DATABASE DESIGN DOCUMENT FOR DENTAL OFFICE

**Table of Contents**

[***Section 1: Project Overview 2***](#_gjdgxs)

[***Section 2: Business Rules 2***](#_30j0zll)

[**2.1 In-Scope Business Rules 2**](#_1fob9te)

[**2.2 Out-Of-Scope Business Rules 4**](#_3znysh7)

[***Section 3: Entity Description 4***](#_2et92p0)

[**3.1 Patient: 4**](#_tyjcwt)

[3.1.1 PatientInfo: 4](#_3dy6vkm)

[3.1.2 InsuranceInfo: 5](#_1t3h5sf)

[3.1.3 MasterMedicalDentalCondition: 5](#_4d34og8)

[3.1.4 MedicalHistory: 6](#_2s8eyo1)

[3.1.5 DentalHistory: 7](#_17dp8vu)

[3.1.6 MasterAllergy: 7](#_3rdcrjn)

[3.1.7 PatientAllergyMapping: 8](#_26in1rg)

[3.1.8 MasterTreatment: 8](#_lnxbz9)

[3.1.9 MasterToothDetails: 8](#_35nkun2)

[3.1.10 DentalRecordDetail: 9](#_1ksv4uv)

[3.1.11 DentalRecord: 9](#_44sinio)

[3.1.12 PatientMedicalConsent: 10](#_2jxsxqh)

[**3.2 Dentist: 10**](#_z337ya)

[3.2.1 DentistInfo: 11](#_3j2qqm3)

[3.2.2 DentistType: 11](#_1y810tw)

[3.2.3 Prescription: 11](#_4i7ojhp)

[3.2.4 Medication: 12](#_2xcytpi)

[**3.3 Office: 12**](#_1ci93xb)

[3.3.1 Office: 12](#_3whwml4)

[3.3.2 Room: 13](#_2bn6wsx)

[3.3.3 RoomAvailability: 14](#_qsh70q)

[3.3.4 EquipmentDetails: 14](#_3as4poj)

[3.3.5 MasterAddress: 15](#_1pxezwc)

[3.3.6 MasterZipCode: 15](#_49x2ik5)

[**3.4 Appointment: 15**](#_2p2csry)

[3.4.1 DoctorAvailability: 16](#_147n2zr)

[3.4.2 AppointmentDetails: 16](#_3o7alnk)

[***Section 4: Entity Relationship Diagram 17***](#_23ckvvd)

# Section 1: Project Overview

In this project, we are working on the database design of the Dental office patient care system. The main objective of the project is to capture the treatment of patients in a dental office. We will be tracking the patient's dental history, appointments, treatments, etc. Within this database design, patients will be able to book an appointment with a different type of dentist available at the particular office location.

We need to think of multiple entities which will play a key role in this database design. Detailed entity description is provided further in the document. Business rules also define the structure of the database.

# Section 2: Business Rules

While designing any database the business rules play an important role to structure the database. We have to define the business rules based on the real-world scenario of the dental office and how we will incorporate every element in our database. Here our focus is on patient care given at the dental office. So based on that there could be In-Scope business rules and Out-Of-Scope business rules. Let’s discuss all the business rules below.

## 2.1 In-Scope Business Rules

## These rules will serve the main purpose of the database. We will consider the below rules while designing tables and their attributes and relationships.

* **A new patient must fill the intake form and provide all the details**

For a patient to be added to the record in the dental clinic, he/she must fill the complete intake form which will have basic details like the name, address, phone number, etc.

* **A patient should mention his/her medical and dental history in the form**

Patient is required to fill the data of his/her medical and dental condition. This will give a better idea to a doctor regarding a patient's current condition and the doctor can treat them accordingly.

* **A patient must mandatorily fill allergies form before every visit**

After fixing an appointment at the dental clinic, a patient must mandatorily fill a form that indicates any allergy that the patient may have. This will help the dentist consult that patient to get more information of any medication/treatment that the patient can have an allergic reaction to and help avoid the same.

* **Dental record must be updated after every visit for the patient**

After each patient visit, the dentist who consulted or treated the patient needs to fill in all the details of treatment to maintain the patient’s dental history.

* **Every appointment must be mapped with one or more dentist in a particular room**

A patient can be scheduled for a procedure/treatment that may require more than one dentist. For instance, a patient can have a tooth replacement procedure and cleaning procedure that requires a prosthodontist and a hygienist. Also, all the appointments are scheduled to take place in a particular room.

* **Tooth Treatment details of the patient should be tracked**

During every appointment the doctor could work on one more tooth. There could be multiple tasks performed on different teeth. So, keeping track of all those activities could be helpful to the doctor for the next appointment of that patient.

* **Consent form must be filled out by a patient every 6 months**

The purpose of this is to ensure that the patient is fully aware of the treatment/procedure being performed and has provided consent for the same.

* **Prescription should be mapped to patient and his/her appointment**

Prescriptions given by the doctor should be mapped with the patient's appointment. This can be used to give it to pharmacies and can be used to track the previous history of prescription.

* **Appointments are based on room availability as well as doctor availability**

We should be tracking room and doctor availability in better format/design. This is important data because based on availability of room and doctor we should be able to book the appointment for patients.

* **A dentist can have availability at different offices on everyday basis**

The same dentist can be available at different offices on a daily basis. Dentist’s date, start and end time, office in which doctor is available on that date will be tracked.

* **In one appointment different treatments can be performed on different teeth.**

During one appointment a dentist can perform different treatment on different teeth, and we will track that data in database teeth wise and Dental record wise.

* **It’s not mandatory for patients to have insurance. He/she can have zero or multiple insurance**

We will track insurance details of the patient. In that case patients might not have any insurance or they could have one or more Insurance. We will track all that data in the database.

* **While booking an appointment office in which a dentist is available and room and office should match.**

At the time of booking an appointment there should be a check that the room in which the doctor is supposed to work on a particular date should match the office that is mentioned in the Doctor's availability schedule.

* **Patients can book an appointment slot of half an hour. They can book multiple slots if needed.**

In the database we are maintaining a slot availability system. Each slot will be of half an hour. If a patient's treatment is big then he can book multiple continuous slots.

* **To book an appointment all equipments in the room should be available**

We will track equipment details and status in the room. In this case, a room will store the Equipment’s working condition whether it is working or not.

## 2.2 Out-Of-Scope Business Rules

* **Employees other than Dentists will not be considered in the database.**

The database focuses on the patient care system at the dental office. So, we need to consider only those employees who are involved in the process.

* **Pharmacy details are not required right now**

Pharmacy could be part of the complete office system, but as our main focus is on activities happening at the office which are limited to dentists and patients.

# 

# Section 3: Entity Description

Entity in DBMS can be a real-world object with an existence. There are multiple entities we can consider for this data model. Below we will discuss some entities and we will expand one of the entities in detail.

## 3.1 Patient:

Patient data cluster will contain different important entities. Some entities that come under this are PatientInfo and InsuranceInfo, etc. Data for all entities under the patient cluster will be stored at the time of patient registration. We will look in detail about all the entities below.

### 3.1.1 PatientInfo:

This table consists of important personal information of a patient that we will require in managing patient care. This data will be acquired when Patient will fill up the form given.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| PatientId | System generated unique Id to register new patient | int | Yes | PK |
| AddressId | FK column value coming from MasterAdddress table | int | Yes | FK |
| PrimaryDentistId | Id of dentist who is handling the patient case. | int | No | FK |
| FirstName | First name of the patient | varchar(50) | Yes |  |
| LastName | Last name of the patient | varchar(50) | Yes |  |
| Gender | Gender of the patient | char(10) | Yes |  |
| Dob | Date of birth of patient | date | Yes |  |
| Email | Email Id of patient | varchar(50) | Yes |  |
| ContactNo | Phone Number of patient | varchar(15) | Yes |  |
| EmergencyContactNo | Emergency Phone No given by patient | varchar(15) | Yes |  |
| DateOfRegistration | Date on which first time patient was registered. | date | No |  |

Example:

| **PatientId** | **AddressId** | **PrimaryDentistId** | **FirstName** | **LastName** | **Gender** |
| --- | --- | --- | --- | --- | --- |
| 21 | 1 | 51 | Rohit | Sharma | Male |
| 22 | 2 | 52 | Sachin | Tendulkar | Male |
| 23 | 3 | 53 | Virat | Kohli | Male |

| **DOB** | **Email** | **ContactNo** | **EmergencyContactNo** |
| --- | --- | --- | --- |
| 1990-01-01 | rohitsharma@gmail.com | 6171234567 | 6172345678 |
| 1991-06-01 | viratkohli@gmail.com | 6173456780 | 6174567890 |
| 1993-07-01 | viratkohli@gmail.com | 6174567890 | 6175678901 |

### 3.1.2 InsuranceInfo:

This table consists of insurance information of a patient. In this table we are also tracking SubscriberName and Patient's relationship with subscriber so that we can track the case where subscriber is different from patient. For example, in case of kids their parents could be subscribers.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| InsuranceId | System generated unique Id to register patient insurance | int | Yes | PK |
| PatientId | FK column value coming from PatientInfo | int | Yes | FK |
| PolicyNumber | Policy number of patient insurance | varchar(50) | Yes |  |
| InsuranceCompany | Company Name | varchar(50) | Yes |  |
| SubscriberName | Original subscriber of that insurance/policy | varchar(50) | Yes |  |
| ReltionshipWithSubscriber | Patient's relationship with policy subscriber | varchar(50) | No |  |
| ExpiryDate | Expiry date of insurance | date | Yes |  |

Example:

| **InsuranceId** | **PatientId** | **PolicyNumber** | **InsuranceCompany** | **SubscriberName** | **ReltionshipWithSubscriber** | **ExpiryDate** |
| --- | --- | --- | --- | --- | --- | --- |
| 151 | 21 | BI12345 | Brit Insurance | Vinit Sharma | Father | 2024-01-10 |
| 152 | 22 | PLI23456 | Pramerica Insurance | Sagar Tendulkar | Father | 2028-06-06 |
| 153 | 23 | ICIC34567 | ICICI Prudential | Pramod Kohli | Father | 2027-06-01 |

### 3.1.3 MasterMedicalDentalCondition:

This is a master table which will contain all different types of Medical and Dental conditions. ConditionId is a unique id of this table that will act as a foreign key in MedicalHistory and DentalHistory tables.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| ConditionId | System generated unique Id to register new Allergy | int | Yes | PK |
| ConditionType | Condition type like Medical or dental | varchar(10) | Yes |  |
| MedicalOrDentalCondition | Any previous Medical or Dental Condition of a patient which we can track for a patient care | varchar(100) | Yes |  |

Example:

| **ConditionId** | **ConditionType** | **MedicalOrDentalCondition** |
| --- | --- | --- |
| 71 | Medical | Low Blood Pressure |
| 72 | Medical | High Blood Pressure |
| 73 | Medical | Lower Back Pain |

### 3.1.4 MedicalHistory:

This table consists of the medical history of a patient. We will add only those medical conditions which the patient has. That means a patient could have multiple medical conditions eg Diabetes, BP, etc.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| PatientId | FK column value coming from PatientInfo | int | Yes | PFK |
| ConditionId | FK column value coming from MasterMedicalDentalCondition | int | Yes | PFK |
| Doctor | Doctor Name who is giving treatment for that medical condition | varchar(50) | No |  |
| EntryDate | Date of entry of this record | date | Yes |  |
| DoctorContact | Phone Number of that doctor | varchar(15) | No |  |
| DateOfDiagnosis | Diagnosis date if any | date | No |  |
| TreatmentInProgress | Current status of treatment | varchar(500) | No |  |
| Remarks | Any comments/Remarks regarding medical condition | varchar(500) | No |  |

Example:

| **PatientId** | **ConditionId** | **Doctor** | **EntryDate** | **DoctorContact** | **DateOfDignosis** | **TreatmentInProgress** | **Remarks** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 71 | Joe Root | 2021-10-10 | 2111234567 | 2000-03-03 | N | Patient does not have Low Blood Pressure |
| 22 | 72 | Dominic Sibley | 2021-09-02 | 2114567890 | 2002-02-02 | N | Patient does not have High Blood Pressure |
| 23 | 73 | NULL | NULL | NULL | NULL | NULL | NULL |

### 3.1.5 DentalHistory:

This table consists of the dental history of patients. We will add only those dental conditions which the patient has. That means a patient could have multiple dental conditions.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| PatientId | FK column value coming from PatientInfo | int | Yes | PFK |
| ConditionId | FK column value coming from MasterMedicalDentalCondition | int | Yes | PFK |
| DentistName | Dentist Name who is giving treatment for that medical condition | varchar(50) | No |  |
| EntryDate | Date of entry of this record | date | Yes |  |
| DentistContact | Phone number of previous Dentist | varchar(15) | No |  |
| Remarks | Any comments/Remarks regarding Dental condition | varchar(500) | No |  |

Example:

| **PatientId** | **ConditionId** | **DentistName** | **EntryDate** | **DentistContact** | **Remarks** |
| --- | --- | --- | --- | --- | --- |
| 21 | 71 | NULL | NULL | NULL | NULL |
| 22 | 72 | NULL | NULL | NULL | NULL |
| 23 | 73 | James Anderson | 2000-02-02 | 2115678901 | Patient Lower Back Pain is acute |

### 3.1.6 MasterAllergy:

This is a master table to track the allergy of patients. AllergyId is a unique id of this table that will act as a foreign key in the PatientAllergyMapping table.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| AllergyId | System generated unique Id to register new Allergy | int | Yes | PK |
| AllergyName | Allergy name which we will add in this master table | varchar(50) | Yes |  |

Example:

| **AllergyId** | **AllergyName** |
| --- | --- |
| 10 | Skin Allergy |
| 20 | Pollen Allergy |
| 30 | Drug Allergy |

### 3.1.7 PatientAllergyMapping:

In this table we are keeping track of patient's different allergies.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| PatientId | FK column value coming from PatientInfo | int | Yes | PFK |
| AllergyId | FK column value coming from MasterAllergy table | int | Yes | PFK |
| Medications | Record of any medications taking for that allergy | varchar(500) | No |  |
| Remarks | Comments if any | varchar(500) | No |  |

Example:

| **PatientId** | **AllergyId** | **Medications** | **Remarks** |
| --- | --- | --- | --- |
| 21 | 10 | Hydrocortisone Cream | NULL |
| 22 | 20 | Benadryl Allergy Ultratab Tablets | NULL |
| 23 | 30 | Drug desensitization | NULL |

### 3.1.8 Treatment:

This is a master table which contains data of different dental treatments. TreatmentId is a unique id of this table that will act as a foreign key in DentalRecordDetail table.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| TreatmentId | System generated unique Id to register new treatment | int | Yes | PK |
| Treatment | Dental treatment name | varchar(500) | Yes |  |

Example:

| **Treatmentid** | **Treatment** |
| --- | --- |
| 101 | Bridges |
| 102 | Crowns |
| 103 | Fillings |

### 3.1.9 MasterToothDetails:

This is a master table which consists of details of each tooth. ToothId is a unique id of this table that will act as a foreign key in DentalRecordDetail table.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| ToothId | System generated unique Id to register new treatment | nvarchar(4) | Yes | PK |
| ToothName | Dental treatment name | varchar(100) | Yes |  |
| Description | Description about tooth | varchar(100) | No |  |

Example:

| **ToothId** | **ToothName** | **Description** |
| --- | --- | --- |
| RU3 | Canine | Right Upper Tooth Number 3 |
| RU4 | 1st Bicuspid | Right Upper Tooth Number 4 |
| LU3 | Canine | Left Upper Tooth Number 3 |

### 3.1.10 DentalRecordDetail:

In this table, we are maintaining a patient's dental record.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| DentalRecordId | FK from DentalRecord | int | Yes | FK |
| ToothId | Fk column value coming from MasterToothDetails table | nvarchar(4) | Yes | FK |
| TreatmentId | FK column value coming from MasterTreatment table | int | Yes | FK |

Example:

| **DentalRecordId** | **ToothId** | **Treatmentid** |
| --- | --- | --- |
| 121 | RU3 | 101 |
| 122 | RU4 | 102 |
| 123 | LU3 | 103 |

### 3.1.11 DentalRecord:

In this table, we are tracking a patient's Dental record history. The table will be updated after every appointment.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| DentalRecordId | System generated Id for each dental record | int | Yes | PK |
| PatientId | FK column value coming from PatientInfo | int | Yes | FK |
| AppointmentId | FK column value coming from AppointmentDetails | int | Yes | FK |
| DentistComments | Any comments regarding the patient treatment after the appointment is done | varchar(500) | No |  |

Example:

| **DentalRecordId** | **PatientId** | **AppointmentId** | **DentistComments** |
| --- | --- | --- | --- |
| 121 | 21 | 141 | Patient has successfully undergone Root Canal treatment, Patient is advised to schedule a new appointment for routine checkup. |
| 122 | 22 | 142 | Patient has successfully undergone Dental Implant treatment, Patient is advised to schedule a new appointment for routine checkup. |
| 123 | 23 | 143 | No Comments |

### 3.1.12 PatientMedicalConsent:

In this table we will track consent given by the patient. Consent of the patient will be collected while filling up the form for the first time.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| ConsentId | System generated unique Id for Patient consent | int | Yes | PK |
| PatientId | FK column value coming from PatientInfo | int | Yes | FK |
| ConsentFor | Consent detail description | varchar(1000) | Yes |  |
| Consented | Consent given by patient value will be yes or no | bit | Yes |  |
| Date | Date on which consent provided by patient | date | Yes |  |

Example:

| **ConsentId** | **PatientId** | **ConsentFor** | **Consented** | **Date** |
| --- | --- | --- | --- | --- |
| 161 | 21 | Consented for Dental Treatment | 1 | 2021-12-12 |
| 162 | 22 | Consented for Dental Treatment | 1 | 2012-12-13 |
| 163 | 23 | Consented for Dental Treatment | 1 | 2021-12-14 |

## 3.2 Dentist:

The Dentist data cluster has entities related to dentist data such as personal details and type. Data for all entities under the dentist cluster will be stored when the dentist enrolls into the system. Below are the entities under Dentist.

### 3.2.1 DentistInfo:

This table will contain Dentist's personal information. There are different types of dentists, therefore DentistTypeId is used to identify the type of dentist.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| DentistId | System generated unique Id to register new Dentist | int | Yes | PK |
| DentistTypeId | FK column value coming from MasterDentistType table | int | Yes | FK |
| FirstName | First name of the dentist | varchar(50) | Yes |  |
| LastName | Last name of the dentist | varchar(50) | Yes |  |
| ContactNo | Phone No of the dentist | varchar(15) | Yes |  |
| EmailId | Email Id of the dentist | varchar(50) | Yes |  |
| Active | Boolean check if the dentist is active in the system or not | bit | Yes |  |

Example:

| **DentistId** | **DentistTypeId** | **FirstName** | **LastName** | **ContactNo** | **EmailId** | **Active** |
| --- | --- | --- | --- | --- | --- | --- |
| 51 | 61 | Aaron | Finch | 1817345678 | aaronfinch@gmail.com | 1 |
| 52 | 62 | David | Warner | 1817456789 | davidwarner@gmail.com | 1 |
| 53 | 63 | Steve | Smith | 1817567890 | stevesmith@gmail.com | 1 |

### 3.2.2 DentistType:

This is a master table for tracking the type of dentist.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| DentistTypeId | System generated unique Id to register Dentist Type | int | Yes | PK |
| DenstistTypeName | Different types of dentists | varchar(50) | Yes |  |

Example:

| **DentistTypeId** | **DenstistTypeName** |
| --- | --- |
| 61 | Periodontists |
| 62 | Prosthodontists |
| 63 | Oral and Maxillofacial Surgeons |

### 3.2.3 Prescription:

This table will contain data of a patient's prescription given by the doctor after their each appointment.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| PatientId | FK column value coming from PatientInfo | int | Yes | PFK |
| AppointmentId | FK column value coming from AppointmentDetails table | int | Yes | PFK |
| MedicationId | FK column value coming from MasterMedication table | int | Yes | PFK |
| MedicationComments | Medication related comments given by dentist | varchar(500) | No |  |

Example:

| **PatientId** | **AppointmentId** | **MedicationId** | **MedicationComments** |
| --- | --- | --- | --- |
| 21 | 141 | 11 | Morning and Night before meal 1 tablet |
| 22 | 142 | 22 | Only after dinner 1 tablet |
| 23 | 143 | 33 | In the afternoon around 2 pm 1 tablet |

### 3.2.4 Medication:

This is a master table for adding the new medications in the system.

| **ColumnName** | **Description** | **DataType** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| MedicationId | System generated unique Id to register new medication | int | Yes | PK |
| MedicationName | Name of the particular medicine or tablet | varchar(100) | Yes |  |

Example:

| **MedicationId** | **MedicationName** |
| --- | --- |
| 11 | Penicillin |
| 22 | Amoxicillin |
| 33 | Peridex |

## 3.3 Office:

The Office data cluster has entities related to office location, address, rooms, etc. Data for all entities under the office cluster will be managed and maintained by the admin. The detailed entities are explained below.

### 3.3.1 Office:

The Office table will contain details of different office locations.

| **ColumnName** | **Description** | **Datatype** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| OfficeId | System generated unique Id to register | int | Yes | PK |
| AddressId | FK column value coming from MasterAddress table | int | Yes | FK |
| ContactNo | Phone Number of the office | varchar(15) | Yes |  |
| Email | Email Id of that particular office | varchar(50) | Yes |  |

Example:

| OfficeId | AddressId | ContactNo | Email |
| --- | --- | --- | --- |
| 81 | 1 | 6171234567 | dentalcare.health@gmail.com |
| 82 | 2 | 6171234568 | dentalhealth.care@gmail.com |
| 83 | 3 | 6171234569 | carehealth.dental@gmail.com |

### 3.3.2 Room:

This contains details of the room like room code and the office id with which the particular room is associated.

| **ColumnName** | **Description** | **Datatype** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| RoomId | System generated unique Id to register new Room | int | Yes | PK |
| OfficeId | FK column value coming from Office table | int | Yes | FK |
| RoomCode | Any room code given | varchar(50) | Yes |  |

Example:

| **RoomId** | **OfficeId** | **RoomCode** |
| --- | --- | --- |
| 91 | 81 | AB01 |
| 92 | 82 | AB12 |
| 93 | 83 | AB23 |

### 3.3.3 RoomAvailability:

The RoomAvailability table keeps track of all the slots of a room according to dates. This contains RoomId and dates and the available different slots of the respective dates. We have decided that the appointment will be for half hour slots. And SlotStatus column will give the information if a slot for a particular room is available or booked.

| **ColumnName** | **Description** | **Datatype** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| RoomAvailabilityId | System generated unique Id to track Room Availability | int | Yes | PK |
| RoomId | FK column value coming from Room table | int | Yes | FK |
| Date | Date on which room is available. We can keep the record of next one month | date | Yes |  |
| StartTime | Start time of room available from the time on mentioned date | time | Yes |  |
| EndTime | End time of room available till the time on mentioned date | time | Yes |  |
| RoomStatus | FK column value coming from MasterAvailabilityStatus table | bit | Yes |  |

Example:

| **RoomAvailabilityId** | **RoomId** | **Date** | **StartTime** | **EndTime** | **RoomStatus** |
| --- | --- | --- | --- | --- | --- |
| 101 | 91 | 2021-12-17 | 9:00:00 AM | 9:30:00 AM | 1 |
| 102 | 91 | 2021-12-17 | 9:30:00 AM | 10:00:00 AM | 1 |
| 103 | 91 | 2021-12-17 | 10:00:00 AM | 10:30:00 AM | 0 |

### 3.3.4 EquipmentDetails:

This table will contain data of different dental equipment used in the room.

| **ColumnName** | **Description** | **Datatype** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| EquipmentId | System generated unique Id to register new Equipment | int | Yes | PK |
| RoomId | RoomId in which that equipment is getting used | int | Yes | FK |
| EquipmentName | Name of that equipment | varchar(50) | Yes |  |
| Comments | Any comments related to equipment | varchar(500) | No |  |
| IsAvailable | Availability Status of equipment | bit | Yes |  |

Example:

| **EquipmentId** | **RoomId** | **EquipmentName** | **Comments** | **IsAvailable** |
| --- | --- | --- | --- | --- |
| 111 | 91 | Extracting Forceps | Basic Equipment | 1 |
| 112 | 92 | Ultrasonic Scaler | Basic Equipment | 1 |
| 113 | 93 | Mouth Mirror | Basic Equipment | 1 |

### 

### 3.3.5 MasterAddress:

This is a master table for tracking the addresses. AddressId is a unique id of this table that will act as a foreign key in other tables.

| **ColumnName** | **Description** | **Datatype** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| AddressId | System generated unique id to register new address | int | Yes | PK |
| ZipCode | Zip code of the area | varchar(5) | Yes | FK |
| AddLine1 | Address line 1 for user address | varchar(100) | Yes |  |
| AddLine2 | Address line 2 for user address | varchar(100) | No |  |

Example:

| **AddressId** | **ZipCode** | **AddLine1** | **AddLine2** |
| --- | --- | --- | --- |
| 1 | 02120 | 30 South Huntington | Boston, MA, USA. |
| 2 | 02120 | 235 Park Drive | Boston, MA, USA. |
| 3 | 02101 | 150 Boylston Street | Boston, MA, USA. |

### 

### 

### 

### 

### 

### 

### 

### 

### 3.3.6 MasterZipCode:

This is a master table of zip code. As city and state are associated with zip code, this table provides better handling of data and reducing redundancy in databases.

| **ColumnName** | **Description** | **Datatype** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| ZipCode | City and State wise unique zip codes | varchar(5) | Yes | PK |
| City | City of the zip code | varchar(50) | Yes |  |
| State | State of zip code | varchar(50) | Yes |  |
| Country | Country zip code | varchar(50) | Yes |  |

Example:

| **ZipCode** | **City** | **State** | **Country** |
| --- | --- | --- | --- |
| 02120 | Boston | MA | USA |
| 02101 | Boston | MA | USA |
| 02102 | Malden | MA | USA |

## 3.4 Appointment:

In this data cluster we will consider the entities like Availability of doctor and Appointment Details. Data for all entities under the office cluster will be managed and maintained by the admin. The detailed entities are given below:

### 

3.4.1 DoctorAvailability:

This table will contain the Doctor's availability schedule. AvailabilityStatus will display if the slot is available or not.

| **ColumnName** | **Description** | **Datatype** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| AvailabilityId | System generated unique Id to register Availability of doctor | int | Yes | PK |
| DentistId | FK column value coming from Dentist table | int | Yes | FK |
| OfficeId | FK column value coming from Office table | int | Yes | FK |
| Date | Date on which doctor is available. We can keep the record of next one month | date | Yes |  |
| StartTime | Start time of each slot for that dentist on mentioned date | time | Yes |  |
| EndTime | End time of each slot for that dentist on mentioned date | time | Yes |  |
| AvailabilityStatus | FK column value coming from MasterAvailabilityStatus table | bit | Yes |  |

Example:

| **AvailabilityId** | **DentistId** | **OfficeId** | **Date** | **StartTime** | **EndTime** | **AvailabilityStatus** |
| --- | --- | --- | --- | --- | --- | --- |
| 131 | 51 | 81 | 2021-12-17 | 9:30:00 | 10:00:00 | 1 |
| 132 | 52 | 82 | 2021-12-17 | 10:00:00 | 10:30:00 | 1 |
| 133 | 53 | 83 | 2021-12-17 | 10:30:00 | 11:0:00 | 1 |

### 3.4.2 AppointmentDetails:

This table contains the details of appointments that are booked. This is mainly the mapping table because it contains FK columns coming from different tables. This table needs to be filled while booking the appointments.

| **ColumnName** | **Description** | **Datatype** | **Required** | **Key** |
| --- | --- | --- | --- | --- |
| AppointmentId | System generated unique Id to register new Appointment | int | Yes | PK |
| PatientId | FK column value coming from PatientInfo table | int | Yes | PFK |
| RoomId | FK column value coming from Room table | int | Yes | FK |
| AvailabilityId | FK column value coming from DoctorAvailability table | int | Yes | FK |

Example:

| **AppointmentId** | **PatientId** | **RoomId** | **AvailabilityId** |
| --- | --- | --- | --- |
| 141 | 21 | 91 | 131 |
| 142 | 22 | 92 | 132 |
| 143 | 23 | 93 | 133 |

# 

# 

# 

# 

# 

# 

# 

# Section 4: Entity Relationship Diagram

Based on the above entities now we can create an entity relationship diagram. ER diagram includes all above entities with detailed attributes. In the ER diagram we can see how each entity relates to other entities. It helps us understand the database in a better way. Below are the screenshots of the ER diagram created for the Dental Office data model.

Diagram

Description automatically generated

Diagram

Description automatically generated