# Project Story

In the ever-evolving world of healthcare, optimizing hospital efficiency is crucial to ensure timely and effective patient care. This mini project, titled 'Hospital Patient Flow Analysis,' is aimed at understanding the patterns of hospital admissions, discharges, and patient stay durations. By leveraging SQL for data extraction and analysis and Python for visualization, we can uncover insights that help reduce congestion and improve resource utilization.

## Project Objective

The main goal of this project is to monitor patient flow within a hospital environment. This includes analyzing admission and discharge trends, calculating patient stay durations, and identifying key performance indicators (KPIs) that reflect the hospital's operational efficiency.

## Journey Through the Project Tasks

We began by loading patient records containing vital information such as admission dates, discharge dates, and the wards they were assigned to. This data laid the foundation for our analysis.

Next, we harnessed the power of SQL to calculate the average length of stay for each patient and to explore admission and discharge patterns over time. By grouping this data day-wise, we could detect trends and peaks in patient traffic.

To bring the data to life, we transitioned into Python. Using pandas, we performed datetime operations to calculate durations and grouped the data for deeper insights. Visualizations such as bar charts and line plots provided a clear picture of bed occupancy and average stay metrics across wards.

## Data Source

The dataset used for this project was obtained from Kaggle and includes anonymized hospital patient statistics. You can find it here: https://www.kaggle.com/datasets/cms/hospital-general-information

## Key Performance Indicators (KPIs)

Throughout the project, we focused on a few important KPIs to measure hospital efficiency:

1. 1. Average Length of Stay: Calculated using the formula AVG(DATEDIFF(discharge\_date, admission\_date)), this KPI indicates the typical duration a patient remains admitted.
2. 2. Bed Utilization Rate: This is the ratio of occupied beds to total available beds, shedding light on resource usage.
3. 3. Admission/Discharge Rate: We monitored how many admissions and discharges occur each day using COUNT operations grouped by day.
4. 4. Readmission Rate: A critical indicator, calculated as COUNT(Readmissions) / Total Discharges, helps assess quality of care.

## SQL Implementation

We designed a table with fields such as patient\_id, ward, admission\_date, and discharge\_date. Using SQL queries, we calculated stay durations and derived daily admission rates to understand hospital load across time.

## Python Implementation

In Python, we used pandas to manipulate dates and grouped the data to observe trends. We visualized key metrics using matplotlib and seaborn, turning raw data into meaningful visuals that highlight ward-level congestion and performance metrics.

## Conclusion

By integrating SQL-based data analysis with Python visualizations, we were able to monitor and understand the flow of patients in a hospital setting. These insights can help hospital administrators make data-driven decisions to improve efficiency, reduce bottlenecks, and enhance patient care.