Hospital Management Database

Project Report

DATABASE DESIGN - CMPS 364

Team Members

• Joel Acosta • Gilir Gabinete • Charlie Komar • Gabe Ridlon

Due Date: 05/02/25

1. Introduction

Purpose – The project designs a relational database that supports day-to-day clinical, billing, and pharmacy workflows for a mid-sized hospital:

- real-time staff/room allocation,
- outpatient and inpatient scheduling,
- medication order tracking, and
- integrated billing & payments.

Key Services

Area	Main Functions
Department & Staff	Maintain departments, prevent orphan deletes, count staff.
Patient & Outpatient	Register patients, schedule / cancel appointments, assign clinicians.
Inpatient & Billing	Track admissions, rooms, bills, payments; auto-discharge on billing.
Orders & Medications	Enter multidisciplinary orders; cap meds per order.

Target Users – hospital administrators, clinicians (doctors, nurses, pharmacists), and the finance team.

2. Business Rules

This section lists the business rules that have been identified based on the discussion with the end-user and the group members.

Module 1 – Department & Staff

Department

- Each department has a unique ID (department_id) (1:1)
- A department may have zero or many staff members (0:M)
- A department cannot be deleted if any active staff are assigned (1:M)

Staff

- Each staff member has a unique ID (staff_id)
- Each staff member belongs to exactly one department (department_id) (M:1)
- staff_role indicates the staff member's classification ("Doctor," "Nurse," etc.) (1:1)
- A staff member cannot be deleted if they are assigned to any ongoing appointment or admission. Relationship enforced through Appointment_Staff and Inpatient_Staff bridge tables (M:M)
- Every staff member must be assigned to a department(1:1)

Module 2 – Patient, Outpatient_Appointment & Appointment_Staff

Patient

- Each patient has a unique id (patient_id) (1:1)
- Patients must have at least a primary way to contact them (phone or email) (1:1)
- A patient **cannot be deleted** if they have active admission, future appointment, or unpaid bills (1:M)

- A patient must have one or more outpatient appointments or inpatient admissions over time (1:M)

Outpatient_Appointment

- Each appointment has a unique id (appointment_id) (1:1)
- Must reference exactly one patient (patient_id) (M:1)
- Cannot be scheduled in the past
- If canceled, set status to “canceled” and record reason

Appointment_Staff (Bridge)

- Staff member cannot be assigned if they are “inactive” (M:M)
- (appointment_id, staff_id) combination is unique (M:M)
- role_in_appointment describes staff’s role for an appointment (e.g., “Primary Doctor”, “Assistant Nurse”) (M:M)

Module 3 – Inpatient_Admission, Inpatient_Staff, Bill & Payment

Inpatient_Admission

- Each Inpatient_Admission record must be associated with exactly one Patient (M:1).
- A Patient may have multiple Inpatient_Admissions (1:M).
- An Inpatient_Admission must have an admit_date, which records the date the patient was admitted (1:1).
- A discharge_date is optional and will be populated when the patient is discharged (0:1).
- Each Inpatient_Admission record may be assigned to one room_number, though this is also optional (0:1).

Inpatient_Staff

- Inpatient_Staff records establish a many-to-many relationship between Staff and Inpatient_Admission (M:M).
- A Staff member can be associated with multiple Inpatient_Admissions (1:M).
- An Inpatient_Admission can have multiple Staff members (1:M).
- Each Inpatient_Staff record must include a role_in_admission, defining the staff's role in the admission (1:1).

Bill

- A Bill must be associated with a Patient via patient_id (M:1).

- A Patient may have multiple Bills, one for each admission or appointment (1:M).
- A Bill is required for each Inpatient_Admission, but can also be linked to an Outpatient_Appointment instead (0:1)
- A Bill can either be linked to an Inpatient_Admission or an Outpatient_Appointment, but not both at the same time.
- Bill records must include amount_due and amount_paid (1:1).
- A Bill may receive multiple partial Payments, tracked through the Payment table (1:M).

Payment

- Each Payment must be associated with exactly one Bill (M:1).
- A Payment must include a payment_date and a payment_amount (1:1).
- A Payment record should update the amount_paid on the associated Bill to reflect total payments received.

Module 4 – Orders, Medication & Order_Medication

Orders

- Every order must have a unique ID. (1:1)
- Each order must be linked to exactly one prescriber/staff and one patient. (1:1)
- Each order can be optionally linked to either:
    - one inpatient admission (0:1)
    - or one outpatient appointment (0:1)
- Every order must include a type (1:1)
- Every order must include the date it was created (1:1)
- Orders can be updated or canceled, tracked using the order_status field.

Medication

- Every medication must have a unique ID and a name. (1:1)
- Medications must include details about their form (e.g., tablet) and strength (e.g., 200mg). (1:1)
- A medication must exist in the system before it can be linked to any order. (1:M) from medication to order_medication.
- A medication cannot be deleted if it is referenced in any active order.

Order_Medication

- The order_medication table links orders to medications. It includes:
    - order_id (FK) – references the order being placed
    - medication_id (FK) – references the medication being ordered
    - dosage_amount, frequency, and duration of administration

- A single order can include one or more medications. (1:M) from orders to order_medication.
- A single medication can be part of multiple orders. (1:M) from medication to order_medication.

3. ERD

This section shows the ERD diagram for our database based on the business rules we decided to implement.

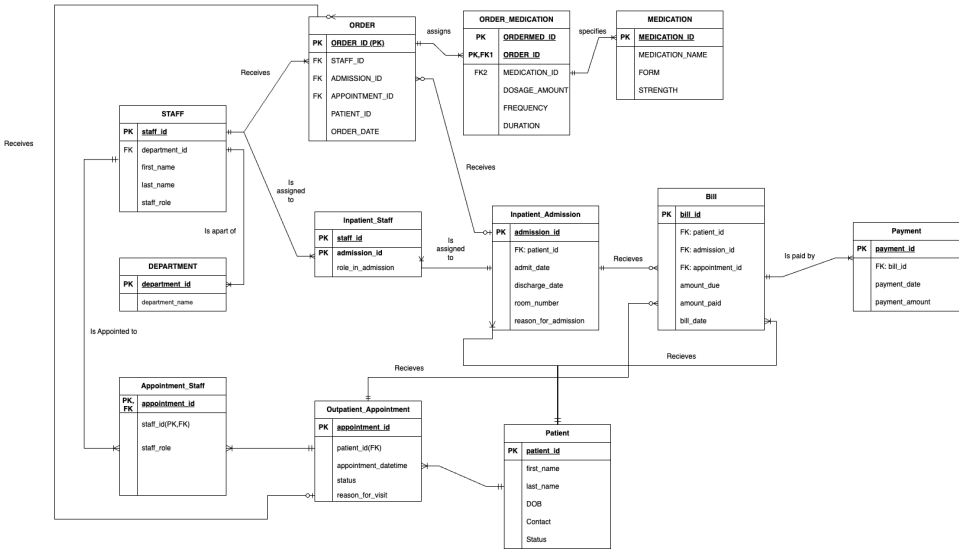

Figure 3.1: Entity Relationship Diagram of Hospital Management System

4. Normalization Tables

This section shows the normalization tables showing how our table attributes are distributed.

Module 1

UNF	1NF	2NF	3NF
staff_id(PK) dept_id first_name last_name staff_role dept_name	staff_id(PK) dept_id first_name last_name staff_role dept_name	**Staff** staff_id(PK) dept_id first_name last_name staff_role dept_id(FK) **Department** dept_id(PK) dept_name	**Staff** staff_id(PK) dept_id first_name last_name staff_role dept_id(FK) **Department** dept_id(PK) dept_name

Module 2

UNF	1NF	2NF	3NF
patient_id(PK) first_name last_name DOB contact status admissions appointments	**PATIENT** patient_id(PK) first_name last_name DOB contact status admission_id(FK) appointment_id(FK)	**PATIENT** patient_id(PK) first_name last_name DOB contact status **Outpatient_ADMISSIONS** admission_id(PK) patient_id(FK) admission_date discharge_date	**PATIENT** patient_id(PK) first_name last_name DOB contact status **Outpatient_ADMISSIONS** admission_id(PK) patient_id(FK) admission_date

		reason_for_admission **APPOINTMENTS_staff** appointment_id(PK) patient_id(FK) appointment_datetime status reason_for_visit	discharge_date reason_for_admission **APPOINTMENTS_Staff** appointment_id(PK) patient_id(FK) appointment_datetime status reason_for_visit

Module 3

UNF	1NF	2NF	3NF
InpatientAdmission admission_id (PK) patient_id (FK) admit_date discharge_date room_numbers reasons_for_admission staff_list roles bill_id patient_id appointment_id admission_id amount_due amount_paid bill_date bill_id payment_entries (e.g., dates & amounts)	**InpatientAdmission** admission_id (PK) patient_id (FK) admit_date discharge_date admission_id room_number reason_for_admission admission_id staff_id role_in_admission **Bill** bill_id (PK) patient_id (FK) appointment_id admission_id amount_due amount_paid bill_date payment_id bill_id payment_date payment_amount	**InpatientAdmission** admission_id (PK) patient_id (FK) admit_date discharge_date room_number reason_for_admission **InpatientStaff** admission_id (FK) staff_id (FK) role_in_admission **Bill** bill_id (PK) patient_id (FK) appointment_id (FK) admission_id (FK) amount_due amount_paid bill_date payment_id bill_id payment_date	**InpatientAdmission** admission_id (PK) patient_id (FK) admit_date discharge_date room_number reason_for_admission **InpatientStaff** admission_id (FK) staff_id (FK) role_in_admission **Bill** bill_id (PK) patient_id (FK) appointment_id (FK) admission_id (FK) amount_due amount_paid bill_date **Payment** payment_id (PK)

		payment_amount	bill_id (FK) payment_date payment_amount

Module 4

ORDERS Table:

Medications UNF	1NF	2NF	3NF
OrderID(PK) PatientID StaffID AdmissionID AppointmentID OrderDate MedicationID Medication_Name Form Strength OrderMedID DosageAmount Frequency Duration	**ORDER** OrderID(PK) PatientID StaffID AdmissionID AppointmentID OrderDate **MEDICATION** MedicationID(PK) Medication_Name Form Strength OrderMedID DosageAmount Frequency Duration	**ORDER** OrderID(PK) PatientID StaffID AdmissionID AppointmentID OrderDate **MEDICATION** MedicationID(PK) Medication_Name Form Strength **ORDER_MEDICATION** OrderMedID (PK) OrderID(PK FK1) MedicationID(FK2) DosageAmount Frequency Duration	**ORDER** OrderID(PK) PatientID StaffID AdmissionID AppointmentID OrderDate **MEDICATION** MedicationID(PK) Medication_Name Form Strength **ORDER_MEDICATION** OrderMedID (PK) OrderID(PK FK1) MedicationID(FK2) DosageAmount Frequency Duration

5. Data Dictionary (Meta Data)

This section shows the data dictionaries for each table showing the attribute names, data types, sizes, and if they can allow null input.

Department

NAME	DATA TYPE	SIZE	ALLOW NULL
department_id	NUMBER	3	NO (PK)
dept_name	VARCHAR2	50	NO

STAFF

NAME	DATA TYPE	SIZE	ALLOW NULL
staff_id	5	5	NO (PK)
first_name	VARCHAR2	30	NO
last_name	VARCHAR2	30	NO
staff_role	VARCHAR2	20	NO
department_id	NUMBER	3	NO (FK)

Patient

NAME	DATA TYPE	SIZE	ALLOW NULL
patient_id	INT	3	No(PK)
first_name	VARCHAR	50	No
last_name	VARCHAR	50	No
DOB	DATE	10	No
contact	VARCHAR	15	Yes

status	VARCHAR	20	No

Admissions

NAME	DATA TYPE	SIZE	ALLOW NULL
admission_id	INT	3	No(PK)
patient_id	VARCHAR	10	No(FK)
admission_date	VARCHAR	10	No
discharge_date	DATE	10	Yes
reason_for_admission	VARCHAR	200	Yes

Appointment

NAME	DATA TYPE	SIZE	ALLOW NULL
appointment_id	INT	3	No(PK)
patient_id	INT	10	No(FK)
appointment_datetime	DATE	10	No
status	VARCHAR	20	No
reason_for_visit	VARCHAR	100	Yes

INPATIENT_ADMISSION

Name	Data Type	Size	Allow Null
ADMISSION_ID	NUMBER	10	NO (PK)
PATIENT_ID	NUMBER	10	NO (FK)
ADMISSION_DATE	DATETIME	-	NO
DISCHARGE_DATE	DATETIME	-	YES

ADMISSION_ROOM

Name	Data Type	Size	Allow Null
ADMISSION_ID	NUMBER	10	NO (FK)
ROOM_NUMBER	VARCHAR2	10	NO

BILL

Name	Data Type	Size	Allow Null
BILL_ID	NUMBER	10	NO (PK)
PATIENT_ID	NUMBER	10	NO (FK)
ADMISSION_ID	NUMBER	10	YES (FK)
APPOINTMENT_ID	NUMBER	10	YES (FK)
AMOUNT_DUE	NUMBER	10,2	NO
AMOUNT_PAID	NUMBER	10,2	NO

PAYMENT

Name	Data Type	Size	Allow Null
PAYMENT_ID	NUMBER	10	NO (PK)
BILL_ID	NUMBER	10	NO (FK)
PAYMENT_DATE	DATETIME	-	NO
PAYMENT_AMOUNT	NUMBER	10,2	NO

ORDERS

Name	Data Type	Size	Allow Null
ORDER_ID	NUMBER	10	NO(PK)
ORDER_DATE	DATETIME	-	NO

STAFF_ID	NUMBER	10	NO (FK)
ADMISSION_ID	NUMBER	10	YES (FK)
APPOINTMENT_ID	NUMBER	10	YES (FK)
ORDER_TYPE	VARCHAR2	50	YES
ORDER_STATUS	VARCHAR2	20	YES

MEDICATION

Name	Data Type	Size	Allow Null
MEDICATION_ID	NUMBER	10	NO(PK)
MED_NAME	VARCHAR2	100	NO
FORM	VARCHAR2	50	YES
STRENGTH	VARCHAR2	50	YES

ORDER_MEDICATION

Name	Data Type	Size	Allow Null
ORDERMED_ID	NUMBER	10	NO(PK)
ORDER_ID	NUMBER	10	NO (FK)
MEDICATION_ID	NUMBER	10	NO (FK)
DOSAGE_AMOUNT	VARCHAR2	100	NO
FREQUENCY	VARCHAR2	100	NO
DURATION	VARCHAR2	50	YES

6. Task Distribution Table

This section shows how we distributed our work for the project.

Name	**Task**
Joel Acosta	All Module 1 sub-tasks
Gilir Gabinete	All Module 2 sub-tasks
Charles Komar	All Module 3 sub-tasks
Gabriel Ridlon	All Module 4 sub-tasks

7. Future Contributions

This section lists out the possible future contributions to improve our project.

Potential Enhancements

- XML data column / views – store lab-result documents in an XMLTYPE column; expose XQuery/XPath views for clinicians who need raw report data.

- User-defined functions (UDFs) – add PL/SQL table-returning functions (e.g., fn_get_patient_balance(p_patient_id)) to simplify reporting.

- Hybrid NoSQL extension – off-load large unstructured items (radiology images, PDF consents) to a document store such as MongoDB or Oracle JSON Collections; keep FK reference in core schema.

- Enhanced transaction management – wrap multi-step workflows (admission → order → bill) in stored procedures with explicit SAVEPOINTs to guarantee

atomicity.

- Fine-grained concurrency control – implement locking via version_no columns on high-traffic tables (orders, payment) to prevent lost updates.