**Medical Appointment Booking Platform**

Applied Database Technologies (DSCI - D532)

Database Schema Description

by

**Ameya Dattaram Parab**

**Trishna Patil**

Indiana University Bloomington

**ER Diagram:**

Diagram

Description automatically generated

The diagram represents the Entity-Relationship diagram for our Medical Appointment Booking Platform project. In this diagram, entities are depicted as real-world objects that possess their own attributes and properties, represented as rectangles. Solid rectangles denote strong entities, while weak entities are indicated with dotted lines. Specifically, Appointments, Doctor Specialties, and Doctor Organizations are weak entities in our diagram. Relationships are portrayed as lines connecting entities, indicating how they are interrelated. Essentially, the ER diagram illustrates how the various entities in the system are associated with one another.

**Database Normalization:**

The raw dataset provided contains 31 columns that exhibit redundancy. Therefore, to ensure data consistency and facilitate database maintenance, it is necessary to decompose the dataset into smaller entities by establishing relationships between them. This can be achieved by applying normalization techniques to the dataset, which will eliminate data redundancy and improve data consistency. By breaking down the dataset into smaller, more focused entities, we can ensure that each entity has a clear purpose and is represented by a unique table in the database. This will also make it easier to manage the dataset and perform complex queries on it, leading to better insights and more informed decision-making.

* **1 NF:**

To convert the data into first normal form, it is necessary to ensure that each column in the table contains atomic values. In this dataset, each doctor has a primary specialty and multiple secondary specialties, which results in redundancy. To address this issue, the database can be transformed by creating a Specialties table that lists each individual specialty. Similarly, a Schools table and a Contacts table can be created to store the medical school names and phone numbers of the doctors, respectively.

* **2 NF:**

All non-key attributes in a table should be dependent on the primary key of that table. In this case, the tables have been split to store relevant information about doctors, with the DoctorID serving as the primary key. The Education table includes educational credentials of the doctors, referenced by the DoctorID. Additionally, it is related to the Schools table by the SchoolID key, which allows for easy retrieval of information about the schools attended by each doctor.

* **3NF:**

To ensure that the data does not have any transitive dependency between non-key columns, it is important to properly structure the database. In this dataset, the information includes addresses for each doctor, some of whom have individual clinics while others belong to organizations. To normalize the data into 3NF form, a separate table is created to store all the addresses with unique Address IDs. These IDs are then referenced in intermediate tables created to store information about the clinics and organizations. These tables also reference the Doctor tables through Doctor IDs, allowing for easy retrieval of all relevant data.

**Entities:**

There are various entities present in the database which are described below:

1. **Doctors**

This table contains the basic information about doctors. Doctors have attributes such as FirstName, MiddleName, LastName and Gender. DoctorID is the primary key for this table.

1. **Patients**

This table contains the basic information about patients. Patients have attributes such as FirstName, LastName, Gender and Email address. PatientID is the primary key for this table.

1. **Appointments**

In this table the data about the appointments booked will be stored. This is a weak entity consisting of attributes such as AppointmentDate and AppointmentTime. It has a primary key of AppointmentID and 2 foreign keys as DoctorID and PatientID.

1. **Specialties**

Individual specialties are stored in this table with the SpecialtyName and a primary key of SpecialtyID.

1. **Doctor Specialties**

The specialties which the doctors have are stored in this table. This is a weak entity and has an attribute of isPrimary and 2 foreign keys namely DoctorID and SpecialtyID.

1. **Organizations**

The table is designed to store information about the organizations/ hospitals. This has a primary key of OrganizationID, attributes are OrganizationName and Members and one foreign key of AddressID.

1. **Doctor Organizations**

It acts as a mapping between doctor and the organizations. This is a weak entity. It has an attribute of isHospital and 2 foreign keys of DoctorID and OrganizationID.

1. **Doctor Clinics**

Similar to the above table this entity maps doctors to their clinics. It has 2 foreign keys – DoctorID and AddressID.

1. **Addresses**

A separate table to store all the addresses in the dataset. It has attributes such as AddressLine1, AddressLine2, City, State and Zip Code. And primary key of AddressID.

1. **Services**

The services provided by the doctors are represented by this entity. Services entity has attributes such as TeleConsultation, IndividualMedicare and GroupMedicare. And foreign key is DoctorID.

1. **Contacts**

This table is designed to store the contact information such as PhoneNumber and DoctorID as the foreign key. There is a possibility of data changing or growing in future with possible attributes such as Email, Socials, etc. hence, it is better to store these information in a separate entity.

1. **Education**

The data about doctor’s educational background is stored in this entity. Education has credentials and graduation year as it’s attributes and 2 foreign keys – DoctorID and SchoolID.

1. **Schools**

The table is broken down to from the Education so that individual schools information can be maintained in it. This has MedicalSchoolName as the attribute and SchoolID as the primary key.

**Relationships:**

* Doctors and Patients have a 1:N relationship; i.e. one doctor can have many patients.
* Doctors and Services have a 1:1 relationship with each other; i.e. one doctor can provide only one type of service.
* Doctors and Specialties have a 1:N relationship with one another and it is optional. So a doctor may or may not have a specialty or can have many specialties as well.
* Similarly, one organization can have many doctors, but each doctor can belong to only one organization.
* One doctor can go to multiple doctor clinics. Doctors can also have many contact numbers.
* Also, Organizations and Doctor Clinics have a 1:1 relationship with Addresses.
* Doctors have a 1:1 relationship with Education and Schools have a 1:N relationship with Education.