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

Exercise 1 :

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.

Exercise 2 :

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.