

# 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:***

The appointment scheduling system for hospitals is a comprehensive solution designed to streamline the process of managing patient appointments, doctor schedules, and related information in a healthcare setting. Here's a short overview of its key features:

##### **1. User Management:**

- Supports the management of various user roles, including administrators, doctors, and staff.
- Provides secure login and authentication for system users.

##### **2. Employee and Role Management:**

- Allows the registration and tracking of hospital employees, assigning them specific roles.
- Defines and manages different roles within the hospital, such as doctors, administrators, etc.

##### **3. Doctor and Speciality Management:**

- Maintains details of doctors, including their names and qualifications.
- Associates doctors with specific medical specialities.

**4.Patient Registration and Medical History:**

- Enables patients to register in the system with their demographic details.
- Captures and maintains the medical history of patients for reference.

**5.Appointment Booking:**

- Facilitates patients in scheduling appointments with doctors.
- Provides real-time availability of doctors and time slots.

**6.Time Slot Management:**

- Manages and displays available time slots for scheduling appointments.

**7.Status Tracking:**

- Tracks the status of appointment requests, helping to monitor the progress of bookings.

**8.Feedback and Follow-Up:**

- Collects feedback from patients after appointments.
- Supports follow-up actions based on patient feedback.

**9.User Authentication and Authorization:**

- Ensures secure access to the system with role-based permissions for different users.

**10.Multi-Role Access:**

- Allows administrators, doctors, and staff to access functionalities relevant to their roles.

**11.Notification and Reminder System:**

- Sends automated notifications and reminders to patients for upcoming appointments.

**12.Integration with External Systems:**

- Integrates with electronic medical records (EMR) and billing systems for seamless operations.

**13.Telemedicine Support:**

- Integrates with telemedicine platforms to support virtual appointments.

**14.Reporting and Analytics:**

- Generates reports on appointment trends, no-shows, and resource utilization.
- Provides analytics for optimizing appointment schedules and improving patient flow.

**15.Accessibility and Language Support:**

- Supports multiple languages for a diverse patient population.
- Incorporates accessibility features for users with disabilities.

**16.Mobile Accessibility:**

Offers a mobile-friendly interface and mobile app for convenient access.

This system enhances the efficiency of hospital operations, improves patient experience, and provides a centralized platform for managing appointments and related information. It is designed to meet the diverse needs of both healthcare providers and patients within a hospital setting.

# Tables

## EmployeeDetails:

- UserID (Primary Key, VARCHAR(50))
- FirstName (VARCHAR(50))
- LastName (VARCHAR(50))
- MailID (VARCHAR(100))
- PhoneNumber (VARCHAR(20))
- Gender (VARCHAR(10))

## EmployeeRoles:

- UserID (Foreign Key referencing EmployeeDetails, VARCHAR(50))
- RoleID (Foreign Key referencing RoleInfo, VARCHAR(50))

## RoleInfo:

- RoleID (Primary Key, VARCHAR(50))
- RoleName (VARCHAR(50))

## DoctorDetails:

- DoctorID (Primary Key, VARCHAR(50))
- FirstName (VARCHAR(50))
- LastName (VARCHAR(50))

## SpecialityInfo:

- SpecialityID (Primary Key, VARCHAR(50))
- SpecialityName (VARCHAR(50))

**DoctorSpecialityDetails:**

- DoctorID (Foreign Key referencing DoctorDetails, VARCHAR(50))
- SpecialityID (Foreign Key referencing SpecialityInfo, VARCHAR(50))

**PatientDetails:**

- PatientID (Primary Key, VARCHAR(50))
- FirstName (VARCHAR(50))
- LastName (VARCHAR(50))
- PhoneNumber (VARCHAR(20))

**BookingRequest:**

- BookingRequestID (Primary Key, VARCHAR(50))
- PatientID (Foreign Key referencing PatientDetails, VARCHAR(50))
- DoctorID (Foreign Key referencing DoctorDetails, VARCHAR(50))
- Date (DATE)
- TimeID (Foreign Key referencing TimeInfo, VARCHAR(50))
- Comments (VARCHAR(255))

**TimeInfo:**

- TimeID (Primary Key, VARCHAR(50))
- TimeName (VARCHAR(50))

**BookingStatusInfo:**

- StatusID (Primary Key, VARCHAR(50))
- Status (VARCHAR(50))

**BookingStatusDetails:**

- BookingStatusDetailsID (Primary Key, VARCHAR(50))

- RequestID (Foreign Key referencing BookingRequest, VARCHAR(50))
- StatusID (Foreign Key referencing BookingStatusInfo, VARCHAR(50))

#### **Feedback:**

- FeedbackID (Primary Key, VARCHAR(50))
- BookingRequestID (Foreign Key referencing BookingRequest, VARCHAR(50))
- Feedback (VARCHAR(255))
- FollowUpDate (DATE)

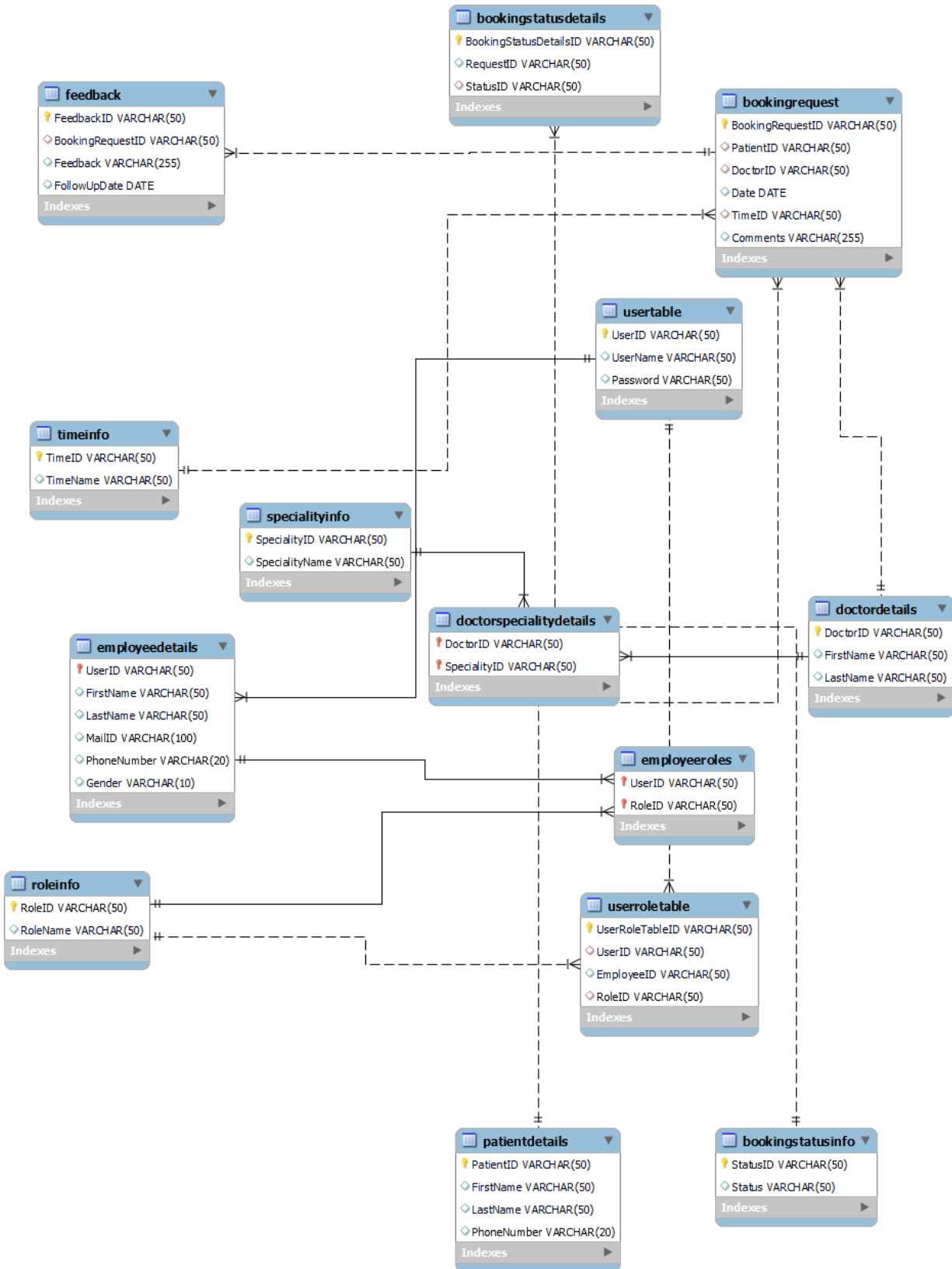
#### **UserTable:**

- UserID (Primary Key, VARCHAR(50))
- UserName (VARCHAR(50))
- Password (VARCHAR(50))

#### **UserRoleTable:**

- UserRoleTableID (Primary Key, VARCHAR(50))
- UserID (Foreign Key referencing UserTable, VARCHAR(50))
- EmployeeID (Foreign Key referencing EmployeeDetails, VARCHAR(50))
- RoleID (Foreign Key referencing RoleInfo, VARCHAR(50))

## **Entity Relationship Diagram**



## **Entities:**

- User Entity: Represents both employees and patients.
- Role Entity: Represents different roles that can be assigned to users.
- Doctor Entity: Represents doctors.
- Speciality Entity: Represents medical specialities.
- Patient Entity: Represents patients.
- Booking Request Entity: Represents requests for appointments.
- Time Slot Entity: Represents available time slots for appointments.
- Booking Status Entity: Represents the status of booking requests.
- Feedback Entity: Represents feedback provided by patients.
- UserRole Entity: Represents the roles assigned to users.

## **Relationships:**

- User-Role Relationship: One user can have multiple roles; a role can be associated with multiple users. (Many-to-Many)
- EmployeeDetails-UserRole Relationship: One employee can have multiple roles over time. (One-to-Many)
- EmployeeRoles-RoleInfo Relationship: One role can be assigned to multiple employees. (One-to-Many)
- DoctorDetails-DoctorSpecialityDetails Relationship: One doctor can have multiple specialities; a speciality can be associated with multiple doctors. (Many-to-Many)
- BookingRequest-DoctorDetails Relationship: One booking request is associated with one doctor. (Many-to-One)
- BookingRequest-TimeInfo Relationship: One booking request is associated with one time slot. (Many-to-One)
- BookingRequest-BookingStatusDetails Relationship: One booking request can have multiple status details; a status detail is associated with one booking request. (One-to-Many)
- BookingStatusDetails-BookingStatusInfo Relationship: One status detail is associated with one status. (Many-to-One)
- Feedback-BookingRequest Relationship: One feedback is associated with one booking request. (One-to-One)



**Attributes:**

- Each entity has its own set of attributes, such as UserID, FirstName, LastName, MailID, PhoneNumber, Gender, etc.

**Table overview**

```
mysql> DESC EmployeeDetails;
```

Field	Type	Null	Key	Default	Extra
UserID	varchar(50)	NO	PRI	NULL	
FirstName	varchar(50)	YES		NULL	
LastName	varchar(50)	YES		NULL	
MailID	varchar(100)	YES		NULL	
PhoneNumber	varchar(20)	YES		NULL	
Gender	varchar(10)	YES		NULL	

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

```
mysql>
```

```
mysql> -- Show the structure of EmployeeRoles table
```

```
mysql> DESC EmployeeRoles;
```

Field	Type	Null	Key	Default	Extra
UserID	varchar(50)	NO	PRI	NULL	
RoleID	varchar(50)	NO	PRI	NULL	

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

```
mysql>
```

```
mysql> -- Show the structure of RoleInfo table
```

```
mysql> DESC RoleInfo;
```

Field	Type	Null	Key	Default	Extra
RoleID	varchar(50)	NO	PRI	NULL	
RoleName	varchar(50)	YES		NULL	

```
mysql> DESC DoctorDetails;
```

Field	Type	Null	Key	Default	Extra
DoctorID	varchar(50)	NO	PRI	NULL	
FirstName	varchar(50)	YES		NULL	
LastName	varchar(50)	YES		NULL	

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

```
mysql>
```

```
mysql> -- Show the structure of SpecialityInfo table
```

```
mysql> DESC SpecialityInfo;
```

Field	Type	Null	Key	Default	Extra
SpecialityID	varchar(50)	NO	PRI	NULL	
SpecialityName	varchar(50)	YES		NULL	

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

```
mysql>
```

```
mysql> -- Show the structure of DoctorSpecialityDetails table
```

```
mysql> DESC DoctorSpecialityDetails;
```

Field	Type	Null	Key	Default	Extra
DoctorID	varchar(50)	NO	PRI	NULL	
SpecialityID	varchar(50)	NO	PRI	NULL	

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

```
mysql> DESC PatientDetails;
```

Field	Type	Null	Key	Default	Extra
PatientID	varchar(50)	NO	PRI	NULL	
FirstName	varchar(50)	YES		NULL	
LastName	varchar(50)	YES		NULL	
PhoneNumber	varchar(20)	YES		NULL	

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

```
mysql> DESC BookingRequest;
```

Field	Type	Null	Key	Default	Extra
BookingRequestID	varchar(50)	NO	PRI	NULL	
PatientID	varchar(50)	YES	MUL	NULL	
DoctorID	varchar(50)	YES	MUL	NULL	
Date	date	YES		NULL	
TimeID	varchar(50)	YES	MUL	NULL	
Comments	varchar(255)	YES		NULL	

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

```
mysql>
```

```
mysql> -- Show the structure of TimeInfo table
```

```
mysql> DESC TimeInfo;
```

Field	Type	Null	Key	Default	Extra
TimeID	varchar(50)	NO	PRI	NULL	
TimeName	varchar(50)	YES		NULL	

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

```
mysql>
```

```
mysql> -- Show the structure of BookingStatusInfo table
```

```
mysql> DESC BookingStatusInfo;
```

Field	Type	Null	Key	Default	Extra
StatusID	varchar(50)	NO	PRI	NULL	
Status	varchar(50)	YES		NULL	

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

```
mysql> DESC BookingStatusDetails;
```

Field	Type	Null	Key	Default	Extra
BookingStatusDetailsID	varchar(50)	NO	PRI	NULL	
RequestID	varchar(50)	YES		NULL	
StatusID	varchar(50)	YES	MUL	NULL	

3 rows in set (0.00 sec)

```
mysql>
```

```
mysql> -- Show the structure of Feedback table
```

```
mysql> DESC Feedback;
```

Field	Type	Null	Key	Default	Extra
FeedbackID	varchar(50)	NO	PRI	NULL	
BookingRequestID	varchar(50)	YES	MUL	NULL	
Feedback	varchar(255)	YES		NULL	
FollowUpDate	date	YES		NULL	

4 rows in set (0.00 sec)

```
mysql>
```

```
mysql> -- Show the structure of UserTable table
```

```
mysql> DESC UserTable;
```

Field	Type	Null	Key	Default	Extra
UserID	varchar(50)	NO	PRI	NULL	
UserName	varchar(50)	YES		NULL	
Password	varchar(50)	YES		NULL	

```
mysql> DESC UserRoleTable;
```

Field	Type	Null	Key	Default	Extra
UserRoleTableID	varchar(50)	NO	PRI	NULL	
UserID	varchar(50)	YES	MUL	NULL	
EmployeeID	varchar(50)	YES		NULL	
RoleID	varchar(50)	YES	MUL	NULL	

4 rows in set (0.00 sec)