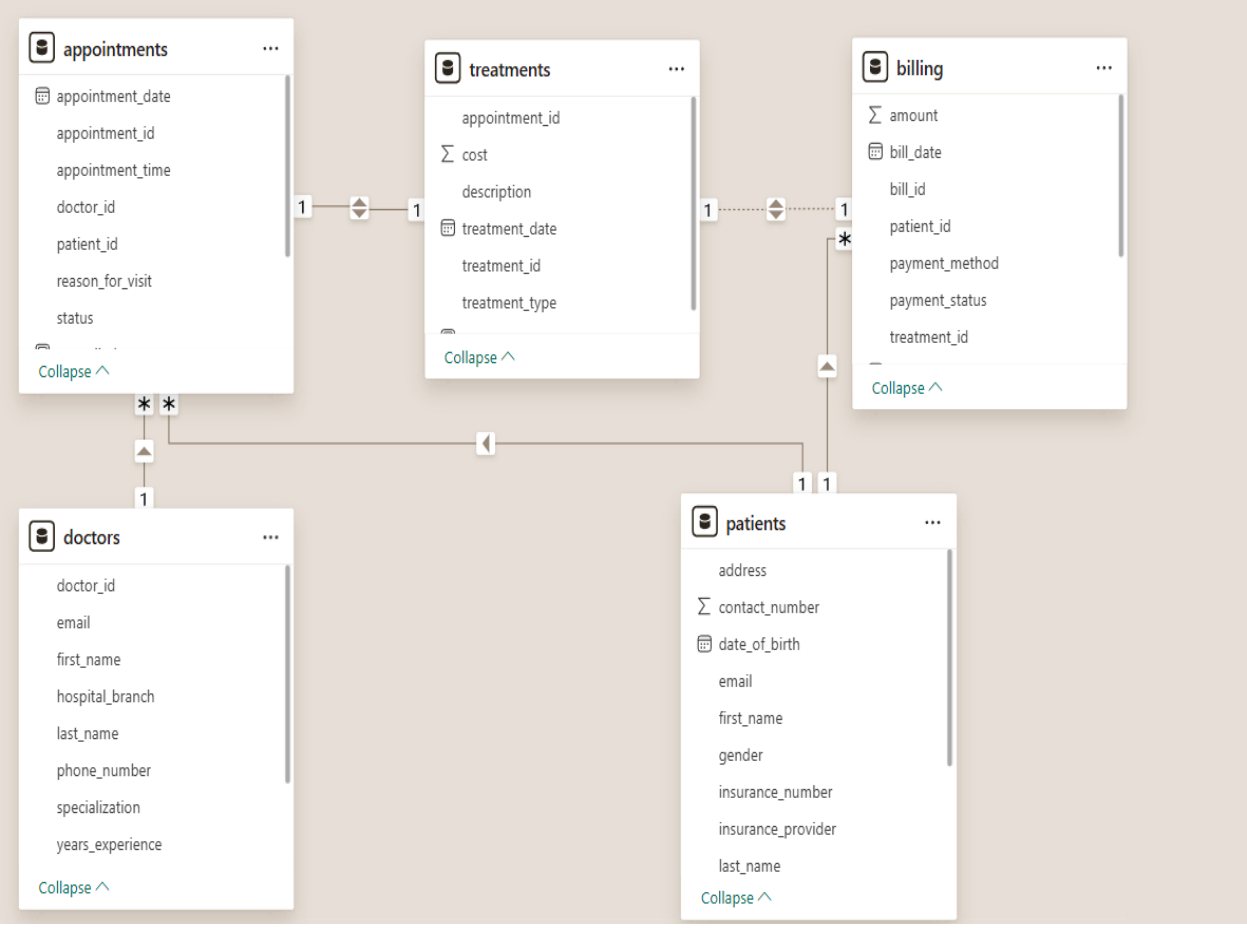
Objective

Analyze and visualize hospital operations to uncover insights about patients, doctors, treatments, appointments, and billing using structured SQL queries and Power BI.

Tools Used

- **MySQL Workbench** for SQL querying
- **Power BI** for dashboard visualization
- **CSV files** for data sources

Dataset Overview

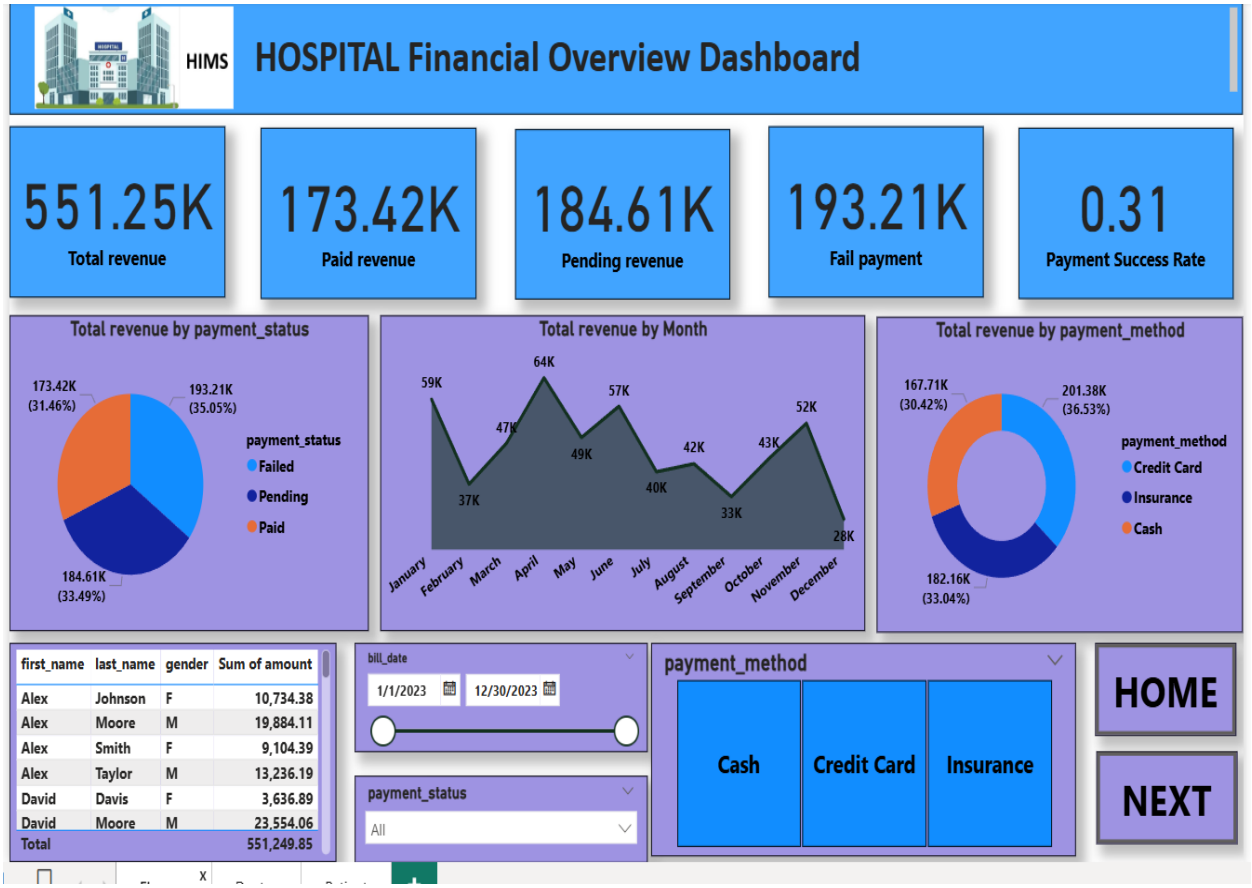


Dashboard Overview (Power BI)

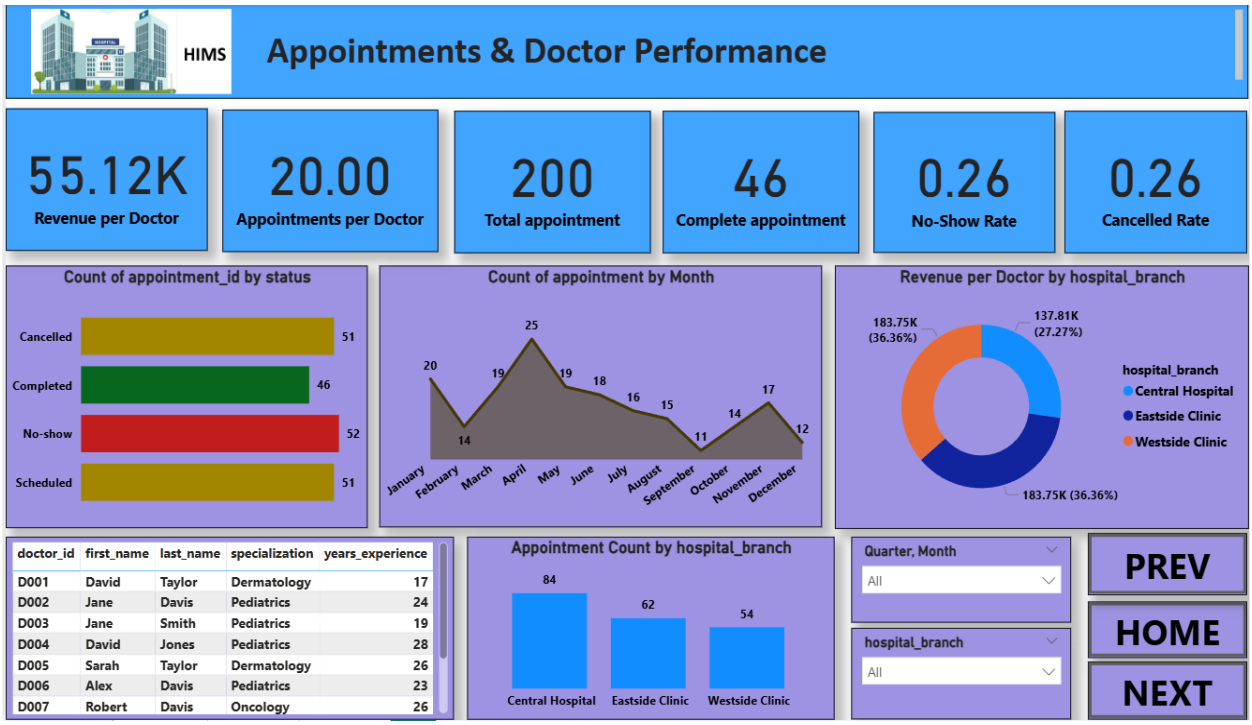
Key Visuals:

- Total Patients, Total Revenue, Total Appointments, Total Doctors
- Gender Distribution (Pie)
- Age Distribution (Histogram)
- Appointment Status (Donut Chart)
- Monthly Revenue Trend (Line)
- Top 5 Patients by Revenue
- Top 5 Experienced Doctors
- Doctors with No Appointments

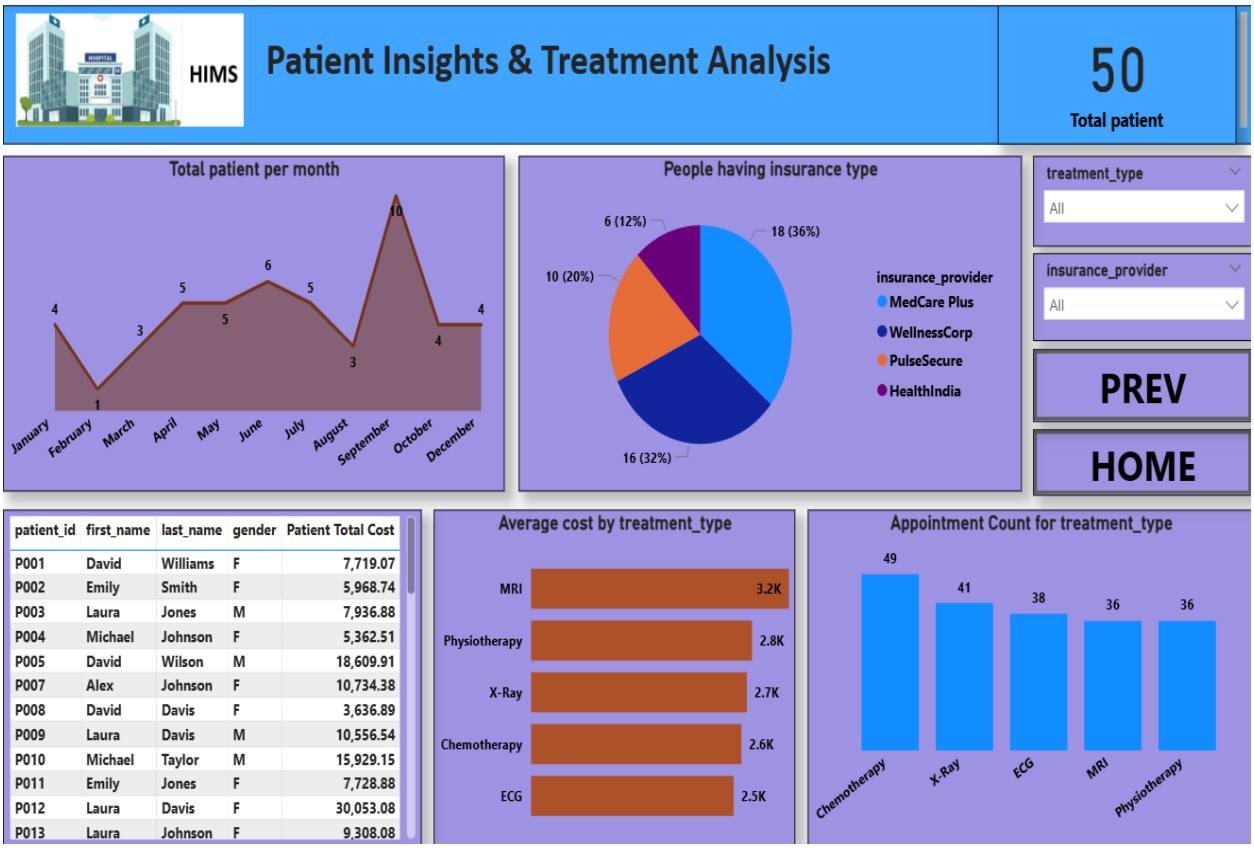
DASHBOARD 1: Hospital Financial Overview Dashboard



DASHBOARD 2: Appointments & Doctor Performance



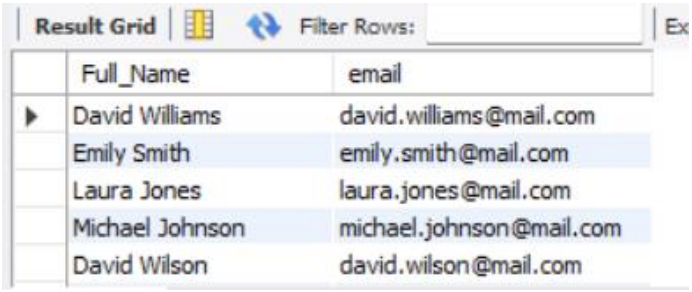
DASHBOARD 3: Patient Insights & Treatment Analysis



Key SQL Queries

1. **List all patients with their full name and email.**

```sql
-- 1. List all patients with their full name and email.
SELECT CONCAT(first_name,' ',last_name) as Full_Name, email
FROM patients;
```



Result Grid | Filter Rows:

Full_Name	email
David Williams	david.williams@mail.com
Emily Smith	emily.smith@mail.com
Laura Jones	laura.jones@mail.com
Michael Johnson	michael.johnson@mail.com
David Wilson	david.wilson@mail.com

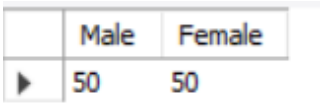
2. **Show all doctors who specialize in "Dermatology".**

```sql
-- 2. Show all doctors who specialize in "Dermatology".
SELECT CONCAT(first_name,' ',last_name) AS full_name
FROM doctors
WHERE LOWER(specialization)= "dermatology";
```

full_name
David Taylor
Sarah Taylor
Linda Brown

3. **Count how many patients are male and female**

```sql
-- 3. Count how many patients are male and female.
SELECT COUNT(LOWER(gender)="male") AS Male, COUNT(LOWER(gender)="female") AS Female
FROM patients;
```



Male	Female
50	50

4. **List the top 5 most experienced doctors.**

```sql
-- 5. List the top 5 most experienced doctors.
SELECT CONCAT(first_name,' ',last_name) as Full_Name, years_experience
FROM doctors
ORDER BY years_experience DESC
LIMIT 5;
```

Result Grid | Filter Rows:

Full_Name	years_experience
David Jones	28
Sarah Taylor	26
Robert Davis	26
Sarah Smith	26
Jane Davis	24

5. List all pending bills from the billing table.

```sql
-- 8. List all pennding bills from the billing table.
SELECT bill_id, patient_id
FROM billing
WHERE LOWER(payment_status)="pending";
```

bill_id	patient_id
B001	P034
B005	P040
B006	P045
B012	P029
B015	P026

6. Find the total revenue generated from each treatment type.

```sql
-- 11. Find the total revenue generated from each treatment type.
SELECT t.treatment_type, ROUND(SUM(b.amount),2) AS total_revenue
FROM treatments t
JOIN billing b ON t.treatment_id=b.treatment_id
GROUP BY t.treatment_type
ORDER BY total_revenue DESC;
```

treatment_type	total_revenue
Chemotherapy	128855.68
MRI	116098.16
X-Ray	110653.67
Physiotherapy	99418.1
ECG	96224.24

7. List patients along with their assigned doctor and appointment date.

```sql
-- 12. List patients along with their assigned doctor and appointment date.
SELECT
    CONCAT (p.first_name,' ',p.last_name) AS Patient_name,
    CONCAT (d.first_name,' ',d.last_name) AS Doctor_name,
    a.appointment_date
FROM appointments a
JOIN doctors d ON a.doctor_id = d.doctor_id
JOIN patients p ON a.patient_id= p.patient_id;
```

	Patient_name	Doctor_name	appointment_date
▶	David Williams	Sarah Taylor	4/1/2023
	David Williams	Alex Davis	1/26/2023
	David Williams	Sarah Smith	1/16/2023
	David Williams	Robert Davis	4/9/2023
	Emily Smith	Jane Davis	4/12/2023

8. Show how many treatments each patients has received.

```
-- 13. Show how many treatments each patient has received.
SELECT
    CONCAT(p.first_name,' ',p.last_name) AS Patient_name,
    COUNT(t.treatment_type) AS Treatments
FROM patients p
JOIN billing b ON p.patient_id = b.patient_id
JOIN Treatments t ON b.treatment_id = t.treatment_id
GROUP BY p.first_name, p.last_name
ORDER BY Treatments DESC;
```

	Patient_name	Treatments
▶	Michael Taylor	16
	David Wilson	15
	Michael Wilson	14
	Laura Davis	14
	David Smith	9

9. Show the monthly revenue trend for the last 6 month

```
-- 17. Show the monthly revenue trend for the last 6 months.
SELECT
    DATE_FORMAT(STR_TO_DATE(bill_date, '%m/%d/%Y'), '%Y-%m') AS month,
    ROUND(SUM(amount), 2) AS total_revenue
FROM billing
GROUP BY month
ORDER BY month;
```

	month	total_revenue
▶	2023-01	58701.23
	2023-02	36669.69
	2023-03	47304.29
	2023-04	64271.54
	2023-05	48791.05

10. Display each patient's age based on their date of birth.

```
-- 20. Display each patient's age based on their date of birth.
SELECT
    CONCAT(p.first_name, ' ', p.last_name) AS Patient_name,
    TIMESTAMPDIFF(YEAR, STR_TO_DATE(p.date_of_birth, '%d/%m/%Y'),CURDATE()) AS Age
FROM patients p
WHERE p.date_of_birth IS NOT NULL
ORDER BY age DESC;
```

	Patient_name	Age
▶	Jane Wilson	74
	David Williams	70
	John Brown	69
	Sarah Johnson	60
	Emily Jones	59

11. Find the doctor who treated the highest number of unique patients.

```
-- 22. Find the doctor who treated the highest number of unique patients.
SELECT
    CONCAT(d.first_name, ' ', d.last_name) AS Doctor_name,
    COUNT(DISTINCT a.patient_id) AS patient_no
FROM doctors d
JOIN appointments a ON d.doctor_id=a.doctor_id
GROUP BY d.first_name, d.last_name
ORDER BY patient_no DESC
LIMIT 1;
```

	Doctor_name	patient_no
▶	Sarah Taylor	23

12. Create a report that shows -patient name – doctor name- treatment name-bill amount-payment status- all in one row.

```
-- 25. Create a report that shows:• patient name • doctor name • treatment name • bill amount • payment status – all in one row.
SELECT
    CONCAT(p.first_name, ' ', p.last_name) AS Patient_name,
    CONCAT(d.first_name, ' ', d.last_name) AS Doctor_name,
    t.treatment_type,b.amount,b.payment_status
FROM patients p
JOIN billing b ON p.patient_id = b.patient_id
JOIN treatments t ON b.treatment_id = t.treatment_id
JOIN appointments a ON t.appointment_id = a.appointment_id
JOIN doctors d ON a.doctor_id = d.doctor_id
```

Patient_name	Doctor_name	treatment_type	amount	payment_status
David Williams	Sarah Taylor	Physiotherapy	975.49	Pending
David Williams	Alex Davis	ECG	2960.14	Paid
David Williams	Sarah Smith	Chemotherapy	3249.41	Failed
David Williams	Robert Davis	Chemotherapy	534.03	Failed
Emily Smith	Jane Davis	Chemotherapy	616.15	Paid

Conclusion

This project demonstrates my ability to work with structured healthcare data using SQL and to transform analytical insights into a meaningful, interactive Power BI dashboard. Through the integration of five key datasets—patients, doctors, appointments, treatments, and billing—I was able to explore and answer real-world business questions.

By writing these SQL queries ranging from basic filtering to advanced joins and aggregations, I gained hands-on experience with data cleaning, relationship modeling, and KPI extraction. The Power BI dashboard added a powerful visualization layer to communicate these insights clearly and effectively.

This end-to-end project has not only enhanced my technical skills in SQL and Power BI but also strengthened my analytical thinking, data storytelling, and dashboard design capabilities—making me better prepared for a data analyst role in healthcare or any data-driven industry.