# Database Design and Development
# on
# Medical Management

-Mann Trivedi

## Table Of Content

## 1. Introduction :

- In this article we are looking to the public health services in the hospitals.
- Public health is a multidisciplinary field focused on improving the health and well-being of populations.
- It encompasses a wide range of activities aimed at preventing disease, promoting health, and extending life through organized efforts and informed choices of society, organizations, communities, and individuals.
- It encompasses a wide range of activities aimed at preventing disease, promoting health, and extending life through organized efforts and informed choices of society, organizations, communities, and individuals.

## 2. Mission :

- Our Mission is to ensure the center continues to provide high-quality care, the database project aims to streamline data and analysis. By improving the availability and accuracy of healthcare data, the center can make more informed decisions to optimize resource allocation and enhance patient outcomes.

## 3. Objectives :

- Our main objectives are to Collect historical data from legacy storage formats and create a pipeline to store it in a more organized database to identify probable causes of resource misuse. Use newly organized to design predictive algorithms and improve diagnosis efficiency.
- Record doctor schedules and resource usage logs in a separate database to improve efficiency and reduce wait times for patients.

## 4. Organization :

- This medical centre was opened its doors to patients in 1994 .
- It is located in the city of Calgary, Alberta, Canada, on the shores of the Glen-more Reservoir and is administered by Alberta Health Services and formerly by the Calgary Health Region.
- The hospital contains over 650 beds and provides medical and surgical services to Calgary and Southern Alberta.
- It includes a 24-hour emergency department, an intensive care unit (ICU), as well as day surgery units.

## 5. Database Design :

- Data design in databases refers to the process of defining how data will be stored, organized, and accessed within a database system.
- To gathering the historical data and store it into the Relational database , we need to create the database of specific clinic .

### 5.1 Why Tables and Fields are important in database ?

- Tables and fields form the backbone of a database, facilitating efficient data storage, retrieval, and management.
- Tables are used for organization of data , Tables can be related to one another through foreign keys, allowing for complex data relationships and enabling the representation of real-world scenarios, data integrity , it enables users to retrieve and manipulate data efficiently.
- Fields are used for attributes of data , define data of each attribute , fields allow for searching and filtering data based on specific criteria , Fields can have constraints (like NOT NULL or UNIQUE) that help maintain data quality by ensuring only valid and required data is stored.
- To create database for clinic , we need to create the separate tables for patient , doctor , department , appointment , doctor’s schedule and clinic in these tables we need to insert the values with some specific data types .
- Each Table has Primary key and Foreign key (which give reference of another table) and have relationship between the tables For Example One To One , One to Many and Many To One.

## 5.2 Fields in database Table :

### 5.2.1. Clinic

| Fields Name | Data Type | Description |
|---|---|---|
| Clinic_Id (PK) | Integer | Make attribute as primary key with integer datatype |
| Clinic_Name | Var char (200) | It has clinic name with mixture of variable and character but it set the word limit till 200. |
| Address | Var char (200) | It mention address of specific clinic with same as above datatype |
| Department_Id (Fk) | Integer | It give reference to the departm-ent table |

### 5.2.2. Doctor

| Fields Name | Data Type | Description |
|---|---|---|
| Doctor_Id (PK) | Integer | It assign as primary key for each doctor |
| Name | Var char(100) | It Specify the doctor name |
| Specialization | Var char(100) | In which field doctor is |

| | | |
|---|---|---|
| | | special for diagnose |
| Clinic_Id (FK) | Integer | It gives reference to doctor from which clinic |
| No of_patient_per day | Integer | It count the no of patient per day the doctor has attended |

### 5.2.3. Department

| Fields Name | Data Type | Description |
|---|---|---|
| Department_Id (PK) | Integer | It indicates the specific department with the Id |
| Intensive_care | var char (50) | It is the department for particular service |
| Emergency_care | var char (50) | It is the department for particular service |
| Surgery | var char (50) | It is the department for particular service |
| Pediatrics | var char (50) | It is the department for particular service |

### 5.2.4. Doctor Schedule Table

| Field Name | Data Type | Description |
|---|---|---|
| Doctor id | INTEGER | Foreign key, links to the doctor |
| Clinic id | INTEGER | Foreign key, links to the clinic |
| Monday | TIME | Doctor's available time on Monday |
| Tuesday | TIME | Doctor's available time on Tuesday |
| Wednesday | TIME | Doctor's available time on Wednesday |
| Thursday | TIME | Doctor's available time on Thursday |
| Friday | TIME | Doctor's available time on Friday |
| Saturday | TIME | Doctor's available time on Saturday |
| Sunday | TIME | Doctor's available time on Sunday |
| Clinic id | INTEGER | Foreign key, links to the clinic |
| Department id | INTEGER | Foreign key, links to the department |

#### 5.2.5.Appointment

| Fields Name | Data Type | Description |
|---|---|---|
| Appointment_Id (PK) | Integer | It gives specific appointment Id based on patient information |
| Clinic_Id (FK) | Integer | It gives the information about the which clinic patient wants to prefer |
| Patient_Id (FK) | Integer | It generate specific id for each patient |
| Doctor_Id (FK) | Integer | It assign specific doctor as per patient diseases |
| Scheduled_time | Timestamp | It defines the schedule time |

| | | |
|---|---|---|
| | | for patient when the doctor is available |
| Diagnosis | Var char (200) | This field defines the diagnose of the patient |

### 5.2.6.Patient

| Fields Name | Data Type | Description |
|---|---|---|
| Patient_Id (PK) | Integer | It indicates specific patient id for each patient |
| Appointment_Id (FK) | Integer | It gives information about appointment |
| Doctor_Id (FK) | Integer | It indicates which attend the patient |
| Name | Var char (50) | It gives patient name |
| Gender | Var char (50) | It shows the gender of patient |
| Age | Integer | It indicates the age of the patient |

### 5.3 Relationship

- In a database, a relationship refers to how data in one table is associated with data in another table. This is fundamental in relational databases, which use tables to store data.

### 5.3.1 why Relationship is important ?

- Relationship is important because of organize data logically, making it easier to understand and manage. Related data can be grouped, which enhances data retrieval and manipulation.

### 5.3.2 How you used relationship on your case study ?

- There are four types or relationships.
- One to One , One to Many , Many to One , Many to Many .
- I have applied these three relationship between tables of clinic , department , doctor ,department , appointment and patient tables .

### 5.3.3. One to Many Relationships :

- **Clinic** ⟶ **Department :** One clinic can have multiple departments, but each department belongs to only one clinic.

  Clinic.Clinic id → Department.Clinic id

- **Clinic** ⟶ **Doctor** One clinic can have many doctors, but each doctor works in only one

- **Clinic** ⟶ **Appointment :**One clinic can have many appointments, but each appointment takes place in only one clinic.

  Clinic.Clinic id → Appointment.Clinic id

- **Doctor** ⟶ **Appointment**: One doctor can have many appointments, but each appointment is assigned to only one doctor.

  Doctor.Doctor id → Appointment.Doctor id

- **Patient** ⟶ **Appointment**: One patient can have multiple appointments, but each appointment is linked to only one patient.

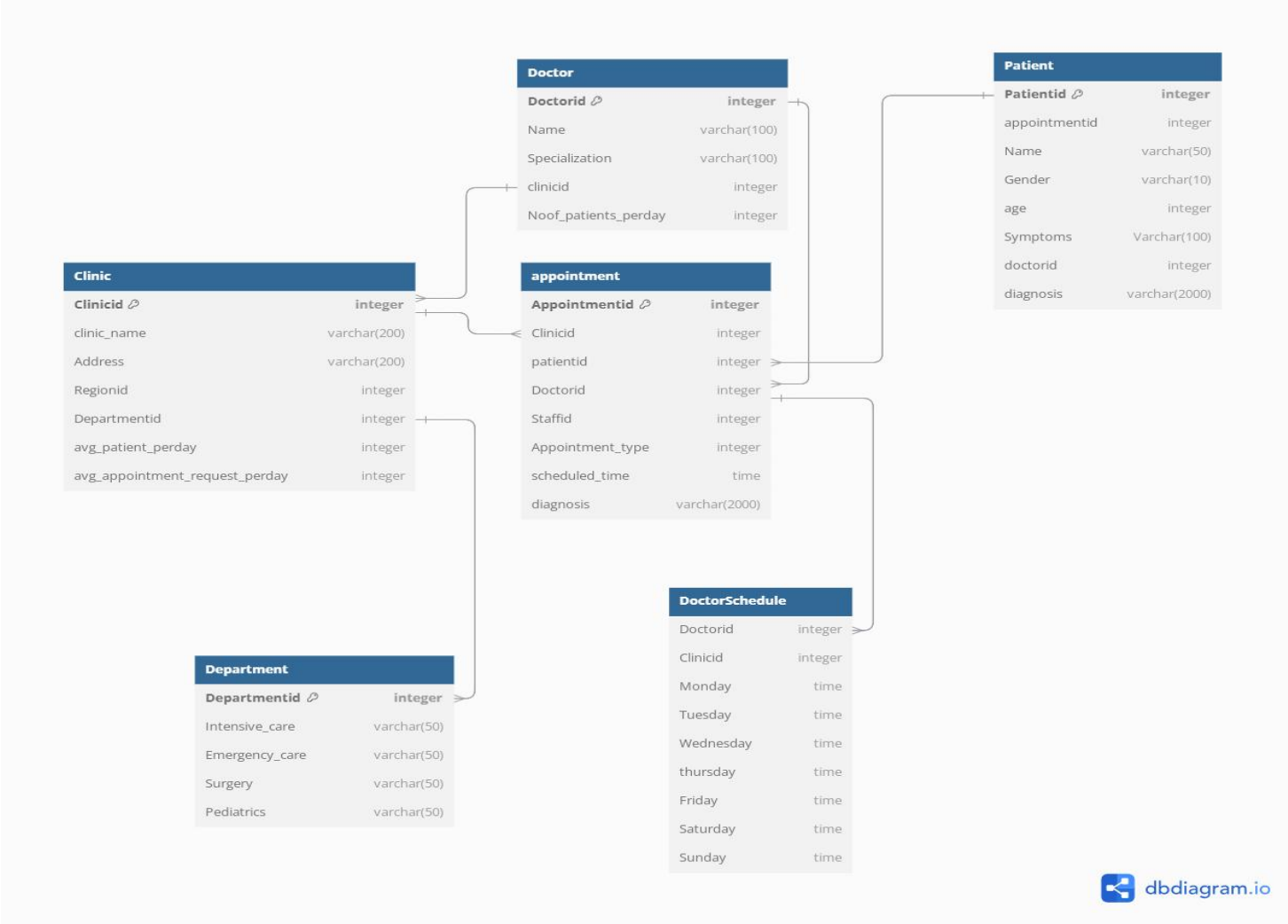
  Patient.Patient id → Appointment.Patient id

**Appointment**: ⟶ **Doctor Schedule :** One appointment have multiple doctors to attend the patient

Appointment . Doctor.id → Doctor schedule . Doctor.id

## 5.4 E R Diagram

- In this E R Diagram the multiple tables are created and add some fields as per the clinic database.
- The source table for the medical center is Appointment tables because from that tables we can get the information about the patient ,diagnose , prefer doctor and schedule time.
- I mention relationships between the tables to understand and manage the data.
- By clearly mapping out the data structure, the ER diagram facilitates efficient database design, improving the clinic's operational efficiency and enhancing patient care.




# 6 Database Development

## 6.1.Clinic Table

```
CREATE TABLE Clinic (
Clinic id INTEGER PRIMARY KEY ,
 clinic_name VARCHAR(100),
Address VARCHAR(200),
Department Id Integer Foreign key );
```

## 6.2. Department Table

```
CREATE TABLE Department (
    Department id INTEGER PRIMARY KEY,
    Clinic id INTEGER,
```

```
    Department_name VARCHAR(100),
    FOREIGN KEY (Clinic id) REFERENCES Clinic(Clinic id)
);
```

### 6.3. Doctor Table

```
CREATE TABLE Doctor (
    Doctor id INTEGER PRIMARY KEY,
    Name VARCHAR(100),
    Specialization VARCHAR(100),
    Clinic id INTEGER,
    Department id INTEGER,
    No of_patients_per day INTEGER,
    FOREIGN KEY (Clinic id) REFERENCES Clinic(Clinic id),
    FOREIGN KEY (Department id) REFERENCES Department(Department id)
```

### 6.4. Patient Table

```
CREATE TABLE Patient (
    Patient id INTEGER PRIMARY KEY,
    Name VARCHAR(100),
    Gender VARCHAR(10),
    Age INTEGER,
    Symptoms VARCHAR(200),
    Diagnosis VARCHAR(2000)
);
```

### 6.5. Appointment Table

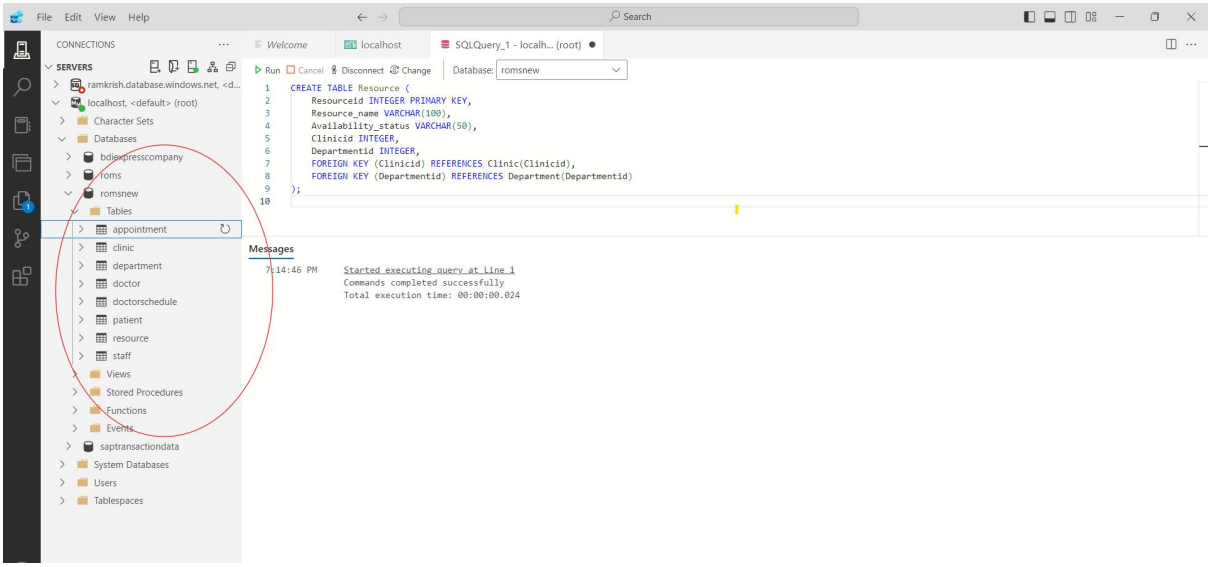
```
CREATE TABLE Appointment (
    Appointment id INTEGER PRIMARY KEY,
    Clinic id INTEGER,
    Patient id INTEGER,
    Doctor id INTEGER,
    Appointment_type VARCHAR(50),
    Scheduled_time TIME,
    Diagnosis VARCHAR(2000),
    FOREIGN KEY (Clinic id) REFERENCES Clinic(Clinic id),
    FOREIGN KEY (Patient id) REFERENCES Patient(Patient id),
    FOREIGN KEY (Doctor id) REFERENCES Doctor(Doctor id)
);
```

## 7.Conclusion :

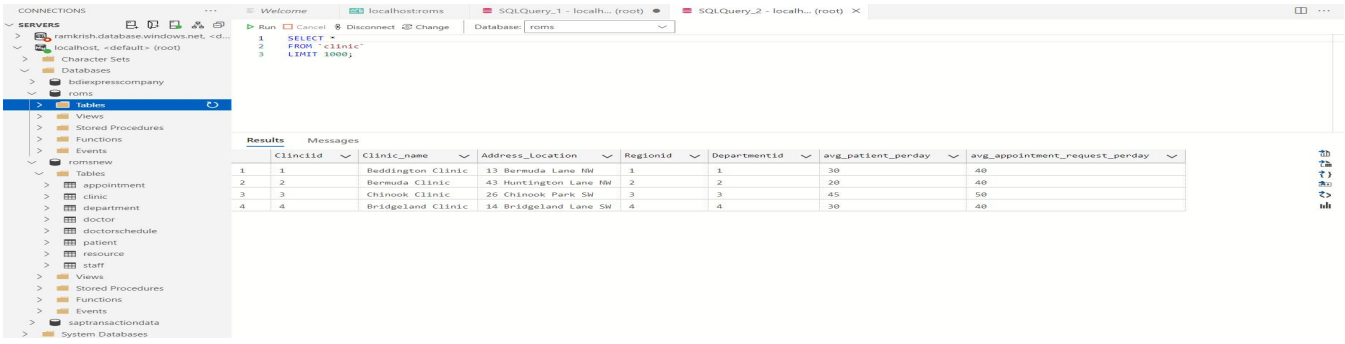
- In this case study we create the database and store historical data into the database which is useful for their optimize operation and to achieve our objectives :
- Reduced wait times
- Optimized resource usage
- Data-driven decision making
- Enhanced patient care and outcomes

## 8.Appendix :

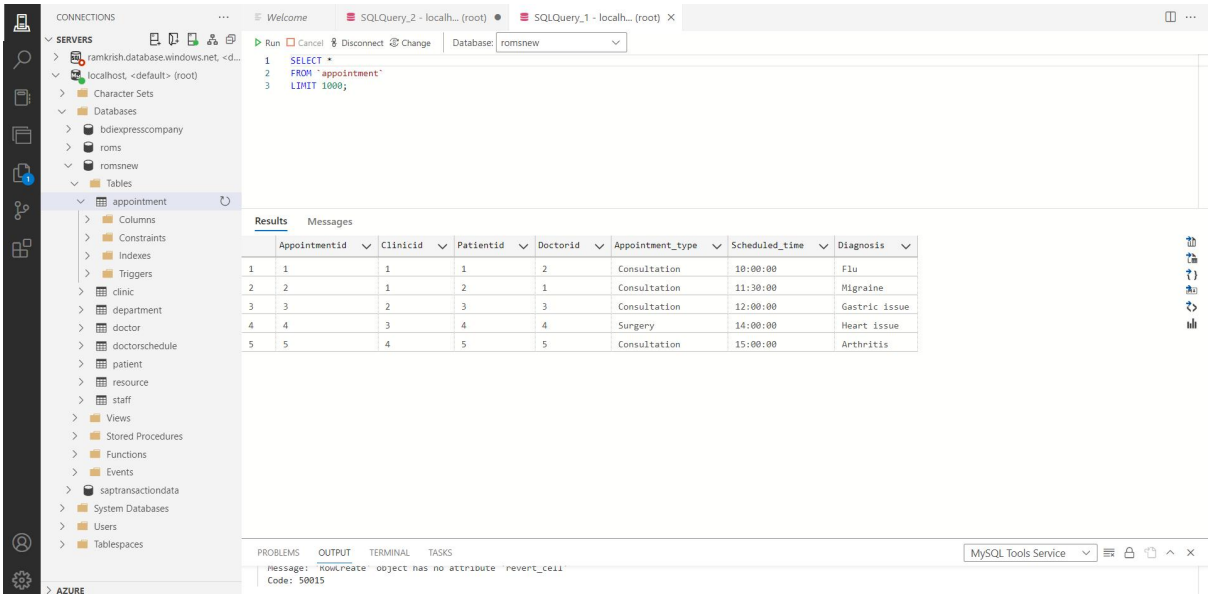
### 8.1 New Database and tables created




### 8.2 Values in Table




### 8.3 Values in the Appointment Table




## 8.4 Inner Join Query to see Appointment