Level: 2nd Year Bachelor's in Computer Science (2025/2026)
MODULE: DATABASES
TD No. 2: RELATIONAL MODEL

<u>Exercise 1 :</u>

Algorithm:Transitive Closure

Remark: Focus on attributes that do not appear on the right-hand side of other FDs already included, to avoid redundant steps.

1. **Start** with a set of attributes X whose closure you want to compute.
2. **Initialize** the closure: $X^+ = X$.
3. **Repeat** until no new attributes can be added:
   - For each functional dependency $Y \rightarrow Z$, if $Y \subseteq X^+$, then add to X^+ **only the attributes in Z that are not already in X^+**.
4. **Stop** when X^+ no longer grows.
5. **Result:** X^+ contains all attributes functionally determined by X.

Use:

- To check if X is a **superkey**: X^+ must include all table attributes.

You are given the following table:

Employee (Emp_ID, Name, Dept_ID, Dept_Name, Manager_ID, Manager_Name) with the fellowing Functional Dependencies (FDs):

1. Emp_ID → Name, Dept_ID
2. Dept_ID → Dept_Name, Manager_ID
3. Manager_ID → Manager_Name

Questions

1. Using the algorithm given above (transitive closure) determine the primary key of the table.
2. Check if the table is in 1NF.
3. Check if the table is in 2NF and justify your answer.
4. Check if the table is in 3NF and justify your answer.
5. If necessary, decompose the table to achieve 3NF.

<u>Exercise 2 :</u>

Let the relation Hospital be defined by the following schema:

Hospitalisation (Patient_ID, Admission_Date, Room_Num, Patient_Name, Doctor_ID, Doctor_Name, Department_ID, Department_Name, Discharge_Date)

Functional Dependencies (FDs):

1. Patient_ID → Patient_Name
2. {Patient_ID, Admission_Date} → Room_Num, Doctor_ID, Department_ID, Discharge_Date
3. Doctor_ID → Doctor_Name, Department_ID
4. Department_ID → Department_Name

Tasks

1. Draw the transitive closure graph of the functional dependencies.
2. Determine the primary key of the relation.
3. Identify the highest normal form of the relation and justify.
4. Propose a decomposition of the relation into 3NF using the .

Exercice 3

Using SQL queries, implement the 3NF Hospital database

1. Create a table Patient with fields Patient_ID (number, primary key) and Patient_Name (text, length 50).
2. Create a table Department with fields Department_ID (number, primary key) and Department_Name (text, length 50).
3. Create a table Doctor with fields Doctor_ID, Doctor_Name, Department_ID and add a foreign key constraint linking Department_ID to Department.
4. Create a table Hospitalisation with fields Patient_ID, Admission_Date, Room_Num, Doctor_ID, Discharge_Date and primary key {Patient_ID, Admission_Date}.
5. Insert a department Cardiology with ID 10.
6. Insert a doctor Dr. Karim with Doctor_ID = 101 working in Department_ID = 10.
7. Insert a Patient with Id=5001 and Name = 'Samir'
8. Insert a hospitalization record for the patient with Patient_ID = 5001, admitted on 2026-02-01, assigned to room 201, treated by Dr.Karim, and discharged on 2026-02-10.
9. Add a column Patient_Birthday (Date) to the Patient table.
10. Remove the table Patient completely from the database.

Solution

Exercise 1:

Q1: We select only Emp_ID because it not appear in the right of DFs.

Compute closure of {Emp_ID}:

1. Start:

X^+ = {Emp_ID}

2. Apply FD1: Emp_ID → Name, Dept_ID → add {Name, Dept_ID}

X^+ = {Emp_ID, Name, Dept_ID}

3. Apply FD2: Dept_ID → Dept_Name, Manager_ID → add {Dept_Name, Manager_ID}

X^+ = {Emp_ID, Name, Dept_ID, Dept_Name, Manager_ID}

4. Apply FD3: Manager_ID → Manager_Name → add {Manager_Name}

X^+ = {Emp_ID, Name, Dept_ID, Dept_Name, Manager_ID, Manager_Name}

Closure includes **all attributes** → Emp_ID is the **primary key**.

Q2: Check 1NF

- 1NF = All attributes **atomic** (no repeating groups)
- All columns contain single values

Consequently, the table is **in 1NF**.

Q3: Check 2NF

- 2NF = 1NF + no partial dependency on part of composite key
- Key = `{Emp_ID}` → single attribute

No partial dependency possible → table **is in 2NF**.

Q4: Check 3NF

- 3NF = 2NF + **no transitive dependency on non-key attributes**
- Look at FDs:
    1. Dept_ID → Dept_Name, Manager_ID → **Dept_Name and Manager_ID depend on Dept_ID**, which is **not a key** → **transitive dependency**
    2. Manager_ID → Manager_Name → Manager_Name depends on Manager_ID → **transitive dependency**

Table is **not in 3NF**.

Q5: decomposition of the table:

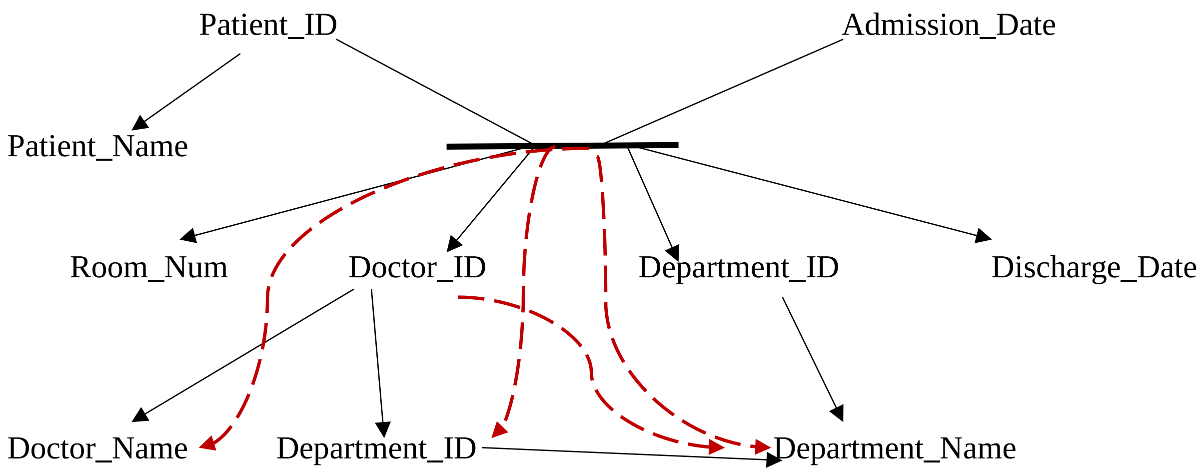
Employee (**Emp_ID**, Name, *Dept_ID)

Department (**Dept_ID**, Dept_Name, *Manager_ID)

Manager (**Manager_ID**, Manager_Name)

<u>Exercise 2</u>

Q1:



Q2: With Transitive closure graph, we can deduce that the primary key of the relation is:

{Patient_ID, Admission_Date}

Q3:

Hospitalisation (**<u>Patient_ID, Admission_Date</u>**, Room_Num, Patient_Name, Doctor_ID, Doctor_Name, Department_ID, Department_Name, Discharge_Date)

Q4:

- All attributes are atomic → **1NF satisfied**
- Patient_Name depends on **Patient_ID only → partial dependency exists!**
  Table is **not in 2NF** because of Patient_ID → Patient_Name
- Already not in 2NF → cannot be in 3NF. Also, transitive dependencies exist:
  - Doctor_ID → Doctor_Name, Department_ID
  - Department_ID → Department_Name

  Table is **not in 3NF**

Q5: Decomposition of the relation (3NF)

Patient(**<u>Patient_ID</u>**, Patient_Name)

Department(**<u>Department_ID</u>**, * Department_Name)

Doctor(**<u>Doctor_ID</u>**, Doctor_Name, Department_ID)

Hospitalisation(**<u>Patient_ID, Admission_Date</u>**, Room_Num, * Doctor_ID, Discharge_Date)

Exercise 3

```
CREATE TABLE Patient (
   Patient_ID INT PRIMARY KEY,
   Patient_Name TEXT(50)
);
--------------------------------------------

CREATE TABLE Department (
   Department_ID INT PRIMARY KEY,
   Department_Name TEXT(50)
);
--------------------------------------------

CREATE TABLE Doctor (
   Doctor_ID INT PRIMARY KEY,
   Doctor_Name TEXT(50),
   Department_ID INT,
   CONSTRAINT FK_Doctor_Department
      FOREIGN KEY (Department_ID)
      REFERENCES Department(Department_ID)
);
--------------------------------------------

CREATE TABLE Hospitalisation (
   Patient_ID INT,
   Admission_Date DATE,
   Room_Num INT,
   Doctor_ID INT,
   Discharge_Date DATE,
   CONSTRAINT PK_Hospitalisation
      PRIMARY KEY (Patient_ID, Admission_Date),
   CONSTRAINT FK_Hosp_Patient
      FOREIGN KEY (Patient_ID)
```

```
        REFERENCES Patient(Patient_ID),
    CONSTRAINT FK_Hosp_Doctor
        FOREIGN KEY (Doctor_ID)
        REFERENCES Doctor(Doctor_ID)
);
-----------------------------------------------

INSERT INTO Department (Department_ID, Department_Name)
VALUES (10, 'Cardiology');
INSERT INTO Doctor (Doctor_ID, Doctor_Name, Department_ID)
VALUES (101, 'Dr. Karim', 10);
-----------------------------------------------

INSERT INTO Doctor (Doctor_ID, Doctor_Name, Department_ID)
VALUES (101, 'Dr. Karim', 10);
-----------------------------------------------

INSERT INTO Patient (Patient_ID, Patient_Name)
VALUES (5001, 'Samir');
-----------------------------------------------

INSERT INTO Hospitalisation
(Patient_ID, Admission_Date, Room_Num, Doctor_ID, Discharge_Date)
VALUES (5001, #2026-02-01#, 201, 101, #2026-02-10#);
-----------------------------------------------

ALTER TABLE Patient
ADD COLUMN Patient_Birthday DATE;
-----------------------------------------------

DROP TABLE Patient;
```