Zohaib Ali

Part 1: Database Schema Design

1. Entity Identification

From the Patient Visit Manager requirements (expanded for database use):

- **Patient** → Stores patient details.
- **Doctor** → Stores doctor details.
- **Visit** → Stores visit-specific information.
- **VisitType** (Lookup) → Predefined categories of visits.
- **FeeSchedule** (Lookup) → Standard fees for each visit type.
- **UserRole** (Lookup) → Defines roles like Admin, Receptionist, Doctor.
- **ActivityLog** → Tracks actions in the system.

- ## 2. Table Schemas
- **Patients**

Column Name	Data Type	Constraints
PatientID	INT	PRIMARY KEY, IDENTITY(1,1)
PatientName	VARCHAR(100)	NOT NULL

- ---
- **Doctors**

Column Name	Data Type	Constraints
DoctorID	INT	PRIMARY KEY, IDENTITY(1,1)
DoctorName	VARCHAR(100)	NOT NULL

- ---
- **VisitTypes (Lookup)**

Column Name	Data Type	Constraints
VisitTypeID	INT	PRIMARY KEY, IDENTITY(1,1)
TypeName	VARCHAR(50)	NOT NULL, UNIQUE

- ---
- **FeeSchedule (Lookup)**

Column Name	Data Type	Constraints
FeeID	INT	PRIMARY KEY, IDENTITY(1,1)
VisitTypeID	INT	FOREIGN KEY REFERENCES VisitTypes(VisitTypeID)
FeeAmount	DECIMAL(10,2)	NOT NULL

-

- **UserRoles (Lookup)**

Column Name	Data Type	Constraints
RoleID	INT	PRIMARY KEY, IDENTITY(1,1)
RoleName	VARCHAR(50)	NOT NULL, UNIQUE

-

- **Visits**

Column Name	Data Type	Constraints
VisitID	INT	PRIMARY KEY, IDENTITY(1,1)
PatientID	INT	FOREIGN KEY REFERENCES Patients(PatientID)
DoctorID	INT	NULL, FOREIGN KEY REFERENCES Doctors(DoctorID)
VisitTypeID	INT	FOREIGN KEY REFERENCES VisitTypes(VisitTypeID)
VisitDate	DATE	NOT NULL
Description	VARCHAR(500)	NULL

-

- **ActivityLog**

Column Name	Data Type	Constraints
LogID	INT	PRIMARY KEY, IDENTITY(1,1)
RoleID	INT	FOREIGN KEY REFERENCES UserRoles(RoleID)
Action	VARCHAR(255)	NOT NULL
ActionDate	DATETIME	NOT NULL

Normalization Steps

Unnormalized Form (UNF)

The initial class held everything in a single record/object:

PatientName	VisitDate	VisitType	Description	DoctorName

Issues:

- Repeated patient names for multiple visits.

- Possible repetition of VisitType text.

Normalization Steps

First Normal Form (1NF)

Rule: Eliminate repeating groups and ensure each field contains atomic values.

Changes Made:

- Each visit becomes a separate row.
- Multi-valued or composite attributes split into separate columns.
- No arrays or comma-separated values in any column.

1NF Table:

PatientName	VisitDate	VisitType	Description	DoctorName	Fee

Problem Remaining:

- Redundancy (e.g., same VisitType string stored multiple times).
- Fee depends on VisitType, not directly on visit record (partial dependency).

Second Normal Form (2NF)

Rule: Remove partial dependencies (non-key attributes should depend on the whole primary key, not part of it).

Changes Made:

- Extract Patient table to store unique patient details.
- Extract Doctor table for unique doctor details.
- Extract VisitType table to store type names.
- Extract FeeSchedule table to store fees per visit type.
- Visit table now references Patient, Doctor, and VisitType by IDs.

Tables in 2NF:

Patient
| PatientID (PK) | PatientName |

Doctor
| DoctorID (PK) | DoctorName |

VisitType
| VisitTypeID (PK) | TypeName |

FeeSchedule
| FeeID (PK) | VisitTypeID (FK) | FeeAmount |

Visit
| VisitID (PK) | PatientID (FK) | VisitDate | VisitTypeID (FK) | Description | DoctorID (FK, nullable) |

Problem Remaining:

- Transitive dependency: FeeAmount depends on VisitType, which is stored separately (already resolved by FeeSchedule table, but also applies to other cases like UserRole).

Third Normal Form (3NF)

Rule: Remove transitive dependencies (non-key attributes should depend only on the key, not on other non-key attributes).

Final Changes Made:

- Introduced UserRole as a separate table for role-based permissions.
- Introduced ActivityLog to track actions, storing only foreign keys and timestamps.
- Removed any remaining derived or dependent fields from main entities.

Final 3NF Schema:

Patient
| PatientID (PK) | PatientName |

Doctor
| DoctorID (PK) | DoctorName |

VisitType
| VisitTypeID (PK) | TypeName |

FeeSchedule
| FeeID (PK) | VisitTypeID (FK) | FeeAmount |

UserRole
| RoleID (PK) | RoleName |

Visit
| VisitID (PK) | PatientID (FK) | VisitDate | VisitTypeID (FK) | Description | DoctorID (FK, nullable) |

ActivityLog
| ActivityID (PK) | UserID (FK) | LogMessage| LogDate|