University Mohammed VI Polytechnic

College of Computing

Moroccan National Health Services/Logical Design

Data Management Course

UM6P College of Computing

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

Team Information

Team Name	**AtlasDB**
Member 1	Ahmed ENNASSIB
Member 2	Abdeljalil EL ACHEHAB
Member 3	Salma EL KADI
Member 4	Omar El BOUKILI
Member 5	Adam El MANNANI
Member 6	Housam EL GOUINA
Repository Link	`https://github.com/BoukiliOmar/DBMS-AtlasDB`

1 Introduction

The problem involves designing a conceptual data model for the Moroccan National Health Services (MNHS) to manage patients, staff, hospitals, departments, appointments, prescriptions, medications, insurance, billing, emergencies, and pharmacy inventory with remarking every considerable relations between all these entities. This deliverable covers the creation of an ER diagram that addresses these core requirements by identifying entities, attributes, relationships, and constraints. It supports operational queries and future analytics , the thing that ensures an organized , enhanced and efficiently working database design for MNHS management.

2 Requirements

The deliverable for this lab consists of a complete and well-documented relational schema based on the provided Entity-Relationship (ER) model, accompanied by SQL scripts demonstrating its implementation. We are required to translate their ER model into relational tables, clearly defining primary and foreign keys, and justifying any composite keys used. We must also state all necessary integrity constraints to maintain data consistency and enforce semantics.
The SQL implementation should include CREATE TABLE statements for at least three core entities, such as Patient, Hospital, and Appointment, along with the insertion of at least two records per table. Additionally, a query must be written to list the names of patients with scheduled appointments in Benguerir.

3 Methodology

In this lab, our group followed a systematic and structured methodology to develop the logical design of the database.
We began by analyzing the conceptual Entity–Relationship (ER) model to identify all relevant entities, their attributes, and the corresponding primary keys.
Each entity was then translated into a relational schema while ensuring normalization and referential integrity.
Relationships between entities were carefully examined to determine the appropriate cardinalities, with composite primary keys used to represent many-to-many associations such as *Stock*, *Covers*, and *Include*.
Foreign keys were defined to maintain referential links and ensure data consistency across tables.
Additionally, we incorporated ISA hierarchies, such as those within the *Staff* entity (*Practitioner*, *Caregiving*, and *Technical*), by applying inheritance principles that preserve total participation in the parent entity.
Domain and integrity constraints were also established to enforce data validity—for instance, restricting enumerated attributes such as *Sex*, *Status*, and *Blood Group* to predefined values, and applying logical checks on numeric attributes like *Total* and *Phone*.
Once the logical schema was finalized, we validated it through SQL implementation, testing key relationships, constraints, and queries to confirm the model's correctness and alignment with the relational design principles outlined in the lab instructions.

4 Implementation & Results

Logical design :

Patient:

→ **Attributes :**

- IID ; **Primary Key**
- CIN ; Domain: CHAR(10), **UNIQUE**, **NOT NULL**
- Name ; Domain: VARCHAR(100), **NOT NULL**
- Sex ; Domain: ENUM('Male', 'Female')
- Birth ; Domain: DATE
- BloodGroup ; Domain: ENUM('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-', 'O+', 'O-')
- Phone ; Domain: VARCHAR(15)

→ **Relationships linked to Patient :**

- **Have :**
  - PatientID : **Foreign Key** (ref *Patient(IID)*)
  - ContactID : **Foreign Key** (ref *Contact(CID)*)
  - (PatientID, ContactID) : **Primary Key**
- **Covers :**
  - PatientID : **Foreign Key** (ref *Patient(IID)*)
  - InsuranceID : **Foreign Key** (ref *Insurance(InsID)*)
  - (PatientID, InsuranceID) : **Primary Key**

ClinicalActivity :

attributes:

- CAID : PRIMARY KEY , NOT NULL , UNIQUE
- Time : TIME
- Date : DATE
- Type : ENUM('Apointment' , 'emergency')

Foreign keys :

- ExID (Foreign Key ,references Expenses)
- IID(Foreign Key, references Patient)
- DEPID(Foreign Key, references Department)

participation:
Each clinical activity is either an appointment or an emergency at a given time and date , it has its own id and generates one or more expenses for a patient , all of this is happening in a single department.
contact location: attributes :

- CLID: Primary key , unique , not null
- City : not null
- Street
- Number
- Province
- Phone
- PostalCod

Relationship :
Has(LocationID , IID , (LocationID ,IID) :Primary key)

- LocationID : foreign key references Location-Contact(LocationID)
- IID : foreign key references Patient(IID)

participation:
Each patient is associated with at least one adress throught the relationship 'has' (0-to many relationship).
Domaine constraintes:

- Phone length $\leq$ 15
- Email must contain "@"
- City NOT NULL

Expense:
attributes:

- ExpenseID: unique ID for each expense
- Total : int
- foreign key CAID references clinical-activity(CAID)

participation:
an expense is attached to a one or many insurances.
each expense is genrated by one exact clinical activity.
Relationship:
Attached (InsID ,ExID , Primary key (InsID ,ExID))

- InsID : foreign key references Insurrance(InsID)

- ExID :foreign key references Expenses(ExID)

Domain constraints (for Expenses entity) :

- Total : int , check(Total > 0)
- ExID : primary key , unique , not null

Insurance:
attributes:

- InsID ; Primary key :int
- Type ; Domain : enum("CNOPS","CNSS","RAMED","Private","None")

Relationships linked to insurance:
Insurance (Covers) Patient :

- InsuranceID : Foreign Key (ref InsID) : int
- atientID : Foreign Key (Ref IID) :int
- (InsuranceID,PatientID) : Primary Key

Insurance (Attached) Expenses:

- ExpenseID : Foreign Key (ref ExID) :int
- InsuranceID : Foreign Key (ref InsID) :int
- (ExpenseID, InsuranceID) : Primary Key

Prticipation :

- Insurance covers a patient (1 to many relation); a plan can cover many patients but the patient's participation is optional.
- Insurance is attached to expenses (1 to 1 relation) ; The insurance only covers the patients bill (A patient's subscription to an insurance doesn't cover somebody else's bill)

Prescription:
attributes:

- PID ; Primary key :int
- DateIssued : Date

Relationships linked to Prescription:
- Prescription (Include) Medication:

- Dosage :int
- Duration :
- PrescriptionID : Foreign key (ref PID) :int

- MedID : Foreign Key (ref DrugID) : int
- (PrescriptionID, MedID) : Primary key

-Clinical activity (Generate) Prescription:

- PrescriptionID : Foreign key (ref PID) : int
- ClinActID : Foreign key (ref CAID) :int
- (PrescriptionID,ClinActID) : Primary Key

participation:

- Prescription includes medication (Many to Many relatio) ; A prescription can have many medications , and a medication can be in many prescriptions
- Clinical activity generates prescription (1 to 1 relation); each clinical activity a patient has , he has to pay for it

Medication:
attributes:

- DrugID ; Primary key :int
- Class : varchar
- Name : varchar
- Form : varchar
- Strength : varchar
- ActiveIngredient : varchar
- Manufacturer : varchar

Relationships linked to Medication:
- Mediation (Stock) Hospital:

- UnitPrice : int
- StockTimestamp : time
- Qty : int
- ReorderLevel : varchar
- MedID : Foreign Key (ref DrugID) : int
- HospID : Foreign Key (ref HID) : int
- (HospID,MedID) : Primary Key

participation:

- Stock between hospital and medication (many to many relation) ; The hospital's stock provides as many medications needed , and many medications can be found in the hospital's stock

All relationships use composite keys to enforce the participation of the entities linked to it (covers , attached, include, generate,stock)

Hospital:

attributes:

- HID : Primary Key
- Name : Domain: VARCHAR(100), NOT NULL.
- City : Domain: ENUM('Benguerir', 'Casablanca', 'Rabat', 'Marrakech', . . .).
- Region : Domain: VARCHAR(100).

Relationships linked to Hospital :

-Hospital (owns) Department :

- HID : Foreign Key (ref Hospital(HID))
- DEPID : Foreign Key (ref Department(DEPID))
- (HID, DEPID) : Composite Primary Key. justified: ensures a department is uniquely associated with one hospital.
- Participation: total for both. A hospital can't exist without a department in it, and a department must belong to a hospital.

Department :

attributes:

- DEPID ; Primary Key
- Name ; Domain: VARCHAR(100), NOT NULL
- Speciality ; Domain: ENUM('Cardiology', 'Radiology', 'Neurology', 'Emergency', 'Paediatrics', 'General', . . .)
- HID ; Foreign Key (ref Hospital(HID))

Relationships linked to Department :

-Department (Employ) Staff :

- DEPID : Foreign Key (ref DEPID)
- STAFFID : Foreign Key (ref Staff(STAFFID))
- (DEPID, STAFFID): Primary Key. justified: multiple staff can work in a department, pair ensures uniqueness
- Participation: total for Staff (there is no staff member who doesn't work in a department), total for Department (at least one staff member in a department)

-Clinical Activity (occurs) Department:

- DEPID: Foreign Key (ref Department (DEPID))
- CAID: Foreign Key (ref Clinical Activity (CAID))
- (DEPID, CAID): Primary Key. justified: one department can host multiple clinical activities, ensures unique activity-department pair.
- Participation: Partial for Clinical Activity (a clinical activity can exist before being assigned to a department), total for Department.

Staff:
attributes:

- STAFFID ; Primary Key
- Name ; Domain: VARCHAR(100), NOT NULL
- Status ; Domain: ENUM('Practitioner', 'Caregiving', 'Technical')
- ISA Hierarchy
  - Practitioner: LicenseNumber (VARCHAR (20)), Speciality (VARCHAR(50))
  - Caregiving: Grade (ENUM('A','B','C')), Ward (VARCHAR(50))
  - Technical: Modality (VARCHAR(50)), Certifications (VARCHAR (100))

Relationships linked to Staff:
- Staff (WorkIn) Department:

- STAFFID: Foreign Key (ref STAFFID)
- DEPID: Foreign Key (ref DEPID)
- (STAFFID, DEPID): Primary Key. justified: a staff can work in multiple departments; each pair is unique.
- Participation: total for Staff and department.

- Staff (LinkedTo) Clinical Activity:

- o STAFFID: Foreign Key (ref Staff (STAFFID))
- CAID: Foreign Key (ref Clinical Activity (CAID))
- (STAFFID, CAID): Primary Key. justified: one staff can participate in multiple clinical activities; each pair is unique.
- Participation: partial for both Staff and Clinical Activity

Implementation's output:

	Name
▶	Sara Belmaktoub

```sql
SELECT p.Name
FROM Patient p
JOIN Appointment a ON p.CAID = a.CAID
JOIN ClinicalActivity ca ON a.CAID = ca.CAID
JOIN Department d ON ca.CAID = d.CAID
JOIN Hospital h ON d.HID = h.HID
WHERE a.Status = 'Scheduled'
  AND h.City = 'Benguerir';
```

5 Discussion

During the development of the logical design, our group encountered several challenges related to the different perspectives and interpretations of the ER model.
Each member initially had a unique understanding of how certain entities and relationships should be represented, especially when deciding on the use of composite keys, participation constraints, and ISA hierarchies.
These differences often led to long discussions and multiple revisions of the schema, as we needed to ensure that every design choice was logically consistent and technically valid.
One of the main difficulties was reaching a consensus on the structure of relationships such as *Covers* and *Stock*, which required balancing normalization principles with practical implementation concerns.
It took a considerable amount of time and collaboration for the entire team to agree on a unified schema that satisfied both the conceptual accuracy and the relational integrity required by the lab.
Through this process, we learned the importance of clear communication, compromise, and collective problem-solving when working as a team on technical projects.
We also gained a deeper understanding of how relational modeling decisions directly affect database consistency, implementation efficiency, and real-world usability.
Although it was sometimes challenging to align our ideas, this experience helped us strengthen our teamwork skills, improve our analytical reasoning, and appreciate the value of collaboration in achieving a coherent and well-structured design.

6 Conclusion

In conclusion, this project allowed our group to apply the main concepts of database design in a practical way.
By going through all the steps from the conceptual model to the logical and physical design, we were able to better understand how data is structured and how relationships are implemented in SQL.
Even though we faced some challenges and spent a lot of time agreeing on certain design choices, we successfully created a consistent and well-organized schema that meets the lab requirements.
This work helped us improve our teamwork, problem-solving, and technical skills, and

gave us a clearer idea of how relational databases are designed and managed in real projects.