



University  
Mohammed VI  
Polytechnic



# Deliverable #: Relational Schema

Data Management Course

UM6P College of Computing

**Professor:** Karima Echihabi    **Program:** Computer Engineering

**Session:** Fall 2025

---

## Team Information

<b>Team Name</b>	QueryMaster
<b>Member 1</b>	El Mehdi Regagui
<b>Member 2</b>	Yasser Jarboua
<b>Member 3</b>	Adam Ibourg-EL Idrissi
<b>Member 4</b>	Salma Mana
<b>Member 5</b>	Hiba Mhirit
<b>Member 6</b>	Sara Qiouame
<b>Member 7</b>	Douaae Mabrouk
<b>Repository Link</b>	<a href="https://github.com/yasserJarboua/QueryMasters/">https://github.com/yasserJarboua/QueryMasters/</a>

# 1 Introduction

The Moroccan National Health Services (MNHS) requires a robust database to manage patients, staff, hospitals, departments, appointments, prescriptions, medications, insurance, billing, and emergencies. This deliverable translates the previously designed ER model into a relational schema, defining primary and foreign keys for all entities. Additionally, we implement a subset of this schema in SQL, including three core tables and a sample query, to demonstrate practical data access.

## 2 Requirements

This deliverable addresses the following tasks:

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:
  - (a) Write CREATE TABLE statements for at least three core entities (e.g., Patient, Hospital, Appointment)
  - (b) Insert at least two tuples per table
  - (c) Write one query that lists the names of patients with scheduled appointments in the city of Benguerir

## 3 Methodology

Our methodology consisted of two main phases: relational schema design and SQL implementation, ensuring a systematic transformation from conceptual model to working database.

### 3.1 Relational Schema Design

In this initial phase, we organized our ER model into a relational schema following the recognized principles of database design. For each entity and relationship, we define attributes, primary keys, and constraints as detailed below:

#### Entities and their attributes

##### 1. Patient

**Attributes:** IID (PK), CIN, Name, Sex, Birth, BloodGroup, Phone.

**Notes:** IID uniquely identifies each patient; CIN appears as a secondary identifier in the diagram, but IID is the central key used for the links.

---

**Foreign keys:** None

**Participation:**

- Total in ClinicalActivity (every activity belongs to a patient).
- Partial in Have and Covers (a patient may have zero or many contacts/insurances).

**Domain checks:**

- CIN → 10–20 alphanumeric chars.
- Name → VARCHAR(100), NOT NULL.
- Sex → 'M' or 'F' only.
- Birth → DATE NOT NULL;
- BloodGroup → string with 5 characters at most, and it has a valid blood group ('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-', 'O+', 'O-').
- Phone → VARCHAR, 10 digits.

## 2. Contact Location

**Attributes:** CLID (PK), Province, City, Street, Number, PostalCode, Phone.

**Notes:** Linked to Patient via “Have” relationship; CLID is the natural key of the contact location entry.

**Foreign keys:** None.

**Participation:**

- Partial on both sides (a patient may have 0–n contact locations; a contact location may belong to multiple patients).

**Domain checks:**

- CLID → INT NOT NULL PK.
- Street → VARCHAR(20).
- Number → alphanumeric.
- City → alphabetic.
- Province → alphabetic.
- PostalCode → VARCHAR(5)
- Phone → VARCHAR, 10 digits.

### 3. Staff

**Attributes:** STAFF\_ID (PK), Name, Status.

**Notes:** Connected to Department and participates in ISA to Practitioner, Caregiving, Technical.

**Foreign keys:** Referenced by ISA subtypes (Practitioner, Caregiving, Technical) and by WorkIn (STAFF\_ID) and ClinicalActivity (Staff\_ID).

**Participation:**

- Total in ISA if every staff member belongs to one of the subtypes.
- Partial otherwise.
- Total in WorkIn

**Domain checks:**

- Staff\_ID INT NOT NULL PK,
- Name → VARCHAR(50) not null.
- Status → VARCHAR(50) NOT NULL.

### 4. Practitioner (ISA of Staff)

**Attributes:** Staff\_ID (PK, also FK to Staff), LicenseNumber, Specialty, Grade, Ward, Certifications.

**Notes:** Uses the parent key STAFF\_ID as its primary key to ensure one-to-one specialization with Staff.

**Foreign keys:** STAFF\_ID → references Staff(Staff\_ID).

**Participation:**

- Total from Practitioner to Staff (each Practitioner is a Staff).
- Partial from Staff to Practitioner (not all staff are Practitioners).

**Domain checks:**

- LicenseNumber : VARCHAR(20).
- Specialty : VARCHAR(20).

---

## 5. Caregiving (ISA of Staff)

**Attributes:** STAFF\_ID (PK, FK to Staff), Specialty, Grade, Ward, Certifications.

**Notes:** Inherits identity from Staff; attributes shown in the ISA cluster apply here as subtype-specific properties for caregiving roles.

**Foreign keys:** STAFF\_ID references Staff(STAFF\_ID).

### Participation:

- Total from Caregiving to Staff.
- Partial from Staff to Caregiving.

### Domain checks:

- Grade: VARCHAR(100)
- Ward: VARCHAR(100)

## 6. Technical (ISA of Staff)

**Attributes:** STAFF\_ID (PK, FK to Staff), Modality, Certifications, Grade.

**Notes:** Shares the Staff identifier to preserve subtype identity and integrity across staff categories.

**Foreign keys:** STAFF\_ID references Staff(STAFF\_ID).

### Participation:

- Total from Technical to Staff.
- Partial from Staff to Technical.

### Domain checks:

- Modality: VARCHAR(100)
- Certifications: VARCHAR(100)

## 7. Department

**Attributes:** DEP\_ID (PK), Name, Specialty.

---

**Notes:** DEP\_ID is unique department identifier.

**Foreign keys:** HID references Hospital(HID).

**Participation:**

- Total in Hospital (each department belongs to one hospital).

**Domain checks:**

- Name: VARCHAR(100)
- Specialty: VARCHAR(100)

## 8. Hospital

**Attributes:** HID (PK), Name, City, Region.

**Notes:** Root organizational entity for departments and stock.

**Foreign keys:** None.

**Participation:**

- Total from Prescription to ClinicalActivity.

**Domain checks:**

- DateIssued: VARCHAR(50), CHECK (LENGTH(DateIssued) >= 4).

## 9. Medication

**Attributes:** DrugID (PK), Class, Name, Form, Strength, ActiveIngredient, Manufacturer.

**Notes:** Central catalog entry for drugs referenced by prescriptions; DrugID is the unique code for a medication record.

**Participation:**

- PARTIAL from Medication to Hospital
- Partial from Medication to Prescription

---

**Domain checks:**

- DrugID:INT NOT NULL
- Name: VARCHAR(100)
- Form: VARCHAR(20)
- Strength: VARCHAR(30)

**10. Prescription**

**Attributes:** PID (PK), DateIssued.

**Notes:** Represents a prescribing event that includes medications with dosage and duration; PID uniquely identifies the prescription document/record.

**Foreign keys:** CAID references ClinicalActivity(CAID).

**Participation:**

- Total from Prescription to ClinicalActivity.

**Domain checks:**

- DateIssued: VARCHAR(50) (date format)

**11. Insurance**

**Attributes:** InsID (PK), Type, Covers.

**Notes:** Linked to Patient via “Has” and to Expense coverage; InsID is the insurer/plan identifier in this context.

**Participation:**

- Partial from insurance to patient
- PARTIAL from insurance to Expense

**Domain checks:**

- INSID: INT NOT NULL
- TYPE:VARCHAR(10) check if(CNOPS, CNSS, RAMED, private, or none)
-

## 12. Expense

**Attributes:** ExID (PK), Total.

**Notes:** Attached to clinical activity or appointment as indicated; ExID is the unique charge/expense record identifier.

**Foreign keys:** InsID references Insurance(InsID), CAID references Clinical Activity(CAID).

**Participation:**

- Partial (not every insurance has an expense).

**Domain checks:**

- Total: DECIMAL(10,2), CHECK (Total >= 0).

## 13. Clinical Activity

**Attributes:** CAID (PK), Time, Date.

**Notes:** Supertype entity for Appointment and Emergency (ISA); CAID is referenced by links to Patient and Staff.

**Foreign keys:** IID references Patient(IID) , DEP\_ID references Departement(DEP\_ID), STAFF\_ID references Staff(STAFF\_ID).

**Participation:**

- Total because each clinical activity occurs exactly one departement(from Clinical Act to clinical activity)
- Total from Clinical Act to patient (each clinical act has exactly one patient)
- Total from Clinical Act to Expense

**Domain checks:**

- CAID INT PRIMARY KEY,
- Date DATE NOT NULL,
- Time TIME NOT NULL,

## 14. Appointment (ISA of Clinical Activity)

**Attributes:** CAID (PK, FK to ClinicalActivity), Reason, Status.



**Notes:** Uses the supertype key CAID as its primary key to maintain one-to-one correspondence with the general clinical activity record.

**Foreign keys:** CAID references Clinical Activity(CAID).

**Domain checks:**

- Reason: Varchar(100)
- Status: varchar (100)

## 15. Emergency (ISA of Clinical Activity)

**Attributes:** CAID (PK, FK to ClinicalActivity), TriageLevel, Outcome.

**Notes:** Inherits the identity CAID from Clinical Activity to keep emergency events aligned with the activity supertype entry.

**Foreign keys:** CAID references Clinical Activity(CAID).

**Domain checks:**

- TriageLevel:VARCHAR(50) NOT NULL,
- Outcome:VARCHAR(50),

## Relationships and their attributes

### 1. Stock

**Attributes:** StockID (PK), UnitPrice, StockTimestamp, Qty, ReorderLevel, HID (FK to Hospital).

**Foreign keys:** HID references Hospital(HID).

**Foreign keys:** Drug\_ID references Medication(Drug\_ID).

### 2. Attached (Expense—Clinical Activity)

**Attributes:** none beyond foreign keys; the diagram labels “Attached” from Expense to Clinical Activity.

**Primary Key:** -If each expense belongs to exactly one clinical activity, the PK can be ExID carried on Expense, and the relationship table is unnecessary; store CAID as a foreign key in Expense. -If modeling a separate link table (to allow flexible associations), use composite (ExID, CAID) so each attachment of a specific expense to a specific activity is unique; this pair naturally identifies the link row and prevents duplicates.

### 3. Covers (Insurance—Clinical Activity or Expense)

**Interpretation from the diagram:** “Covers” radiates from Insurance toward the Clinical Activity/Expense flow, indicating coverage of costs generated by activities and attached as expenses.

**Attributes:** optionally CoverageType, CoveragePercent, Copay, Authorization-Code if captured at the link; none are explicitly drawn, so treat as none unless extending the model.

**Primary Key:** -If coverage is recorded per expense line, use composite (InsID, ExID) to state that a particular insurance covers a particular expense exactly once; the pair is the natural minimal identifier of the coverage record. -If coverage is recorded at the activity level instead, use composite (InsID, CAID) to register that an insurance covers a given clinical activity; again, the pair uniquely identifies the coverage entry without a surrogate.

**Composite Key Justification:** In some implementations Stock can use a composite key (HID, DrugID) to represent per-hospital inventory of a medication; this composite uniquely identifies the stock row without a surrogate and reflects the natural business rule “one stock level per medication per hospital”.

**Notes:** The diagram shows Stock associated to Hospital and Medication; if a surrogate StockID is preferred, keep DrugID and HID as foreign keys with a unique constraint on (HID, DrugID) to enforce the same rule.

**Foreign keys:** InsID references Insurance(InsID), IID references Patient(IID).

### 4. Have (Patient—Contact Location)

**Attributes:** none beyond FKs.

**Primary Key:** Can be composite (IID, CLID) when modeled as an associative table to allow multiple addresses per patient and reuse of locations; this composite is justified because the pair uniquely identifies each association instance without requiring a surrogate.

---

**Foreign keys:** IID references Patient(IID), CLID references Contact Location(CLID).

5. **Has (Patient—Insurance)**

**Attributes:** none beyond FKs.

**Primary Key:** Composite (IID, InsID) to allow a patient to hold multiple policies and prevent duplicates of the same patient-policy link; the pair forms a natural unique identifier of the relationship row.

6. **Linked (Clinical Activity—Patient)**

**Attributes:** none beyond FKs.

**Primary Key:** CAID when each clinical activity links to exactly one patient, or composite (CAID, IID) if the model allows group encounters; the composite, when used, ensures each patient linkage to a given activity is unique.

7. **Occurs (Clinical Activity—Hospital/Department)**

**Attributes:** Date, Time; the relationship itself in the diagram carries no extra attributes.

**Primary Key:** CAID when each activity occurs in exactly one organizational location, or composite (CAID, DEP\_ID) if recording departmental occurrence explicitly; the composite ensures a single occurrence record per activity-department pair without duplicates.

8. **Work In (Staff—Department)**

**Attributes:** none shown.

**Primary Key:** Composite (STAFF\_ID, DEP\_ID) to support staff working in multiple departments and avoid duplicate assignments; the pair is the natural identifier of a staff-department assignment.

**Foreign keys:** CAID references ClinicalActivity(CAID), DEP\_ID references Department(DEP\_ID).

9. **Generates (Clinical Activity—Expense)**

**Attributes:** none shown.

**Primary Key:** ExID if one-to-one, or composite (CAID, ExID) to allow multiple expenses per activity while preserving uniqueness of each linkage.

#### 10. Generate (Clinical Activity/Appointment—Prescription)

**Attributes:** none shown.

**Primary Key:** PID if one-to-one, or composite (CAID, PID) where multiple prescriptions can stem from a single activity; the pair uniquely identifies each linkage without a surrogate.

#### 11. Include (Prescription—Medication)

**Attributes:** dosage, duration.

**Primary Key:** Composite (PID, DrugID) justified because a prescription can include multiple medications and the same medication should not be repeated within the same prescription; the pair naturally and minimally identifies each line item, while supporting line attributes dosage and duration on the association.

**Foreign keys:** PID references Prescription(PID), Drug\_ID references Medication(Drug\_ID).

### 3.2 SQL Implementation Methodology

The SQL implementation of the database was developed based on the previously designed relational schema. Each entity and relationship was translated into a corresponding table with carefully defined attributes and appropriate data types reflecting their domains. Primary keys were established to uniquely identify each record. To maintain referential integrity, foreign key constraints were implemented to enforce the relationships between tables, with consideration of participation constraints by using NOT NULL where total participation was required. Domain integrity was ensured through the use of CHECK constraints that restrict attribute values to valid ranges or sets, such as limiting gender to specific characters. Additional constraints, including UNIQUE and NOT NULL, were added to prevent duplicate or incomplete data. The use of cascading actions on foreign keys was carefully applied to handle dependent records during deletions. Finally, sample data was inserted respecting all constraints to validate the schema's correctness and consistency. This approach guarantees a robust and reliable database structure aligned with the conceptual design.

## 4 Implementation & Results

### 4.1 SQL Code Implementation

```

1  -- Question 3.1 Creating the different needed tables --
2  CREATE DATABASE IF NOT EXISTS MNHS_DB;
3  USE MNHS_DB;
4
5  CREATE TABLE IF NOT EXISTS Patient (
6      IID INT PRIMARY KEY,
7      CIN VARCHAR(20) UNIQUE,
8      Name VARCHAR(100) NOT NULL,
9      Sex CHAR(1)
10         CHECK (Sex IN ('M', 'F')),
11      Birth DATE NOT NULL,
12      BloodGroup VARCHAR(5)
13         CHECK (BloodGroup IN ('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-',
14                                ', 'O+', 'O-')),
15      Phone VARCHAR(15)
16  );
17
18  CREATE TABLE IF NOT EXISTS Staff (
19      Staff_ID INT PRIMARY KEY,
20      Name VARCHAR(50) NOT NULL,
21      Status VARCHAR(50) NOT NULL
22  );
23
24  CREATE TABLE IF NOT EXISTS Practitioner (
25      Staff_ID INT PRIMARY KEY,
26      LicenseNumber VARCHAR(20) NOT NULL,
27      Specialty VARCHAR(50) NOT NULL,
28      FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
29         ON DELETE CASCADE
30  );
31
32  CREATE TABLE IF NOT EXISTS Caregiving (
33      Staff_ID INT PRIMARY KEY,
34      Ward VARCHAR(100),
35      Grade VARCHAR(100),
36      FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
37         ON DELETE CASCADE
38  );
39
40  CREATE TABLE IF NOT EXISTS Technical (
41      Staff_ID INT PRIMARY KEY,
42      Modality VARCHAR(100),
43      Certifications VARCHAR(100),
44      Grade VARCHAR(100),
45      FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
46         ON DELETE CASCADE
47  );
48
49  CREATE TABLE IF NOT EXISTS Hospital (
50      HID INT PRIMARY KEY,

```

```
50     Name VARCHAR(50) NOT NULL,
51     City VARCHAR(50),
52     Region VARCHAR(50)
53 );
54
55 CREATE TABLE IF NOT EXISTS Department (
56     DEP_ID INT PRIMARY KEY,
57     Name VARCHAR(100) NOT NULL,
58     Specialty VARCHAR(100),
59     HID INT NOT NULL,
60     FOREIGN KEY (HID) REFERENCES Hospital(HID)
61         ON DELETE CASCADE
62 );
63
64 CREATE TABLE IF NOT EXISTS WorkIn(
65     Staff_ID INT NOT NULL,
66     DEP_ID INT NOT NULL,
67     PRIMARY KEY (Staff_ID, DEP_ID),
68     FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
69         ON DELETE CASCADE,
70     FOREIGN KEY (DEP_ID) REFERENCES Department(DEP_ID)
71         ON DELETE CASCADE
72 );
73
74 CREATE TABLE IF NOT EXISTS ClinicalActivity (
75     CAID INT PRIMARY KEY,
76     'Date' DATE NOT NULL,
77     'Time' TIME NOT NULL,
78     IID INT NOT NULL,
79     Staff_ID INT NOT NULL,
80     DEP_ID INT NOT NULL,
81     FOREIGN KEY (IID) REFERENCES Patient(IID)
82         ON DELETE NO ACTION,
83     FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
84         ON DELETE NO ACTION,
85     FOREIGN KEY (DEP_ID) REFERENCES Department(DEP_ID)
86         ON DELETE NO ACTION
87 );
88
89 CREATE TABLE IF NOT EXISTS Appointment (
90     CAID INT PRIMARY KEY,
91     Reason VARCHAR(255),
92     Status VARCHAR(50) NOT NULL,
93     FOREIGN KEY (CAID) REFERENCES ClinicalActivity(CAID)
94         ON DELETE CASCADE
95 );
96
97 CREATE TABLE IF NOT EXISTS Emergency(
98     CAID INT PRIMARY KEY,
99     TriageLevel VARCHAR(50) NOT NULL,
100     Outcome VARCHAR(50),
```

```

101     FOREIGN KEY (CAID) REFERENCES ClinicalActivity(CAID)
102     ON DELETE CASCADE
103 );
104
105 -- Question 2.2 Inserting different tuples to the different
106    tables --
107 INSERT INTO Hospital (HID, Name, City, Region)
108 VALUES
109 (1, 'CHU Mohammed VI', 'Benguerir', 'Marrakech-Safi'),
110 (2, 'CHU Ibn Rochd', 'Casablanca', 'Casablanca-Settat'),
111 (3, 'Hopital Militaire', 'Rabat', 'Rabat-Sale-Kenitra'),
112 (4, 'CHU Mohammed VI', 'Marrakech', 'Marrakech-Safi'),
113 (5, 'Hopital Mohamed V', 'Tangier', 'Tanger-Tetouan-Al Hoceima');
114
115 INSERT INTO Patient (IID, CIN, Name, Sex, Birth, BloodGroup,
116    Phone)
117 VALUES
118 (1, 'BB532415', 'Salma', 'F', '2006-04-14', 'O+', '0612452318'),
119 (2, 'BJ234567', 'Hiba', 'F', '2006-06-26', 'O-', '0614567833'),
120 (3, 'BH123456', 'Sara', 'F', '2006-07-26', 'A+', '0616234590'),
121 (4, 'BH125457', 'Amine', 'M', '1982-07-26', 'A-', '0616224596'),
122 (5, 'BJ113426', 'Mohammed', 'M', '1998-07-26', 'A+', '0617234594'
123    );
124
125 INSERT INTO Department (DEP_ID, Name, Specialty, HID)
126 VALUES
127 (1, 'Radiology', 'Medical imaging', 1),
128 (2, 'Cardiology', 'Cardiovascular Medicine', 2),
129 (3, 'Neurology', 'Brain and nervous system', 3),
130 (4, 'Oncology', 'Cancer Treatment', 4),
131 (5, 'Neurosurgery', 'Brain and Nervous System', 5);
132
133 INSERT INTO Staff (Staff_ID, Name, Status)
134 VALUES
135 (1, 'Dr Youssef Bennani', 'Active'),
136 (2, 'Dr Fatima Zahra Alami', 'Active'),
137 (3, 'Dr Hassan El Idrissi', 'Active'),
138 (4, 'Dr Amina Tazi', 'Active'),
139 (5, 'Dr Omar Kettani', 'On Leave'),
140 (6, 'Karima Belkadi', 'Active'),
141 (7, 'Said Moussaoui', 'Active'),
142 (8, 'Laila Benjelloun', 'Active'),
143 (9, 'Ahmed Tahiri', 'Active'),
144 (10, 'Sanaa Cherkaoui', 'Active');
145
146 INSERT INTO Practitioner (Staff_ID, LicenseNumber, Specialty)
147 VALUES
148 (1, 'MED-RAB-2015-1234', 'Radiology'),
149 (2, 'MED-CAS-2012-5678', 'Cardiology'),
150 (3, 'MED-RAB-2010-9012', 'Neurology'),
151 (4, 'MED-MAR-2018-3456', 'Oncology'),

```

```

149 (5, 'MED-TAN-2016-7890', 'Neurosurgery');
150
151 INSERT INTO Caregiving (Staff_ID, Ward, Grade)
152 VALUES
153 (6, 'Ward A - Cardiology', 'Senior Nurse'),
154 (7, 'Ward B - Radiology', 'Nurse'),
155 (8, 'Ward C - Neurology', 'Head Nurse');
156
157 INSERT INTO Technical (Staff_ID, Modality, Certifications, Grade)
158 VALUES
159 (9, 'MRI and CT Scan', 'Certified Radiologic Technologist', '
    Senior Technician'),
160 (10, 'Ultrasound', 'Diagnostic Medical Sonographer', 'Technician'
    );
161
162 INSERT INTO WorkIn (Staff_ID, DEP_ID)
163 VALUES
164 (1, 1),
165 (2, 2),
166 (3, 3),
167 (4, 4),
168 (5, 5),
169 (6, 2),
170 (7, 1),
171 (8, 3),
172 (9, 1),
173 (10, 1);
174
175 INSERT INTO ClinicalActivity (CAID, 'Date', 'Time', IID, Staff_ID
    , DEP_ID)
176 VALUES
177 (1, '2025-10-15', '09:00:00', 1, 1, 1),
178 (2, '2025-10-16', '10:30:00', 2, 2, 2),
179 (3, '2025-10-17', '14:00:00', 3, 3, 3),
180 (4, '2025-10-18', '11:00:00', 4, 4, 4),
181 (5, '2025-10-20', '15:30:00', 5, 5, 5),
182 (6, '2025-10-22', '08:30:00', 1, 1, 1),
183 (7, '2025-10-25', '13:00:00', 3, 2, 2),
184 (8, '2025-10-16', '18:45:00', 2, 2, 2),
185 (9, '2025-10-17', '22:15:00', 3, 3, 3),
186 (10, '2025-10-20', '03:30:00', 5, 5, 5);
187
188 INSERT INTO Appointment (CAID, Reason, Status)
189 VALUES
190 (1, 'X-Ray examination', 'scheduled'),
191 (2, 'ECG and heart checkup', 'completed'),
192 (3, 'Neurological assessment', 'scheduled'),
193 (4, 'Cancer screening', 'scheduled'),
194 (5, 'Pre-surgery consultation', 'cancelled'),
195 (6, 'Follow-up scan', 'scheduled'),
196 (7, 'Cardiac stress test', 'completed');

```



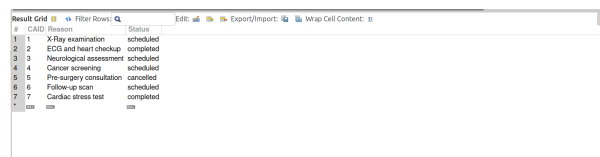
```

197
198 INSERT INTO Emergency (CAID, TriageLevel, Outcome)
199 VALUES
200 (8, 'Level 2 - High Priority', 'Admitted to ICU'),
201 (9, 'Level 3 - Medium Priority', 'Treated and Released'),
202 (10, 'Level 1 - Critical', 'Transferred to Surgery');
203
204 -- Question 3.3 --
205 SELECT p.Name
206 FROM Patient p
207 JOIN ClinicalActivity ca ON p.IID = ca.IID
208 JOIN Appointment a ON ca.CAID = a.CAID
209 JOIN Department d ON ca.DEP_ID = d.DEP_ID
210 JOIN Hospital h ON d.HID = h.HID
211 WHERE h.City = 'Benguerir' AND a.Status = 'scheduled';
212
213 -- We Display the Results --
214 SHOW TABLES;
215 SELECT * FROM Patient;
216 SELECT * FROM Staff;
217 SELECT * FROM Practitioner;
218 SELECT * FROM Caregiving;
219 SELECT * FROM Technical;
220 SELECT * FROM Hospital;
221 SELECT * FROM Department;
222 SELECT * FROM WorkIn;
223 SELECT * FROM ClinicalActivity;
224 SELECT * FROM Appointment;
225 SELECT * FROM Emergency;

```

## 4.2 Screenshots of Outputs

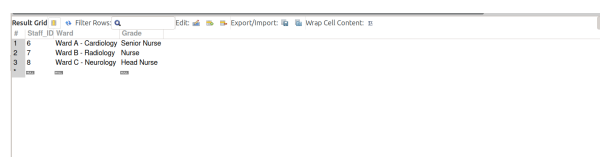
### 1. Appointment Table



CAID	Reason	Status
1	X-Ray examination	scheduled
2	ECG and heart checkup	completed
3	Neurological assessment	scheduled
4	Cancer screening	scheduled
5	Pre-surgery consultation	cancelled
6	Follow-up scan	scheduled
7	Cardiac stress test	completed

Figure 1: Appointment Table

### 2. Caregiving Table



Staff_ID	Ward	Grade
6	Ward A - Cardiology	Senior Nurse
7	Ward B - Radiology	Nurse
8	Ward C - Neurology	Head Nurse

Figure 2: Caregiving Table

### 3. Clinical Activity Table

#	CAID	Date	Time	Value
1	1	2025-10-15	09:00:00	1
2	2	2025-10-16	10:30:00	2
3	3	2025-10-17	14:00:00	3
4	4	2025-10-18	11:00:00	4
5	5	2025-10-20	15:30:00	5
6	6	2025-10-22	08:30:00	1
7	7	2025-10-25	13:00:00	2
8	8	2025-10-16	18:45:00	2
9	9	2025-10-17	22:15:00	3
10	10	2025-10-20	03:30:00	5

Figure 3: Clinical Activity Table

### 4. Department Table

#	ID	Name	Speciality
1	1	Radiology	Medical imaging
2	2	Cardiology	Cardiovascular Medicine
3	3	Neurology	Brain and nervous system
4	4	Oncology	Cancer Treatment
5	5	Neurosurgery	Brain and Nervous System

Figure 4: Department Table

### 5. Emergency Table

#	CAID	TriageLevel	Outcome
1	1	Level 2 - High Priority	Admitted to ICU
2	2	Level 3 - Medium Priority	Treated and Released
3	3	Level 1 - Critical	Transferred to Surgery

Figure 5: Emergency Table

### 6. Hospital Table

#	ID	Name	City	Region
1	1	CHU Mohammed VI	Benguerat	Marrakech-Safi
2	2	CHU Ibn Rochd	Casablanca	Casablanca-Settat
3	3	Hôpital Militaire	Rabat	Rabat-Sale-Kenitra
4	4	CHU Mohammed VI	Marrakech	Marrakech-Safi
5	5	Hôpital Mohamed V	Tanger	Tanger-Tetouan-Al Hoceima

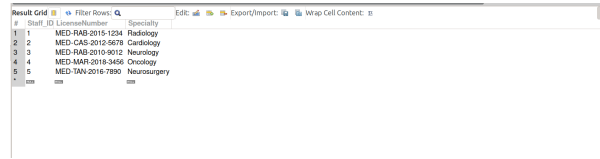
Figure 6: Hospital Table

### 7. Patient Table

#	ID	CIN	Name	Sex	Birth	BloodGroup
1	1	BB302415	Salma	F	2006-04-14	O+
2	2	BL204567	Hiba	F	2006-06-26	O-
3	3	BH123456	Sara	F	2006-07-26	A+
4	4	BH123457	Amine	M	1982-07-26	A-
5	5	BL112420	Mohammed	M	1999-07-26	A+

Figure 7: Patient Table

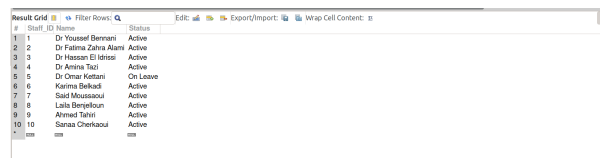
## 8. Practitioner Table



#	Staff_ID	LicenseNumber	Speciality
1	1	MED-RAB 2015-1254	Radiology
2	2	MED-CAS 2012-5678	Cardiology
3	3	MED-RAB 2010-9012	Neurology
4	4	MED-MAR 2016-3456	Oncology
5	5	MED-TAN 2016-7890	Neurosurgery

Figure 8: Practitioner Table

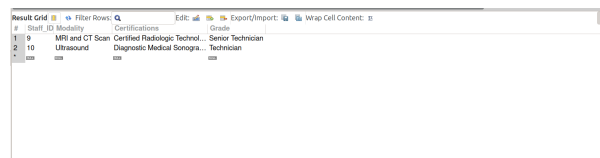
## 9. Staff Table



#	Staff_ID	Name	Status
1	1	Dr Youssef Bernani	Active
2	2	Dr Fatima Zahra Alami	Active
3	3	Dr Hassan El Jorjori	Active
4	4	Dr Amine Tazi	Active
5	5	Dr Omar Kettani	On Leave
6	6	Karima Bekkadi	Active
7	7	Said Moussaoui	Active
8	8	Laila Benmiloud	Active
9	9	Ahmed Tahiri	Active
10	10	Samia Cherkaoui	Active

Figure 9: Staff Table

## 10. Technical Table



#	Staff_ID	Modality	Certifications	Grade	Senior
1	9	MRI and CT Scan	Certified Radiologic Technol...	Senior	Technician
2	10	Ultrasound	Diagnostic Medical Sonograp...	Technician	

Figure 10: Technical Table

# 5 Discussion

During the implementation of this deliverable, we encountered several challenges related to database normalization and constraint enforcement. Ensuring referential integrity between multiple entities such as Patient, ClinicalActivity, and Department required careful attention to foreign key order and data insertion sequence. Additionally, designing ISA (inheritance) relationships between Staff, Practitioner, Caregiving, and Technical demanded consistent primary key reuse to maintain one-to-one mappings.

Another challenge was implementing realistic domain constraints, such as ensuring valid birth dates and enforcing unique identifiers like CIN and LicenseNumber. Testing SQL inserts helped verify constraint correctness and prevent logical inconsistencies.

Through this process, we gained a deeper understanding of relational schema design, normalization, and how conceptual models translate into physical database structures. The hands-on SQL implementation highlighted the importance of incremental testing, constraint validation, and careful data ordering.

---

## 6 Conclusion

This deliverable successfully translated the conceptual ER model of the Moroccan National Health Services (MNHS) system into a robust relational schema and validated its correctness through SQL implementation. The schema ensures data integrity, minimizes redundancy, and supports key relationships among patients, staff, hospitals, and clinical activities.

The implemented query demonstrates how information can be efficiently retrieved using relational joins, showing the practical use of the design in real-world healthcare management. Overall, this project strengthened our understanding of database modeling principles, SQL implementation, and the importance of structured data representation in healthcare systems.