

Moroccan National Health Services (MNHS)

Data Management Course

Mohammed VI Polytechnic University (UM6P)

Professor: Karima Echihabi

Teaching Assistant: Hasnae Zerouaoui

Program: Computer Engineering

Session: Fall 2025

Team Information

Team Name:	Team1 / Section 2
Member 1:	Soukaina EL KESSIRI
Member 2:	Soukaina EL JAOUHARI
Member 3:	Alaa Allah EL BAZZAZ
Member 4:	Mohamed Ziad EL ABBASSI
Member 5:	Youness EL HACHIMI
Member 6:	Oussama EL HILALI
Repository Link:	https://github.com/hash-gdel

Prepared as part of the Data Management Project.

1- Introduction

This project is a teamwork-based initiative that focuses on building the **Database Management System (DBMS)** for the **Moroccan National Health Services (MNHS)**. Through this deliverable, we design the **relational model** of the different entities and relationships involved in managing the Health Service.

Our primary goal is to guarantee **accurate data organization**, **efficient access to information**, and to ensure **reliable query handling**.

2- Requirements

1. For each entity and relationship, list attributes and primary keys. Justify any composite keys.
2. Specify foreign keys, participation, and domain checks.
3. Implement part of the schema in SQL:
 - Write `CREATE TABLE` statements for at least three core entities (e.g., `Patient`, `Hospital`, `Appointment`).
 - Insert at least two tuples per table.
 - Write one query that lists the names of patients with scheduled appointments in the city of **Benguerir**.

3- Strategy

We developed our relational model based on the given ERD, carefully considering the integrity constraints and key constraints.

4- Implementation & Results

Entity Attributes, Keys, and Relationships

- **ContactLocation:**
 - **Attributes:** CLID, City, Province, Street, contact_number, Postal_code, contact_Phone
 - **Primary Key:** CLID

- **Justification:**
 1. No two distinct tuples can have the same identifier.
 2. This is not true for any subset of the key.
 - **Composite Key:** No composite key for this entity.
 - **Foreign Key:** No foreign keys for this entity.
 - **Domain Checks:**
 - * CLID VARCHAR(50)
 - * City VARCHAR(20)
 - * Province VARCHAR(50)
 - * Street INT
 - * contact_number INT
 - * Postal_code VARCHAR(30)
 - * contact_Phone VARCHAR(20)
 - **Participation:** Partial – The relationship between Patient and ContactLocation is 0-M.
- **Have:**
 - **Attributes:** CLID, IID
 - **Primary Key:** Composed Key
 - **Justification:** No single attribute is sufficient to uniquely identify a row
 - **Composite Key:**
 1. No two distinct tuples can have the same combination of values.
 2. This is not true for any subset of the key.
 - **Foreign Key:** CLID and IID
 - **Justification:** To identify a row, we refer to the local primary keys CLID and IID respectively from Contact_Location and Patient.
 - **Domain Checks:**
 - * CLID VARCHAR(100)
 - * IID VARCHAR(100)
 - **Insurance:**
 - **Attributes:** InsID, Type_Ins

- **Primary Key:** InsID
 - **Justification:**
 1. No two distinct tuples can have the same identifier.
 2. This is not true for any subset of the key.
 - **Composite Key:** No composite key for this entity.
 - **Foreign Key:** No foreign keys for this entity.
 - **Domain Checks:**
 - * InsID VARCHAR(100)
 - * Type_Ins VARCHAR(100)
 - **Participation:** Partially participates in the relationship "covers" with the patient , and samely in the relationship "attached" with expense.
- **ClinicalActivity:**
- **Attributes:** CAID, activity_date, activity_time, contact_number, generates_expense, generate_prescription
 - **Primary Key:** CAID
 - **Justification:**
 1. No two distinct tuples can have same clinical_activity identifier.
 - **Composite Key:** No composite key for this table, as it could be uniquely identified by one.
 - **Domain Checks:**
 - * CAID VARCHAR(100)
 - * activity_time TIME
 - * activity_date DATE
 - * generates_expense VARCHAR(100)
 - * generate_prescription VARCHAR(100)
 - * **Note:** generate_prescription is an attribute that represents the relationship between ClinicalActivity Entity and Prescription. It was chosen like this to avoid the redundancy .
 - * **Note:** generates_expense is an attribute used to represent the relationship between Expense and ClinicalActivity. Since this is a 1-1 relationship , then there is no need to separate the realtionship as a table , it is convenient to store it as an attribute in the ClinicalActivity entity.

- **Foreign Keys:**
 - * DEP_ID VARCHAR(100) – In order to identify the row department, we refer to it using the local primary key (DEP_ID).
 - * Staff_ID VARCHAR(20) – The row Staff is identified using the key (Staff_ID).
 - * IID VARCHAR(20) – The Key IID is used to identify the row Patient.
 - **Participation:** Total participation in all of the relationships with Staff (via Linked), Department (via Occurs), Patient (via Has).
- **Appointment:**
 - **Attributes:** CAID , Reason , Appointment_Status.
 - **Primary Key:** CAID. This is an entity related to the entity : ClinicalActivity using the hierarchy ISA . Hence, the Primary Key is inherited from the superclass ClinicalActivity.
 - **Composite Key:** No composite key for this table , as it could be uniquely identified by one (the one inherited by the superclass).
 - **Domain Checks:**
 - * PCAID VARCHAR(100)
 - * Reason VARCHAR(100)
 - * Appointment_Status VARCHAR(100)
 - **Foreign Keys:** The attribute "CAID" is used as a foreign key to uniquely identify the ISA entity referring to the superclass . In addition to this , the constraint "On delete Cascade" is added to ensure the relative deletion between the superclass data and the data of the ISA entity .
 - **Participation:** Same participation as ClinicalActivity above
 - **Emergency:**
 - **Attributes:** CAID , Triage_Level , Outcome..
 - **Primary Key:**CAID . This is an entity related to the entity : ClinicalActivity using the hierarchy ISA . Hence, the Primary Key is inherited fromthe superclass ClinicalActivity
 - **Composite Key:** No composite key for this table , as it could be uniquely identified by one (the one inherited by the superclass).
 - **Domain Checks:**
 - * CAID VARCHAR(100)

- * Triage_Level VARCHAR(100)
- * Outcome VARCHAR(100)
- **Foreign Keys:** The attribute "CAID" is used as a foreign key to uniquely identify the ISA entity referring to the superclass. In addition to this, the constraint "On delete Cascade" is added to ensure the relative deletion between the superclass data and the data of the ISA entity.
- **Participation:** Same participation inherited by ClinicalActivity.
- **Prescription_generate:**
 - **Attributes:** PID, DateIssued
 - **Primary Key:** PID (the attribute uniquely identifies the generated prescription)
 - **Composite Key:** No composite key
 - **Domain Checks:**
 - * PID VARCHAR(100) (can be a combination of characters and ints)
 - * DateIssued DATE, (because it's a date)
 - **Participation:**
 - * a prescription is generated by one clinical activity - Total
 - * a clinical activity generates a prescription - Partial
- **Expense_Attached:**
 - **Attributes:** ExID, Total, InsID
 - **Primary Key:** ExID (uniquely identifies the expense)
 - **Foreign Key:** InsID - This is a key used to identify the Insurance entity in its relationship with the expense.
 - **Domain Checks:**
 - * Total VARCHAR(100)
 - * ExID VARCHAR(100)
 - * InsID VARCHAR(100)
 - **Participation:**
 - * Exactly one expense is generated by the clinical activity - Total participation
 - * Every insurance is attached to 0 or many expenses - Partial participation

- **Covers:**
 - **Attributes:** No attributes
 - **Domain Checks:** The domain checks are the ones inherited from the foreign keys used
 - **Foreign Keys:** IID, InsID
 - * IID : attribute needed to identify the patient in the relationship with the entity Insurance .
 - * InsID : attribute needed to identify the insurance in the relationship with the entity Patient
 - **Primary Key (composite key):** (IID, InsID)
 - **Justification:** only one of them does not uniquely identify the tuple since the relationship is M-M.
- **Department_belongs:**
 - **Attributes:** DEP_ID, Name_Dep, Specialty
 - **Primary Key:** DEP_ID
 - **Justification:**
 1. No two distinct tuples have the same DEP_ID
 - **Foreign Keys:** HID
 - **Composite Key:** No composite key
 - **Domain Checks:**
 - * DEP_ID VARCHAR(100) NOT NULL
 - * Name_Dep VARCHAR(100)
 - * Specialty VARCHAR(100)
 - * HID VARCHAR(100)
 - * **Participation:** Total participation with the entity Hospital because every department must belong to a hospital. Total with the entity named ClinicalActivity, since we always need a department where to perform the clinical activity. Total with the entity Staff because a department must have at least one member of the staff, a department cannot work on its own.
- **Work_In:**

- **Attributes:** No attributes
- **Primary Key:** Staff_ID
- **Justification:** Each tuple has a unique Staff_ID
- **Foreign Keys:**
 - * Staff_ID - Staff(Staff_ID)
 - * DEP_ID - Department_belongs(DEP_ID)

The two foreign keys are used to identify the entities involved in this relationship .

- **Composite Key:** No composite key
- **Domain Checks:**
 - * Staff_ID VARCHAR(20)
 - * DEP_ID VARCHAR(100)

- **include:**

- **Attributes:** duration, Dosage
- **Primary Key:** PID and Drug_ID
- **Composite Key:** Composed of PID and Drug_ID
- **Justification:** No two distinct tuples can have the same combination of primary keys (PID, Drug_ID), however this is not true for any subset of the key
- **Foreign Keys:**
 - * PID - Prescription(PID)
 - * Drug_ID - Medication(Drug_ID)

The two foreign keys are used to identify the entities involved in this relationship .

- **Domain Checks:**
 - * PID VARCHAR(100)
 - * Drug_ID VARCHAR(100)
 - * duration INT
 - * Dosage VARCHAR(100)

- **Medication:**

- **Attributes:** Drug_ID, Class, Med_Name, Form, Strength, Manufacturer, Active_ingredient

- **Primary Key:** Drug_ID
 - **Justification:**
 1. No two distinct tuples can have the same Drug_ID
 - **Composite Key:** No composite key for this table
 - **Domain Checks:**
 - * Drug_ID VARCHAR(100)
 - * Class VARCHAR(100)
 - * Med_Name VARCHAR(100)
 - * Form VARCHAR(100)
 - * Strength VARCHAR(100)
 - * Manufacturer CHAR(100)
 - * Active_ingredient VARCHAR(100)
 - **Participation:** Partial in relationships: Stock and include. Because a medication can exist in the catalog without necessarily being stocked in any hospital or included in any prescription. It's possible to have medication definitions that aren't currently used.
- **Stock:**
 - **Attributes:** Drug_ID, HID, Unit_Price, Stock_Timestamp, Qty, Reorder_level
 - **Primary Key:** (Drug_ID, HID)
 - **Justification:**
 1. No two distinct tuples can have the same combination of Drug_ID and HID
 2. This is not true for any individual attribute (neither Drug_ID nor HID alone can uniquely identify tuples)
 - **Foreign Keys:**
 - * Drug_ID : Medication(Drug_ID)
 - * HID : Hospital(HID)
 - **Composite Key:** Composed of Drug_ID and HID
 - **Domain Checks:**
 - * Drug_ID VARCHAR(100)
 - * HID VARCHAR(100)

- * Unit_Price INT
- * Stock_Timestamp INT
- * Qty INT
- * Reorder_level VARCHAR(100)

- **Hospital:**

- **Attributes:** HID, Name_H, City, Region
- **Primary Key:** HID
- **Justification:**
 1. No two distinct tuples can have the same HID
- **Composite Key:** No composite key for this table
- **Domain Checks:**
 - * HID VARCHAR(100)
 - * Name_H VARCHAR(100)
 - * City VARCHAR(100)
 - * Region VARCHAR(100)
- **Participation:** Total in Department_belongs and Stock. Because every hospital must have at least one department (cannot exist as a hospital without departments) and must have some stock of medications. A hospital without departments or inventory wouldn't be functional.

- **Patient:**

- **Attributes:** IID, CIN, pa_name, Birth, Sex, BloodGroup, Phone
- **Primary Key:** IID
- **Justification:** No composite key: IID uniquely identifies each patient.
- **Domain Checks:**
 - * IID (VARCHAR(20), PRIMARY KEY, NOT NULL)
 - * CIN (VARCHAR(100), NOT NULL)
 - * pa_name (VARCHAR(100), NOT NULL)
 - * Birth (DATE)
 - * Sex (VARCHAR(10))
 - * BloodGroup (VARCHAR(5))
 - * Phone (VARCHAR(20))

- **Participation:**
 - * Total in ClinicalActivity (every patient must have at least one clinical activity)
 - * Partial in Have and Covers (not every patient must have contact or insurance)
- **Staff:**
 - **Attributes:** Staff_ID, Staff_Name, Staff_Status
 - **Primary Key:** Staff_ID
 - **Justification:** No composite key — Staff_ID uniquely identifies each staff.
 - **Domain Checks:**
 - * Staff_ID (VARCHAR(20), PRIMARY KEY, NOT NULL)
 - * Staff_Name(VARCHAR(100), NOT NULL)
 - * Staff_Status (VARCHAR(100))
 - **Participation:**
 - * Total in Caregiving, Technical, Practitioner (since they are related by ISA relationship)
 - * Partial in the relationship with department (Work_In) and the relationship with ClinicalActivity
- **Caregiving (is-a Staff):**
 - **Attributes:** Staff_ID, Grade, Ward
 - **Primary Key:** Staff_ID (inherited from Staff)
 - **Justification:** Staff_ID uniquely identifies each caregiving because each caregiving corresponds to exactly one Staff member.
 - **Domain Checks:**
 - * Staff_ID: VARCHAR(20)
 - * Grade: INT
 - * Ward: VARCHAR(100)
 - **Foreign Key:** Staff_ID REFERENCES Staff(Staff_ID) → ensures each caregiving is also a Staff member.
 - **Participation:** Total participation from caregiving in Staff (each caregiving must belong to a Staff, not all staff are caregiving.)

- **Technical (is-a Staff):**

- **Attributes:** Staff_ID, Modality, Certifications
- **Primary Key:** Staff_ID (same as FK, inherited from Staff)
- **Domain Checks:**
 - * Staff_ID (VARCHAR(20))
 - * Modality VARCHAR(100)
 - * Certifications VARCHAR(100)
- **Foreign Key:** Staff_ID REFERENCES Staff(Staff_ID) → ensures each Technical is also a Staff member.
- **Participation:** Total participation from Technical in Staff (each Technical must belong to a Staff, not all staff are technical.)
- **Justification of keys:** Staff_ID uniquely identifies each Technical because each Technical corresponds to exactly one Staff member.

- **Practitioner (is-a Staff):**

- **Attributes:** Staff_ID, licenseNumber, Specialty
- **Primary Key:** Staff_ID (inherited from Staff)
- **Justification:** Staff_ID uniquely identifies each Practitioner because each Practitioner corresponds to exactly one Staff member.
- **Domain Checks:**
 - * Staff_ID (VARCHAR(20), PRIMARY KEY)
 - * Specialty (VARCHAR(100), NOT NULL)
 - * licenseNumber INT
- **Foreign Key:** Staff_ID REFERENCES Staff(Staff_ID) → ensures each Practitioner is also a Staff member.
- **Participation:** Total participation from Practitioner in Staff (each Practitioner must belong to a Staff, not all staff are Practitioner.)

Core Database Entities

```
CREATE TABLE ClinicalActivity(  
  CAID VARCHAR(100),  
  activity_time TIME,
```

```

activity_date DATE,
DEP_ID VARCHAR(100) NOT NULL,
Staff_ID VARCHAR(20) NOT NULL,
IID VARCHAR(20) NOT NULL,
generates_expense VARCHAR(100),
generate_prescription VARCHAR(100),
PRIMARY KEY (CAID),
FOREIGN KEY (IID) REFERENCES Patient (IID),
FOREIGN KEY (DEP_ID) REFERENCES Department_belongs (DEP_ID),
FOREIGN KEY (Staff_ID) REFERENCES Staff (Staff_ID)
);

CREATE TABLE Appointment (
    CAID VARCHAR(100),
    Reason VARCHAR(100),
    Appointment_Status VARCHAR(100),
    PRIMARY KEY (CAID),
    FOREIGN KEY (CAID) REFERENCES ClinicalActivity (CAID) ON DELETE
        CASCADE
);

CREATE TABLE Patient (
    IID VARCHAR(20) NOT NULL,
    CIN VARCHAR(100),
    pa_name VARCHAR(100),
    Birth DATE,
    Sex VARCHAR(100),
    BloodGroup VARCHAR(5),
    Phone VARCHAR(20),
    PRIMARY KEY (IID)
);

CREATE TABLE Emergency (
    CAID VARCHAR(100),
    Triage_Level VARCHAR(100),
    Outcome VARCHAR(100),
    PRIMARY KEY (CAID),
    FOREIGN KEY (CAID) REFERENCES ClinicalActivity (CAID) ON DELETE
        CASCADE

```

```
);

CREATE TABLE Hospital (
  HID VARCHAR(100) NOT NULL,
  Name_H VARCHAR(100),
  City VARCHAR(100),
  Region VARCHAR(100),
  PRIMARY KEY (HID)
);

CREATE TABLE Department_belongs (
  Name_Dep VARCHAR(100),
  DEP_ID VARCHAR(100) NOT NULL,
  Specialty VARCHAR(100),
  HID VARCHAR(100),
  FOREIGN KEY (HID) REFERENCES Hospital (HID),
  PRIMARY KEY (DEP_ID)
);
```

Listing 1: CREATE TABLE statements for core entities

Insert Sample Data

```
-- Patient
INSERT INTO Patient (IID, pa_name, Birth, Sex) VALUES
('P001', 'Karim El Amrani', '1985-01-12', 'M'),
('P002', 'Yasmine Bouziane', '1992-06-23', 'F');

-- ClinicalActivity
INSERT INTO ClinicalActivity (CAID, activity_date, DEP_ID, Staff_ID,
  IID) VALUES
('CA001', '2025-03-10', 'D001', 'S001', 'P001'),
('CA002', '2025-04-15', 'D002', 'S002', 'P002');

-- Department_belongs
INSERT INTO Department_belongs (DEP_ID, Name_Dep, HID) VALUES
('D001', 'Cardiology', 'H001'),
('D002', 'Neurology', 'H002');
```

```

-- Hospital
INSERT INTO Hospital (HID, Name_H, City) VALUES
('H001', 'Marrakech General Hospital', 'Marrakech'),
('H002', 'Benguerir Clinic', 'Benguerir');

-- Appointment
INSERT INTO Appointment (CAID, Reason, Appointment_Status) VALUES
('CA001', 'Checkup', 'Done'),
('CA002', 'Consultation', 'Pending');

-- Emergency
INSERT INTO Emergency (CAID, Triage_Level, Outcome) VALUES
('CA001', 'High', 'Stable'),
('CA002', 'Medium', 'Recovered');

```

Listing 2: Sample data insertion for tables

Query: Patients in Benguerir

```

SELECT DISTINCT p.pa_name
FROM Patient p
JOIN ClinicalActivity ca ON p.IID = ca.IID
JOIN Department_belongs d ON ca.DEP_ID = d.DEP_ID
JOIN Hospital h ON d.HID = h.HID
JOIN Appointment a ON a.CAID = ca.CAID
WHERE h.City = 'Benguerir';

```

Listing 3: Select patients with clinical activities in Benguerir

5- Discussion

5.0.1 Challenges Faced

- Finding an efficient methodology to divide the work among team members.
- Clearly expressing some of the constraints proved challenging.
- Initial misunderstandings of the requirements caused some difficulties.

5.0.2 Lessons learned

- Effective communication are crucial in a team project
- Carefully reading and clarifying requirements
- Asking questions early saves time during implementation.

6- Conclusion

In conclusion, our team successfully developed the relational model for the Moroccan National Health Services (MNHS) database. By carefully analyzing the requirements and thoroughly understanding the ER diagram, we were able to define each entity and its attributes clearly. This project improved our practical skills in database design, reinforced our understanding of integrity and key constraints.