DnA Project Phase 1

Team 46

1 Introduction

Our mini-world focuses on the internal database requirements of a hospital. It is used for maintaining and keeping track of the current employees and patients in the hospital and manage the critical data and operations related to patient care, medical staff, and administrative functions.

2 Purpose of the Database

The purpose of the database is to manage hospital's resources and ensure smooth functioning. This structured system will enable efficient tracking of patient information, medical treatments, prescriptions, and billing, while also streamlining employee roles, inventory management, and departmental coordination.

3 Users of the Database

- **Doctors and Nurses** - Access patient medical histories, prescriptions, and lab results. Update treatment notes, prescribe medication, and schedule follow-ups.
- **Administrative Staff** - Manage patient appointments, admissions, discharges, and billing. Update and organize department and facility information.
- **Pharmacy Staff** - Access prescription records to manage and track medicine inventory. Update inventory for medications dispensed, ensure stock availability.
- **Lab Technicians** - Record lab test details, input test results, and access patient test history. Manage scheduling and track inventory of lab-related materials.
- **Finance Department** - Access billing information, generate invoices, update payment statuses. Track revenue generated from treatments and patient care.
- **IT/Database Administrators** - Maintain and manage database security, permissions, and data integrity. Implement updates, monitor usage, and ensure data compliance with medical privacy laws.
- **Hospital Management** - Access summaries and reports on department efficiency, patient counts, facility utilization. Analyze data for decision-making on staffing, resource allocation, and patient care improvements

4 Application

The system supports users in performing a variety of tasks, including:

1. Access Patient Medical Records.
2. Schedule and Manage Appointments.
3. Monitor and Update Inventory.
4. Generate Patient and Financial Reports.
5. Organize Departmental Data.
6. Produce Summaries and Analytics.

5 Database Requirements

Entity Type	Attribute	Attribute Type	Data Type	Constraints
PATIENT	PatientID	Key Attribute	Integer	Not null, Primary key
	Name	Composite	Varchar	Not null
	Age	Derived	Integer	Calculated from DOB
	Date of Birth	Composite	Integer	Not null
	Gender	Simple	Enum Male, Female, Other	Not null
	Contact Number	Multi-valued	Varchar	Not null
	Aadhar Number	Simple	Varchar	Not null
	Insurance ID	Simple	Varchar	Optional
	Blood Type	Simple	Enum A+, A-, B+, B-, AB+, AB-, O+, O-	Not null
	Emergency Contact	Simple	Varchar	Not null
EMPLOYEE (superclass of Doctor, Nurse)	EmployeeID	Primary Key	Integer	Not null, Primary key
	Name	Simple	Varchar	Not null
	Role	Simple	Enum Doctor, Nurse, Administrator, Support Staff	Not null
	ContactNumber	Simple	Varchar	Not null
	Gender	Simple	Enum Male, Female, Other	Not null
	DepartmentID	Foreign Key	Integer (References DepartmentID from DEPARTMENT)	Not null
	SupervisorID	Foreign Key	Integer (References EmployeeID from Employee)	
	Date of Joining	Composite	Integer	Not null
DOCTOR	Specialization	Simple	Varchar	Not null
	LicenseNumber	Simple	Integer	Not null
	MedicalQualifications	Simple	Varchar	Not null
NURSE	Specialization	Simple	Varchar	Not null
	MedicalQualifications	Simple	Varchar	Not null
DEPARTMENT	DepartmentID	Primary Key	Integer	Not null, Primary key
	Name	Simple	Varchar	Not null
	Location	Simple	Varchar	Not null
	Head of Department (HoD)	Foreign Key	Integer (References EmployeeID from EMPLOYEE)	Not null
	Number of Employees	Derived	Integer	Calculated based on assigned employees
	Contact Number	Simple	Varchar	Not null
INVENTORY	InventoryID	Primary Key	Integer	Not null, Primary key
	DepartmentID	Foreign Key	Integer (References DepartmentID from DEPARTMENT)	Not null
	Name	Simple	Varchar	Not null
	Quantity	Simple	Integer	Not null
	ManifacutreDate	Composite	Integer	Not null
	ExpiryDate	Composite	Integer	Optional
EQUIPMENT	EquipmentID	Partial Key	Integer	Not null
	DepartmentID	Foreign Key	Integer (References DepartmentID from DEPARTMENT)	Not null, Primary Key (Composite)
	Name	Simple	Varchar	Not null
	Type	Simple	Enum Diagnostic, Surgical, Therapeutic	Not null
	Purchase Date	Composite	Integer	Not null
	Condition	Simple	Enum New, Good, Fair, Poor	Not null
FACILITY	FacilityID	Primary Key	Integer	Not null, Primary key
	Type	Simple	Enum Room, ICU, Ambulance, Operation Theater, Waiting Area	Not null
	Capacity	Simple	Integer	Not null
	Location	Simple	Varchar	Not null
	DepartmentID	Foreign Key	Integer (References DepartmentID from DEPARTMENT)	Optional
	Status	Simple	Enum Available, Under Maintenance, Out of Service	Not null

Table 1: Strong Entities

Entity Type	Attribute	Attribute Type	Data Type	Constraints
TREATMENT	TreatmentID	Partial Key	Integer	Not null
	PatientID	Foreign Key	Integer (References PatientID from PATIENT)	Not null
	DoctorID	Foreign Key	Integer (References EmployeeID from EMPLOYEE where Role = Doctor)	Not null
	Start Date	Composite	Integer	Not null
	End Date	Composite	Integer	Optional
	Treatment Notes	Simple	Text	Optional
PRESCRIPTION	PrescriptionID	Partial Key	Integer	Not null
	TreatmentID	Foreign Key	Integer (References TreatmentID from TREATMENT)	Not null
	PrescribedBy	Foreign Key	Integer (References EmployeeID from EMPLOYEE where Role = Doctor)	Not null
	InventoryID	Multi-valued Foreign Key	Integer (References InventoryID from INVENTORY)	Not null
	Quantity Prescribed	Simple	Integer	Not null
	Dosage Instructions	Simple	Varchar	Not null
	Date Prescribed	Composite	Integer	Not null
BILLING	BillID	Partial Key	Integer	Not null
	PatientID	Foreign Key	Integer (References PatientID from PATIENT)	Not null
	TreatmentID	Foreign Key	Integer (References TreatmentID from TREATMENT)	Optional
	PrescriptionID	Foreign Key	Integer (References PrescriptionID from PRESCRIPTION)	Optional
	Total Amount	Simple	Decimal	Not null
	Payment Status	Simple	Enum Pending, Paid, Overdue	Not null
	Date Issued	Composite	Integer	Not null
	Date Paid	Composite	Integer	Optional
LAB-TEST	TestID	Partial Key	Integer	Not null
	PatientID	Foreign Key	Integer (References PatientID from PATIENT)	Not null, Primary Key (Composite)
	LabName	Simple	Varchar	Not null
	DepartmentID	Foreign Key	Integer (References DepartmentID from DEPARTMENT)	Not null
	TestName	Simple	Varchar	Not null
	TestDate	Composite	Integer	Not null
	Results	Simple	Text	Optional
APPOINTMENT	AppointmentID	Partial Key	Integer	Not null
	PatientID	Foreign Key	Integer (References PatientID from PATIENT)	Not null
	DoctorID	Foreign Key	Integer (References DoctorID from EMPLOYEE)	Not null
	AppointmentDate	Composite	Integer	Not null
	AppointmentTime	Composite	Integer	Not null
	Status	Simple	Enum Scheduled, Completed, Cancelled	Not null

Table 2: Weak Entities

Relationship Type	Degree	Entities	Cardinality	Constraints
Receives	2	Patient receives Treatment	1:N	Patient(0,N), Treatment(1,1)
Administers	2	Employee with Role=Doctor administers Treatment	1:N	Employee(0,N), Treatment(1,1)
Prescribes	3	Employee with Role=Doctor prescribes Prescription for Patient	1:N:1	Employee(0,N), Prescription(1,1), Patient(0,N)
Manages	4	Departments manage Inventory, Facility and Equipment	1:N:M:L	Department(1,N), Facility(1,1) ,Inventory(1,1), Equipment(1,1)
Conducts	3	Department conducts Lab-tests for Patients	1:N:1	Department(0,N), Labtest(1,1), Patient(0,N)
Pays	2	Patient pays Billing	1:N	Patient(0,N), Billing(1,1)
Scheduled	2	Appointment is scheduled for Patient and Doctor	N:1:1	Appointment(1,1), Patient(0,N), Doctor(0,M)
Supervises	2	Employee supervises Employee	1:N	Supervisor(0,N), Supervisee(0, 1)
Works-for	2	Employee works for Department	N:1	Employee(1,1), Department(1, N)
Heads	2	Employee (Head of Department) heads Department	1:1	Employee(0,1), Department(1, 1)

Table 3: Relationship Types

6 Functional Requirements

The following are some examples of queries that may be used by end users such as receptionists, doctors and nurses, administration department, logistics department etc.

6.1 Retrieval

- **Query**
  - List EMPLOYEE from [DEPARTMENT]
    - lists complete data of all employees from a specified department
  - List PATIENT treated by [DOCTOR]
    - lists Patient ID and names of all patients who have been treated by a specified doctor
  - List APPOINTMENT from [DATE] to [DATE]
    - lists all appointments between two given dates
  - List INVENTORY with MIN(Quantity)
    - lists the inventory item(s) with lowest quantity
  - Search for DOCTOR with name matching string
    - searches for doctors whose names have a specified string as as a subpart
- **Analysis**
  - List all names from PATIENT who have TREATMENT with Overdue BILLING Payment Status
  - List all names from PATIENT who have been prescribed [MEDICINE] by [DOCTOR]
    - lists names of all patients who have been prescribed medicine with a specified name from a specified doctor

6.2 Modification

- Insert EMPLOYEE into the table EMPLOYEE
- Insert EQUIPMENT into the table EQUIPMENT
- Insert new PRESCIPTION
- Insert new BILLING
- Update Contact Number of EMPLOYEE

- Update Contact Number of PATIENT
- Update Date and Time of APPOINTMENT
- Update Status of APPOINTMENT
- Update Quantity of INVENTORY
- Delete EQUIPMENT from table EQUIPMENT
  - may be necessary for equipment that is no longer usable
- Delete entry from INVENTORY
  - needed for expired/used up inventory

7 Summary

We have built a database for the mini-world of a hospital. It is used for maintaining and keeping track of all entities - employees, departments, patients, etc. with its unique set of attributes. Along with the strong entities, it incorporates weaker entity categories like prescriptions, treatments, appointments. etc.

The users of the database comprises of Doctors, Nurses, Administrative Staff, Pharmacy Staff, Lab Technicians, Finance Department, IT/Database Administrators and Hospital Management.
The database is to maintained with clean and meaningful data, which is updated regularly to keep track of all important information of the current state of hospital.