



# Deliverable #: Lab-3/ Relational Schema

## Data Management Course

UM6P College of Computing

Professor: Karima Echihabi Program: Computer Engineering

Session: Fall 2025

## **Team Information**

Team Name	StarsDB
Member 1	Nouha Talssi
Member 2	Mariam Said
Member 3	Tilamsane Wissal
Member 4	Alae Sabir
Member 5	Amina Sidki
Repository Link	https://github.com/StarsDB





#### 1 -Introduction

This deliverable aims to transform the conceptual schema of the MNHS (Moroccan National Health Services) database into a logical design model by creating tables, attributes, primary and foreign keys, and specifying integrity constraints to efficiently handle and organize critical healthcare information.

## 2 -Requirements

In this deliverable, we base our work on the ER diagram developed in the previous lab. Each entity and its relationships are translated into tables ,columns and foreign keys depending on the type of relationship (one-to-one, one-to-many, or many-to-many). Below is the summary of the main entities and their relationships.

- Patient: This entity Patient with the Attributes:cin,name,sex,birth,phone,blood group, phone,id will be implemented as a table where each column of this table corresponds to one attribute .
  - Relationships:
    - Relation many-to-many with the Entity Insurance.
    - Relation many-to-many with the Entity Contact Location.
    - Relation one-to-many with the Clinical Activity.
- Staff: The entity Staff with the Attributes: Staff-id, name, status. Sub-entities(ISA relationship): Practitioner, Caregiving, Technical each with their specific attributes. Relationships:
  - Total participation with the Entity department.
  - Relation one-to-many with the Entity department.
- Prescription: Attributes : PID , DateIssued.
  - Relationships:
    - Relation many-to-many with the Entity Medication.
    - Relation one-to-many with the Entity Clinical Activity.
- Expenses: Attributes : InsID , type.
  - Relationships:
    - Relation many-to-many with the Entity Insurance.
    - Relation one-to-many with the Entity Clinical Activity.
- **Hospital**: Attributes : hid,name , city ,region . Relationships :
  - Relation one-to-many with the Entity department.
  - Relation many-to-many with the Entity Medication.

The logical model is then tested in SQL and the required query is implemented.





# 3 -Methodology

We broke down the process of converting the ERD into a logical design into smaller steps, which are the following:

- We converted the entities into tables. Each table has its own primary key and the appropriate schema.
- For the Staff hierarchy, we represented it by putting a foreign key that points back to the superclass table in each subclass, and we did the same for the Clinical Activity hierarchy: we put the Clinical Activity's primary key as a foreign key in each subclass.
- For the many-to-many relationships, we added a third relation to represent it, like: Stock, Cover, Include. For each new relation, the primary key was composed of the primary keys of the two entities involved in the relationship.
- For the one-to-many relationships, we put the primary key of the "one" side in the table of the "many" side, so that we avoid the redundancy of the data. For example, we put the primary key of the Hospital in the Department table as a foreign key.
- And finally, for the one-to-one relationship, we chose one of the two entities participating in the relationship to have the primary key of the other as a foreign key. This is exactly the case in the relationship between the Clinical Activity and Expenses. (Generates).

## 4 -Implementation & Results

We first started by writing the logical design which is as follows:

- Patient(<u>IID</u> CHAR(10), CIN CHAR(8), Name VARCHAR(20), Sex VARCHAR(10), Birth DATE ,Blood\_Group CHAR(3), Phone CHAR(11))
- Contact\_Location(<u>CLID</u> CHAR(10), City VARCHAR(50), Province VARCHAR(50), Street VARCHAR(50), Number INTEGER, Postal\_Code INTEGER, Phone CHAR(11)
- Have\_location(<u>IID</u> CHAR(10), <u>CLID</u> CHAR(10)) FOREIGN KEY (IID) referencing Patient(IID) and FOREIGN KEY (CLID) referencing Contact\_location(CLID)
- Insurance(<u>InsID</u> CHAR(20), Type VARCHAR(50))
- Covers(<u>IID</u> CHAR(10), <u>InsID</u> CHAR(20)) FOREIGN KEY (IID) referencing Patient(IID) and FOREIGN KEY (InsID) referencing Insurance(InsID)





- Staff(Staff\_Id CHAR(12), Name VARCHAR(50), Status VARCHAR(50))
- Practioner (Staff\_Id CHAR(12), Specialty VARCHAR(50), License  $_Number CHAR(12)$ ) FOREIGN
- Caregiving(Staff\_id CHAR(12),Grade VARCHAR(20), Ward VARCHAR(20)) FOR-EIGN KEY (Staff\_id) referencing Staff(Staff\_id)
- Technical(Staff\_id CHAR(12), Modality VARCHAR(50), Certifications VARCHAR(50)) FOREIGN KEY (Staff\_id) referencing Staff(Staff\_id)
- Hospital(<u>HID</u> CHAR(12), Name VARCHAR(50), city VARCHAR(50), region VARCHAR(50))
- Department(<u>DepID</u> CHAR(12), name VARCHAR(50), Specialty VARCHAR(50), HID CHAR(12)) FOREIGN KEY (HID) referencing Hospital(HID)
- WORK\_IN(<u>Staff\_id</u> CHAR(12), <u>DepID</u> CHAR(12)) FOREIGN KEY (Staff\_id) referencing Staff(Staff\_id) and FOREIGN KEY (DepID) referencing Department(DepID)
- ClinicalActivity(underlineCAID CHAR(12),Time TIME,Date DATE,Staff\_ID CHAR(12),DepID CHAR(12),IID CHAR (10)) FOREIGN KEY (Staff\_id) referencing Staff(Staff\_id),
   FOREIGN KEY (DepID) referencing Department(DepID) and FOREIGN KEY (IID) referencing Patient(IID)
- Expenses(<u>ExID</u> CHAR(12), Total FLOAT, InsID CHAR(20), CAID CHAR(12)) FOREIGN KEY (InsID) referencing Insurance(InsID) and FOREIGN KEY (CAID) referencing ClinicalActivity(CAID)
- Appointment(<u>CAID</u> CHAR(12), Reason VARCHAR(100), Status VARCHAR(20)) FOREIGN KEY (CAID) referencing Clinical Activity(CAID)
- Emergemcy(<u>CAID</u> CHAR(12), Triage\_Level INTEGER, Status VARCHAR(20)) FOR-EIGN KEY (CAID) referencing Clinical Activity(CAID)





- Prescription(<u>PID</u> CHAR(12), CAID CHAR(12), Date\_Issued DATE)) FOREIGN KEY (CAID) referencing ClinicalActivity(CAID)
- Medication(<u>DrugID</u> CHAR(12), Class VARCHAR(50), Name VARCHAR(50), Form VARCHAR(50), Strength VARCHAR(50), Manufacturer VARCHAR(50), Active\_Ingredient VARCHAR(50))
- Include(<u>DrugID</u> CHAR(12), <u>PID</u> CHAR(12), Dosage FLOAT, Duration FLOAT) FOREIGN KEY (DrugID) referencing Medication(DrugID) and FOREIGN KEY (PID) referencing Prescription(PID)
- Stock(<u>DepID</u> CHAR(12), <u>DrugID</u>, UnitPrice FLOAT, Restock\_TimeStamp TIME, Quantity FLOAT, Reorder\_Level VARCHAR(50)) FOREIGN KEY (DepID) referencing Department(DepID) and FOREIGN KEY (DrugID) referencing Medication(DrugID)

After that, we implemented the logical design in MySQL, and here is the full code:

```
CREATE TABLE Patient (
     IID
            CHAR (10)
                       PRIMARY
                                 KEY,
2
     CIN
            CHAR (8),
3
            VARCHAR (20),
     Name
            VARCHAR(10) CHECK (Gender IN ('Female', 'Male')),
     Sex
     Birth DATE,
6
     Blood_Group CHAR(3) CHECK (BloodGroup IN ('A+', 'A-', 'B+', 'B
         -', 'AB+', 'AB-', 'O+', 'O-')),
     Phone CHAR (11)
  );
  CREATE TABLE Contact_location(
11
      CLID CHAR (10) PRIMARY KEY,
12
      City VARCHAR (50),
      Province VARCHAR (50),
14
      Street VARCHAR (50),
      Number INT,
16
      Postal_Code INT CHECK (Postal_Code BETWEEN 10000 AND 99999),
      Phone CHAR (11)
18
  );
19
  CREATE TABLE Have_location(
21
      IID CHAR (10),
22
      CLID CHAR (10),
23
      PRIMARY KEY(IID, CLID),
24
      FOREIGN KEY (IID) REFERENCES Patient(IID) ON DELETE CASCADE,
      FOREIGN KEY (CLID) REFERENCES Contact_location(CLID) ON
          DELETE NO ACTION
27 );
```





```
28
  CREATE TABLE Insurance (
       InsID CHAR (50) PRIMARY KEY,
30
       Type VARCHAR (50)
  );
32
33
  CREATE TABLE Covers (
34
       IID CHAR (10),
       InsID CHAR (12),
       PRIMARY KEY (IID, InsID),
37
       FOREIGN KEY (IID) REFERENCES Patient(IID) ON DELETE CASCADE,
38
       FOREIGN KEY (InsID) REFERENCES Insurance(InsID) ON DELETE SET
39
           NULL
  );
40
41
  CREATE TABLE Staff(
42
       Staff_ID CHAR(12) PRIMARY KEY,
43
       Name VARCHAR (50),
44
       Status VARCHAR (50)
45
  );
47
  CREATE TABLE Practionner(
48
       Staff_ID CHAR(12) PRIMARY KEY,
49
       Specialty VARCHAR (20),
       License_Number CHAR(12),
51
       FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID) ON DELETE
          CASCADE
  );
53
54
  CREATE TABLE Caregiving(
55
       Staff_ID CHAR(12) PRIMARY KEY,
56
       Grade VARCHAR (20),
57
       Ward VARCHAR (20),
       FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID) ON DELETE
59
          CASCADE
60
  );
  CREATE TABLE Technical(
       Staff_ID CHAR(12) PRIMARY KEY,
63
       Modality VARCHAR (50),
64
       Certifications VARCHAR (50),
65
       FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID) ON DELETE
66
          CASCADE
67
  );
68
  CREATE TABLE Hospital(
69
       HID CHAR (12) PRIMARY KEY,
70
       Name VARCHAR (50),
71
       City VARCHAR (50),
72
       Region VARCHAR (50)
73
74 );
```





```
75
   CREATE TABLE Departement (
76
       DepID CHAR (12) PRIMARY KEY,
77
       Name VARCHAR (50),
       Specialty VARCHAR (50),
79
       HID CHAR (12) NOT NULL,
80
       FOREIGN KEY (HID) REFERENCES Hospital(HID) ON DELETE NO
81
          ACTION
  );
82
83
   CREATE TABLE WORK_IN(
84
       DepID CHAR (12) NOT NULL,
85
       Staff_ID CHAR(12),
86
       PRIMARY KEY (DepID, Staff_ID),
       FOREIGN KEY (DepID) REFERENCES Departement(DepID) ON DELETE
88
          CASCADE,
       FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID) ON DELETE
89
          CASCADE
  );
90
91
   CREATE TABLE Clinical Activity (
92
       CAID CHAR (12) PRIMARY KEY,
93
       Time TIME,
94
       Date DATE,
95
       Staff_ID CHAR (12) NOT NULL,
96
       DepID CHAR (12) NOT NULL,
       IID CHAR (10) NOT NULL,
98
       FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID) ON DELETE
99
          RESTRICT,
       FOREIGN KEY (DepID) REFERENCES Departement(DepID)ON DELETE
100
          RESTRICT,
       FOREIGN KEY (IID) REFERENCES Patient(IID)ON DELETE RESTRICT
  );
   CREATE TABLE Expenses (
104
       EXID CHAR (12) PRIMARY KEY,
       Total FLOAT,
106
       InsID CHAR (50),
       CAID CHAR (12) NOT NULL,
108
       FOREIGN KEY (InsID) REFERENCES Insurance(InsID) ON DELETE SET
109
           NULL,
       FOREIGN KEY (CAID) REFERENCES Clinical Activity (CAID) ON
          DELETE RESTRICT
  );
   CREATE TABLE Appointment (
       CAID CHAR (12) PRIMARY KEY,
114
       Reason VARCHAR (100),
       Status VARCHAR (20),
       FOREIGN KEY (CAID) REFERENCES Clinical Activity (CAID) ON
117
          DELETE CASCADE
```





```
);
118
   CREATE TABLE Emergency (
120
       CAID CHAR (12) PRIMARY KEY,
       Triage_Level INT,
       Outcome VARCHAR (100),
123
       FOREIGN KEY (CAID) REFERENCES Clinical Activity (CAID) ON
124
          DELETE CASCADE
  );
125
   CREATE TABLE Prescription (
       PID CHAR (12) PRIMARY KEY,
128
       CAID CHAR (12) NOT NULL,
       Date_Issued VARCHAR(50),
130
       FOREIGN KEY (CAID) REFERENCES Clinical Activity (CAID) ON
131
          DELETE CASCADE
  );
   CREATE TABLE Medication(
134
       DrugID CHAR (12) PRIMARY KEY,
135
       Class VARCHAR (50),
136
       Name VARCHAR (50),
       Form VARCHAR (50),
138
       Strength VARCHAR (50),
       Manufacturer VARCHAR (50),
140
       Active_Ingredient VARCHAR(50)
141
  );
142
143
   CREATE TABLE Include (
144
       DrugID CHAR (12),
145
       PID CHAR (12),
146
       Dosage FLOAT,
147
       Duration FLOAT,
148
       PRIMARY KEY (DrugID, PID),
149
       FOREIGN KEY
                     (DrugID) REFERENCES Medication(DrugID), ON
            CASCADE
       FOREIGN KEY
                     (PID) REFERENCES Prescription(PID) ON DELETE
          CASCADE
  );
153
   CREATE TABLE Stock (
154
       UnitPrice FLOAT,
       Restock_TimeStamp TIME,
156
157
       Quantity FLOAT,
       Reorder_Level VARCHAR (50),
158
       DrugID CHAR (12),
159
       DepID CHAR (12),
       PRIMARY KEY (DepID, DrugID),
161
       FOREIGN KEY (DepID) REFERENCES Departement(DepID) ON DELETE
          RESTRICT,
       FOREIGN KEY (DrugID) REFERENCES Medication(DrugID) ON DELETE
163
```





CASCADE

64 );

After implementing the logical design in MySQL, we created the tables and executed the required query, which gave us the following output :(look at page 10)

#### 5 Discussion

while working on this deliverable, we faced some challenges especially on the physical design, such as:

- The first thing that we struggled with was choosing the right way to represent the ISA relationship. It was a bit tricky to decide where to place the foreign key, but after discussing it as a team, we agreed that putting the foreign key in the ISA child tables was the best approach.
- After that we found it a bit hard to choose the right way to represent the relationships, especially the many-to-many ones(the placement of the foreign keys). For the one-to-one relationship, it seemed logical to put a foreign key in both tables, but we chose to put it in only one of them.
- Another challenge we faced was the NOT NULL constraints and where to apply them exactly. For example, in the "Works In" relationship, we only knew that the participation of the Staff entity is total, but we had no information about the Department's participation.
- Furthermore, we had to make decisions about the data types and sizes for each attribute, which required careful consideration of the expected data and its constraints.
- We were also challenged by the placement of the triggers, we found it confusing to choose the right table to put them in
- Lastly, we faced some challenges with the Implementation in Mysql, especially with the foreign keys and the triggers.

#### 6 Conclusion

In this deliverable, we transformed the conceptual schema of the MNHS database into a logical model. Each entity, relationship and constraint identified in the ER diagram is mapped into tables with appropriate primary and foreign keys to ensure data integrity through the correct use of constraints .





	IID	CIN	name	sex	Birth	Blood_Group	Phone
•	P000000001	EA345678	Ranya El Kamali	Female	1989-12-03	AB+	0612345678
	P000000002	HA890456	Hassan Mounib	Male	2002-04-17	AB-	0623456789

Figure 1: Patient table

	Staff_ID	Name	Status
•	S000000001	Dr. Yassine	Ophthalmologist
	S000000002	Dr. Salma	Ophthalmologist

Figure 2: Staff table

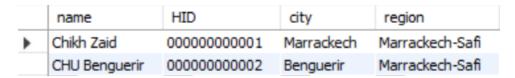


Figure 3: Hospital table



Figure 4: Department table

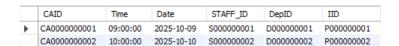


Figure 5: ClinicalActivity table

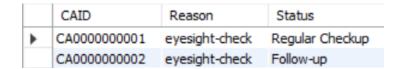


Figure 6: Appointment table

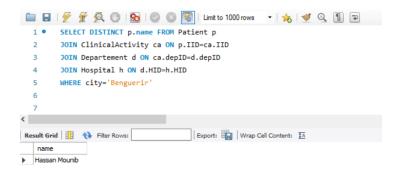


Figure 7: The code and output of the query