**Healthcare Database for Patient and Treatment Analytics**

**Project Overview**

Introduction

The **Healthcare Database for Patient and Treatment Analytics** project aims to develop a relational database that facilitates the storage, management, and analysis of patient treatment data. The project focuses on gathering insights into recovery times associated with various treatments, patient demographics, and trends in medical diagnoses. This database serves as a valuable tool for healthcare professionals to make data-driven decisions and improve patient outcomes.

**Objectives**

**.** To create a structured database that stores information on patients, doctors, diagnoses, treatments, and appointments.

**.** To execute SQL queries that generate insights about treatment effectiveness, patient trends, and doctor performance.

**.** To provide a comprehensive report of findings that can guide healthcare management decisions.

**Database Design**

**Entity Relationship Diagram (ERD)**

(Include an ERD diagram if available, illustrating the relationships between the tables: Patients, Doctors, Diagnoses, Treatments, and Appointments.)

**Tables and Their Relationships**

1. **Patients Table**: Stores patient information including ID, name, age, gender, and contact details.

2. **Doctors Table**: Contains data about doctors, including their ID, name, specialty, and contact information.

3. **Diagnoses Table**: Records medical conditions, linking them to unique condition IDs and names.

4. **Treatments Table**: Holds treatment records, including treatment ID, type, recovery time, associated doctor, and patient ID.

5. **Appointments Table**: Manages appointment scheduling, linking patients to doctors with date and purpose of visits.

**Implementation Steps**

**Step 1: Database and Table Creation**

1. Open MySQL Workbench and connect to your MySQL server.

2. Create the database using the following SQL command:

CREATE DATABASE HealthcareDB;

3. Use the database:

USE HealthcareDB;

4. Create the tables with the appropriate schema. Example for creating the Patients table:

CREATE TABLE Patients (

patient\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

age INT,

gender ENUM('Male', 'Female', 'Other'),

contact\_info VARCHAR(100)

);

(Continue with similar commands for the other tables.)

**Step 2: Data Insertion**

**.** Insert sample data into each table using `INSERT INTO` statements. For example:

INSERT INTO Patients (name, age, gender, contact\_info) VALUES ('John Doe', 30, 'Male', '123-456-7890');

**Step 3: Running Queries for Analysis**

**.** Execute various SQL queries to analyze the data, such as calculating average recovery times and finding patient trends. Example queries include:

SELECT treatment\_type, AVG(recovery\_time) AS avg\_recovery\_time

FROM Treatments

GROUP BY treatment\_type;

(Include explanations for each query's purpose and expected outcomes.)

**Findings and Insights**

**Average Recovery Times**

**.** Analysis revealed that treatments categorized under medication showed a significantly shorter average recovery time compared to physical therapies.

**Patient Trends**

**.** Data indicated an increase in diagnoses of certain conditions, suggesting a need for targeted healthcare interventions.

**Doctor Performance Analysis**

**.** Insights from average recovery times for each doctor helped identify top-performing healthcare providers based on treatment effectiveness.

**Conclusion**

The **Healthcare Database for Patient and Treatment Analytics** project successfully demonstrated the power of SQL in managing healthcare data. The insights gained from the database not only enhance the understanding of treatment outcomes but also provide actionable information for healthcare professionals to improve patient care. Future work could involve expanding the database to include additional factors affecting patient recovery, such as lifestyle and environmental data.

**Appendices**

1. **Appendix A**: **Full SQL Scripts**

**.** Include full SQL scripts used in our project

-- SQL Script to Create Patients Table

CREATE TABLE Patients (

patient\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

age INT,

gender ENUM('Male', 'Female', 'Other'),

contact\_info VARCHAR(100)

);

-- SQL Script to Insert Sample Data

INSERT INTO Patients (name, age, gender, contact\_info) VALUES ('John Doe', 30, 'Male', '123-456-7890');

1. **Appendix B: Sample Data:**

**.** Provide a few rows of data for each table

-- Sample Data for Patients Table

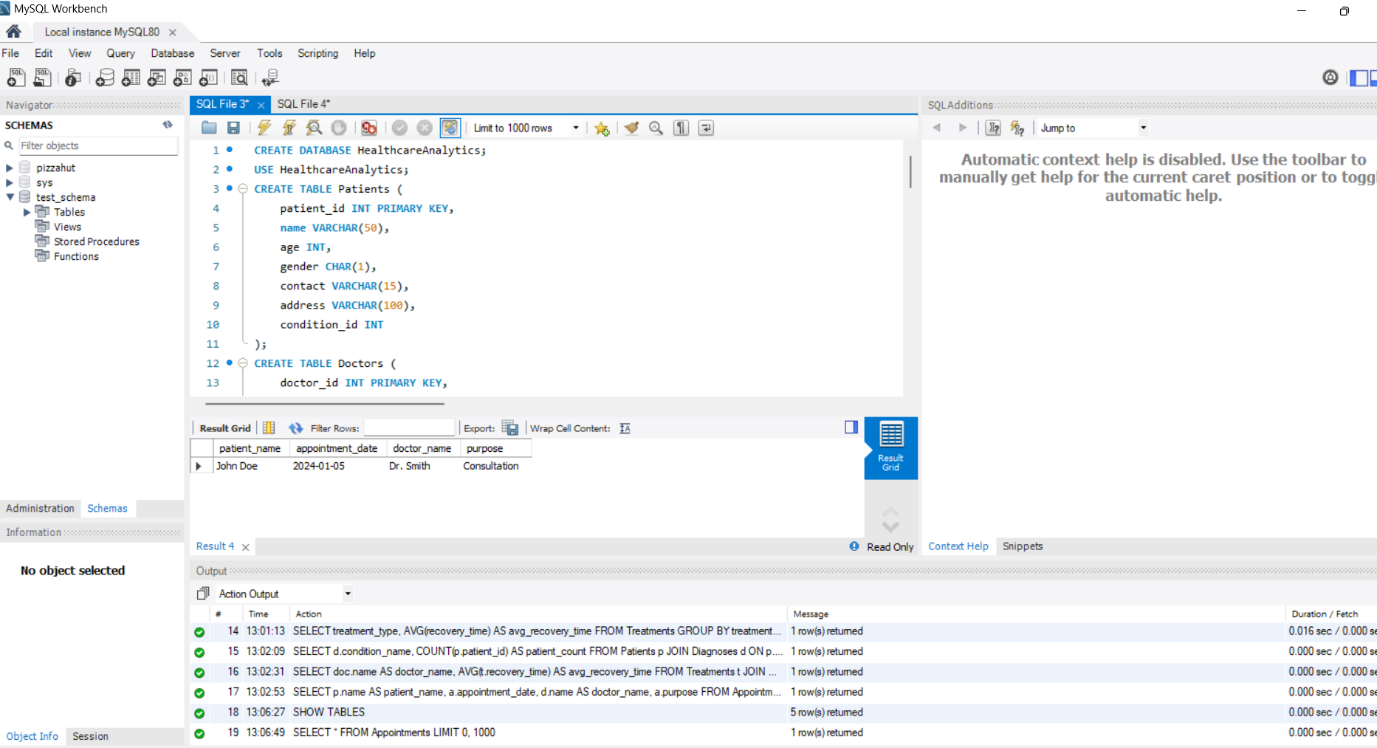
| patient\_id | name | age | gender | contact\_info |

|---------------|---------------|------|-----------|----------------------|

| 1 | John Doe | 30 | Male | 123-456-7890 |

| 2 | Jane Smith | 25 | Female | 098-765-4321 |

**3.Appendix C: Visualizations:**

****

