

# Experion ILP-Batch 1

## DBMS ASSIGNMENT - 2

Varkey E Njaliath

### Case Study

#### Details

Design a comprehensive database with well-normalized tables for a real-time domain of your choice

**Selected Domain:** Appointment scheduling system for Doctors

#### ***Features:***

- Real-time availability: Doctor availability is updated based on confirmed appointments.
- Flexible scheduling: Patients can choose from various doctors and schedule slots based on their needs.
- Automatic notifications: System can send appointment reminders and confirmations via email or SMS.
- Centralized data management: All patient and doctor information is stored in one place.
- Scalability: The database can be easily expanded to accommodate more patients and doctors.

**Overall:**

- The database schema ensures a clear and efficient way to store and retrieve information about patients, doctors, appointments, and schedules.
- The relationships between tables allow for easy querying and reporting on various aspects of the appointment system.
- The design is scalable and can be easily expanded to accommodate more patients, doctors, and features in the future.

# Tables

## 1. Patients

- PatientID (PK, INT): Unique identifier for each patient.
- FirstName (VARCHAR): Patient's first name.
- LastName (VARCHAR): Patient's last name.
- DateOfBirth (DATE): Patient's date of birth.
- Phone (VARCHAR): Patient's phone number.
- Email (VARCHAR): Patient's email address.
- MedicalHistory (TEXT): (Optional) Patient's medical history.

### Relationships:

\*One patient can have many appointments (one-to-many).

## 2. Doctors

- DoctorID (PK, INT): Unique identifier for each doctor.
- FirstName (VARCHAR): Doctor's first name.
- LastName (VARCHAR): Doctor's last name.
- Specialty (VARCHAR): Doctor's specialty (e.g., Cardiology, Dermatology).
- ScheduleID (FK, INT): Foreign key referencing DoctorSchedule table.

### Relationships:

\*One doctor can have many appointments and one active schedule (one-to-many and one-to-one).

\*Each doctor's schedule defines their available appointment slots.

### 3.Doctors Schedule

- ScheduleID (PK, INT): Unique identifier for each doctor's schedule.
- DoctorID (FK, INT): Foreign key referencing Doctors table.
- Day (INT): Day of the week (1-7).
- StartTime (TIME): Start time of available slots.
- EndTime (TIME): End time of available slots.
- SlotDuration (INT): Duration of each appointment slot (minutes).

#### Relationships:

\*One schedule belongs to only one doctor (one-to-one).

\*Each schedule defines the available slots for a specific day for a specific doctor.

\*Appointments are booked within a specific schedule's time slots.

### 4.Appointments

- AppointmentID (PK, INT): Unique identifier for each appointment.
- PatientID (FK, INT): Foreign key referencing Patients table.
- DoctorID (FK, INT): Foreign key referencing Doctors table.
- ScheduleID (FK, INT): Foreign key referencing DoctorSchedule table.
- SlotTime (DATETIME): Exact date and time of the appointment.
- Status (VARCHAR): Status of the appointment (e.g., Pending, Confirmed, Canceled, No-show).
- Reason (TEXT): (Optional) Reason for appointment cancellation (if applicable).

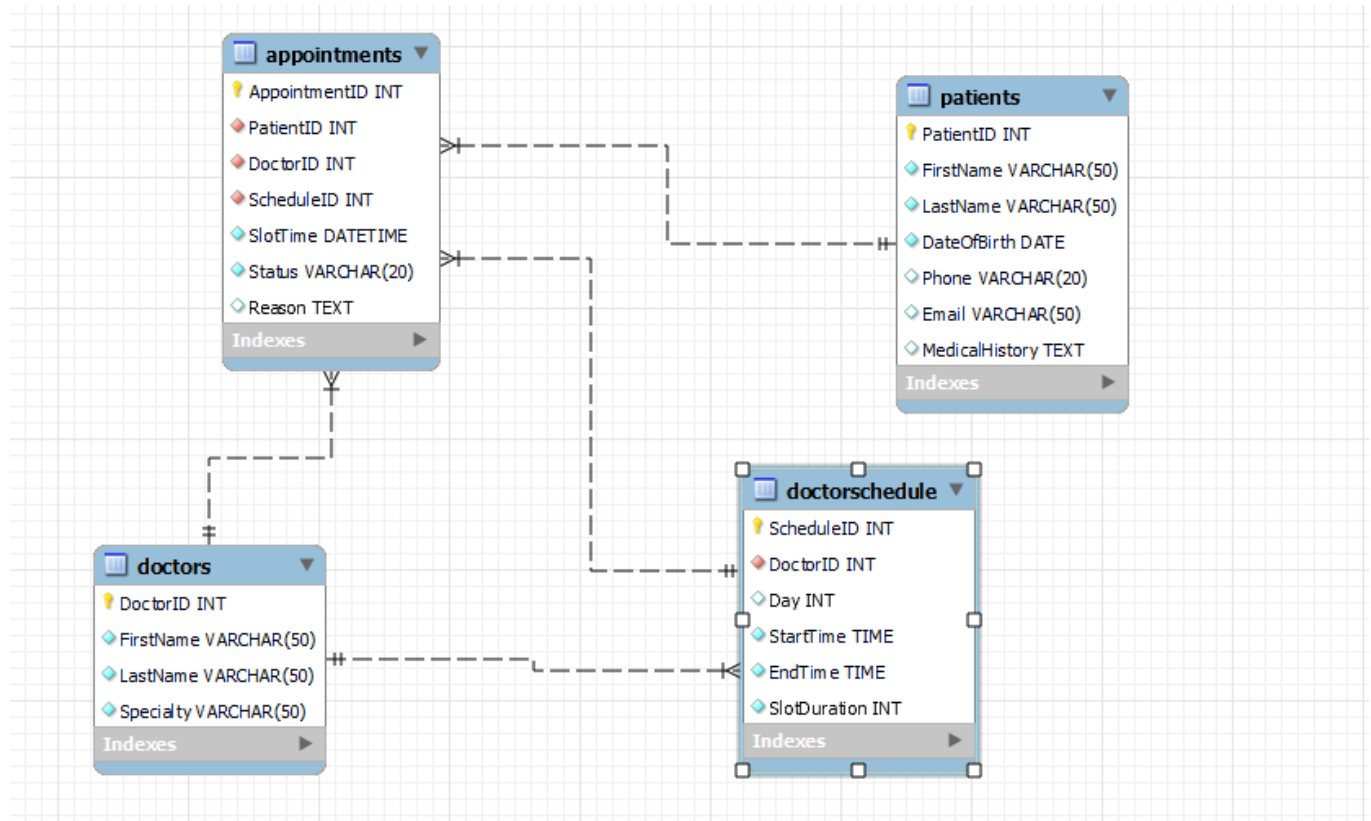
#### Relationships:

\*One appointment belongs to only one patient, doctor, and schedule (one-to-many).

\*Appointments are booked within a specific schedule's available slots.

\*The status and reason provide information about the appointment's booking/cancellation state.

# Entity Relationship Diagram



## Entities

### 1. Patients:

Primary Key: PatientID (unique identifier for each patient)

Attributes: FirstName, LastName, DateOfBirth, Phone, Email, MedicalHistory (optional)

### 2. Doctors:

Primary Key: DoctorID (unique identifier for each doctor)

Attributes: FirstName, LastName, Specialty

### **3.DoctorSchedule:**

Primary Key: ScheduleID (unique identifier for each doctor's schedule)

Foreign Key: DoctorID (references Doctors table)

Attributes: Day, StartTime, EndTime, SlotDuration

### **4.Appointments:**

Primary Key: AppointmentID (unique identifier for each appointment)

Foreign Keys: PatientID (references Patients table), DoctorID (references Doctors table), ScheduleID (references DoctorSchedule table)

Attributes: SlotTime, Status, Reason (optional)

## **Relationships:**

### **1.One-to-Many:**

Patients: One patient can have many appointments (one-to-many).

Doctors: One doctor can have many appointments (one-to-many).

DoctorSchedule: One schedule belongs to only one doctor (one-to-one).

### **2.Many-to-Many (indirect):**

Patients: Patients can have appointments with many doctors through the DoctorSchedule table (many-to-many).

Doctors: Doctors can have appointments with many patients through the DoctorSchedule table (many-to-many).

# Normalization

## 1. Patients

```
mysql> Select * from Patients;
```

PatientID	FirstName	LastName	DateOfBirth	Phone	Email	MedicalHistory
11	John	Smith	1980-01-01	555-123-4567	john.smith@example.com	Cough
12	Alice	Brown	1985-07-14	555-765-4321	alice.brown@example.com	Cancer
13	Mark	Johnson	1978-02-25	555-987-6543	mark.johnson@example.com	HIV
14	Emily	Garcia	1990-03-10	555-345-7890	emily.garcia@example.com	Bronchitis
15	David	Wilson	1982-05-18	555-098-7654	david.wilson@example.com	Malaria
16	Olivia	Miller	1995-04-22	555-123-1234	olivia.miller@example.com	Dengue
17	Matthew	Lopez	1988-06-06	555-456-2345	matthew.lopez@example.com	NULL
18	Sophia	Lee	1992-08-20	555-789-3456	sophia.lee@example.com	NULL
19	Daniel	Rodriguez	1979-09-03	555-567-8901	daniel.rodriguez@example.com	NULL
20	Jessica	Clark	1987-10-17	555-890-0123	jessica.clark@example.com	Family History of Amnesia

10 rows in set (0.00 sec)

**1NF:** The table is in 1NF. All attributes are atomic.

**2NF:** The table is in 2NF. There are no partial dependencies on the primary key.

**3NF:** The table is in 3NF. There are no transitive dependencies.

## 2.Doctors

```
mysql> select *from Doctors;
```

DoctorID	FirstName	LastName	Specialty	ScheduleID
1	John	Smith	Cardiology	101
2	Emily	Jones	Dermatology	102
3	Michael	Chen	Orthopedics	103
4	Sarah	Miller	Pediatrics	104
5	David	Wong	Ophthalmology	105

```
5 rows in set (0.00 sec)
```

**1NF:** The table is in 1NF. All attributes are atomic.

**2NF:** The table is in 2NF. There are no partial dependencies on the primary key.

**3NF:** The table is in 3NF. There are no transitive dependencies.



### 3.DoctorsSchedule

```
mysql> select *from DoctorSchedule;
```

ScheduleID	DoctorID	Day	StartTime	EndTime	SlotDuration
101	1	1	08:00:00	17:00:00	30
102	2	2	09:00:00	16:30:00	20
103	3	3	10:00:00	18:00:00	15
104	4	4	11:00:00	17:30:00	30
105	5	5	13:00:00	19:00:00	25

```
5 rows in set (0.00 sec)
```

**1NF:** The table is in 1NF. All attributes are atomic.

**2NF:** The table is in 2NF. There are no partial dependencies on the primary key.

**3NF:** The table is in 3NF. There are no transitive dependencies

## 4.Appointments

```
mysql> select * from Appointments;
```

AppointmentID	PatientID	DoctorID	ScheduleID	SlotTime	Status	Reason
1	11	1	101	2023-01-15 10:00:00	Confirmed	NULL
2	12	2	102	2023-01-16 14:30:00	Confirmed	NULL
3	13	3	103	2023-01-17 11:30:00	Confirmed	NULL
4	14	4	104	2023-01-18 15:00:00	Confirmed	NULL
5	15	5	105	2023-01-19 17:30:00	Confirmed	NULL
6	16	1	101	2023-01-20 09:45:00	Confirmed	NULL
7	17	2	102	2023-01-21 13:15:00	Confirmed	NULL
8	18	3	103	2023-01-22 16:45:00	Confirmed	NULL
9	19	4	104	2023-01-23 18:30:00	Confirmed	NULL
10	20	5	105	2023-01-24 12:00:00	Confirmed	NULL

```
10 rows in set (0.00 sec)
```

**1NF:** The table is in 1NF. All attributes are atomic.

**2NF:** The table is in 2NF. There are no partial dependencies on the primary key.

**3NF:** The table is in 3NF. There are no transitive dependencies.