Moroccan National Health Sysytem (LAB 2)

Data Management Course
UM6P College of Computing

Professor: Karima Echihabi   **Program:** Computer Engineering
Session: Fall 2025

Team Information

Team Name	The Avengers
Member 1	Taha Tahiri
Member 2	Achraf Tata
Member 3	Mouhamadou Taha Thiam
Member 4	Noura Riahi El Idrissi
Member 5	Adam Yassin
Repository Link	https://github.com/...

UM6P
University
Mohammed VI
Polytechnic

College of
Computing

1　Introduction

This problem aims to describe the relationship between different entities (e.g., patients, staff, billing) within the Moroccan National Health Services. Our solution covers how different organs of the MNHS interact with each other.

The problem we aim to solve is how to represent and conceive a database that encompasses the different components of a functioning hospital structure with its departments, staff members, and interactions with patients.

2　Requirements

Entities and Attributes:

- **Staff:**

    - Technical staff (Certification, record, modality, id)
    - Caregiving staff (ward, record grade, id)
    - Practitioners (specialty, license number, id)

- **Patient:** Full name, id, CIN, phone, date of birth, sex, blood group

- **Department:** did

- **Bills:** bill id

- **Medical Activity:**

    - Appointments (status, date, reason, time)
    - Emergency (triage level, admission timestamp, outcome, handler, record patient)
    - Prescription (date)

- **Insurance:** CNOPS, CNSS, RAMED, Private, None

- **Hospital:** Name, city, region

- **Pharmacy Inventory:** None

- **Address:** city, street, postal code, province, optional phone

- **Medication:** drug id, name, form, strength, manufacturer, therapeutic class, active ingredients

Relationships:

- Staff ↔ Patient: many-to-many (relationship: interact)

- Staff ↔ Department: many-to-many (relationship: works in)

- Staff + Patient (aggregation) ↔ Medical Activity: many-to-many (relationship: intervene)

UM6P
University
Mohammed VI
Polytechnic

College of
Computing

- Bills ↔ Medical Activity: many-to-many (relationship: produce)

- Bills → Insurance: one-to-many (relationship: covers)

- Patient ↔ Insurance: many-to-many (relationship: insured by)

- Patient ↔ Address: many-to-many (relationship: located in)

- Hospital ↔ Pharmacy Inventory: many-to-many (relationship: track)

- Hospital → Department: one-to-many (relationship: belongs)

- Medical Activity → Department: one-to-many (relationship: all in)

3 Methodology

In this part, we explain our methodology for designing the ER diagram.

First of all, **Patient** is an entity with national identity as its primary key alongside the rest of its attributes. We chose to associate it with the **Address** entity using a relationship of type zero-to-many called *"located in"*, because each patient can have several addresses, and different patients can live at the same address.

The patients interact with the **Staff** entity via a relationship called *"interact"* of type zero-to-many in both directions. Since **Staff** has three subtypes (Caregiver, Technical, Practitioner), we represented this with an ISA relationship, each subtype having its own attributes.

In our ER diagram, we aggregated the **Patient** and **Staff** entities and linked this aggregation with another entity, **Clinical Activities**, through a relationship called *"intervene"*. This is of type zero-to-many and one-to-one: a couple (patient, staff) can participate in multiple clinical activities, while a clinical activity is tied to a single (patient, staff) pair.

The **Clinical Activity** entity has an ISA component, since it can be either an **Emergency**, an **Appointment**, or a **Prescription**. Each of these carries specific attributes, such as the date of the prescription. Because a clinical activity occurs in exactly one department, we linked it to the **Department** entity through a relationship *"all in"* of type one-to-one and zero-to-many: a department can have zero or many clinical activities.

A **Prescription** is a special subtype, since it includes medications with dosage and duration. We therefore linked it to the **Medication** entity via an *"includes"* relationship of type zero-to-many in both directions: a prescription can contain many medications, and a medication can appear in many prescriptions. The dosage and duration are attributes of this relationship.

We implemented an aggregation because a clinical activity cannot be linked solely to a patient or solely to a staff member.

The requirements specify that bills are generated after a consultation or prescription. We therefore designed **Bills** as a weak entity dependent on **Clinical Activity**, linked by a *"produce"* relationship of type one-to-one and zero-to-one: a clinical activity may or may not generate a bill, but a bill is generated by exactly one clinical activity.

Since each bill is covered by a single insurance, but patients can have multiple insurances, we represented **Insurance** as an entity. Bills are linked to insurance through a *"covers"* relationship of type zero-to-many and one-to-one: one insurance can cover

multiple bills, but a bill can only be covered by one insurance. Meanwhile, **Patient** and **Insurance** are linked by a *"has"* relationship of type zero-to-many in both directions: an insurance can be assigned to multiple patients, and a patient can have multiple insurances. Types of insurance (RAMED, CNSS, etc.) are modeled with an ISA relationship.

Moving back to staff, since members can be assigned to more than one department, we linked **Staff** to **Department** with a *"work in"* relationship of type zero-to-many in both directions.

Department is represented as a weak entity dependent on **Hospital**, linked by an *"in"* relationship of type one-to-one and zero-to-many: a department must belong to a hospital, but a hospital can have zero or many departments.

Because hospitals need to track their medication supplies, we represented **Pharmacy Inventory** as a weak entity dependent on the hospital, linked by a *"tracks for"* relationship of type one-to-one and zero-to-many: a hospital keeps track of multiple inventories, but an inventory belongs to only one hospital.

Finally, **Medication** is linked to **Pharmacy Inventory** via a *"tracked by"* relationship of type zero-to-many in both directions: the same medication can appear in multiple inventories, and an inventory keeps track of multiple medications.

4   Implementation & Results
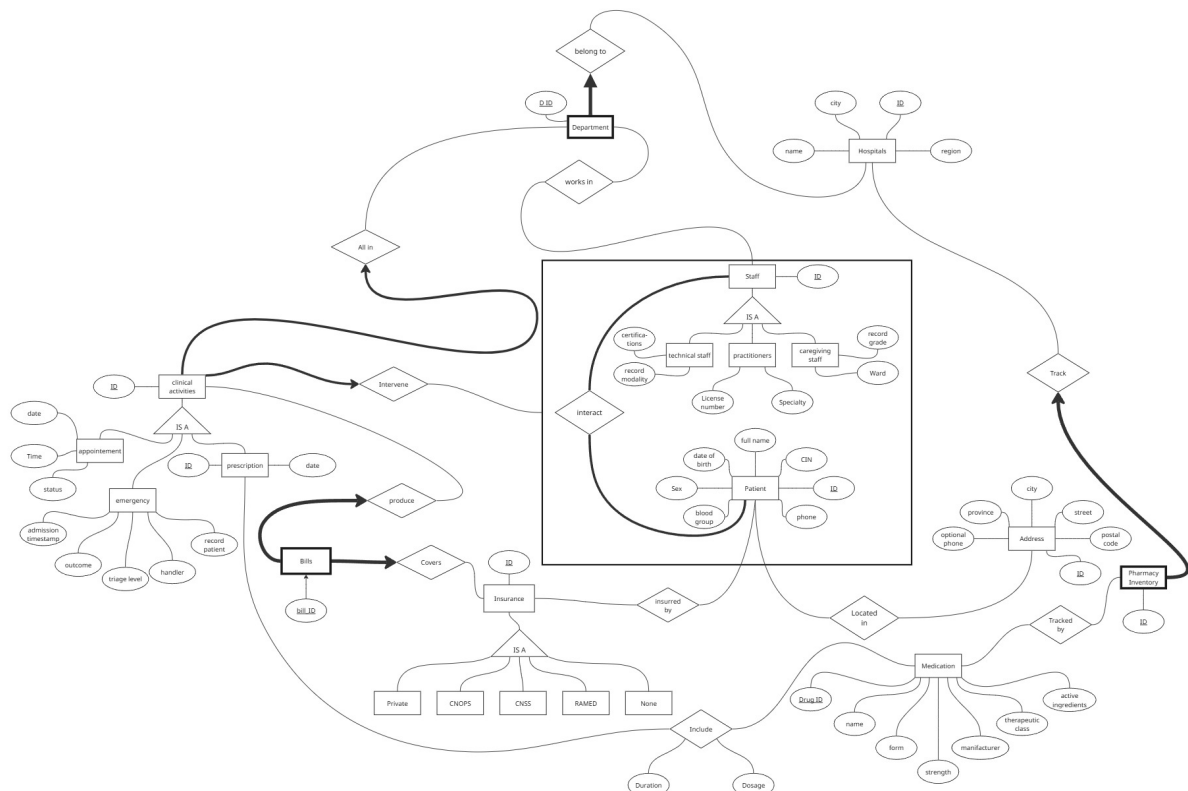
5   Discussion

The work in our group was very enriching. We interacted with each other very well, but faced challenges when convincing one another of different points of view. For example, when discussing the type of relationship between staff, patients, and departments, one

UM6P
University
Mohammed VI
Polytechnic

College of
Computing

member suggested a ternary relationship, another proposed an aggregation, and others argued for multiple binary relationships. In the end, the most logical argument prevailed.

We observed that different perspectives could arise, but uniting our ideas allows us to solve complex problems effectively. Teamwork encourages critical thinking, alternative consideration, and objective evaluation.

We also learned the importance of fully understanding requirements and paying attention to details. A single word can change the meaning of a diagram.

Clear communication prevents misunderstandings, ensures everyone is on the same page, and fosters an atmosphere of trust. Good communication strengthens cooperation.

6   Conclusion

Our group worked together several times to fully understand the requirements and discuss perspectives. We tested several ER diagrams before agreeing on the final version, which reflects the complexity of hospital databases in Morocco and shows the organization needed for them to function effectively. The lab highlighted the value of collaboration, discussion, and attention to detail in reaching a solid outcome.