University
Mohammed VI
Polytechnic

College of Computing

Deliverable #: Relational Schema

Data Management Course

UM6P College of Computing

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

Team Information

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

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:

- HID: INT NOT NULL,
- NAME: VARCHAR(100) NOT NULL,
- CITY: VARCHAR(100),
- REGION : VARCHAR(100)

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.
- Partial from Prescription tp Medication

Domain checks:

- PID : INT NOT NULL,
- 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 exacly one departement(from Clinical Act to clinical activity)
- Total from Clinical Act to patient (each clinal 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, AuthorizationCode 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 Departement(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

```sql
-- Question 3.1 Creating the different needed tables --
CREATE DATABASE IF NOT EXISTS MNHS_DB;
USE MNHS_DB;

CREATE TABLE IF NOT EXISTS Patient (
    IID INT PRIMARY KEY,
    CIN VARCHAR(20) UNIQUE,
    Name VARCHAR(100) NOT NULL,
    Sex CHAR(1)
        CHECK (Sex IN ('M', 'F')),
    Birth DATE NOT NULL,
    BloodGroup VARCHAR(5)
        CHECK (BloodGroup IN ('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-
            ', 'O+', 'O-')),
    Phone VARCHAR(15)
);

CREATE TABLE IF NOT EXISTS Staff (
    Staff_ID INT PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Status VARCHAR(50) NOT NULL
);

CREATE TABLE IF NOT EXISTS Practitioner (
    Staff_ID INT PRIMARY KEY,
    LicenseNumber VARCHAR(20) NOT NULL,
    Specialty VARCHAR(50) NOT NULL,
    FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
        ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Caregiving (
    Staff_ID INT PRIMARY KEY,
    Ward VARCHAR(100),
    Grade VARCHAR(100),
    FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
        ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Technical (
    Staff_ID INT PRIMARY KEY,
    Modality VARCHAR(100),
    Certifications VARCHAR(100),
    Grade VARCHAR(100),
    FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
        ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Hospital (
    HID INT PRIMARY KEY,
```

```
    Name VARCHAR(50) NOT NULL,
    City VARCHAR(50),
    Region VARCHAR(50)
);

CREATE TABLE IF NOT EXISTS Department (
    DEP_ID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Specialty VARCHAR(100),
    HID INT NOT NULL,
    FOREIGN KEY (HID) REFERENCES Hospital(HID)
        ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS WorkIn(
    Staff_ID INT NOT NULL,
    DEP_ID INT NOT NULL,
    PRIMARY KEY (Staff_ID, DEP_ID),
    FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
        ON DELETE CASCADE,
    FOREIGN KEY (DEP_ID) REFERENCES Department(DEP_ID)
        ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS ClinicalActivity (
    CAID INT PRIMARY KEY,
    `Date` DATE NOT NULL,
    `Time` TIME NOT NULL,
    IID INT NOT NULL,
    Staff_ID INT NOT NULL,
    DEP_ID INT NOT NULL,
    FOREIGN KEY (IID) REFERENCES Patient(IID)
        ON DELETE NO ACTION,
    FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
        ON DELETE NO ACTION,
    FOREIGN KEY (DEP_ID) REFERENCES Department(DEP_ID)
        ON DELETE NO ACTION
);

CREATE TABLE IF NOT EXISTS Appointment (
    CAID INT PRIMARY KEY,
    Reason VARCHAR(255),
    Status VARCHAR(50) NOT NULL,
    FOREIGN KEY (CAID) REFERENCES ClinicalActivity(CAID)
        ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Emergency(
    CAID INT PRIMARY KEY,
    TriageLevel VARCHAR(50) NOT NULL,
    Outcome VARCHAR(50),
```

```
    FOREIGN KEY (CAID) REFERENCES ClinicalActivity(CAID)
        ON DELETE CASCADE
);

-- Question 2.2 Inserting different tuples to the different tables --
INSERT INTO Hospital (HID, Name, City, Region)
VALUES
(1, 'CHU Mohammed VI', 'Benguerir', 'Marrakech-Safi'),
(2, 'CHU Ibn Rochd', 'Casablanca', 'Casablanca-Settat'),
(3, 'Hopital Militaire', 'Rabat', 'Rabat-Sale-Kenitra'),
(4, 'CHU Mohammed VI', 'Marrakech', 'Marrakech-Safi'),
(5, 'Hopital Mohamed V', 'Tangier', 'Tanger-Tetouan-Al Hoceima');

INSERT INTO Patient (IID, CIN, Name, Sex, Birth, BloodGroup, Phone)
VALUES
(1, 'BB532415', 'Salma', 'F', '2006-04-14', 'O+', '0612452318'),
(2, 'BJ234567', 'Hiba', 'F', '2006-06-26', 'O-', '0614567833'),
(3, 'BH123456', 'Sara', 'F', '2006-07-26', 'A+', '0616234590'),
(4, 'BH125457', 'Amine', 'M', '1982-07-26', 'A-', '0616224596'),
(5, 'BJ113426', 'Mohammed', 'M', '1998-07-26', 'A+', '0617234594');

INSERT INTO Department (DEP_ID, Name, Specialty, HID)
VALUES
(1, 'Radiology', 'Medical imaging', 1),
(2, 'Cardiology', 'Cardiovascular Medicine', 2),
(3, 'Neurology', 'Brain and nervous system', 3),
(4, 'Oncology', 'Cancer Treatment', 4),
(5, 'Neurosurgery', 'Brain and Nervous System', 5);

INSERT INTO Staff (Staff_ID, Name, Status)
VALUES
(1, 'Dr Youssef Bennani', 'Active'),
(2, 'Dr Fatima Zahra Alami', 'Active'),
(3, 'Dr Hassan El Idrissi', 'Active'),
(4, 'Dr Amina Tazi', 'Active'),
(5, 'Dr Omar Kettani', 'On Leave'),
(6, 'Karima Belkadi', 'Active'),
(7, 'Said Moussaoui', 'Active'),
(8, 'Laila Benjelloun', 'Active'),
(9, 'Ahmed Tahiri', 'Active'),
(10, 'Sanaa Cherkaoui', 'Active');

INSERT INTO Practitioner (Staff_ID, LicenseNumber, Specialty)
VALUES
(1, 'MED-RAB-2015-1234', 'Radiology'),
(2, 'MED-CAS-2012-5678', 'Cardiology'),
(3, 'MED-RAB-2010-9012', 'Neurology'),
(4, 'MED-MAR-2018-3456', 'Oncology'),
```

```
(5, 'MED-TAN-2016-7890', 'Neurosurgery');

INSERT INTO Caregiving (Staff_ID, Ward, Grade)
VALUES
(6, 'Ward A - Cardiology', 'Senior Nurse'),
(7, 'Ward B - Radiology', 'Nurse'),
(8, 'Ward C - Neurology', 'Head Nurse');

INSERT INTO Technical (Staff_ID, Modality, Certifications, Grade)
VALUES
(9, 'MRI and CT Scan', 'Certified Radiologic Technologist', '
   Senior Technician'),
(10, 'Ultrasound', 'Diagnostic Medical Sonographer', 'Technician'
   );

INSERT INTO WorkIn (Staff_ID, DEP_ID)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 2),
(7, 1),
(8, 3),
(9, 1),
(10, 1);

INSERT INTO ClinicalActivity (CAID, `Date`, `Time`, IID, Staff_ID
   , DEP_ID)
VALUES
(1, '2025-10-15', '09:00:00', 1, 1, 1),
(2, '2025-10-16', '10:30:00', 2, 2, 2),
(3, '2025-10-17', '14:00:00', 3, 3, 3),
(4, '2025-10-18', '11:00:00', 4, 4, 4),
(5, '2025-10-20', '15:30:00', 5, 5, 5),
(6, '2025-10-22', '08:30:00', 1, 1, 1),
(7, '2025-10-25', '13:00:00', 3, 2, 2),
(8, '2025-10-16', '18:45:00', 2, 2, 2),
(9, '2025-10-17', '22:15:00', 3, 3, 3),
(10, '2025-10-20', '03:30:00', 5, 5, 5);

INSERT INTO Appointment (CAID, Reason, Status)
VALUES
(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');
```

```sql

INSERT INTO Emergency (CAID, TriageLevel, Outcome)
VALUES
(8, 'Level 2 - High Priority', 'Admitted to ICU'),
(9, 'Level 3 - Medium Priority', 'Treated and Released'),
(10, 'Level 1 - Critical', 'Transferred to Surgery');

-- Question 3.3 --
SELECT p.Name
FROM Patient p
JOIN ClinicalActivity ca ON p.IID = ca.IID
JOIN Appointment a ON ca.CAID = a.CAID
JOIN Department d ON ca.DEP_ID = d.DEP_ID
JOIN Hospital h ON d.HID = h.HID
WHERE h.City = 'Benguerir' AND a.Status = 'scheduled';

-- We Display the Results --
SHOW TABLES;
SELECT * FROM Patient;
SELECT * FROM Staff;
SELECT * FROM Practitioner;
SELECT * FROM Caregiving;
SELECT * FROM Technical;
SELECT * FROM Hospital;
SELECT * FROM Department;
SELECT * FROM WorkIn;
SELECT * FROM ClinicalActivity;
SELECT * FROM Appointment;
SELECT * FROM Emergency;
```

4.2 Screenshots of Outputs

1. Appointment Table

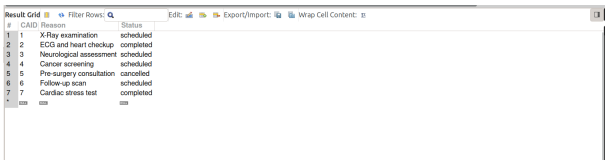


Figure 1: Appointment Table

2. Caregiving Table

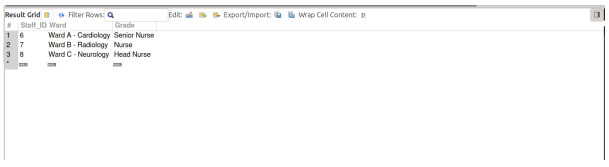
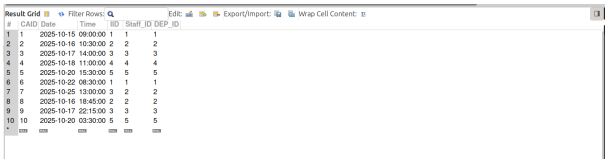


Figure 2: Caregiving Table

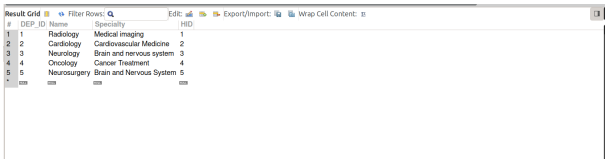
3. Clinical Activity Table



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

Figure 3: Clinical Activity Table

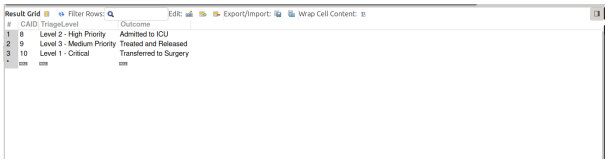
4. Department Table



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

Figure 4: Department Table

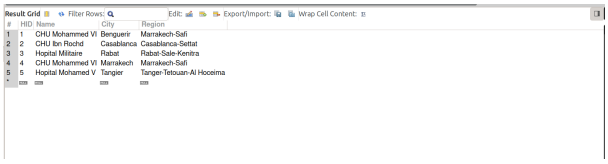
5. Emergency Table



#	CAID	TriageLevel	Outcome
1	8	Level 2 - High Priority	Admitted to ICU
2	9	Level 3 - Medium Priority	Treated and Released
3	10	Level 1 - Critical	Transferred to Surgery

Figure 5: Emergency Table

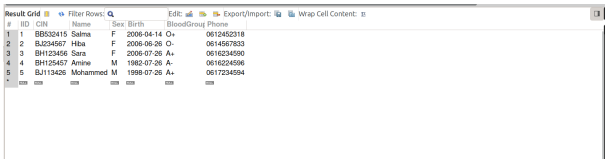
6. Hospital Table



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

Figure 6: Hospital Table

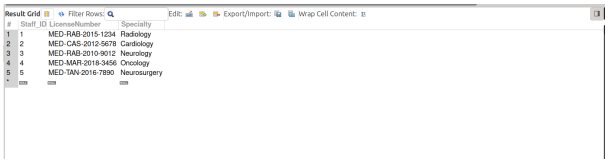
7. Patient Table



#	HID	CIN	Name	Sex	Birth	BloodGroup	Phone
1	1	BB532415	Salma	F	2006-04-14	O+	0612452318
2	2	BJ234567	Hiba	F	2006-06-26	O-	0614567833
3	3	BH123456	Sara	F	2006-07-26	A+	0616234590
4	4	BH125457	Amine	M	1982-07-26	A-	0616224596
5	5	BJ113426	Mohammed	M	1998-07-26	A+	0617234594

Figure 7: Patient Table

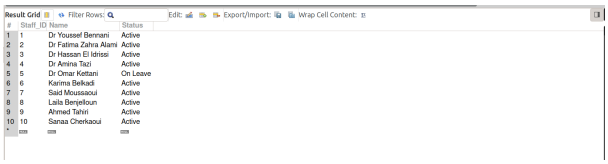
8. Practitioner Table



#	Staff_ID	LicenseNumber	Specialty
1	1	MED-RAB-2015-1234	Radiology
2	2	MED-CAS-2012-5678	Cardiology
3	3	MED-RAB-2010-9012	Neurology
4	4	MED-MAR-2018-3456	Oncology
5	5	MED-TAN-2016-7890	Neurosurgery

Figure 8: Practitioner Table

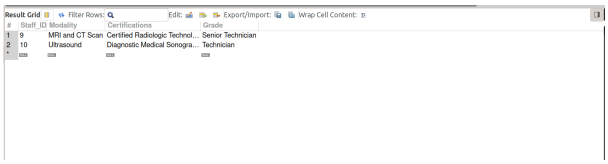
9. Staff Table



#	Staff_ID	Name	Status
1	1	Dr Youssef Bennani	Active
2	2	Dr Fatima Zahra Alami	Active
3	3	Dr Hassan El Idrissi	Active
4	4	Dr Amina Tazi	Active
5	5	Dr Omar Kettani	On Leave
6	6	Karima Belkadi	Active
7	7	Said Moussaoui	Active
8	8	Laila Benjelloun	Active
9	9	Ahmed Tahiri	Active
10	10	Sanaa Cherkaoui	Active

Figure 9: Staff Table

10. Technical Table



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