Hospital Patient Analytics Using SQL

This project analyzes a hospital dataset with over 7500+ records, applying intermediate to advanced SQL concepts to extract meaningful insights related to patients, diagnoses, treatments, and bed occupancy.

❖ Dataset Overview

Table	Description	Rows
patients	Patient demographics and stay info	1500
diagnoses	Medical diagnoses with severity	1500
treatments	Treatment type, duration, cost	3045
beds	Bed type and occupancy info	1500

➢ Sample SQL Queries & Insights

1. Top 5 Cities by Average Treatment Cost

⍰ Objective: Identify cities with the highest average treatment cost.

- SQL Query:

```
SELECT p.city, ROUND(AVG(t.cost), 2) AS avg_cost
FROM patients p
JOIN treatments t ON p.patient_id = t.patient_id
GROUP BY p.city
ORDER BY avg_cost DESC
LIMIT 5;
```

⍰ Insight: These cities generate the most revenue per treatment.

2. ICU Patients with High Severity Diagnoses

⍰ Objective: Count patients in ICU beds with high severity diagnoses.

- SQL Query:

```
SELECT COUNT(DISTINCT p.patient_id) AS high_severity_icu_cases
FROM patients p
JOIN beds b ON p.patient_id = b.patient_id
JOIN diagnoses d ON p.patient_id = d.patient_id
WHERE b.bed_type = 'ICU' AND d.severity_level = 'High';
```

□ Insight: Highlights critical patient load and ICU resource pressure.

3. Classify Patients by Stay Duration

□ Objective: Use CASE to classify patients based on their hospital stay length.

- SQL Query:

```
SELECT patient_id,
    DATEDIFF(discharge_date, admit_date) AS stay_days,
    CASE
      WHEN DATEDIFF(discharge_date, admit_date) <= 3 THEN 'Short Stay'
      WHEN DATEDIFF(discharge_date, admit_date) <= 7 THEN 'Medium Stay'
      ELSE 'Long Stay'
    END AS stay_category
FROM patients;
```

□ Insight: Useful for operational planning and patient management.

4. Doctors Ranked by Revenue Generated

□ Objective: Rank doctors based on total treatment cost handled.

- SQL Query:

```
SELECT doctor_id,
    SUM(cost) AS total_revenue,
    RANK() OVER (ORDER BY SUM(cost) DESC) AS rank
FROM treatments
GROUP BY doctor_id;
```

□ Insight: Finds top-performing doctors by revenue contribution.

5. Patients With Above-Average Treatment Cost

□ Objective: List patients whose total treatment cost is above the dataset average.

- SQL Query:

```
SELECT t.patient_id, SUM(t.cost) AS total_cost
FROM treatments t
GROUP BY t.patient_id
HAVING SUM(t.cost) > (SELECT AVG(cost) FROM treatments);
```

📊 Insight: Identifies high-value patients for specialized services.

6. Treatment Cost Trend by Month

 Objective: Analyze monthly treatment cost trends.

- SQL Query:

```
SELECT
   DATE_FORMAT(start_date, '%Y-%m') AS month,
   ROUND(SUM(cost), 2) AS total_monthly_cost
FROM
   treatments
GROUP BY
   month
ORDER BY
   month;
```

 Insight: Shows seasonal trends in treatment costs for budgeting and planning.

7. Average Stay by Bed Type

 Objective: Evaluate how long patients stay by bed type.

- SQL Query:

```
SELECT
   bed_type,
   ROUND(AVG(DATEDIFF(occupancy_end, occupancy_start)), 2) AS avg_stay_days
FROM
   beds
GROUP BY
   bed_type;
```

 Insight: Helps understand resource turnover for each bed type.

8. Most Common Diagnoses by Age Group

 Objective: Identify common diagnoses in different age groups.

- SQL Query:

```
SELECT
   CASE
      WHEN p.age BETWEEN 0 AND 18 THEN '0-18'
      WHEN p.age BETWEEN 19 AND 40 THEN '19-40'
      WHEN p.age BETWEEN 41 AND 60 THEN '41-60'
      ELSE '60+'
   END AS age_group,
   d.diagnosis,
   COUNT(*) AS total_cases
FROM
    patients p
```

```
JOIN
   diagnoses d ON p.patient_id = d.patient_id
GROUP BY
   age_group, d.diagnosis
ORDER BY
   age_group, total_cases DESC;
```

⍰ Insight: Useful for designing age-specific health programs.

9. Rank Doctors by Revenue (Window Function)

⍰ Objective: Find top-performing doctors by total treatment revenue.

- SQL Query:

```
SELECT
   t.doctor_id,
   ROUND(SUM(t.cost), 2) AS total_revenue,
   RANK() OVER (ORDER BY SUM(t.cost) DESC) AS revenue_rank
FROM
   treatments t
GROUP BY
   t.doctor_id;
```

⍰ Insight: Identifies highest earning doctors—useful for incentives and management.

10. Patients Spending More Than Their City's Average

⍰ Objective: Identify outlier patients with higher-than-average treatment cost within their city.

- SQL Query:

```
SELECT
   p.patient_id,
   p.city,
   t.cost
FROM
   patients p
JOIN
   treatments t ON p.patient_id = t.patient_id
WHERE
   t.cost > (
      SELECT
         AVG(t2.cost)
      FROM
         patients p2
       JOIN
         treatments t2 ON p2.patient_id = t2.patient_id
      WHERE
```

```
        p2.city = p.city
    );
```

□ Insight: Useful to flag premium patients or potential billing anomalies.