**[HealthData Management System (DBMS)]**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of 

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**A picture containing text

Description automatically generated**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522240**

**[May, 2024]**

# **INDEX**

1. Abstract
2. Aim
3. Key Objectives
4. Expected Outcomes
5. Benefits and Impact
6. ER Diagram
7. Database for ER Diagram
8. Normalization of Tables
9. Normalized SQL Code

# **"HealthData: Managing Medical Records with a DBMS"**

**Abstract:**

The "HealthData" project focuses on developing a cutting-edge solution for managing medical records through the utilization of a sophisticated Database Management System (DBMS). This project addresses the escalating challenges posed by the increasing volume and intricacy of medical data in modern healthcare settings. By transitioning from manual, paper-based systems to electronic health records (EHRs) and leveraging the capabilities of a DBMS, HealthData aims to enhance the storage, retrieval, and analysis of medical records while upholding stringent security and privacy standards.

**Aim:** To develop a comprehensive and secure system for managing medical records within healthcare institutions.

**Key Objectives:**

1. **Centralized Data Repository:** Implementing a centralized database system to store all medical records securely. This repository will serve as a single source of truth for patient information, enabling healthcare providers to access comprehensive records efficiently.
2. **Data Security and Privacy:** Implementing robust security measures to safeguard sensitive patient data against unauthorized access, breaches, and data loss. Compliance with regulatory standards such as HIPAA (Health Insurance Portability and Accountability Act) will be a top priority.
3. **Efficient Data Retrieval:** Designing efficient data retrieval mechanisms to enable healthcare professionals to quickly access relevant patient information during consultations, procedures, or emergencies. Indexing, query optimization, and data caching will be utilized to enhance retrieval speed.
4. **Interoperability:** Ensuring interoperability with existing healthcare systems and standards to facilitate seamless exchange of medical data between different healthcare providers, laboratories, and departments. Integration with standards such as HL7 (Health Level Seven) will be explored.
5. **Scalability and Performance:** Building a scalable architecture capable of accommodating the growing volume of medical data while maintaining optimal system performance. This involves employing techniques such as sharding, replication, and load balancing.
6. **User-Friendly Interface:** Developing an intuitive user interface tailored to the needs of healthcare professionals, including physicians, nurses, and administrative staff. The interface will prioritize ease of use, accessibility, and customization options.

**Expected Outcomes:**

* Enhanced patient care through timely access to comprehensive medical records.
* Improved operational efficiency and productivity within healthcare institutions.
* Strengthened compliance with regulatory requirements and data protection standards.
* Facilitated research and analysis through access to aggregated anonymized patient data.
* Reduced administrative burden associated with manual record-keeping processes.

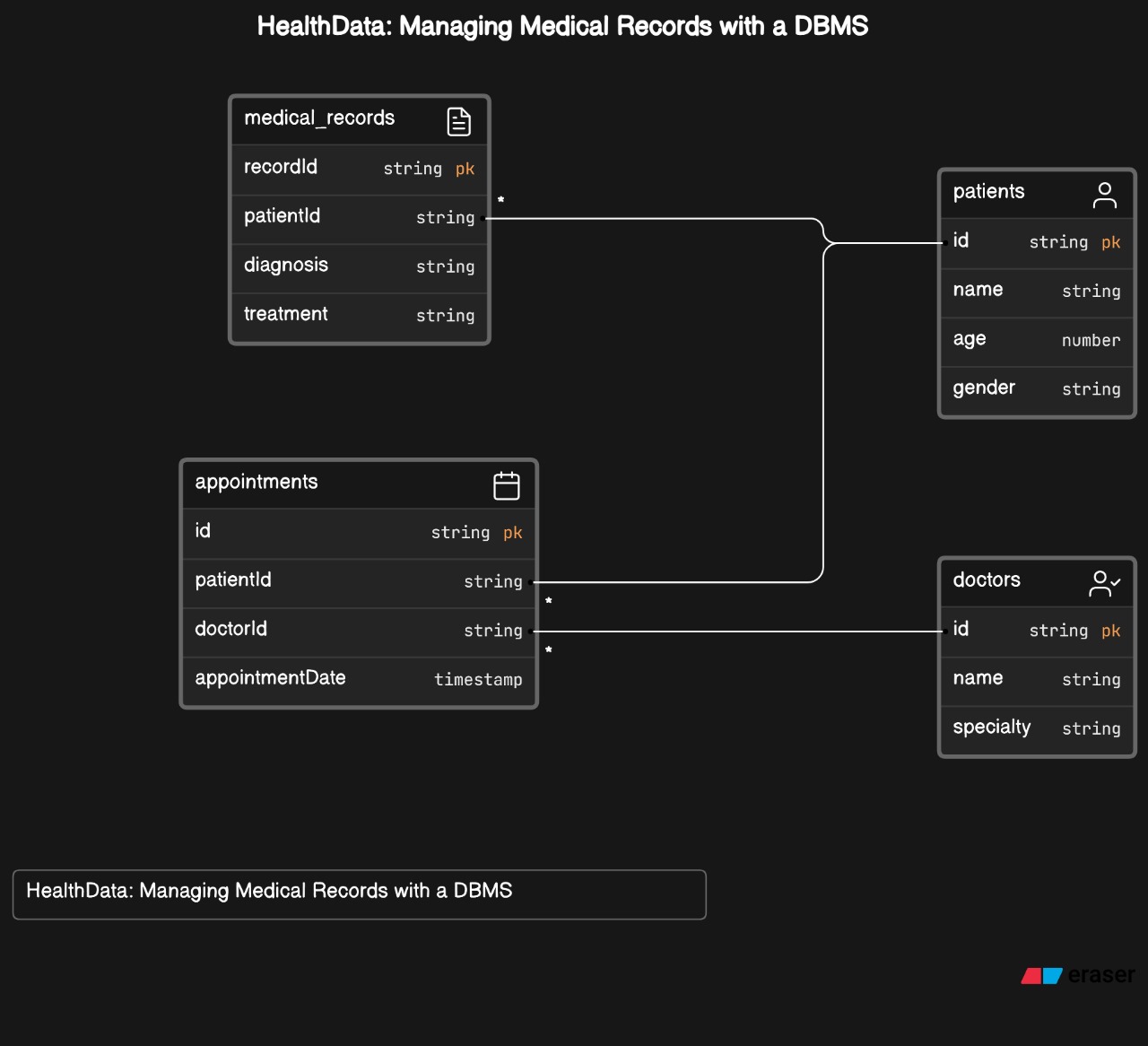
The "HealthData" project represents a significant step towards modernizing healthcare information management and ensuring the delivery of high-quality, patient-centric care in the digital age.

**Benefits and Impact:**

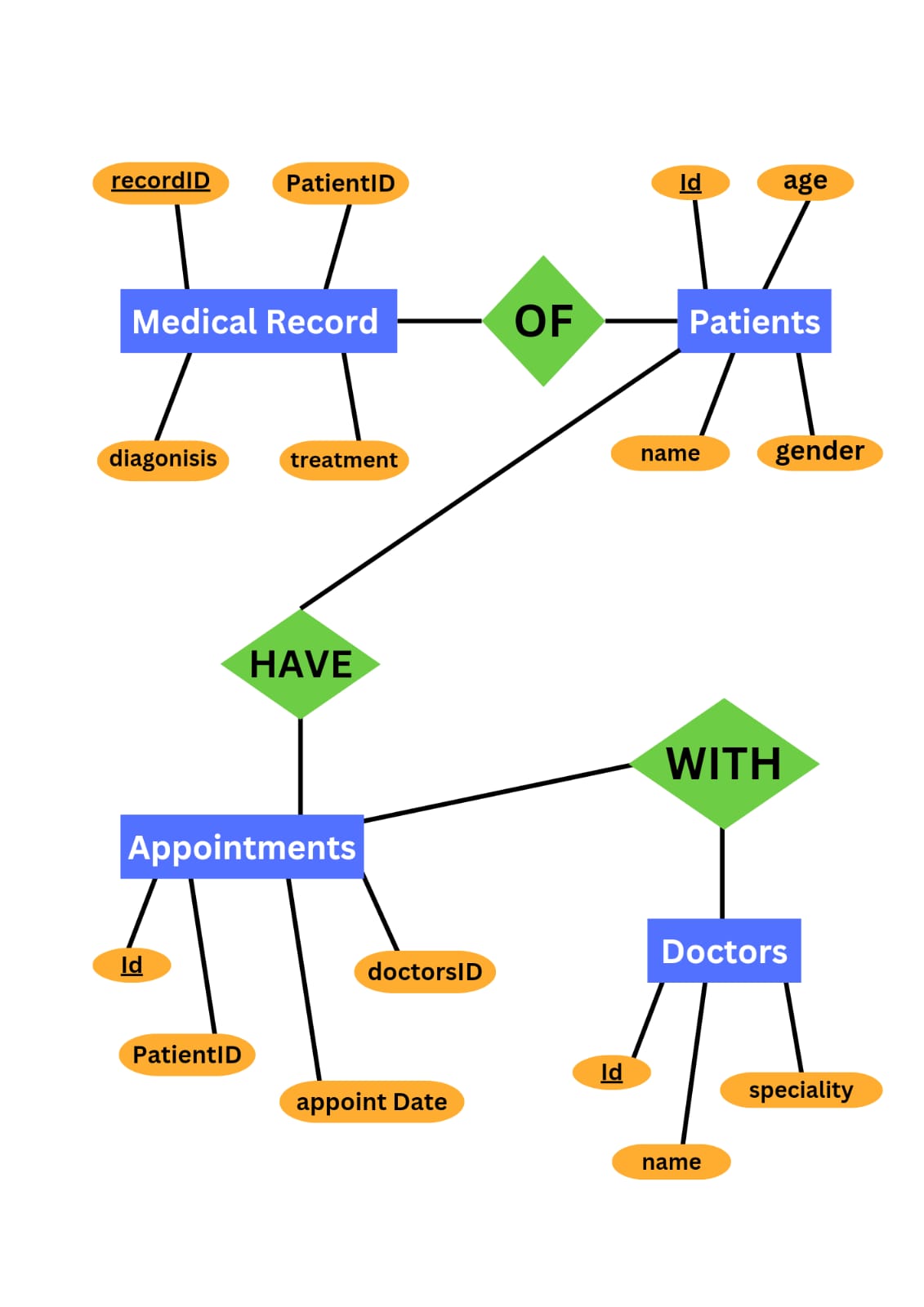
* **Improved Patient Care:** HealthData enables healthcare professionals to access comprehensive patient records promptly, leading to more informed decision-making and improved patient outcomes.
* **Enhanced Efficiency:** By streamlining medical data management processes, HealthData reduces administrative overhead, allowing healthcare institutions to allocate resources more efficiently and focus on delivering quality care.
* **Data-Driven Insights:** HealthData provides valuable insights into patient populations, disease trends, treatment outcomes, and clinical best practices through data analytics and reporting capabilities, fostering evidence-based medicine and continuous quality improvement.
* **Compliance and Security:** By adhering to stringent security and compliance standards, HealthData mitigates the risk of data breaches and ensures patient privacy and confidentiality, fostering trust among patients and regulatory.
* **Cost Savings:** HealthData helps healthcare institutions reduce costs associated with paper-based record-keeping, duplicate tests, and inefficient processes, resulting in significant cost savings over time.

**Description:**

The "HealthData: Managing Medical Records with a DBMS" project represents a significant advancement in healthcare information management, leveraging modern database technologies to revolutionize the way medical records are stored, accessed, and utilized. By empowering healthcare professionals with timely access to accurate and comprehensive patient information, HealthData aims to enhance patient care, optimize operational efficiency, and drive innovation in healthcare delivery.

**Table: **

**ER Diagram:**

****

**Description of ER Diagram:**

* **Entities/Tables:**
* **Patients:** Contains personal details of patients.
* **Medical Records:** Stores medical history linked to each patient.
* **Appointments:** Records appointments between patients and doctors.
* **Doctors:** Holds information about the doctors.
* **Attributes:**
* **Patients:** Attributes include id, name, age, and gender.
* **Medical Records:** Attributes include recordId, patientId, diagnosis, and treatment.
* **Appointments:** Attributes include id, patientId, doctorId, and appointmentDate.
* **Doctors:** Attributes include id, name, and specialty.
* **Relationships:**
* **Medical Records to** **Patients:** A foreign key from medical\_records references the id in patients, indicating which patient the record belongs to.
* **Appointments to** **Patients:** A foreign key from appointments references the id in patients, showing which patient has the appointment.
* **Appointments** **to Doctors:** A foreign key from appointments references the id in doctors, showing which doctor is involved in the appointment.

**Created DATABASE for ER Diagram: (conversion of ER Diagram into tables)**

**-- Create database if not exists**

**CREATE DATABASE IF NOT EXISTS healthcare;**

**USE healthcare;**

**-- Table: patients**

**CREATE TABLE IF NOT EXISTS patients (**

**id VARCHAR(255) PRIMARY KEY,**

**name VARCHAR(255),**

**age INT,**

**gender VARCHAR(50)**

**);**

**-- Table: doctors**

**CREATE TABLE IF NOT EXISTS doctors (**

**id VARCHAR(255) PRIMARY KEY,**

**name VARCHAR(255),**

**specialty VARCHAR(255)**

**);**

**-- Table: appointments**

**CREATE TABLE IF NOT EXISTS appointments (**

**id VARCHAR(255) PRIMARY KEY,**

**patientId VARCHAR(255),**

**doctorId VARCHAR(255),**

**appointmentDate TIMESTAMP,**

**FOREIGN KEY (patientId) REFERENCES patients(id),**

**FOREIGN KEY (doctorId) REFERENCES doctors(id),**

**INDEX (patientId),**

**INDEX (doctorId)**

**);**

**-- Table: medical\_records**

**CREATE TABLE IF NOT EXISTS medical\_records (**

**recordId VARCHAR(255) PRIMARY KEY,**

**patientId VARCHAR(255),**

**diagnosis TEXT,**

**treatment TEXT,**

**FOREIGN KEY (patientId) REFERENCES patients(id)**

**);**

**-- Insert sample data into the patients table**

**INSERT INTO patients (id, name, age, gender) VALUES**

**('P001', 'John Doe', 35, 'Male'),**

**('P002', 'Jane Smith', 45, 'Female'),**

**('P003', 'Michael Johnson', 28, 'Male');**

**-- Insert sample data into the doctors table**

**INSERT INTO doctors (id, name, specialty) VALUES**

**('D001', 'Dr. Smith', 'Cardiology'),**

**('D002', 'Dr. Johnson', 'Orthopedics');**

**-- Insert sample data into the appointments table**

**INSERT INTO appointments (id, patientId, doctorId, appointmentDate) VALUES**

**('A001', 'P001', 'D001', '2024-04-15 10:00:00'),**

**('A002', 'P002', 'D002', '2024-04-16 11:00:00'),**

**('A003', 'P003', 'D001', '2024-04-17 12:00:00');**

**-- Insert sample data into the medical\_records table**

**INSERT INTO medical\_records (recordId, patientId, diagnosis, treatment) VALUES**

**('R001', 'P001', 'Hypertension', 'Prescription medication and lifestyle changes'),**

**('R002', 'P002', 'Fractured arm', 'Surgery and physical therapy'),**

**('R003', 'P003', 'Bronchitis', 'Antibiotics and rest');**

**-- Select all data from the patients table**

**SELECT \* FROM patients;**

**-- Select all data from the doctors table**

**SELECT \* FROM doctors;**

**-- Select all data from the appointments table**

**SELECT \* FROM appointments;**

**-- Select all data from the medical\_records table**

**SELECT \* FROM medical\_records;**

**Output of above DATABASE:**

**A screenshot of a computer

Description automatically generated**

**Table Description:**

* **Patients Table:** This table is designed to store the personal details of patients. Each patient has a unique identifier (id) which is used to link their medical records and appointments. The table also includes the patient’s name, age, and gender.
* **Medical Records Table:** This table contains the medical history of patients. Each record has a unique identifier (recordId) and is linked to a patient through the patientId. The table records the diagnosis and treatment details for each medical entry.
* **Appointments Table:** This table tracks the appointments scheduled between patients and doctors. It includes a unique identifier for each appointment (id), the patient’s identifier (patientId), the doctor’s identifier (doctorId), and the date and time of the appointment (appointmentDate).
* **Doctors Table:** This table holds information about the doctors. Each doctor has a unique identifier (id), along with their name and specialty. This table is linked to the appointments table to show which doctor is seeing which patient.  
    
  These tables are interconnected through foreign keys that establish relationships, ensuring that the data is consistent and that the integrity of the database is maintained. For example, the patientId in the medical\_records and appointments tables references the id in the patients table, creating a link between a patient’s records, appointments, and personal details. Similarly, the doctorId in the appointments table references the id in the doctors table, connecting doctors to their appointments.  
    
  The design of these tables allows for efficient management and retrieval of medical records, patient details, and appointment schedules within a healthcare database management system.

**Normalization of Tables:**

* **First Normal Form (1NF):**
* Each table cell should contain a single value.
* Each record needs to be unique.
* **Second Normal Form (2NF):**
* The table must be in 1NF.
* All non-key attributes must be fully functional dependent on the primary key.
* **Third Normal Form (3NF):**
* The table must be in 2NF.
* There must be no transitive functional dependencies (i.e., non-key attributes must be dependent only on the primary key).

**Let’s apply these rules in the tables:**

* **Patients Table:**
* Meets 1NF: Each cell holds a single value.
* Meets 2NF: No composite primary key, so all attributes depend on the primary key directly.
* Meets 3NF: No transitive dependencies, all attributes depend solely on the primary key.
* **Medical Records Table:**
* Meets 1NF: Each cell holds a single value.
* May or may not meet 2NF: Depends on whether patientId is part of the primary key. If it's part of a composite key with recordId, then it's in 2NF. Needs further analysis.
* Likely meets 3NF: No indication of non-key attributes depending on other non-key attributes. Seems to be in 3NF.
* **Appointments Table:**
* Meets 1NF: Each cell holds a single value.
* Meets 2NF: Non-key attribute (appointmentDate) depends on the primary key (id).
* Meets 3NF: No transitive dependencies.
* **Doctors Table:**
* Meets 1NF: Each cell holds a single value.
* Meets 2NF: No composite primary key, so all attributes depend on the primary key directly.
* Meets 3NF: No transitive dependencies, all attributes depend solely on the primary key.

**Normalized SQL code :-**

**-- Patients Table**

**CREATE TABLE patients (**

**id VARCHAR(255) PRIMARY KEY,**

**name VARCHAR(255),**

**age INT,**

**gender VARCHAR(50)**

**);**

**-- Medical Records Table**

**CREATE TABLE medical\_records (**

**recordId VARCHAR(255) PRIMARY KEY,**

**patientId VARCHAR(255),**

**diagnosis TEXT,**

**treatment TEXT,**

**FOREIGN KEY (patientId) REFERENCES patients(id)**

**);**

**-- Appointments Table**

**CREATE TABLE appointments (**

**id VARCHAR(255) PRIMARY KEY,**

**patientId VARCHAR(255),**

**doctorId VARCHAR(255),**

**appointmentDate TIMESTAMP,**

**FOREIGN KEY (patientId) REFERENCES patients(id),**

**FOREIGN KEY (doctorId) REFERENCES doctors(id)**

**);**

**-- Doctors Table**

**CREATE TABLE doctors (**

**id VARCHAR(255) PRIMARY KEY,**

**name VARCHAR(255),**

**specialty VARCHAR(255)**

**);**

**-- Inserting data into Patients Table**

**INSERT INTO patients (id, name, age, gender) VALUES**

**('P001', 'John Doe', 35, 'Male'),**

**('P002', 'Jane Smith', 45, 'Female'),**

**('P003', 'Michael Johnson', 28, 'Male');**

**-- Inserting data into Medical Records Table**

**INSERT INTO medical\_records (recordId, patientId, diagnosis, treatment) VALUES**

**('MR001', 'P001', 'Hypertension', 'Prescribed medication and lifestyle changes'),**

**('MR002', 'P002', 'Diabetes Type 2', 'Insulin therapy and dietary changes'),**

**('MR003', 'P003', 'Influenza', 'Bed rest and antiviral medication');**

**-- Inserting data into Appointments Table**

**INSERT INTO appointments (id, patientId, doctorId, appointmentDate) VALUES**

**('A001', 'P001', 'D001', '2024-04-15 10:00:00'),**

**('A002', 'P002', 'D002', '2024-04-16 14:30:00'),**

**('A003', 'P003', 'D003', '2024-04-17 11:15:00');**

**-- Inserting data into Doctors Table**

**INSERT INTO doctors (id, name, specialty) VALUES**

**('D001', 'Dr. Emily Brown', 'Cardiology'),**

**('D002', 'Dr. David Miller', 'Endocrinology'),**

**('D003', 'Dr. Sarah Johnson', 'General Practitioner');**

**SELECT name, age**

**FROM patients;**

**SELECT p.name AS patient\_name, m.diagnosis, m.treatment**

**FROM patients p**

**JOIN medical\_records m ON p.id = m.patientId;**

**SELECT name, specialty**

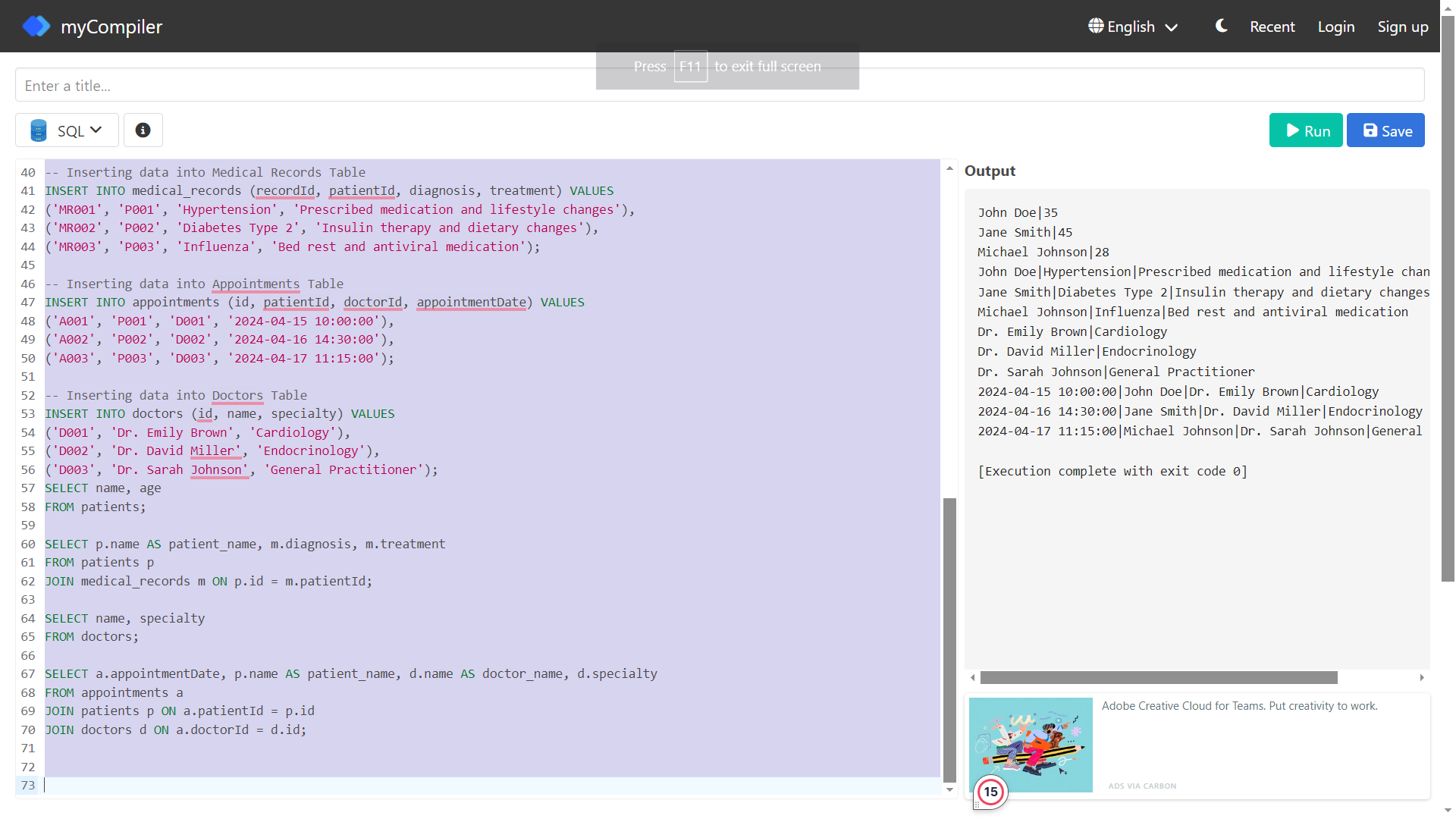
**FROM doctors;**

**SELECT a.appointmentDate, p.name AS patient\_name, d.name AS doctor\_name, d.specialty**

**FROM appointments a**

**JOIN patients p ON a.patientId = p.id**

**JOIN doctors d ON a.doctorId = d.id;**

**Output: **

**Justifying 3NFs followed in above code:**

The table structure provided follows the principle of 3NF. Let’s break it down:

* **First Normal Form (1NF):**
* Each table has a unique identifier (primary key), and each field contains atomic values.
* There are no repeating groups or arrays within the fields.

**Example:** In the patients table, each patient's information is stored in separate rows, and each field (id, name, age, gender) contains single values.

* **Second Normal Form (2NF):**
* Each non-key attribute (field) is fully dependent on the entire primary key.

**Example:** In the medical\_records table, the diagnosis and treatment are attributes fully dependent on the recordId primary key, which identifies unique medical records.

* **Third Normal Form (3NF):**
* No transitive dependencies exist. That is, no non-key attribute depends on another non-key attribute.

**Example:** In the appointments table, the appointment date (appointmentDate) depends only on the appointment (id), and the doctor's specialty (specialty) in the doctors table depends solely on the doctor's ID (id), not on any other non-key attribute.

**SQL Query created for the table are :**

1. **Query to select name and age from patients table**

**SELECT name, age**

**FROM patients;**

**2.Query to select patient name, diagnosis, and treatment from patients and medical\_records tables:**

**SELECT p.name AS patient\_name, m.diagnosis, m.treatment**

**FROM patients p**

**JOIN medical\_records m ON p.id = m.patientId;**

**3.Query to select name and specialty from doctors table:**

**SELECT name, specialty**

**FROM doctors;**

**4.Query to select appointmentDate, patient name, doctor name, and doctor specialty from appointments, patients, and doctors tables:**

**SELECT a.appointmentDate, p.name AS patient\_name, d.name AS doctor\_name, d.specialty**

**FROM appointments a**

**JOIN patients p ON a.patientId = p.id**

**JOIN doctors d ON a.doctorId = d.id;**

**Views that are created using the TABLES:-**

**1.Patient Medical Records View**: This view combines patient information with their medical records.

**CREATE VIEW patient\_medical\_records AS**

**SELECT p.name AS patient\_name, p.age, p.gender, m.diagnosis, m.treatment**

**FROM patients p**

**JOIN medical\_records m ON p.id = m.patientId;**

**2.Doctor Appointments View**: This view combines appointment information with doctor details.

**CREATE VIEW doctor\_appointments AS**

**SELECT a.appointmentDate, p.name AS patient\_name, d.name AS doctor\_name, d.specialty**

**FROM appointments a**

**JOIN patients p ON a.patientId = p.id**

**JOIN doctors d ON a.doctorId = d.id;**

**3.Patients by Gender View**: This view groups patients by their gender and provides a count.

**CREATE VIEW patients\_by\_gender AS**

**SELECT gender, COUNT(\*) AS total\_patients**

**FROM patients**

**GROUP BY gender;**

**4.Patients by Age Group View**: This view groups patients into age ranges and provides a count.

**CREATE VIEW patients\_by\_age\_group AS**

**SELECT**

**CASE**

**WHEN age BETWEEN 0 AND 20 THEN '0-20'**

**WHEN age BETWEEN 21 AND 40 THEN '21-40'**

**WHEN age BETWEEN 41 AND 60 THEN '41-60'**

**ELSE 'Above 60'**

**END AS age\_group,**

**COUNT(\*) AS total\_patients**

**FROM patients**

**GROUP BY age\_group;**

**5.Doctors by Specialty View**: This view groups doctors by their specialty and provides a count.

**CREATE VIEW doctors\_by\_specialty AS**

**SELECT specialty, COUNT(\*) AS total\_doctors**

**FROM doctors**

**GROUP BY specialty;**