PREVIOUSLY:

What You're Aiming For

The objective is to design the Entity-Relationship (ER) model for the employee participation information system, which manages the assignment of employees to various projects within the organization.

Instructions

To build this model, consider the following details:

- ***Each employee is characterized by their number (Num_E), name, position, salary, and the department they are assigned to. An employee can only work in one department.***
- ***A department is identified by a number (Num_S) and is characterized by its label, and the name of its manager. A department can have multiple employees.***
- ***A project is identified by a project number (Num_P) and is characterized by its title, start date, end date, and the department to which it is assigned. A project is specific to one department, and a department can have multiple projects.***

Additionally, the role of each employee in a project must be tracked.

Tasks

1. ***Identify Elements:***
    - ***Identify entities, attributes, and relationships based on the provided information.***
2. ***Specify Relationships:***
    - ***Clearly define all relationships' cardinalities (e.g., one-to-many, many-to-many).***

1. Entities and Attributes

i. Entity: Employee

- Primary Key: Num_E
- Attributes:
  - Name
  - Position
  - Salary
- Relationship: Assigned to one Department
  - Participates in many Projects

ii. Entity: Department

- Primary Key: Num_S
- Attributes:
  - Label
  - ManagerName
- Relationships:
  - Has many Employees
  - Has many Projects

iii. Entity: Project

- Primary Key: Num_P
- Attributes:
  - Title
  - StartDate
  - EndDate
- Relationships:
  - Assigned to one Department
  - Involves many Employees (each with a specific Role)

2. Relationship Cardinalities

Relationship	Entities Involved	Cardinality	Notes
Works_For	Employee → Department	Many-to-One (N:1)	Each employee works in one department
Has_Employees	Department → Employee	One-to-Many (1:N)	A department has many employees
Assigned_To	Project → Department	Many-to-One (N:1)	Each project is assigned to one department
Has_Projects	Department → Project	One-to-Many (1:N)	A department can manage multiple projects
Participation	Employee ↔ Project	Many-to-Many (M:N)	With attribute Role

USING THAT:

Construct the relational schema using the ER diagram created during the previous checkpoint.

For each entity and relationship, define:

- ***Table names***
- ***Columns (attributes) with appropriate primary keys and foreign keys***

1. For each Entity

Table Name: Employee

Column(Attributes):

Column Name	Data Type	Constraints
Num_E	INT	PRIMARY KEY
Name	VARCHAR	NOT NULL
Position	VARCHAR	NOT NULL
Salary	DECIMAL	NOT NULL

Num_S	INT	FOREIGN KEY → Department(Num_S)

Table Name: Department

Column Name	Data Type	Constraints
Num_S	INT	PRIMARY KEY
Label	VARCHAR	NOT NULL
ManagerName	VARCHAR	NOT NULL

Table Name: Project

Column Name	Data Type	Constraints
Num_P	INT	PRIMARY KEY
Title	VARCHAR	NOT NULL
StartDate	DATE	NOT NULL
EndDate	DATE	NOT NULL
Num_S	INT	FOREIGN KEY → Department(Num_S)

2. Table for Many-to-Many Relationship

Participation

Column Name	Data Type	Constraints
Num_E	INT	FOREIGN KEY → Employee(Num_E), part of PK
Num_P	INT	FOREIGN KEY → Project(Num_P), part of PK
Role	VARCHAR	NOT NULL

Primary Key: (Num_E, Num_P)

- Employee → Department: Many-to-One via Num_S FK in Employee
- Project → Department: Many-to-One via Num_S FK in Project
- Employee ↔ Project: Many-to-Many captured through Participation table with additional attribute Role