# SQL-Based Healthcare Data Analysis

**Author:** Amit Singh  
**Email:** [ikarannegi4919@gmail.com](mailto:ikarannegi4919@gmail.com)  
**LinkedIn:** [www.linkedin.com/in/amit-singh2003](https://www.linkedin.com/in/amit-singh2003)  
**Dataset Source:** [Kaggle - Healthcare Dataset](https://www.kaggle.com/datasets/prasad22/healthcare-dataset/code)  
**Dataset Size:** 55,500 records

## Project Overview

This project performs SQL-based analysis on a large healthcare dataset. The dataset includes patient-level admission records across multiple hospitals. The purpose of this analysis is to extract key business insights that can aid hospital management in decision-making.

Tools Used:

* **Excel:** Data Cleaning
* **ERD:** Designed from raw structure for clarity in MySQL Workbench
* **MySQL:** Querying & Analysis

**Excel Data Cleaning Process:-**

Before importing the dataset into MySQL, we performed extensive cleaning and preparation using Excel:

* **Removed titles** ("Mr.", "Mrs.", "Dr.") from patient and doctor names.
* **Trimmed spaces** and removed duplicate rows for hospitals, doctors, and patients etc.
* **Generated unique IDs** for:
  + Patients (e.g., P0001, P0002)
  + Doctors (e.g., D0001, D0002)
  + Hospitals (e.g., H0001, H0002)
  + Records (e.g., R0001, R0002)
* Ensured **date formats** were standardized (e.g., YYYY-MM-DD).

After cleaning, the structured data was saved as a CSV and imported into MySQL Workbench.

**Database Design & ER Diagram**

After data cleaning, the dataset was structured into a relational database.

**The raw flat data was broken into logical entities:**

-Patients Table: Contain details about Patients like Patient\_id, Name , Gender etc.

- Doctor Table: Contain details about Doctor like Doctor\_id, Doctor\_Name etc.

- Hospitals Table: Contains Hospitals details like Hospital\_id, Hospital\_Name etc .

-Insurance Table: Contains Insurance details like Insurance\_id, Insurance\_Provider etc .

- Admissions table: Contains Healthcare Details and links other tables.

**Primary & Foreign Keys :**

- Patients\_id → primary key in Patients, foreign key in Admissions.

- Doctor\_id → primary key in Doctors, foreign key in Admissions.

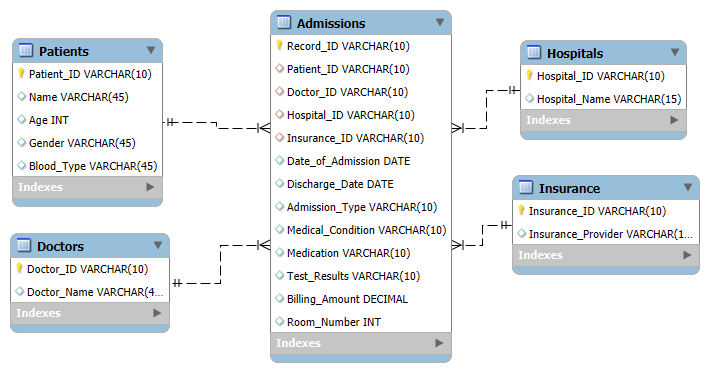
- Hospital\_id → primary key in Hospitals, foreign key in Admissions.

- Insurance\_id → primary key in Insurance, foreign key in Admissions.

## 

## **ER – DIAGRAM :**

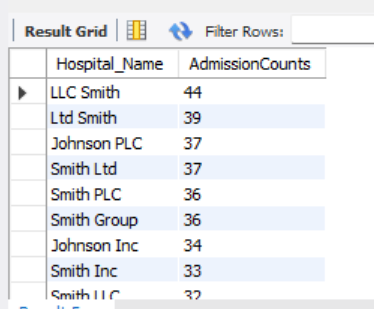
The Entity Relationship Diagram (ERD) was created using MySQL Workbench. It visually represents the relationships between tables.



## Business Problems and SQL Insights

### 1. **Most Visited Hospitals**

SELECT h.Hospital\_Name, COUNT(a.Patient\_ID) AS AdmissionCounts  
FROM admissions a  
JOIN hospitals h ON a.Hospital\_ID = h.Hospital\_ID  
GROUP BY Hospital\_Name  
ORDER BY AdmissionCounts DESC;

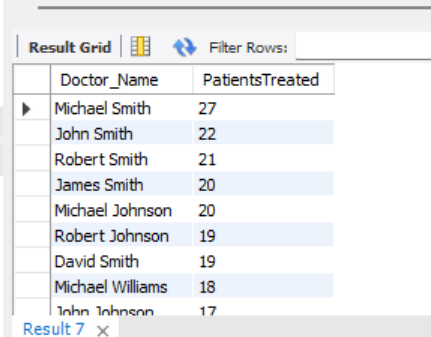


**Insight:**

* Identifies hospitals with the highest patient traffic.
* Useful for resource allocation, funding, and infrastructure planning.

### 2. **Doctors with Highest Patient Count**

SELECT d.Doctor\_Name, COUNT(a.Patient\_ID) AS PatientsTreated  
FROM admissions a  
JOIN doctors d ON a.Doctor\_ID = d.Doctor\_ID  
GROUP BY Doctor\_Name  
ORDER BY PatientsTreated DESC;

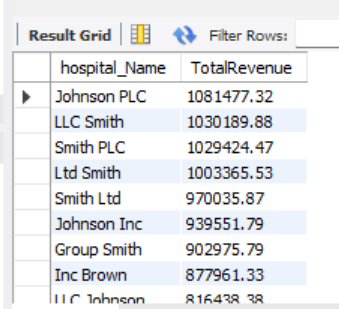


**Insight:**

* Reveals the top-performing doctors by patient volume.
* Can help in recognizing overburdened or popular medical professionals.

### 3. **Total Revenue by Hospital**

SELECT h.Hospital\_Name, SUM(Billing\_Amount) AS TotalRevenue  
FROM admissions a  
JOIN hospitals h ON a.Hospital\_ID = h.Hospital\_ID  
GROUP BY Hospital\_Name  
ORDER BY TotalRevenue DESC;

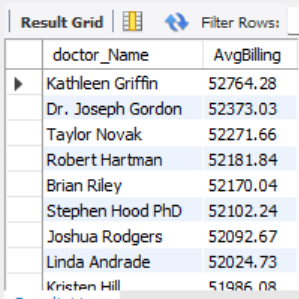


**Insight:**

* Shows which hospitals generate the most billing revenue.

### 4. **Average Billing per Doctor**

SELECT d.Doctor\_Name, ROUND(AVG(Billing\_Amount), 2) AS AvgBilling  
FROM admissions a  
JOIN doctors d ON a.Doctor\_ID = d.Doctor\_ID  
GROUP BY Doctor\_Name  
ORDER BY AvgBilling DESC;

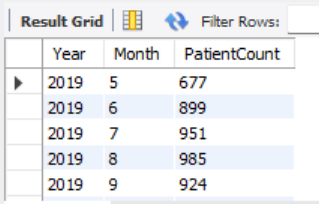


**Insight:**

* Highlights doctors associated with the highest treatment costs.

### 5. **Monthly Admission Trend (with Year)**

SELECT  
 YEAR(Date\_of\_Admission) AS Year,  
 MONTH(Date\_of\_Admission) AS Month,  
 COUNT(Patient\_ID) AS PatientCount  
FROM admissions  
GROUP BY YEAR(Date\_of\_Admission), MONTH(Date\_of\_Admission)  
ORDER BY Year, Month;

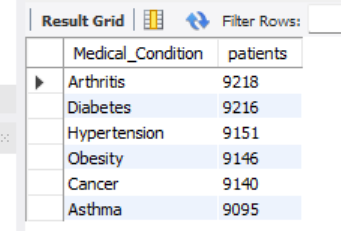


**Insight:**

* Tracks seasonality in patient visits.

### 6. **Most Common Medical Conditions**

SELECT Medical\_Condition, COUNT(Patient\_ID) AS Patients  
FROM admissions  
GROUP BY Medical\_Condition  
ORDER BY Patients DESC;

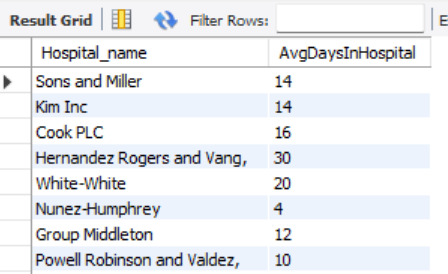


**Insight:**

* Identifies top health concerns among the population.
* Useful for public health campaigns and department staffing.

### 7. **Average Stay Duration per Hospital**

SELECT h.Hospital\_Name, CEIL(AVG(DATEDIFF(Discharge\_Date, Date\_of\_Admission))) AS AvgDaysInHospital  
FROM admissions a  
JOIN hospitals h ON a.Hospital\_ID = h.Hospital\_ID  
GROUP BY Hospital\_Name;

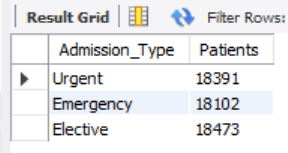


**Insight:**

* Measures hospital efficiency and patient recovery time.

### 8. **Most Common Admission Types**

SELECT Admission\_Type, COUNT(Patient\_ID) AS Patients  
FROM admissions  
GROUP BY Admission\_Type;

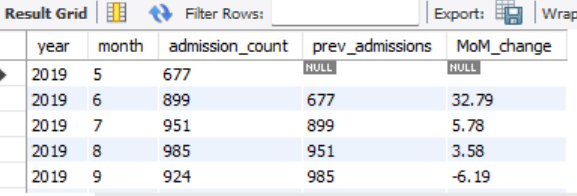


**Insight:**

* Segregates patient entries by type: Emergency, Urgent, Elective.

### 9. **Month-over-Month Growth in Admissions**

SELECT \*, ROUND((admission\_count - prev\_admissions)/prev\_admissions \* 100, 2) AS MoM\_Change  
FROM (  
 SELECT   
 YEAR(Date\_of\_Admission) AS Year,  
 MONTH(Date\_of\_Admission) AS Month,  
 COUNT(\*) AS admission\_count,  
 LAG(COUNT(\*)) OVER (ORDER BY YEAR(Date\_of\_Admission), MONTH(Date\_of\_Admission)) AS prev\_admissions  
 FROM admissions  
 GROUP BY YEAR(Date\_of\_Admission), MONTH(Date\_of\_Admission)  
) t;

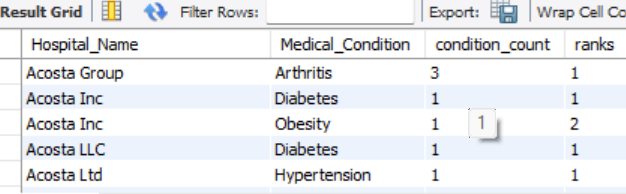


**Insight:**

* Identifies growth trends in admissions month over month.
* Accounts for different years, ensuring accurate comparisons.

### 10. **Top 3 Medical Conditions Per Hospital**

SELECT \* FROM (  
 SELECT h.Hospital\_Name, Medical\_Condition, COUNT(\*) AS Condition\_Count,  
 ROW\_NUMBER() OVER(PARTITION BY h.Hospital\_Name ORDER BY COUNT(\*) DESC) AS Rank  
 FROM admissions a  
 JOIN hospitals h ON a.Hospital\_ID = h.Hospital\_ID  
 GROUP BY h.Hospital\_Name, Medical\_Condition  
) t  
WHERE Rank <= 3;

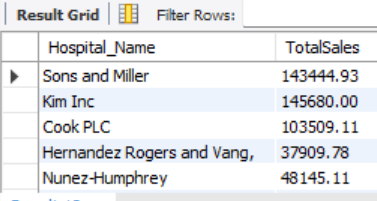


**Insight:**

* Gives a breakdown of frequent diseases per hospital.

### 11. **Hospitals with Above Average Revenue**

WITH totalsales AS (  
 SELECT h.Hospital\_Name, SUM(a.Billing\_Amount) AS TotalSales  
 FROM admissions a  
 JOIN hospitals h ON a.Hospital\_ID = h.Hospital\_ID  
 GROUP BY h.Hospital\_Name  
),  
avg\_sales AS (  
 SELECT AVG(TotalSales) AS avg\_sales FROM totalsales  
)  
SELECT t.\*  
FROM totalsales t  
JOIN avg\_sales a ON t.TotalSales > a.avg\_sales;

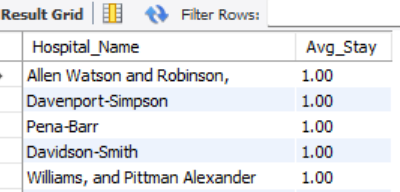


**Insight:**

* Highlights hospitals whose revenue exceeds the average.

### 12. **Hospital with Fastest Discharge Rate**

SELECT h.Hospital\_Name, ROUND(AVG(DATEDIFF(a.Discharge\_Date, a.Date\_of\_Admission)), 2) AS AvgStay  
FROM admissions a  
JOIN hospitals h ON a.Hospital\_ID = h.Hospital\_ID  
GROUP BY h.Hospital\_Name  
ORDER BY AvgStay ASC  
LIMIT 1;

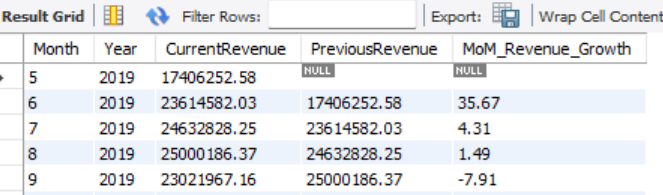


**Insight:**

* Identifies the hospital with the most efficient discharge rate.
* Can indicate faster recovery.

### 13. **Month-over-Month Revenue Growth**

SELECT \*, ROUND((revenue - prev\_revenue)/prev\_revenue \* 100, 2) AS RevenueGrowth  
FROM (  
 SELECT   
 YEAR(Date\_of\_Admission) AS Year,  
 MONTH(Date\_of\_Admission) AS Month,  
 SUM(Billing\_Amount) AS revenue,  
 LAG(SUM(Billing\_Amount)) OVER (ORDER BY YEAR(Date\_of\_Admission), MONTH(Date\_of\_Admission)) AS prev\_revenue  
 FROM admissions  
 GROUP BY YEAR(Date\_of\_Admission), MONTH(Date\_of\_Admission)  
) t;

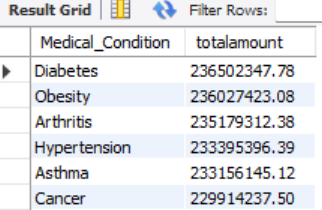


**Insight:**

* Tracks monthly changes in total revenue.
* Ensures accuracy by separating months across years.

### 14. **Revenue per Medical Condition**

SELECT Medical\_Condition, ROUND(SUM(Billing\_Amount), 2) AS TotalRevenue  
FROM admissions  
GROUP BY Medical\_Condition  
ORDER BY TotalRevenue DESC;



**Insight:**

* Reveals which conditions generate the most revenue.
* Helps identify high-cost diseases.

## Conclusion

This analysis provided data-backed insights for healthcare operations, including:

* Hospital traffic and revenue trends
* Doctor performance
* Disease prevalence
* Admission trends and types

These queries are essential for administrators, policymakers, and healthcare strategists aiming to improve operational efficiency and patient care.

For further collaboration or academic reference, feel free to reach out to me at [**ikarannegi4919@gmail.com**](mailto:ikarannegi4919@gmail.com) or connect via [**LinkedIn**](https://www.linkedin.com/in/amit-singh2003).