Healthcare Database Analytics

Improving Hospital Operations and Patient Care Using SQL

1. Project Overview

The **Healthcare Analytics** project focuses on designing a relational healthcare database and performing structured SQL analysis to extract insights related to patient demographics, hospital utilization, treatment costs, and operational performance. The project simulates a real-world hospital information system by integrating patient, hospital, admission, and treatment data.

The primary objective is to demonstrate how **SQL-based analytics** can support healthcare decision-making by improving operational efficiency, optimizing resource utilization, and enhancing patient care outcomes.

2. Database Design & Architecture

A relational database named **HealthcareDB** was created using SQL, consisting of the following core tables:

Tables Created

- **PATIENTS** – Stores patient demographics (age, gender, address)
- **HOSPITALS** – Stores hospital details (location, capacity)
- **ADMISSIONS** – Tracks hospital admissions, discharge dates, and admission reasons
- **TREATMENTS** – Records medical procedures, costs, and outcomes

Key Design Features

- Primary and foreign key constraints to maintain data integrity
- One-to-many relationships between:
    - Patients → Admissions
    - Hospitals → Admissions
    - Admissions → Treatments
- Proper data types for dates, costs, and categorical attributes

This schema supports scalable healthcare analytics and mimics real-world hospital databases.

3. Data Population

Sample data was inserted into all tables to simulate hospital operations, including:

- 10 patients with varied demographics
- 5 hospitals across different locations
- Multiple admissions with discharge tracking
- Treatments with procedure costs and outcomes

4. SQL Analytics Performed

4.1 Patient Demographics Analysis

- Calculated patient count and average age grouped by gender
- Helped identify demographic distribution of hospital patients

4.2 Hospital Utilization

- Identified hospitals with the highest number of admissions
- Highlighted potential capacity and resource demand issues

4.3 Treatment Cost Analysis

- Computed total treatment costs per hospital
- Identified high-revenue hospitals and cost-intensive care areas

4.4 Length of Stay Analysis

- Calculated average patient length of stay by hospital
- Identified extended hospital stays contributing to higher costs

4.5 Advanced Filtering & Business Queries

- Patients staying longer than 7 days
- Treatments performed frequently across hospitals
- Combined admission and treatment data for complete patient histories
- Admission reason analysis (e.g., surgery vs therapy)

5. Advanced SQL Techniques Used

Subqueries

- Identified hospitals with the highest average treatment cost

Views

- Created **HospitalPerformance** view to summarize:
    - Number of admissions
    - Average length of stay

- Total revenue per hospital

Window Functions

- RANK() to rank hospitals by total revenue
- DENSE_RANK() to rank treatments by frequency

These techniques demonstrate advanced SQL skills used in real-world analytics and reporting.

6. Code

```
-- Creating a database named HealthcareDB.

CREATE DATABASE HealthcareDB



-- creating tables within healthcaredb

USE HealthcareDB



CREATE TABLE PATIENTS(

PatientID INT AUTO_INCREMENT PRIMARY KEY,

FullName VARCHAR(20) NOT NULL,

Age INT,

Gender VARCHAR(20),

Address VARCHAR(200)

);

ALTER TABLE PATIENTS

CHANGE Gender Gender VARCHAR(20)

CREATE TABLE HOSPITALS(

HospitalID INT AUTO_INCREMENT PRIMARY KEY,

HospitalName VARCHAR(100) NOT NULL,

Location VARCHAR(255),
```

```
Capacity INT

)


CREATE TABLE ADMISSIONS(

AdmissionID INT AUTO_INCREMENT PRIMARY KEY,

PatientID INT ,

HospitalID INT ,

AdmissionDate DATE NOT NULL,

DischargeDate DATE ,

ReasonForAdmission VARCHAR(200),

CONSTRAINT fk_patients FOREIGN KEY (PatientID) REFERENCES PATIENTS(PatientID),

CONSTRAINT fk_hospitals FOREIGN KEY (HospitalID) REFERENCES
HOSPITALS(HospitalID)

)


CREATE TABLE TREATMENTS(

TreatmentID INT AUTO_INCREMENT PRIMARY KEY,

AdmissionID INT,

ProcedureName VARCHAR(200),

Cost DECIMAL,

Outcome VARCHAR(100),

FOREIGN KEY (AdmissionID) REFERENCES ADMISSIONS(AdmissionID)

)


-- Insert data into Patients table
```

```
INSERT INTO Patients (FullName, Age, Gender, Address) VALUES

('John Doe', 45, 'Male', '123 Elm Street'),

('Jane Smith', 34, 'Female', '456 Oak Avenue'),

('Sam Brown', 29, 'Male', '789 Pine Road'),

('Lisa White', 52, 'Female', '321 Maple Lane'),

('Tom Green', 67, 'Male', '654 Birch Blvd'),

('Alice Johnson', 40, 'Female', '987 Willow Court'),

('Robert Black', 60, 'Male', '564 Cypress Road'),

('Emily Davis', 25, 'Female', '321 Cedar Avenue'),

('Michael Scott', 50, 'Male', '742 Birch Lane'),

('Sarah Taylor', 33, 'Female', '159 Spruce Drive');


-- Insert data into Hospitals table

INSERT INTO Hospitals (HospitalName, Location, Capacity) VALUES

('General Hospital', 'New York', 500),

('City Clinic', 'Los Angeles', 200),

('Central Medical Center', 'Chicago', 300),

('Regional Health Facility', 'Houston', 150),

('Sunrise Hospital', 'Phoenix', 400);


-- Insert data into Admissions table

INSERT INTO Admissions (PatientID, HospitalID, AdmissionDate, DischargeDate,
ReasonForAdmission) VALUES

(1, 1, '2024-11-01', '2024-11-05', 'Surgery'),

(2, 2, '2024-11-03', '2024-11-08', 'Therapy'),
```

(3, 3, '2024-11-10', '2024-11-15', 'Accident'),

(4, 4, '2024-11-12', '2024-11-19', 'Routine Checkup'),

(5, 5, '2024-12-01', '2024-12-08', 'Infection'),

(6, 1, '2024-12-01', NULL, 'Surgery'),

(7, 2, '2024-12-02', '2024-12-05', 'Fracture Repair'),

(8, 3, '2024-12-03', NULL, 'Chronic Illness'),

(9, 4, '2024-12-03', '2024-12-18', 'Therapy'),

(10, 5, '2024-12-04', '2024-12-18', 'Infection');

-- Insert data into Treatments table

INSERT INTO Treatments (AdmissionID, ProcedureName, Cost, Outcome) VALUES

(1, 'Appendectomy', 1500.00, 'Successful'),

(2, 'Physical Therapy', 800.00, 'Ongoing'),

(3, 'Fracture Repair', 3000.00, 'Successful'),

(4, 'Blood Test', 200.00, 'Pending'),

(5, 'Antibiotics', 500.00, 'Improved'),

(6, 'Gallbladder Surgery', 4000.00, 'Successful'),

(7, 'X-Ray', 300.00, 'Successful'),

(8, 'Chemotherapy', 5000.00, 'Ongoing'),

(9, 'MRI Scan', 1200.00, 'Pending'),

(10, 'Diabetes Treatment', 700.00, 'Improved');

Healthcare Analytics Queries:

Patient Demographics: Retrieve the number of patients grouped by gender and calculate the average age of patients.

SELECT COUNT(PatientID) AS NumberofPatients, Gender, AVG(AGE) AS AverageAge

FROM PATIENTS

GROUP BY Gender;

```
96    # Healthcare Analytics Queries:
97
98    # Patient Demographics: Retrieve the number of patients grouped by gender and calculate the average age of patients.
99
100   SELECT COUNT(PatientID) AS NumberofPatients, Gender, AVG(AGE) AS AverageAge
101   FROM PATIENTS
102   GROUP BY Gender;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

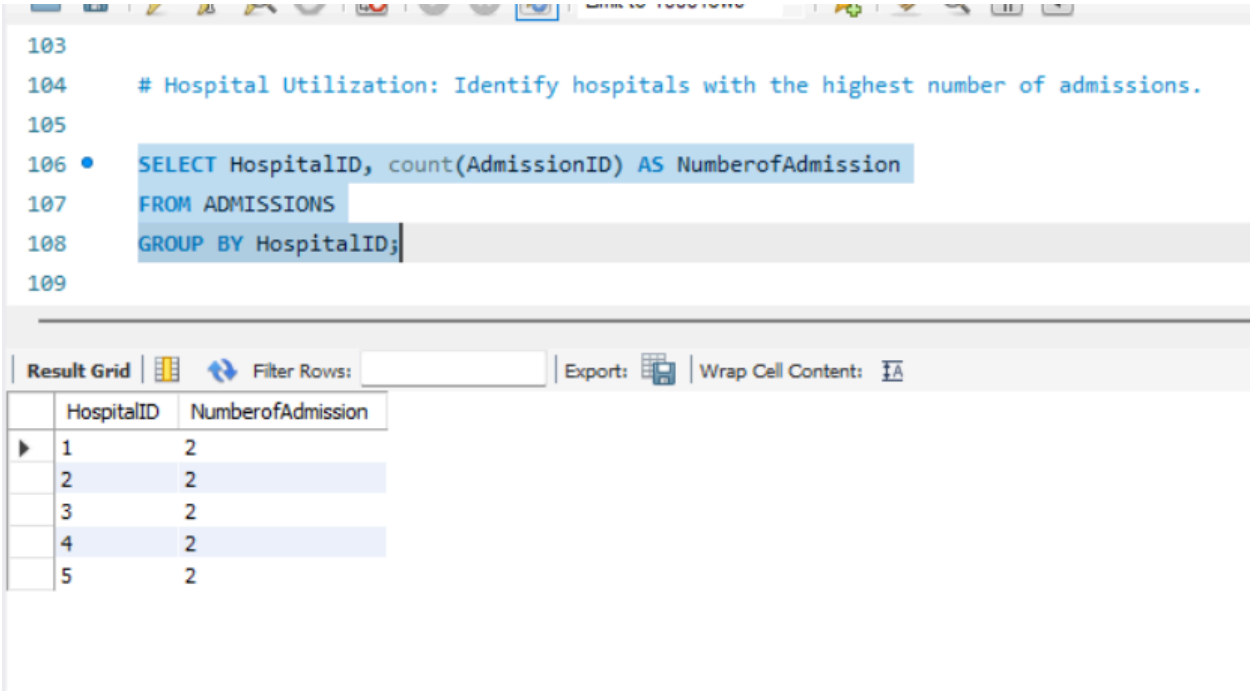
NumberofPatients	Gender	AverageAge
5	Male	50.2000
5	Female	36.8000

Hospital Utilization: Identify hospitals with the highest number of admissions.

SELECT HospitalID, count(AdmissionID) AS NumberofAdmission

FROM ADMISSIONS

GROUP BY HospitalID;



```
103
104     # Hospital Utilization: Identify hospitals with the highest number of admissions.
105
106     SELECT HospitalID, count(AdmissionID) AS NumberofAdmission
107     FROM ADMISSIONS
108     GROUP BY HospitalID;
109
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

HospitalID	NumberofAdmission
1	2
2	2
3	2
4	2
5	2

Treatment Costs: Calculate the total cost of treatments provided at each hospital.

```
SELECT admissions.hospitalid,  sum(treatments.cost) AS totalCost
FROM TREATMENTS
join admissions
on admissions.admissionid = treatments.admissionid
GROUP BY admissions.HospitalID;
```

```
110     # Treatment Costs: Calculate the total cost of treatments provided at each hospital.
111
112     SELECT admissions.hospitalid,  sum(treatments.cost) AS totalCost
113     FROM TREATMENTS
114     join admissions
115     on admissions.admissionid = treatments.admissionid
116     GROUP BY admissions.HospitalID;
117
118     # Length of Stay Analysis: Extract the average length of stay for patients grouped by hospital.
119
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hospitalid	totalCost
1	5500
2	1100
3	8000
4	1400
5	1200

Length of Stay Analysis: Extract the average length of stay for patients grouped by hospital.

SELECT hospitalid, avg(datediff(dischargedate,admissiondate)) as avgLengthofStay

FROM admissions

GROUP BY hospitalid;

```
118     # Length of Stay Analysis: Extract the average length of stay for patients grouped by hospital.
119
120     select hospitalid, avg(datediff(dischargedate,admissiondate)) as avgLengthofStay
121     from admissions
122     group by hospitalid;
123
124     # Advanced Filtering:
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hospitalid	avgLengthofStay
1	4.0000
2	4.0000
3	5.0000
4	11.0000
5	10.5000

Advanced Filtering:

List all patients who stayed longer than 7 days in any hospital.

SELECT hospitalid, datediff(dischargedate, admissiondate) as LengthofStay

FROM admissions

WHERE datediff(dischargedate, admissiondate) > 7;

```
123
124     # Advanced Filtering:
125
126     # List all patients who stayed longer than 7 days in any hospital.
127
128 •   select hospitalid, datediff(dischargedate, admissiondate) as LengthofStay
129     from admissions
130     where datediff(dischargedate, admissiondate) > 7;
131
132     # Identify treatments that have been performed more than 5 times across all hospitals
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hospitalid	LengthofStay
4	15
5	14

Identify treatments that have been performed more than 5 times across all hospitals

SELECT admissions.hospitalid, count(treatments.procedurename) as NoofProcedures

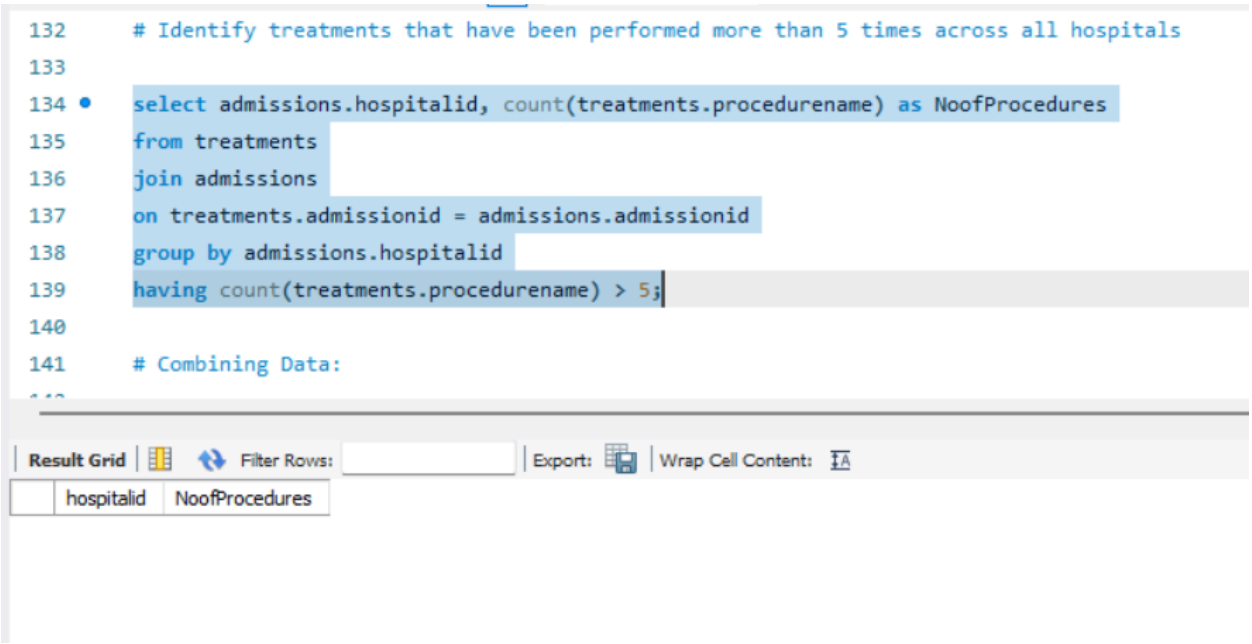
FROM treatments

JOIN admissions

ON treatments.admissionid = admissions.admissionid

GROUP BY admissions.hospitalid

HAVING count(treatments.procedurename) > 5;



Combining Data:

Combineadmission and treatment data to display complete patient histories.

SELECT *

FROM admissions

JOIN treatments

ON admissions.admissionid = treatments.admissionid;

```
141    # Combining Data:
142    
143    # Combineadmission and treatment data to display complete patient histories.
144
145    Select *
146    from admissions
147    join treatments
148    on admissions.admissionid = treatments.admissionid;
149
150    # Combinelists of patients admitted for different reasons (e.g., surgery and therapy)
```

Execute the selected portion of the script or everything, if there is no selection

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AdmissionID	PatientID	HospitalID	AdmissionDate	DischargeDate	ReasonForAdmission	TreatmentID	AdmissionID	ProcedureName	Cost	Outcome
1	1	1	2024-11-01	2024-11-05	Surgery	1	1	Appendectomy	1500	Successful
2	2	2	2024-11-03	2024-11-08	Therapy	2	2	Physical Therapy	800	Ongoing
3	3	3	2024-11-10	2024-11-15	Accident	3	3	Fracture Repair	3000	Successful
4	4	4	2024-11-12	2024-11-19	Routine Checkup	4	4	Blood Test	200	Pending
5	5	5	2024-12-01	2024-12-08	Infection	5	5	Antibiotics	500	Improved
6	6	1	2024-12-01	NULL	Surgery	6	6	Gallbladder Surgery	4000	Successful
7	7	2	2024-12-02	2024-12-05	Fracture Repair	7	7	X-Ray	300	Successful
8	8	3	2024-12-03	NULL	Chronic Illness	8	8	Chemotherapy	5000	Ongoing
9	9	4	2024-12-03	2024-12-18	Therapy	9	9	MRI Scan	1200	Pending
10	10	5	2024-12-04	2024-12-18	Infection	10	10	Diabetes Treatment	700	Improved

Combinelists of patients admitted for different reasons (e.g., surgery and therapy)

SELECT admissions.reasonforadmission , patients.fullname, patients.patientid

FROM admissions

JOIN patients

ON admissions.patientid = patients.patientid; # group by admissions.reasonforadmission

```
149
150     # Combinelists of patients admitted for different reasons (e.g., surgery and therapy)
151
152     select admissions.reasonforadmission , patients.fullname, patients.patientid
153     from admissions
154     join patients
155     on admissions.patientid = patients.patientid; # group by admissions.reasonforadmission
156
157     # Subqueries and Views:
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

reasonforadmission	fullname	patientid
Surgery	John Doe	1
Therapy	Jane Smith	2
Accident	Sam Brown	3
Routine Checkup	Lisa White	4
Infection	Tom Green	5
Surgery	Alice Johnson	6
Fracture Repair	Robert Black	7
Chronic Illness	Emily Davis	8
Therapy	Michael Scott	9
Infection	Sarah Taylor	10

Subqueries and Views:

Useasubquery to find the hospital with the highest average treatment cost.

SELECT HOSPITALS.HOSPITALID, HOSPITALS.HOSPITALNAME, HOSPITALS.LOCATION, COST AS AVERAGETREATMENTCOST

FROM ADMISSIONS

JOIN HOSPITALS

ON ADMISSIONS.HOSPITALID = HOSPITALS.HOSPITALID

JOIN TREATMENTS

ON ADMISSIONS.ADMISSIONID = TREATMENTS.ADMISSIONID

WHERE COST = (

SELECT AVG(COST) AS AVERAGECOST

FROM TREATMENTS

GROUP BY ADMISSIONID

ORDER BY AVERAGECOST DESC LIMIT 1

);

```
157  # Subqueries and Views:
158
159  # Useasubquery to find the hospital with the highest average treatment cost.
160
161  SELECT HOSPITALS.HOSPITALID, HOSPITALS.HOSPITALNAME, HOSPITALS.LOCATION, COST AS AVERAGETREATMENTCOST
162  FROM ADMISSIONS
163  JOIN HOSPITALS
164  ON ADMISSIONS.HOSPITALID = HOSPITALS.HOSPITALID
165  JOIN TREATMENTS
166  ON ADMISSIONS.ADMISSIONID = TREATMENTS.ADMISSIONID
167  WHERE COST = (
168                SELECT AVG(COST) AS AVERAGECOST
169                FROM TREATMENTS
170                GROUP BY ADMISSIONID
171                ORDER BY AVERAGECOST DESC LIMIT 1
172            );
173
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

HOSPITALID	HOSPITALNAME	LOCATION	AVERAGETREATMENTCOST
3	Central Medical Center	Chicago	5000

Create a view named HospitalPerformance to display the total number of admissions, average length of stay, and total revenue generated for each hospital.

CREATE VIEW HOSPITALPERFORMANCE AS

SELECT HOSPITALS.HOSPITALID,

HOSPITALS.HOSPITALNAME,

HOSPITALS.LOCATION,

COUNT(ADMISSIONS.ADMISSIONID) AS NUMOFADMISSIONS,

AVG(DATEDIFF(DISCHARGEDATE, ADMISSIONDATE)) AS AVELENGTHOFSTAY,

SUM(COST) AS TOTALREVENUE

FROM HOSPITALS

JOIN ADMISSIONS

ON HOSPITALS.HOSPITALID = ADMISSIONS.HOSPITALID

JOIN TREATMENTS

ON TREATMENTS.ADMISSIONID = ADMISSIONS.ADMISSIONID

GROUP BY HOSPITALS.HOSPITALID;

```
174  # Create a view named HospitalPerformance to display the total number of admissions, average length of stay, and total revenue generated for each hospital.
175
176  CREATE VIEW HOSPITALPERFORMANCE AS
177
178  SELECT HOSPITALS.HOSPITALID,
179         HOSPITALS.HOSPITALNAME,
180         HOSPITALS.LOCATION,
181        COUNT(ADMISSIONS.ADMISSIONID) AS NUMOFADMISSIONS,
182         AVG(DATEDIFF(DISCHARGEDATE, ADMISSIONDATE)) AS AVELENGTHOFSTAY,
183         SUM(COST) AS TOTALREVENUE
184  FROM HOSPITALS
185  JOIN ADMISSIONS
186     ON HOSPITALS.HOSPITALID =  ADMISSIONS.HOSPITALID
187  JOIN TREATMENTS
188     ON TREATMENTS.ADMISSIONID = ADMISSIONS.ADMISSIONID
189  GROUP BY HOSPITALS.HOSPITALID;
190
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

HOSPITALID	HOSPITALNAME	LOCATION	NUMOFADMISSIONS	AVELENGTHOFSTAY	TOTALREVENUE
1	General Hospital	New York	2	4.0000	5500
2	City Clinic	Los Angeles	2	4.0000	1100
3	Central Medical Center	Chicago	2	5.0000	8000
4	Regional Health Facility	Houston	2	11.0000	1400
5	Sunrise Hospital	Phoenix	2	10.5000	1200

DROP VIEW HOSPITALPERFORMANCE;

SELECT * FROM HOSPITALPERFORMANCE

```
192
193 •  SELECT * FROM HOSPITALPERFORMANCE
194
195    # Window Functions:
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

HOSPITALID	HOSPITALNAME	LOCATION	NUMOFADMISSIONS	AVELENGTHOFSTAY	TOTALREVENUE
1	General Hospital	New York	2	4.0000	5500
2	City Clinic	Los Angeles	2	4.0000	1100
3	Central Medical Center	Chicago	2	5.0000	8000
4	Regional Health Facility	Houston	2	11.0000	1400
5	Sunrise Hospital	Phoenix	2	10.5000	1200

Window Functions:

Use the RANK function to rank hospitals based on their total revenue.

```
SELECT HOSPITALNAME, TOTALREVENUE,
        RANK() OVER (ORDER BY TOTALREVENUE) AS RANKBYTOTALREVENUE
FROM HOSPITALPERFORMANCE
```

```
195     # Window Functions:
196
197     # Use the RANK function to rank hospitals based on their total revenue.
198
199     SELECT HOSPITALNAME, TOTALREVENUE,
200             RANK() OVER (ORDER BY TOTALREVENUE) AS RANKBYTOTALREVENUE
201     FROM HOSPITALPERFORMANCE
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

HOSPITALNAME	TOTALREVENUE	RANKBYTOTALREVENUE
City Clinic	1100	1
Sunrise Hospital	1200	2
Regional Health Facility	1400	3
General Hospital	5500	4
Central Medical Center	8000	5

Use DENSE_RANK to rank treatments based on their frequency

SELECT TREATMENTID, PROCEDURENAME, COUNT(PROCEDURENAME) AS COUNTOFPROCEDURES,

DENSE_RANK() OVER (ORDER BY COUNT(PROCEDURENAME)) RANKBYFREQUENCYOFTREATMENTS

FROM TREATMENTS

GROUP BY TREATMENTID

7. Key Insights

- Certain hospitals generate significantly higher revenue due to treatment complexity and patient volume
- Longer patient stays directly increase operational costs and reduce room availability
- A small subset of treatments contributes disproportionately to total revenue
- Hospital performance varies widely, indicating opportunities for efficiency improvement

8. Business Impact

- Enables hospital administrators to identify high-cost operations
- Supports better capacity and resource planning
- Helps optimize treatment pricing and operational workflows
- Provides a foundation for healthcare dashboards and reporting systems

9. Conclusion

This project demonstrates how **SQL-driven healthcare analytics** can transform raw hospital data into actionable insights. By combining relational database design, business queries, views, and window functions, the analysis supports informed decision-making for hospital operations, financial planning, and patient care optimization.

The project showcases strong SQL fundamentals along with real-world analytical thinking applicable to healthcare, business intelligence, and data analyst roles.