End to End Hospital Analytics

Overview

This project delivers a comprehensive hospital analytics project developed using SQL for data processing and querying, and Power BI for data modeling and visualization. Designed to reflect the complexities of real world hospital operations, it provides insights into patient flow, physician performance, financial trends, and operational challenges through effective data storytelling.

With a background in healthcare, I approached this project with a focus on business relevance and user centric dashboard design. The outcome is a practical and actionable analytics resource that enables hospital management to track key performance indicators, identify inefficiencies, and support data driven decision making.

Thought Process & Project Objectives

My starting point was to ask: *"What would a hospital administrator or stakeholder need to see daily to improve performance?"* From there, I outlined the following objectives:

- Consolidate diverse datasets covering clinical, operational, and financial data
- Validate data quality and ensure referential integrity across all available data
- Use SQL to uncover trends in patient behavior, treatment outcomes, doctor productivity, and billing health
- Design a Power BI dashboard that is easy to interpret, with distinct pages for clinical and financial insights, making it actionable for hospital stake holders
- Use storytelling to transform numbers into insight and strategy

Data Sources

The analysis was based on five interconnected datasets:

- `patients`: basic demographics, insurance information, registration date
- `doctors`: specialization, experience, branch
- `appointments`: dates, reasons for visit, assigned doctor, attendance status
- `treatments`: type, cost, description, and date
- `billing`: amount, payment method, payment status

Tools & Technologies

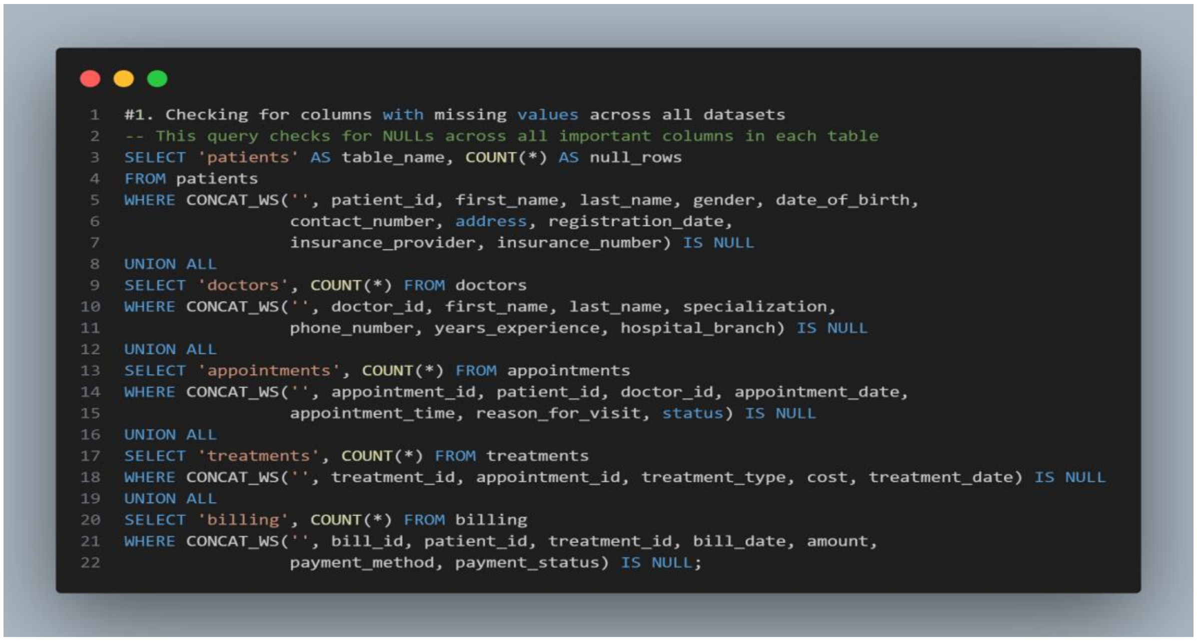
- **SQL (MySQL)**: data cleaning, integrity checks, KPI computation, complex joins
- **Power BI**: dashboard building and DAX for aggregations
- **Excel**: initial exploration and sample mock-ups

Data Cleaning and Validation

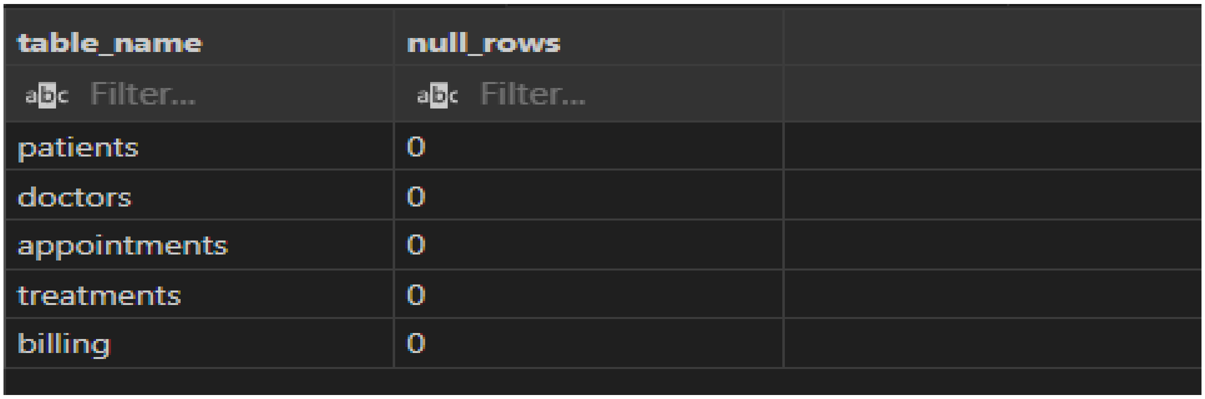
To ensure the accuracy and reliability of the hospital dataset, a thorough data cleaning and validation process was performed across all key tables: `patients`, `doctors`, `appointments`, `treatments`, and `billing`. The following validation steps were implemented:

Missing Values Check

Each table was scanned for rows with missing critical fields using a combination of `CONCAT_WS()` and `IS NULL`. This allowed identification of any records with incomplete information across key attributes such as patient details, appointment metadata, or billing amounts.



```sql
#1. Checking for columns with missing values across all datasets
-- This query checks for NULLs across all important columns in each table
SELECT 'patients' AS table_name, COUNT(*) AS null_rows
FROM patients
WHERE CONCAT_WS('', patient_id, first_name, last_name, gender, date_of_birth,
                contact_number, address, registration_date,
                insurance_provider, insurance_number) IS NULL
UNION ALL
SELECT 'doctors', COUNT(*) FROM doctors
WHERE CONCAT_WS('', doctor_id, first_name, last_name, specialization,
                phone_number, years_experience, hospital_branch) IS NULL
UNION ALL
SELECT 'appointments', COUNT(*) FROM appointments
WHERE CONCAT_WS('', appointment_id, patient_id, doctor_id, appointment_date,
                appointment_time, reason_for_visit, status) IS NULL
UNION ALL
SELECT 'treatments', COUNT(*) FROM treatments
WHERE CONCAT_WS('', treatment_id, appointment_id, treatment_type, cost, treatment_date) IS NULL
UNION ALL
SELECT 'billing', COUNT(*) FROM billing
WHERE CONCAT_WS('', bill_id, patient_id, treatment_id, bill_date, amount,
                payment_method, payment_status) IS NULL;
```



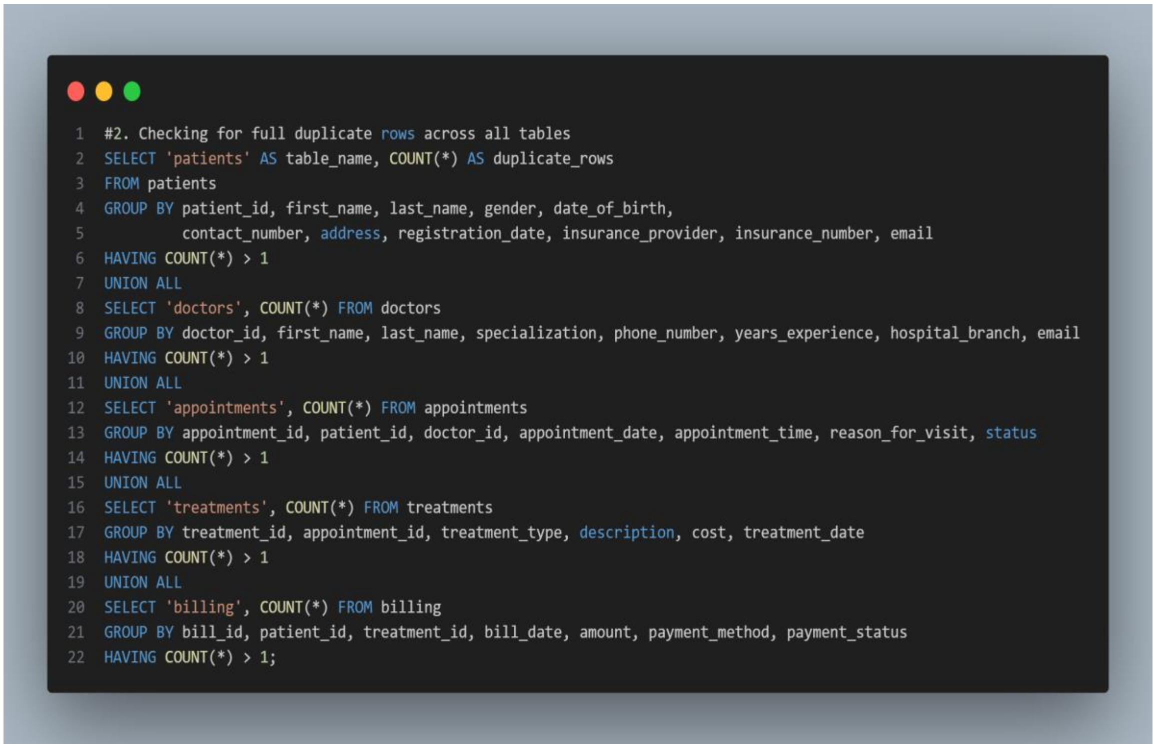
Filter...
Filter...

table_name	null_rows
patients	0
doctors	0
appointments	0
treatments	0
billing	0

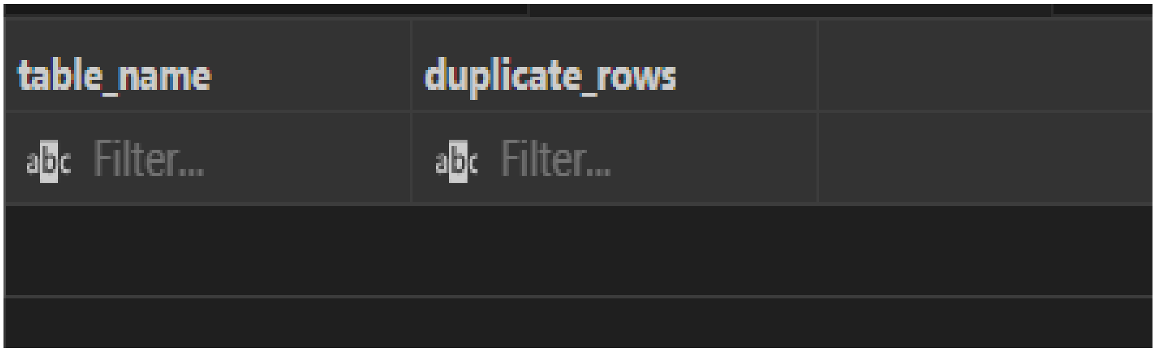
Result: No missing values were found in any table.

Duplicate Record Detection

A duplicate check was executed by grouping each table on all its columns and identifying any repeated rows using the `HAVING COUNT(*) > 1` clause. This helped ensure that no redundant records were stored in the system.



```sql
#2. Checking for full duplicate rows across all tables
SELECT 'patients' AS table_name, COUNT(*) AS duplicate_rows
FROM patients
GROUP BY patient_id, first_name, last_name, gender, date_of_birth,
         contact_number, address, registration_date, insurance_provider, insurance_number, email
HAVING COUNT(*) > 1
UNION ALL
SELECT 'doctors', COUNT(*) FROM doctors
GROUP BY doctor_id, first_name, last_name, specialization, phone_number, years_experience, hospital_branch, email
HAVING COUNT(*) > 1
UNION ALL
SELECT 'appointments', COUNT(*) FROM appointments
GROUP BY appointment_id, patient_id, doctor_id, appointment_date, appointment_time, reason_for_visit, status
HAVING COUNT(*) > 1
UNION ALL
SELECT 'treatments', COUNT(*) FROM treatments
GROUP BY treatment_id, appointment_id, treatment_type, description, cost, treatment_date
HAVING COUNT(*) > 1
UNION ALL
SELECT 'billing', COUNT(*) FROM billing
GROUP BY bill_id, patient_id, treatment_id, bill_date, amount, payment_method, payment_status
HAVING COUNT(*) > 1;
```

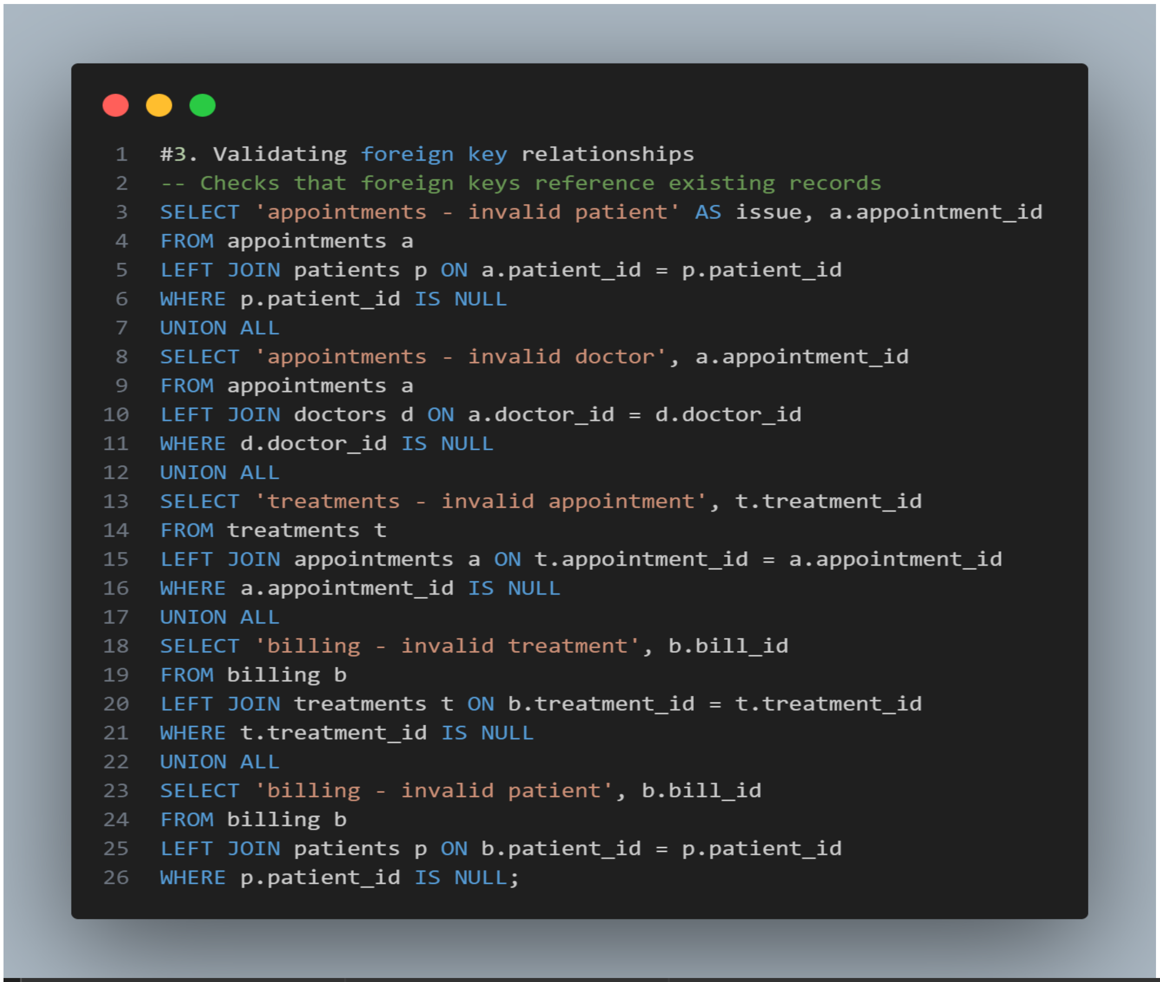


table_name	duplicate_rows

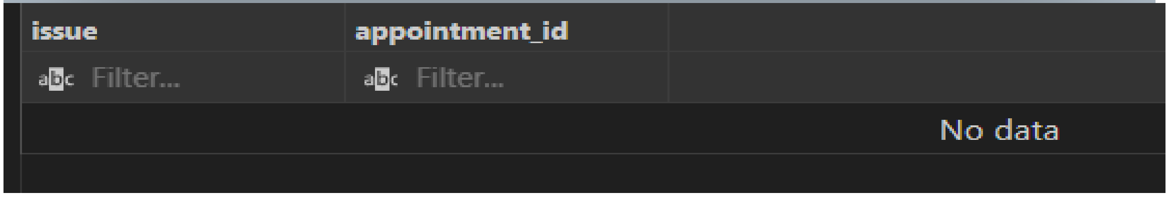
Result: All tables were free from full duplicate rows.

Foreign Key Validation

To maintain referential integrity, foreign key relationships were validated across tables using `LEFT JOIN` checks. This ensured, for example, that every appointment was linked to a valid patient and doctor, and each billing entry referenced a valid treatment and patient.



```sql
#3. Validating foreign key relationships
-- Checks that foreign keys reference existing records
SELECT 'appointments - invalid patient' AS issue, a.appointment_id
FROM appointments a
LEFT JOIN patients p ON a.patient_id = p.patient_id
WHERE p.patient_id IS NULL
UNION ALL
SELECT 'appointments - invalid doctor', a.appointment_id
FROM appointments a
LEFT JOIN doctors d ON a.doctor_id = d.doctor_id
WHERE d.doctor_id IS NULL
UNION ALL
SELECT 'treatments - invalid appointment', t.treatment_id
FROM treatments t
LEFT JOIN appointments a ON t.appointment_id = a.appointment_id
WHERE a.appointment_id IS NULL
UNION ALL
SELECT 'billing - invalid treatment', b.bill_id
FROM billing b
LEFT JOIN treatments t ON b.treatment_id = t.treatment_id
WHERE t.treatment_id IS NULL
UNION ALL
SELECT 'billing - invalid patient', b.bill_id
FROM billing b
LEFT JOIN patients p ON b.patient_id = p.patient_id
WHERE p.patient_id IS NULL;
```



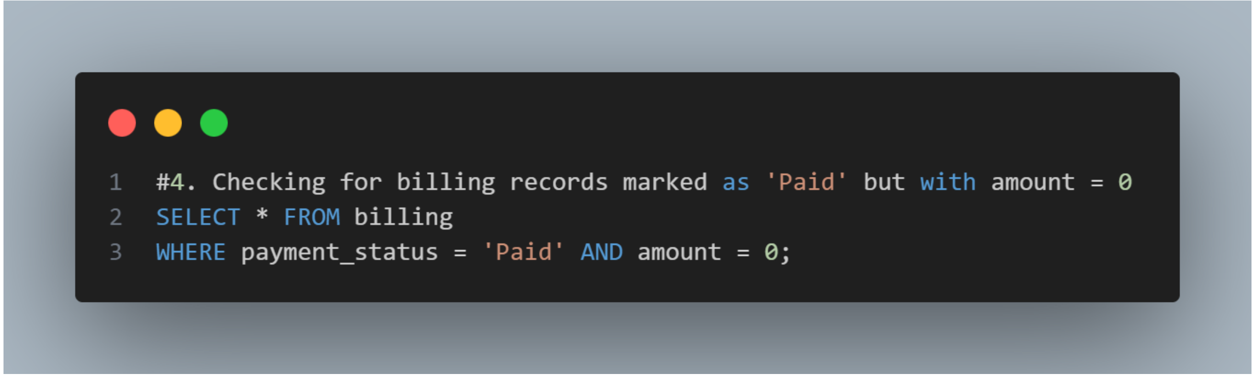
issue	appointment_id

No data

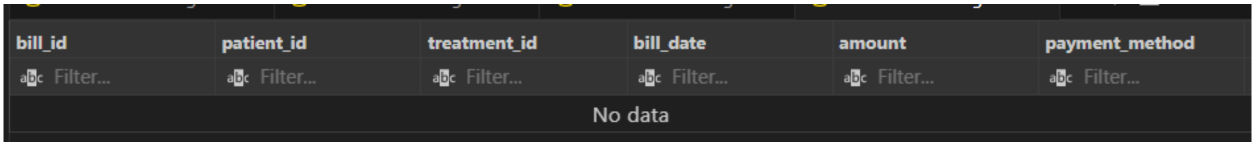
Result: No foreign key mismatches were identified.

Billing Consistency Check

Billing records marked as 'Paid' with an amount of zero were flagged for review, as they could indicate data entry or transactional errors.



```sql
#4. Checking for billing records marked as 'Paid' but with amount = 0
SELECT * FROM billing
WHERE payment_status = 'Paid' AND amount = 0;
```



bill_id	patient_id	treatment_id	bill_date	amount	payment_method
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...

No data

Result: No such inconsistencies were found.

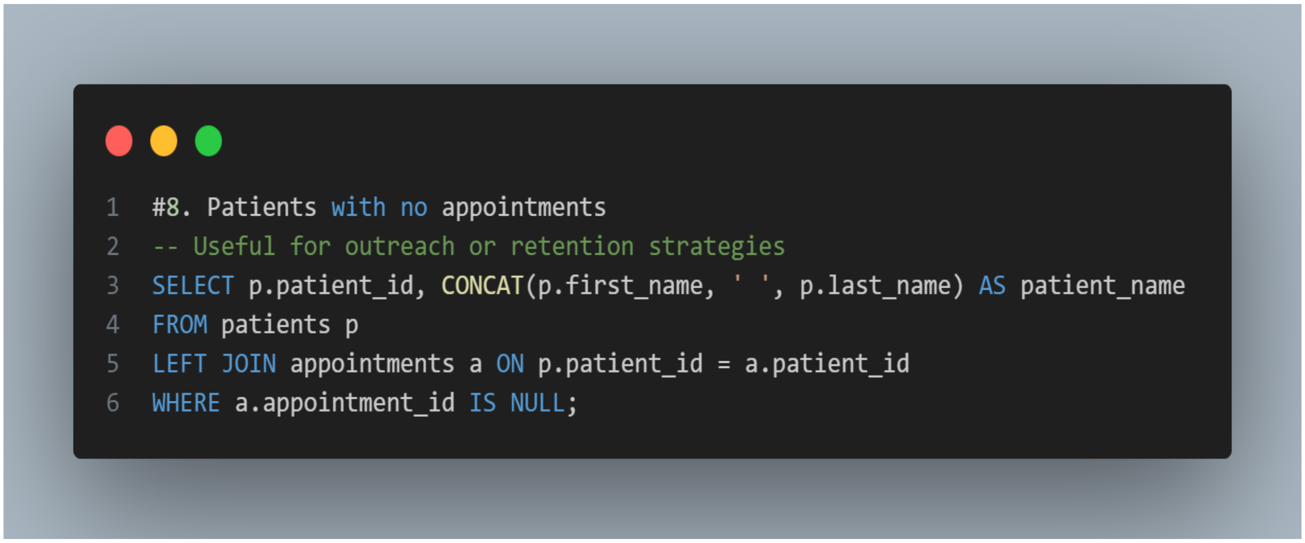
Insights & KPIs

Using SQL, I answered key business questions:

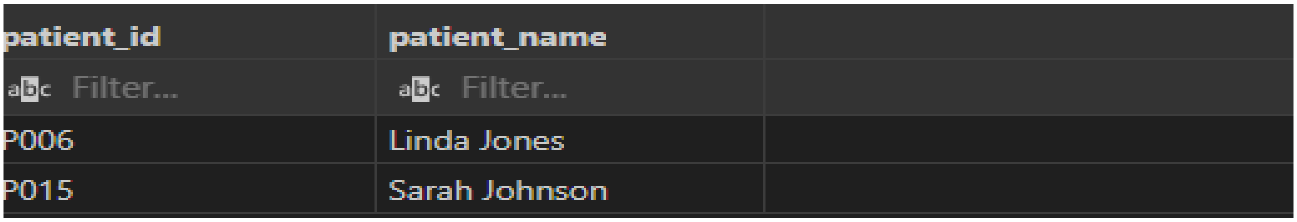
- **Total Patients**: 50
- **Total Appointments**: 200
- **Average Appointments per Patient**: 4.17
- **Total Revenue (Paid)**: 551,000
- **Average Revenue per Patient**: 11,500
- **Failed Payment Rate**: 33%

Other queries revealed:

Patients who never returned after registration (retention issue)



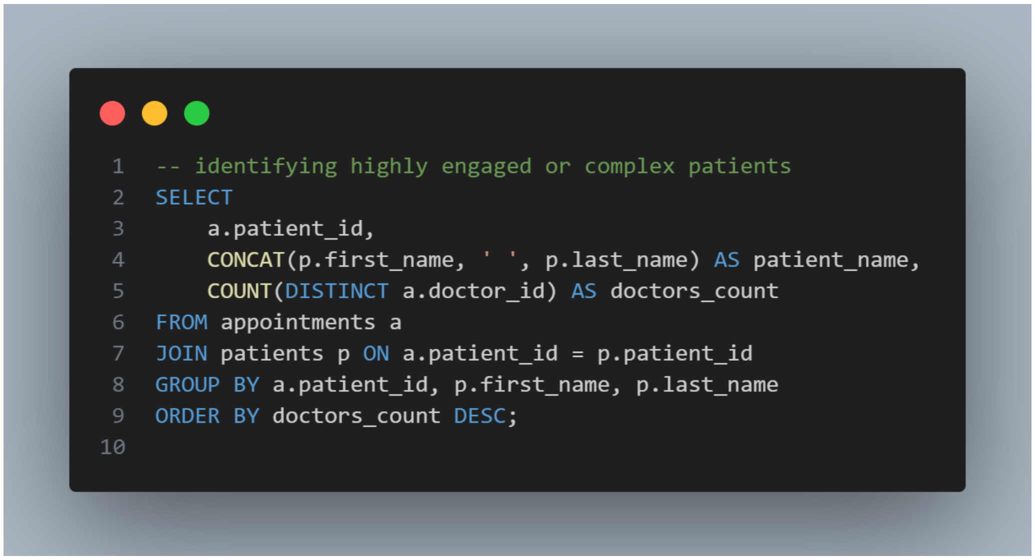
```sql
#8. Patients with no appointments
-- Useful for outreach or retention strategies
SELECT p.patient_id, CONCAT(p.first_name, ' ', p.last_name) AS patient_name
FROM patients p
LEFT JOIN appointments a ON p.patient_id = a.patient_id
WHERE a.appointment_id IS NULL;
```



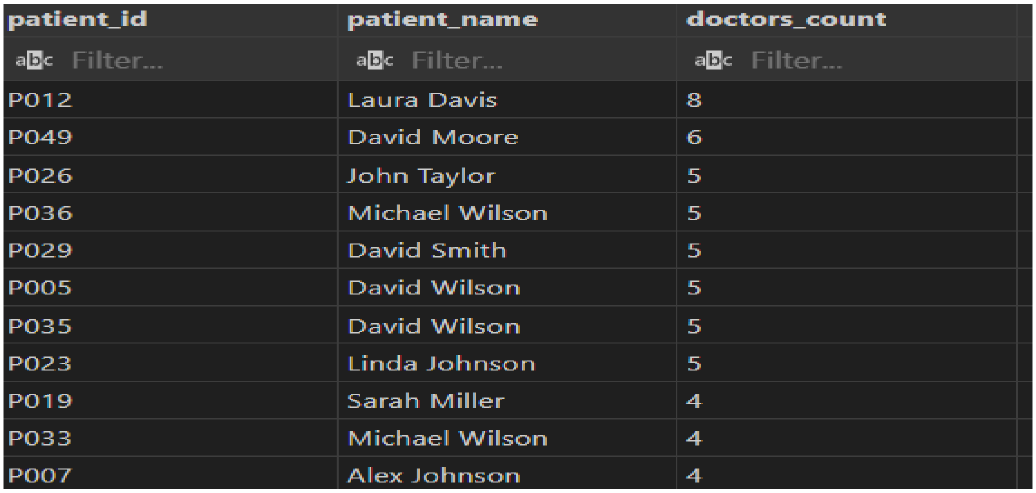
patient_id	patient_name
P006	Linda Jones
P015	Sarah Johnson

This query was designed to identify registered patients who never scheduled an appointment. Highlighting these gaps supports targeted follow up efforts and helps the hospital improve patient engagement and retention.

Top patients who visited multiple doctors (fragmented care or complex needs)



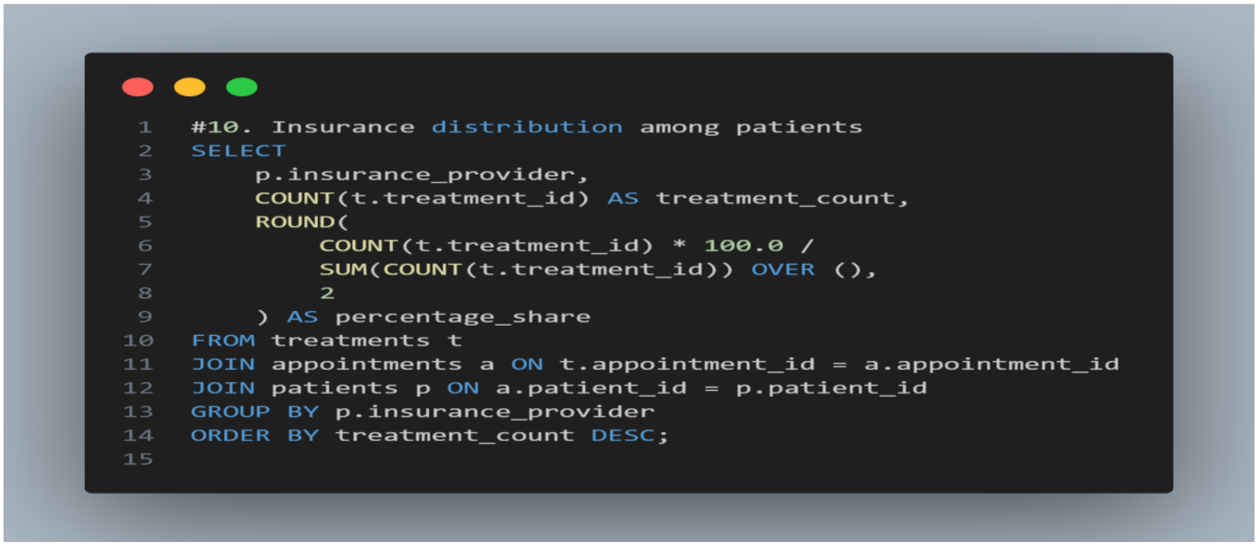
```sql
-- identifying highly engaged or complex patients
SELECT
    a.patient_id,
    CONCAT(p.first_name, ' ', p.last_name) AS patient_name,
    COUNT(DISTINCT a.doctor_id) AS doctors_count
FROM appointments a
JOIN patients p ON a.patient_id = p.patient_id
GROUP BY a.patient_id, p.first_name, p.last_name
ORDER BY doctors_count DESC;

```



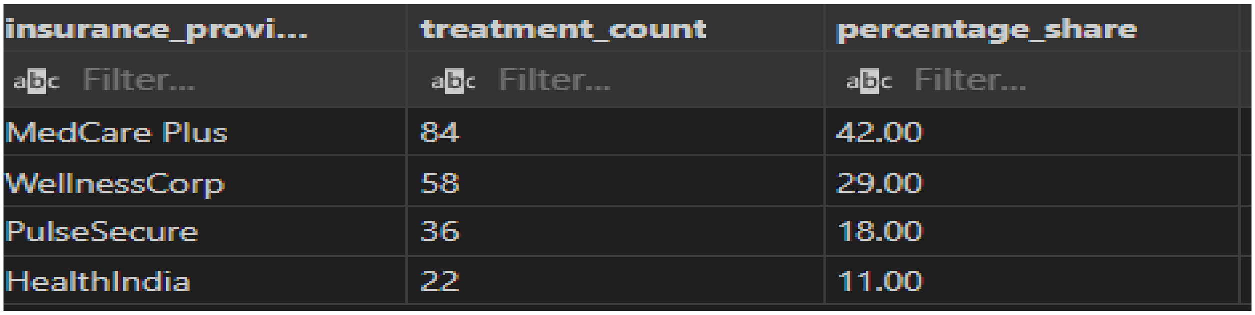
patient_id	patient_name	doctors_count
abc Filter...	abc Filter...	abc Filter...
P012	Laura Davis	8
P049	David Moore	6
P026	John Taylor	5
P036	Michael Wilson	5
P029	David Smith	5
P005	David Wilson	5
P035	David Wilson	5
P023	Linda Johnson	5
P019	Sarah Miller	4
P033	Michael Wilson	4
P007	Alex Johnson	4

This analysis identifies patients who have seen multiple doctors, which may indicate complex medical needs, fragmented care, or inconsistent follow up. For instance, Patient P012 (Laura Davis) consulted with eight different doctors; suggesting the need to review her care pathway. Recognizing such patterns helps improve care coordination and can support targeted case management strategies.

Insurance provider dominance across treatments



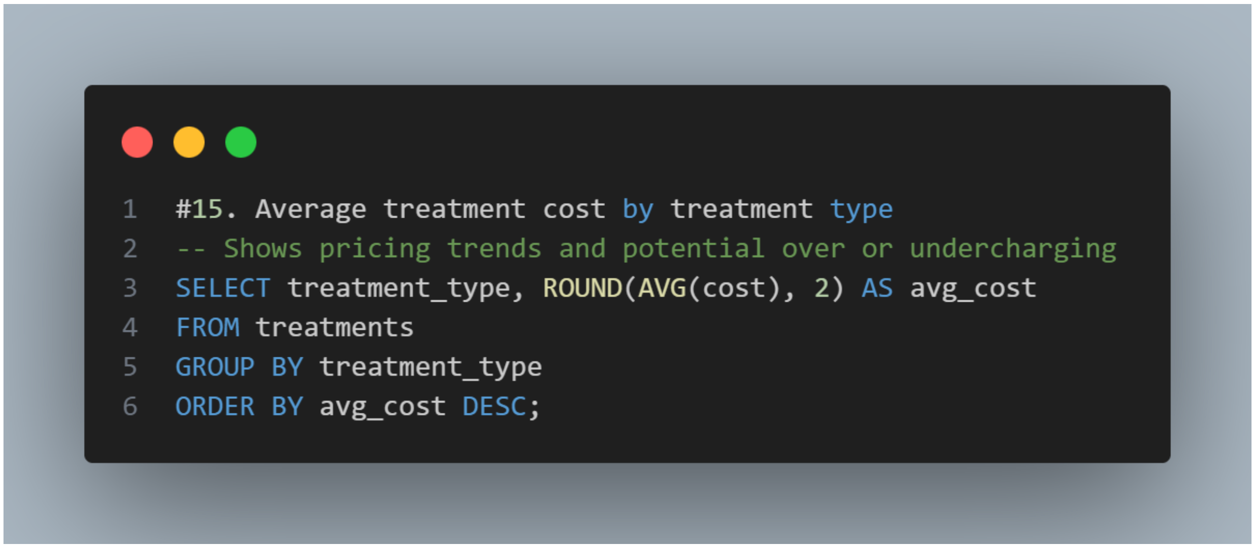
```
#10. Insurance distribution among patients
SELECT
    p.insurance_provider,
    COUNT(t.treatment_id) AS treatment_count,
    ROUND(
        COUNT(t.treatment_id) * 100.0 /
        SUM(COUNT(t.treatment_id)) OVER (),
        2
    ) AS percentage_share
FROM treatments t
JOIN appointments a ON t.appointment_id = a.appointment_id
JOIN patients p ON a.patient_id = p.patient_id
GROUP BY p.insurance_provider
ORDER BY treatment_count DESC;
```



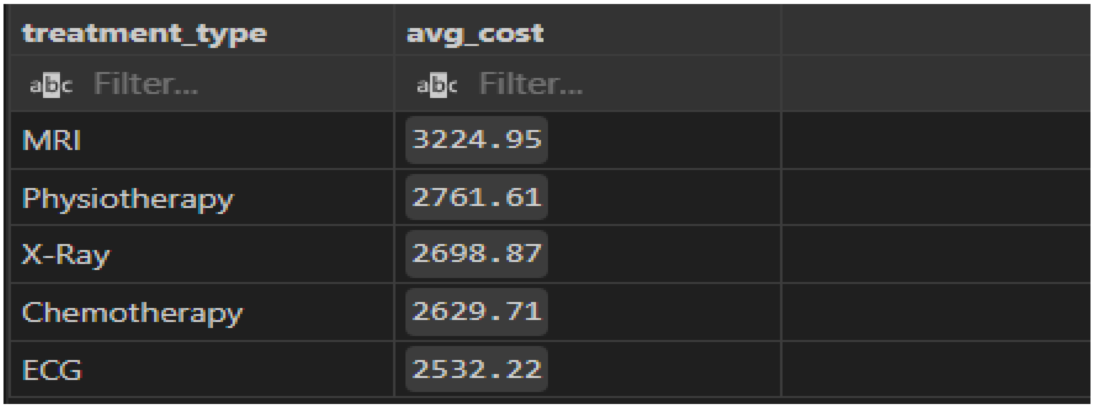
insurance_provi...	treatment_count	percentage_share
MedCare Plus	84	42.00
WellnessCorp	58	29.00
PulseSecure	36	18.00
HealthIndia	22	11.00

This query analyzes treatment volume by insurance provider to understand payer impact on hospital services. MedCare Plus accounts for 42% of all treatments; highlighting its strategic importance for contract negotiations and service planning.

Treatment cost variability and which treatments drove revenue



```sql
#15. Average treatment cost by treatment type
-- Shows pricing trends and potential over or undercharging
SELECT treatment_type, ROUND(AVG(cost), 2) AS avg_cost
FROM treatments
GROUP BY treatment_type
ORDER BY avg_cost DESC;
```



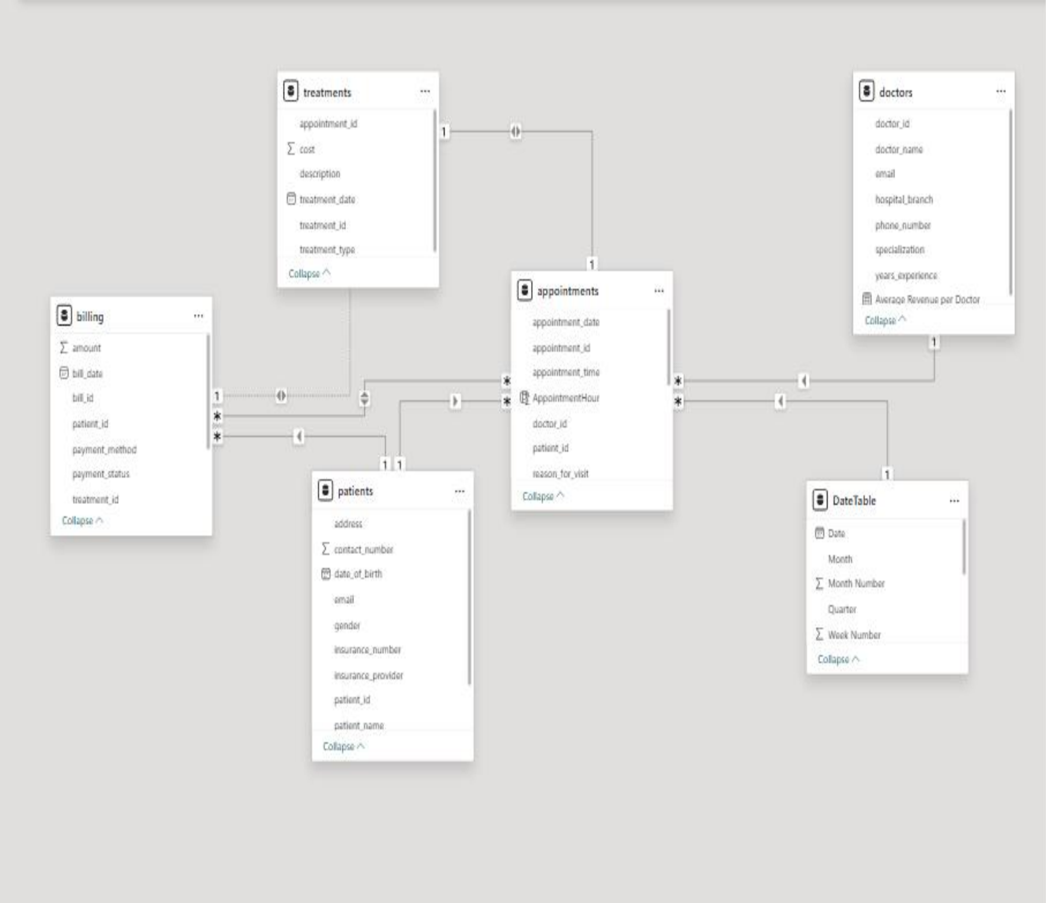
abc Filter...
abc Filter...

treatment_type	avg_cost
MRI	3224.95
Physiotherapy	2761.61
X-Ray	2698.87
Chemotherapy	2629.71
ECG	2532.22

This query calculates the average cost per treatment type to uncover pricing patterns. MRI and Physiotherapy are the most expensive on average, which may warrant a review of resource allocation, pricing strategy, or patient accessibility for these services.

Dashboard Structure in Power BI

After confirming that all tables were clean, I validated foreign key connections between them. I built a **star schema** model with `appointments` and `billing` as central fact tables supported by dimensional patient and doctor information.



The image above showcases the star schema used in the dashboard. Fact tables appointments connected to dimension tables patients, doctors, billing and treatments, enabling efficient filtering and scalable reporting across clinical and financial use cases.

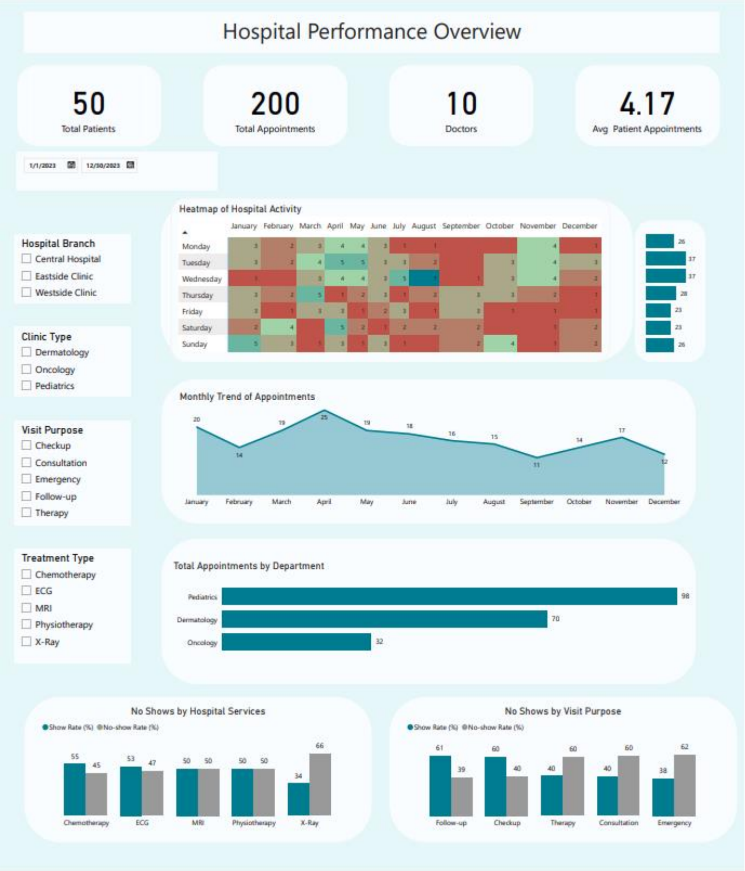
To ensure clarity and usability, the Power BI dashboard was split into two main pages:

Page 1: Clinical Performance & Patient Flow

This page answers the question: *“How are patients using our services and where are we losing efficiency?”*

Visuals:

- KPI Cards: total patients, average appointments, total appointments
- Heatmap: appointments by weekday (Tuesday and Wednesday were busiest)
- Line Graph: monthly appointment trend (peak in April, drop in September)
- Column Charts:
  - Appointments by specialization (Pediatrics had the highest volume)
  - Show vs No-Show rates by treatment type (X-ray 66% no-show)
  - Attendance by visit reason (Emergency and Therapy both ~60% no-show)
- Slicers: by hospital branch, doctor specialization, treatment type, date



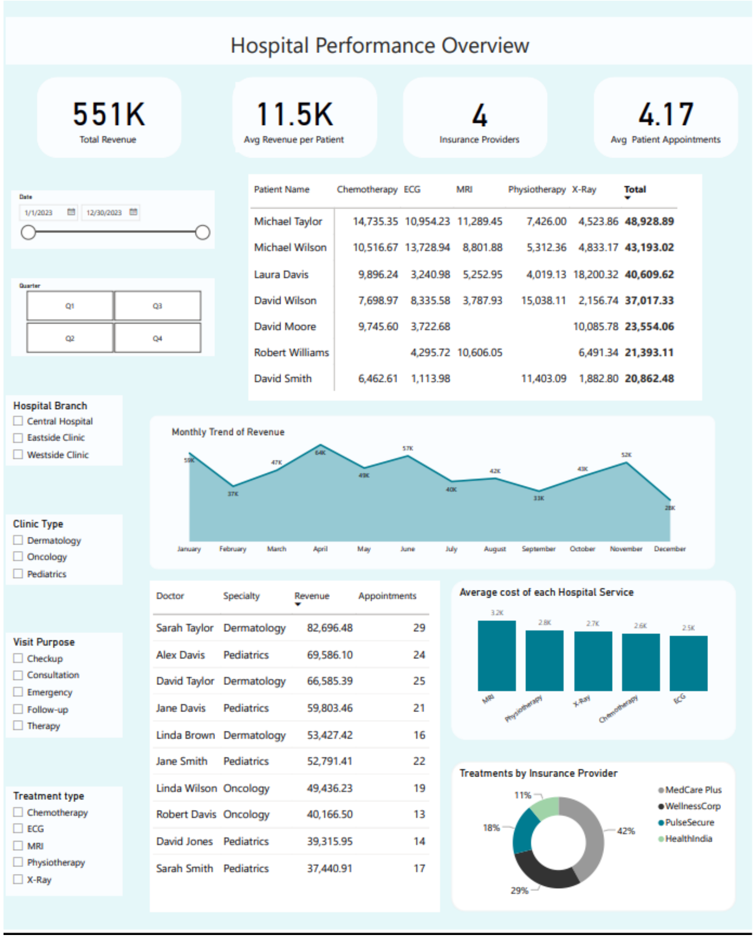
Hospital Overview and Operations Page
This page visualizes patient activity, no-show trends, and departmental efficiency to help identify operational bottlenecks and improve clinical workflow.

Page 2: Financial & Strategic View

This page answers: *"Where is our revenue coming from, and which services, doctors, and branches drive it?"*

Visuals:

- Total and average revenue cards
- Donut Chart: Insurance provider coverage (Medicare Plus covers 40%+)
- Column Chart: revenue by treatment (Chemotherapy is highest earner)
- Line Chart: monthly revenue fluctuations
- Table: Top 7 revenue-contributing patients and services used
- Table: Doctor specialization, appointments, patient volume, total revenue



Hospital Performance Overview

551K
Total Revenue

11.5K
Avg Revenue per Patient

4
Insurance Providers

4.17
Avg Patient Appointments

Date
1/1/2023 12/30/2023

Quarter
Q1 Q3
Q2 Q4

Hospital Branch
□ Central Hospital
□ Eastside Clinic
□ Westside Clinic

Clinic Type
□ Dermatology
□ Oncology
□ Pediatrics

Visit Purpose
□ Checkup
□ Consultation
□ Emergency
□ Follow-up
□ Therapy

Treatment type
□ Chemotherapy
□ ECG
□ MRI
□ Physiotherapy
□ X-Ray

Patient Name	Chemotherapy	ECG	MRI	Physiotherapy	X-Ray	**Total**
Michael Taylor	14,735.35	10,954.23	11,289.45	7,426.00	4,523.86	**48,928.89**
Michael Wilson	10,516.67	13,728.94	8,801.88	5,312.36	4,833.17	**43,193.02**
Laura Davis	9,896.24	3,240.98	5,252.95	4,019.13	18,200.32	**40,609.62**
David Wilson	7,698.97	8,335.58	3,787.93	15,038.11	2,156.74	**37,017.33**
David Moore	9,745.60	3,722.68			10,085.78	**23,554.06**
Robert Williams		4,295.72	10,606.05		6,491.34	**21,393.11**
David Smith	6,462.61	1,113.98		11,403.09	1,882.80	**20,862.48**

Doctor	Specialty	Revenue	Appointments
Sarah Taylor	Dermatology	82,696.48	29
Alex Davis	Pediatrics	69,586.10	24
David Taylor	Dermatology	66,585.39	25
Jane Davis	Pediatrics	59,803.46	21
Linda Brown	Dermatology	53,427.42	16
Jane Smith	Pediatrics	52,791.41	22
Linda Wilson	Oncology	49,436.23	19
Robert Davis	Oncology	40,166.50	13
David Jones	Pediatrics	39,315.95	14
Sarah Smith	Pediatrics	37,440.91	17

Financial Dashboard Page
This view presents financial performance by branch, doctor, and treatment type, enabling strategic revenue planning and resource allocation.

Insights

This analysis uncovered key trends across clinical operations and financial performance:

High No-Show Rates in Critical Services: Departments like **X-ray (66%)** and **Emergency (~60%)** report the highest no-show rates. These are not only operational inefficiencies but potential risks for delayed care.

Patient Care Fragmentation: Several patients saw **3+ doctors**, and one consulted with **8 specialists**. This signals either complex health needs or disjointed care coordination both of which require process improvement and better patient tracking systems.

Midweek Service Overload: The **busiest appointment days are Tuesdays and Wednesdays**, creating potential bottlenecks in scheduling, staff availability, and service delivery. Spreading appointments more evenly could ease staff workload and enhance patient experience.

Insurance Dependence: All patients are insured, with **MedCare Plus alone covering 42%** of treatments. This dominance makes it a critical financial partner.

Revenue Performance Gaps: While **Chemotherapy** generates the most revenue, treatments like **MRI and Physiotherapy** are among the most expensive. This raises questions about pricing alignment, margins, and accessibility.

Branch-Level Opportunity: The **Central branch delivers 41.6% of total revenue**, yet lacks oncology; revealing a strong case for expanding high value services like chemotherapy to that location.
Irregular Monthly Revenue: Revenue shows volatile month over month trends. This may relate to seasonal patient flow, billing delays, or lack of proactive appointment targeting.

Final Recommendations

Reduce No-Shows in Key Departments: Investigate root causes of absenteeism in X-ray and Emergency services. A mix of SMS/Email reminders, pre-visit calls, and workflow audits could boost attendance rates and operational throughput.

Strengthen Care Coordination: Patients seeing 5+ doctors may benefit from care navigation support or dedicated case managers. This will reduce redundancy, improve outcomes, and enhance patient satisfaction.

Redistribute Appointment Load: Encourage off-peak bookings through digital nudges or flexible hours to reduce the midweek bottleneck. This improves staff scheduling and balances demand across the week.

Expand Oncology in High Performing Branches: The Central branch's strong financial return without oncology indicates prime potential for expansion. A feasibility study on service rollout is recommended.

Leverage Insurance Partnerships: Collaborate with MedCare Plus and other insurers on bundled payments or preventive care incentives, as over 40% of patients are concentrated under one payer.

Forecast & Stabilize Revenue Cycles: Address month-to-month revenue swings by creating predictive models for appointment load and implementing consistent billing practices.

Conclusion

This case study demonstrates how healthcare analytics can turn operational complexity into actionable insights. By connecting data across patient experiences, provider performance, and financial results, we can uncover patterns that inform both daily decisions and long term strategy.

From understanding what drives patient engagement to identifying the doctors and services fueling revenue, the analysis offers a full picture view of hospital performance, positioning leadership to be proactive, data driven, and aligned with care outcomes.