## STEP 1: Topic/Domain Selection

**Selected Topic:** Hospital Management System

## Introduction:

Our topic is focused on the Hospital Management System. In the dynamic realm of healthcare, the efficient management of information is fundamental to providing quality patient care and ensuring streamlined hospital operations. The Hospital Management System (HMS) outlined in this database management project offers a comprehensive framework for storing and organizing crucial data related to patients, doctors, appointments, medical records, prescriptions, healthcare providers, staff, departments, financial transactions. This system is designed to enhance the overall efficiency and effectiveness of a medical facility by establishing a structured and interconnected database that supports various aspects of hospital administration.

At its essence, the HMS strives to elevate the efficiency and effectiveness of healthcare services by establishing a structured database that fosters collaboration and data-driven decision-making. By creating a centralized repository for various facets of hospital management, the system empowers healthcare professionals with the tools needed to optimize patient care, streamline administrative processes, and ensure transparency in financial transactions.

## Advantages and contributions:

**Comprehensive Patient Management:**

The Patient Table captures essential patient information, including demographics, contact details, age, and insurance details. This aids in creating a comprehensive patient profile, enabling healthcare providers to deliver personalized and targeted care.

**Efficient Appointment Scheduling:**

The Appointment Table facilitates seamless scheduling and management of appointments. With references to both the Patient and Doctor tables, the system ensures accurate tracking of patient appointments, leading to improved patient-doctor coordination.

**Holistic Medical Record Keeping:**

The Medical Record Table integrates patient data with diagnostic and symptomatic details, offering a complete medical history. This enhances the accuracy of diagnoses and treatments, contributing to improved healthcare outcomes.

**Prescription Management:**

The Prescription Table enables the systematic recording of prescribed medications, dosages, and frequencies. This promotes adherence to treatment plans and provides a valuable resource for both patients and healthcare providers.

**Comprehensive Doctor and Staff Information:**

The Doctor and Staff tables store critical information about healthcare professionals, including their specialization, experience, roles, and contact details. This assists in optimizing workforce management and ensuring the right personnel are assigned to specific tasks.

**Departmental Organization:**

The Department Table, along with Department ID references in the Doctor and Staff supports efficient departmental organization. This structure aids in the assignment of staff, tracking of patients within specific departments.

**Financial Transaction Tracking:**

The Transaction Table records financial transactions associated with patient care. This supports transparent financial management, allowing for precise tracking of expenses, payments, and payment modes.

## Uses Cases.

**Patient-Centric Care:**

The HMS is designed to enhance patient-centric care by providing a centralized repository of patient information, facilitating personalized treatment plans, and efficient communication between healthcare providers and patients.

**Operational Efficiency:**

The system contributes to operational efficiency by streamlining appointment scheduling, optimizing staff allocation, and providing comprehensive medical records. This, in turn, leads to smoother hospital workflows and improved resource utilization.

**Financial Transparency:**

The HMS supports financial transparency by tracking and managing financial transactions associated with patient care. This aids in budgeting, auditing, and ensuring financial accountability within the healthcare institution.

**Quality Improvement:**

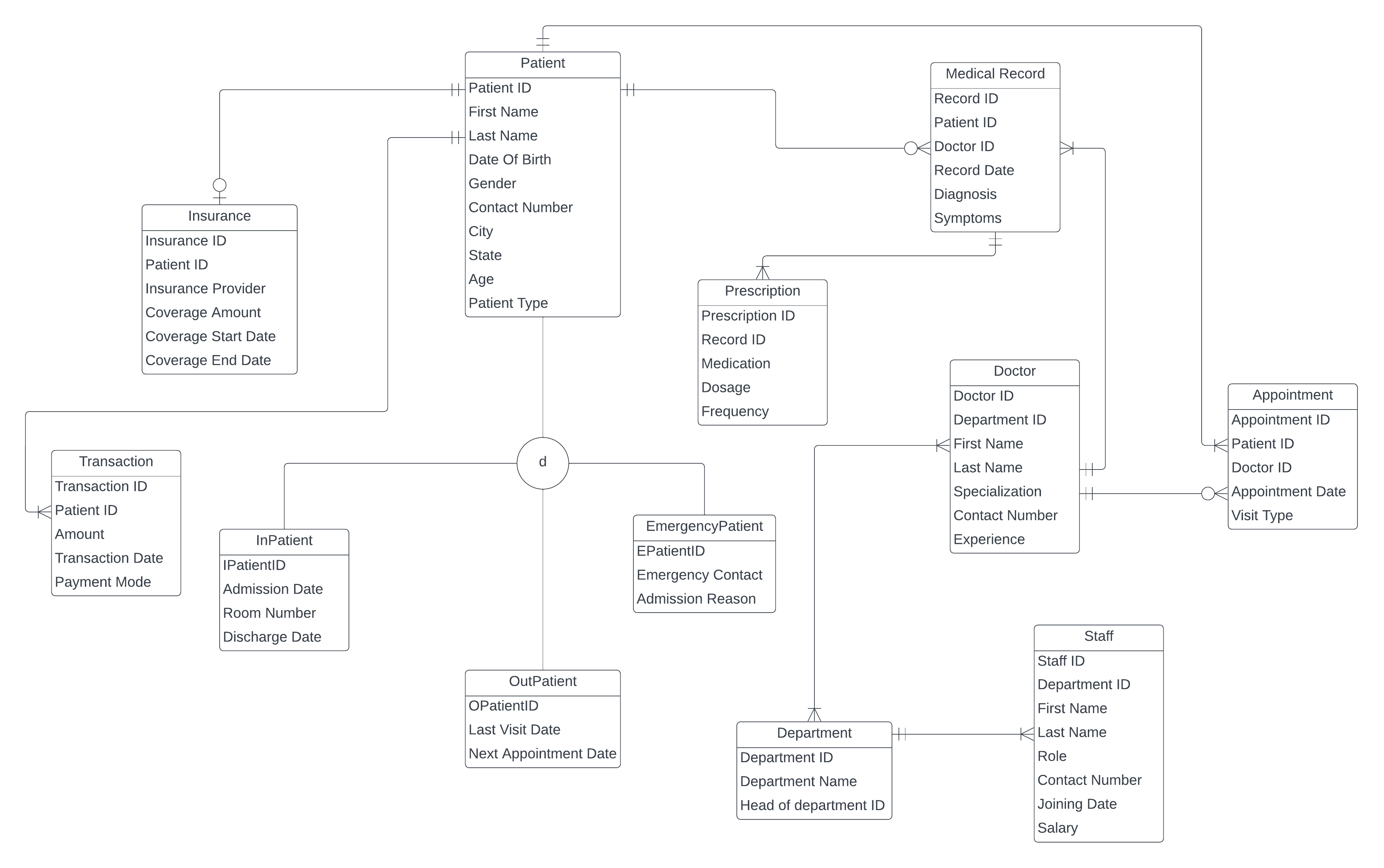
Patient feedback and medical record analysis contribute to ongoing quality improvement initiatives. Identifying areas for enhancement based on patient experiences and treatment outcomes helps in refining healthcare delivery practices.

**STEP 2: Conceptual Data Modeling and Database Design**

**Business rules:**

1. Each patient can only have one insurance.
2. More than one transaction is allowed to be performed by one patient.
3. A patient might have zero or more than one record.
4. One department will have one or more than one staff member.
5. A doctor may have 0 or more appointments.
6. A department can have one or more than one doctor.
7. One doctor will have one or more medical record.
8. A medical record of a patient will have one or more prescription.

**ER/EER DIAGRAM:**



**Relational Model (3NF-Form):**

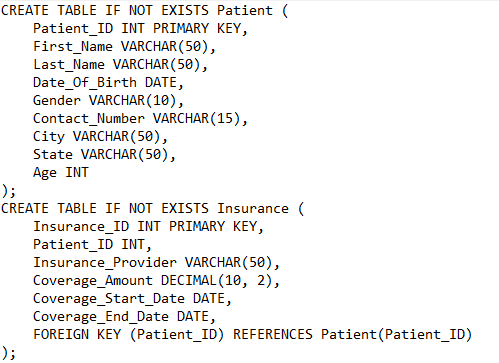
A diagram of a company

Description automatically generated

**STEP 3: Database Implementation**

**SQL Commands to Create Tables for Database:**

Below are the SQL commands that were used to create tables for the database.



A computer code with black text

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

A computer code with black text

Description automatically generated

A computer screen shot of a code

Description automatically generated

A screenshot of a computer program

Description automatically generated

## SQL Commands to Insert Data into Database:

We have inserted the dummy data. The data consists of data for the tables of Patients, Appointments, Medical Records, Prescriptions, Doctors, Staff, Departments, Transactions, Insurance. The corresponding INSERT commands can be found in the columns to the right of the data. These INSERT INTO commands were then used to populate the database on MySQL Workbench. A copy of all the insert-into commands can be found in the attached text file ([Hospital Management Data](https://docs.google.com/document/d/1WbHkD10hAQg8OVJmlZDstlM8EFSokGa4jZi5u9e3MgM/edit))

## Analytical Questions and SQL Queries:

1. **“Doctors” with highest experience of each department**

select Department.Department\_Name,

concat(Doctor.First\_Name,' ',Doctor.Last\_Name) AS Name,

max(Experience) AS Experience\_In\_Yrs

FROM Doctor INNER JOIN Department on Doctor.Department\_ID = Department.Department\_ID

group by Doctor.Department\_ID;

A screenshot of a computer

Description automatically generated

1. **“Department” with highest no of doctors**

SELECT Department.Department\_Name, count(Doctor.Doctor\_ID) AS Number\_of\_Doctors

FROM Doctor INNER JOIN Department ON Doctor.Department\_ID = Department.Department\_ID

group by Department.Department\_Name order by Number\_of\_Doctors DESC LIMIT 1;

A screenshot of a computer

Description automatically generated

1. **“No of patients” in each patient type**

SELECT Patient.PatientType, COUNT(Patient.Patient\_ID) AS PatientCount

FROM Patient

WHERE Patient.PatientType IS NOT NULL

GROUP BY Patient.PatientType;

A screenshot of a computer

Description automatically generated

1. **Patients who paid their hospital bill and how much they paid**

select Patient.Patient\_ID, concat(Patient.First\_Name,' ',Patient.Last\_Name) AS Name, Patient.City, Patient.Contact\_Number, sum(Transactions.Amount) AS Total\_Bill\_Paid from Patient inner join Transactions on Patient.Patient\_ID = Transactions.Patient\_ID group by Patient.Patient\_ID order by Patient.Patient\_ID;

A screenshot of a computer

Description automatically generated

1. **Emergency patient Medical records**

select Patient.Patient\_ID, Patient.Last\_Name, Patient.Date\_Of\_Birth, Patient.Age, Patient.PatientType, EmergencyPatient.Emergency\_Contact, EmergencyPatient.Admission\_Reason, Medical\_Record.Diagnosis, Medical\_Record.Symptoms from Patient INNER JOIN EmergencyPatient ON Patient.Patient\_ID = EmergencyPatient.EPatientID INNER JOIN Medical\_Record ON Patient.Patient\_ID = Medical\_Record.Patient\_ID;

A screenshot of a computer

Description automatically generated

1. **Inpatient details with no of days treated in hospital and their room charge ex: room charge is 500 per day**

WITH Inpatient\_details AS(

SELECT concat(Patient.First\_Name,' ',Patient.Last\_Name) AS Name,

Patient.PatientType, InPatient.RoomNumber,

InPatient.AdmissionDate,

DATEDIFF(InPatient.DischargeDate, InPatient.AdmissionDate) AS Days\_Stayed\_In\_Room

FROM InPatient INNER JOIN Patient ON InPatient.IPatientID = Patient.Patient\_ID

)

select \*,(Days\_Stayed\_In\_Room \* 500) as Room\_charge from Inpatient\_details order by AdmissionDate;

A screenshot of a computer

Description automatically generated

1. **Insurance Providers with highest coverage amount**

select Insurance.Insurance\_Provider, Insurance.Coverage\_Amount

From Insurance order by Insurance.Coverage\_Amount DESC LIMIT 2;

A screenshot of a computer

Description automatically generated

1. **Staff with highest salary and their experience**

Select Department.Department\_Name, CONCAT(Staff.First\_Name, ' ', Staff.Last\_Name) AS Name, Staff.Staff\_Role, round((DATEDIFF(CURDATE(), Staff.Joining\_Date)/365),1) AS Experience, max(Staff.Salary) as Salary from Staff INNER JOIN Department ON Staff.Department\_ID = Department.Department\_ID group by Staff.Department\_ID;

A screenshot of a computer

Description automatically generated

1. **Patients with their medications and their diagnosis results**

select Patient.Patient\_ID, concat(Patient.First\_Name, ' ', Patient.Last\_Name) AS Name,

Medical\_Record.Diagnosis, Medical\_Record.Symptoms, Prescription.Medication, Prescription.Dosage

FROM Patient INNER JOIN Medical\_Record ON Patient.Patient\_ID = Medical\_Record.Patient\_ID

INNER JOIN Prescription ON Medical\_Record.Record\_ID = Prescription.Record\_ID;

A screenshot of a computer

Description automatically generated

1. **No of appointments for each year and for each department**

select Department.Department\_Name, count(Appointment.Appointment\_ID) AS Total\_Appointments,

year(Appointment.Appointment\_Date) AS Year FROM Appointment INNER JOIN Doctor ON Appointment.Doctor\_ID = Doctor.Doctor\_ID INNER JOIN Department ON Doctor.Department\_ID = Department.Department\_ID group by Department.Department\_Name, year order by year;

A screenshot of a computer

Description automatically generated

## STEP 4: Enterprise (web) Database Dashboard

The analytical group dashboard can be found under the following link:

<https://colab.research.google.com/drive/1Q4FKaAEbPq8t6jqGw-NHhTYHkEg7roCQ?usp=sharing>

A screenshot of a computer

Description automatically generated

A close-up of a graph

Description automatically generated

A close-up of a graph

Description automatically generated

A graph and a graph

Description automatically generated with medium confidence

A screenshot of a graph

Description automatically generated