Project Title

Healthcare Operations & Patient Analytics Using SQL

Prepared By-

Madhusudan

(Pursuing PGA- Data Science and Analytics

From- Imarticus Learning)

Project Abstract

- This project focuses on analyzing healthcare operations and patient treatment data using SQL.
- By studying hospital performance, treatment outcomes, patient demographics, visit frequency, and cost patterns, the project delivers insights on revenue optimization, patient management, and treatment quality improvement.
- The analysis identifies high-performing hospitals, disease distribution trends, returning patients, treatment success rates, and financial cost segmentation, enabling efficient healthcare decision-making and operational planning.

Introduction

- Healthcare organisations generate massive amounts of patient and treatment data every day.
  However, without structuring and analysing this data, hospitals fail to:
- Recognize disease trends
- Evaluate treatment success and patient recovery
- Measure hospital performance & utilization
- Identify revenue concentration areas
- Detect high-risk or repeat patients
- To bridge this gap, **SQL-based analytics** is used to convert raw tabular data into actionable insights. This project uses three relational datasets: *patients, hospitals, and treatments* — establishing connections among them to explore operational performance, revenue impact, and patient behavior.

Problem Statement

- Healthcare administrators need structured insights into **patient behavior, revenue flow, cost patterns, and treatment results**, but raw medical data is rarely analyzed effectively.
  The lack of visibility into hospital load, recovery efficiency, disease frequency, and returning visitors leads to:
- Unequal workload across hospitals and doctors
- Poor capacity planning
- Inefficient treatment cost regulation
- Unidentified chronic patient cycles
- Revenue leak & resource imbalance
- This project aims to derive **strong analytical conclusions from SQL** to support data-driven healthcare decisions.

Database Model – Tables & Attributes

Patients Table

Column Name	Description
patient_id (PK)	Unique identity of patient
name	Patient name
age	Age of patient
gender	Male/Female
city	Patient residential city

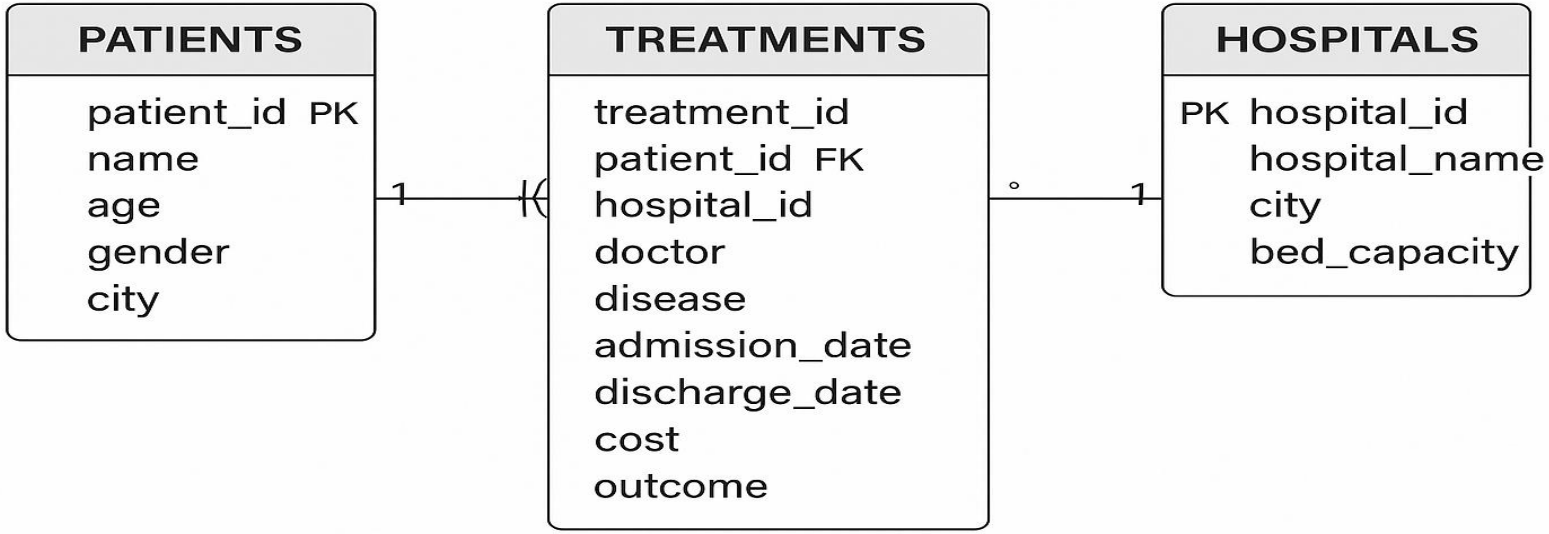
Hospitals Table

Column Name	Description
hospital_id (PK)	Unique identity of hospital
hospital_name	Name of hospital
city	Location of hospital
bed_capacity	Number of available beds

Treatments Table

Column Name	Description
treatment_id (PK)	Unique treatment record
patient_id (FK)	Linked to Patients table
hospital_id (FK)	Linked to Hospitals table
doctor	Doctor handling treatment
disease	Disease diagnosed
admission_date	Admission timestamp
discharge_date	Discharge timestamp
cost	Billing amount of treatment
outcome	Recovered/Death/Referred etc.

Database Model Diagram (Schema Layout)



PATIENTS

patient_id PK
name
age
gender
city

TREATMENTS

treatment_id
patient_id FK
hospital_id
doctor
disease
admission_date
discharge_date
cost
outcome

HOSPITALS

PK hospital_id
hospital_name
city
bed_capacity

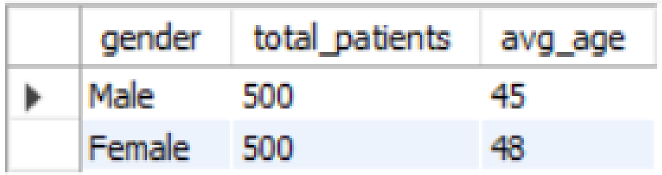
Case Study 1- Patient Demographics & Target Population Study

Problem: Identify who the hospital is serving the most (age + gender)

Query-

```
SELECT gender,
       COUNT(*) AS total_patients,
       ROUND(AVG(age),0) AS avg_age
FROM patients
GROUP BY gender;
```

Output



gender	total_patients	avg_age
Male	500	45
Female	500	48

Insight:
Age-related or chronic care demand likely to be high.

Recommendation:
Invest in cardiology/diabetes units, geriatric care OPD, health monitoring programs.

Case Study 2- Patient Distribution Across Cities

Problem: Determine which cities generate the highest patient inflow.

Query

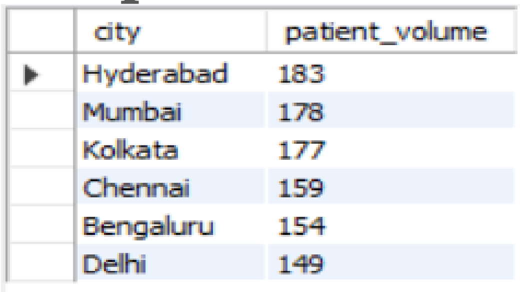
```
SELECT city, COUNT(*) AS patient_volume

FROM patients

GROUP BY city

ORDER BY patient_volume DESC;
```

Output



city	patient_volume
Hyderabad	183
Mumbai	178
Kolkata	177
Chennai	159
Bengaluru	154
Delhi	149

Insight:
Cities with maximum patient count represent core service zones.

Recommendation:
Set up outreach clinics, ambulance network & pharmacy partners in top 2–3 cities.

Case Study 3- Most Common & Recurring Diseases

Problem: Identify diseases with the highest hospitalisation count.

Query

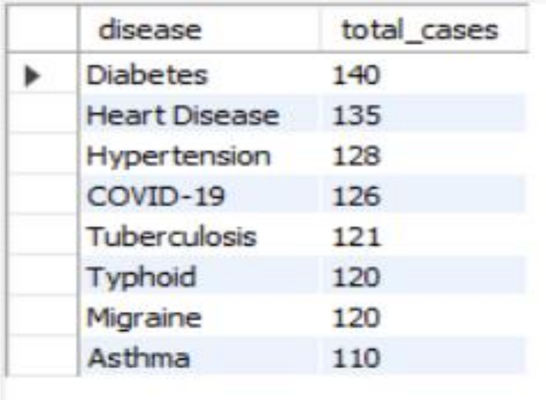
```
SELECT disease, COUNT(*) AS total_cases

FROM treatments

GROUP BY disease

ORDER BY total_cases DESC;
```

Output



	disease	total_cases
▶	Diabetes	140
	Heart Disease	135
	Hypertension	128
	COVID-19	126
	Tuberculosis	121
	Typhoid	120
	Migraine	120
	Asthma	110

Insight:
High-frequency diseases (Typhoid, Diabetes, TB…) = primary burden on infrastructure.

Recommendation:
Conduct preventive awareness drives, vaccination & early screening for top 3 diseases.

Case Study 4-Treatment Cost Analysis (Disease-wise Cost Drivers)

Problem: Identify which diseases require the greatest financial resources.

Query

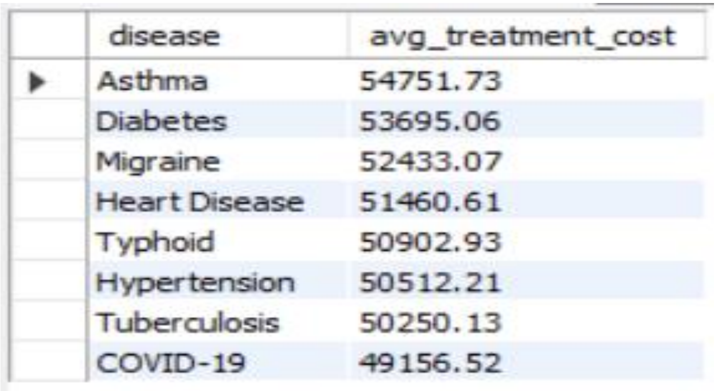
```
SELECT disease,
ROUND(AVG(cost),2) AS
avg_treatment_cost

FROM treatments

GROUP BY disease

ORDER BY avg_treatment_cost
DESC;
```

Output



	disease	avg_treatment_cost
▶	Asthma	54751.73
	Diabetes	53695.06
	Migraine	52433.07
	Heart Disease	51460.61
	Typhoid	50902.93
	Hypertension	50512.21
	Tuberculosis	50250.13
	COVID-19	49156.52

Insight:
High-cost diseases are financial pressure points.

Recommendation:
Introduce insurance partnerships, subsidy plans, and package-based billing.

Case Study 5-Doctor Workload & Patient Load Distribution

Problem: Find whether patient allocation is balanced across doctors.

Query

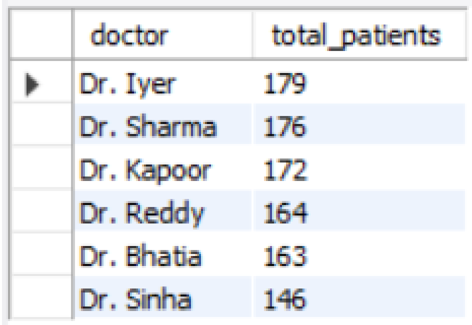
```
SELECT doctor, COUNT(*) AS total_patients

FROM treatments

GROUP BY doctor

ORDER BY total_patients DESC;
```

Output



doctor	total_patients
Dr. Iyer	179
Dr. Sharma	176
Dr. Kapoor	172
Dr. Reddy	164
Dr. Bhatia	163
Dr. Sinha	146

- **Insight:**
  If 1–2 doctors handle most patients, burnout & longer wait times may occur.
- **Recommendation:**
  Workload balancing + junior specialists training program.

Some Advanced Case Studies

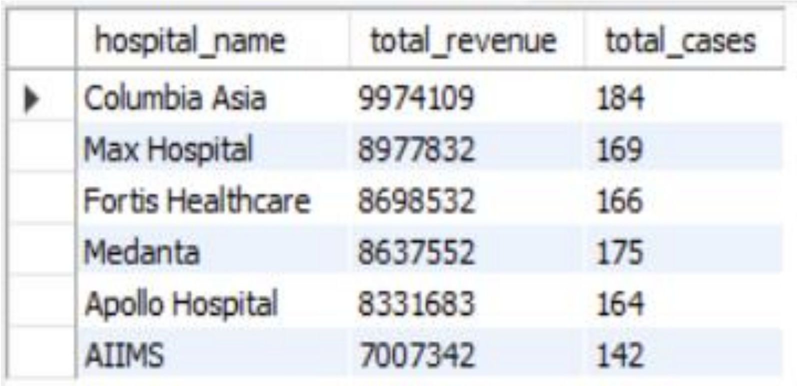
Case Study 6- Revenue Contribution by Hospital

Problem: Identify which hospitals generate the highest revenue.

Query

```
SELECT
    h.hospital_name,
    SUM(t.cost) AS total_revenue,
    COUNT(t.treatment_id) AS total_cases
FROM hospitals h
JOIN treatments t USING(hospital_id)
GROUP BY h.hospital_name
ORDER BY total_revenue DESC;
```

Output



hospital_name	total_revenue	total_cases
Columbia Asia	9974109	184
Max Hospital	8977832	169
Fortis Healthcare	8698532	166
Medanta	8637552	175
Apollo Hospital	8331683	164
AIIMS	7007342	142

Insight

Shows total revenue + patient volume per hospital.

A hospital may have lower cases but higher revenue per case (specialty, ICU cases).

Recommendation

Invest more resources & marketing support into top-earning hospitals.

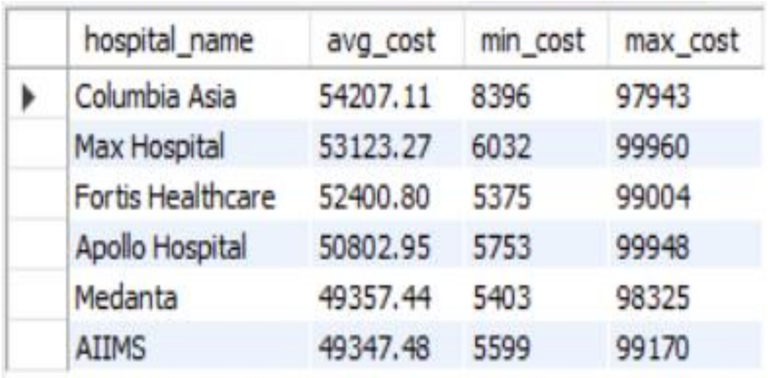
Case Study 7-Average Treatment Cost by Hospital

Problem: Compare pricing across hospitals.

Query

```
SELECT
   h.hospital_name,
   ROUND(AVG(t.cost),2) AS avg_cost,
   MIN(t.cost) AS min_cost,
   MAX(t.cost) AS max_cost
FROM hospitals h
JOIN treatments t USING(hospital_id)
GROUP BY h.hospital_name
ORDER BY avg_cost DESC;
```

Output



	hospital_name	avg_cost	min_cost	max_cost
▶	Columbia Asia	54207.11	8396	97943
	Max Hospital	53123.27	6032	99960
	Fortis Healthcare	52400.80	5375	99004
	Apollo Hospital	50802.95	5753	99948
	Medanta	49357.44	5403	98325
	AIIMS	49347.48	5599	99170

Insight

Shows hospitals with highest charging patterns.

A large cost gap could indicate inconsistent pricing models.

Recommendation

Standardize billing packages & negotiate insurance support for expensive treatments.

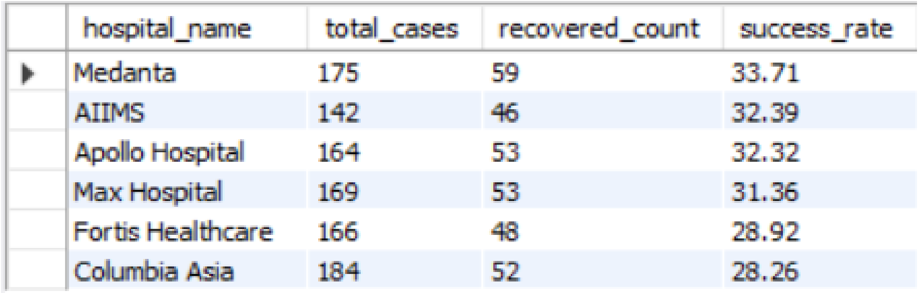
Case Study 8 -Treatment Success Rate (Outcome Evaluation)

Problem: Measure healthcare effectiveness across hospitals.

Query

```
SELECT
   h.hospital_name,
   COUNT(*) AS total_cases,
   SUM(CASE WHEN outcome = 'Recovered' THEN 1 ELSE 0 END) AS recovered_count,
   ROUND(
      SUM(CASE WHEN outcome='Recovered' THEN 1 ELSE 0 END)*100.0/COUNT(*),
      2
   ) AS success_rate
FROM treatments t
JOIN hospitals h USING(hospital_id)
GROUP BY h.hospital_name
ORDER BY success_rate DESC;
```

Output



	hospital_name	total_cases	recovered_count	success_rate
▶	Medanta	175	59	33.71
	AIIMS	142	46	32.39
	Apollo Hospital	164	53	32.32
	Max Hospital	169	53	31.36
	Fortis Healthcare	166	48	28.92
	Columbia Asia	184	52	28.26

Insight

Shows which hospitals deliver better outcomes.

Low recovery hospitals may require protocol improvements.

Recommendation

Share best clinical practices from high-success hospitals with weaker ones.

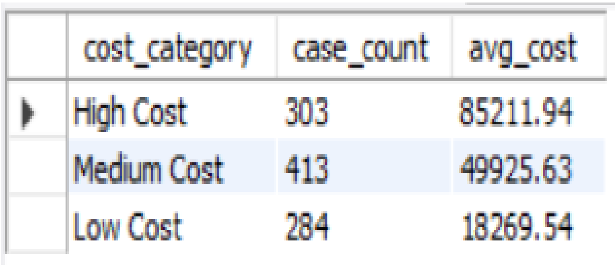
Case Study 9- Cost Distribution Category – Low / Mid / High Cost Cases

Problem: Analyse treatment cost distribution across the dataset.

Query

```
SELECT
    CASE
        WHEN cost < 30000 THEN 'Low Cost'
        WHEN cost BETWEEN 30000 AND 70000 THEN 'Medium Cost'
        ELSE 'High Cost'
    END AS cost_category,
    COUNT(*) AS case_count,
    ROUND(AVG(cost),2) AS avg_cost
FROM treatments
GROUP BY cost_category
ORDER BY avg_cost DESC;
```

Output



cost_category	case_count	avg_cost
High Cost	303	85211.94
Medium Cost	413	49925.63
Low Cost	284	18269.54

Insight

Segmenting cost levels shows what percentage of treatments fall under each bracket.

Helps identify affordability vs high-risk, expensive cases.

Recommendation

Create insurance-linked packages for high-cost treatments.
Promote preventive care for medium-risk categories.

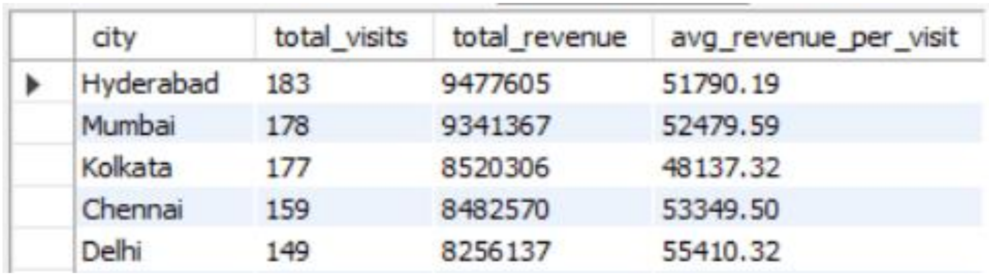
Case Study 10-City-wise Revenue & Patient Volume

Problem - Which **cities** bring the highest **revenue and patient traffic** to the hospital network?

Query

```
SELECT
   p.city,
   COUNT(t.treatment_id) AS total_visits,
   SUM(t.cost) AS total_revenue,
   ROUND(AVG(t.cost), 2) AS avg_revenue_per_visit
FROM patients p
JOIN treatments t USING (patient_id)
GROUP BY p.city
ORDER BY total_revenue DESC;
```

Output



city	total_visits	total_revenue	avg_revenue_per_visit
Hyderabad	183	9477605	51790.19
Mumbai	178	9341367	52479.59
Kolkata	177	8520306	48137.32
Chennai	159	8482570	53349.50
Delhi	149	8256137	55410.32

Insights (example)

2–3 cities may contribute **majority of revenue & visits**.

Some cities might show **fewer visits but higher average revenue**, indicating costlier treatments.

Business recommendation

Focus **marketing, outreach camps, partner clinics, and ambulance network** in high-revenue cities.

For high-volume but low-revenue cities, design **affordable packages** to increase accessibility while preserving margins.

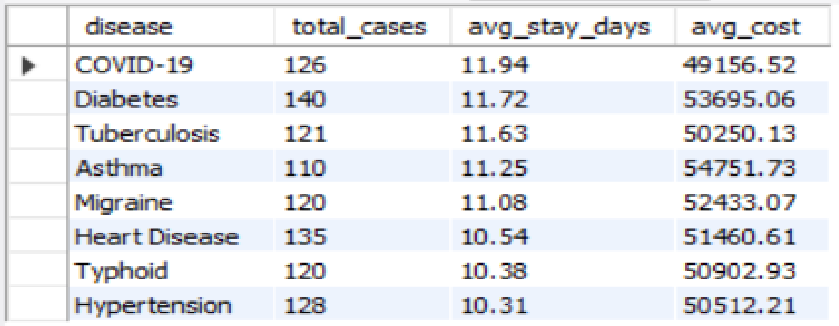
Case Study 11-Average Length of Stay (LOS) by Disease

Problem-Which **diseases** keep patients admitted for longer, increasing bed occupancy and cost?

Query

```
SELECT
    t.disease,
    COUNT(*) AS total_cases,
    ROUND(AVG(DATEDIFF(t.discharge_date, t.admission_date)), 2) AS avg_stay_days,
    ROUND(AVG(t.cost), 2) AS avg_cost
FROM treatments t
WHERE t.discharge_date IS NOT NULL
GROUP BY t.disease
ORDER BY avg_stay_days DESC;
```

Output



disease	total_cases	avg_stay_days	avg_cost
COVID-19	126	11.94	49156.52
Diabetes	140	11.72	53695.06
Tuberculosis	121	11.63	50250.13
Asthma	110	11.25	54751.73
Migraine	120	11.08	52433.07
Heart Disease	135	10.54	51460.61
Typhoid	120	10.38	50902.93
Hypertension	128	10.31	50512.21

Insights (example)

Diseases with **long LOS and high cost** are major drivers of bed occupancy and expenses.

Some diseases may have **short stays but high cost** (e.g., surgeries / ICU).

Business recommendation

Prioritize **care pathways and early discharge protocols** for long-stay diseases.

Plan **bed management, ICU allocation, and nursing staff** based on these high-LOS conditions.

Conclusion

- The SQL-based Healthcare Operations & Patient Analytics project successfully transforms raw medical data into measurable business insights.
- Through revenue evaluation, patient flow tracking, treatment outcome analysis, and high-value hospital identification, the study provides decision support for improving hospital services, resource efficiency, cost management, and clinical quality.
- This framework can be extended into dashboards (Power BI/Tableau), machine-learning-based risk prediction, and real-time hospitalisation trends, proving SQL as a strong foundation for data-driven healthcare strategy.

Thank You