

Database Management System

Case Studies

Case Study 5

A relational database is to be designed for a medium sized Company dealing with industrial applications of computers. The Company delivers various products to its customers ranging from a single application program through to complete installation of hardware with customized software. The Company employs various experts, consultants and supporting staff. All personnel are employed on long-term basis, i.e. there is no short-term or temporary staff. Although the Company is somehow structured for administrative purposes (that is, it is divided into departments headed by department managers) all projects are carried out in an interdisciplinary way. For each project a project team is selected, grouping employees from different departments, and a Project Manager (also an employee of the Company) is appointed who is entirely and exclusively responsible for the control of the project, quite independently of the Company's hierarchy. The following is a brief statement of some facts and policies adopted by the Company.

- Each employee works in some department.
- An employee may possess a number of skills
- Every manager (including the MD) is an employee
- A department may participate in none/one/many projects.
- At least one department participates in a project.
- An employee may be engaged in none/one/many projects

- Project teams consist of at least one member.

For the above business stories, you are expected to create the following.

1. Analyze the data required.
2. Normalize the attributes.
3. Create the logical data model (ER diagrams).

Analyze the Data Required

From the given case study, the key entities and their relationships are identified as follows:

1. Employees

- Each employee works in one department.
- An employee may have multiple skills.
- Employees include managers and the managing director (MD).
- An employee may participate in multiple projects.

2. Departments

- A department may participate in multiple projects.
- Each department is headed by a manager who is also an employee.

3. Projects

- A project involves at least one department.
- A project team consists of at least one employee.
- Each project has a project manager who is an employee.

4. Skills

- Employees may possess multiple skills.

2. Normalize the Attributes

First Normal Form (1NF)

- Ensure each table has a primary key and each column contains atomic values.

Second Normal Form (2NF)

- Ensure all non-key attributes are fully functional dependent on the primary key.

Third Normal Form (3NF)

- Ensure there are no transitive dependencies.

Entities and Attributes

1. Employees

- **EmployeeID** (Primary Key)
- **EmployeeName**
- **DepartmentID** (Foreign Key)
- **Position** (e.g., Staff, Manager, MD)
- **ContactInfo**

2. Departments

- **DepartmentID** (Primary Key)
- **DepartmentName**
- **ManagerID** (Foreign Key)

3. Projects

- **ProjectID** (Primary Key)
- **ProjectName**
- **ProjectManagerID** (Foreign Key)
- **StartDate**
- **EndDate**

4. Skills

- **SkillID** (Primary Key)
- **SkillName**

5. EmployeeSkills

- **EmployeeID** (Primary Key, Foreign Key)
- **SkillID** (Primary Key, Foreign Key)

6. ProjectDepartments

- **ProjectID** (Primary Key, Foreign Key)
- **DepartmentID** (Primary Key, Foreign Key)

7. **ProjectTeams**

- **ProjectID** (Primary Key, Foreign Key)
- **EmployeeID** (Primary Key, Foreign Key)

3. Create the Logical Data Model (ER Diagrams)

Entity-Relationship Diagram (ERD)

1. **Employees:**

- **EmployeeID** (Primary Key)
- **EmployeeName**
- **DepartmentID** (Foreign Key)
- **Position**
- **ContactInfo**

2. **Departments:**

- **DepartmentID** (Primary Key)
- **DepartmentName**
- **ManagerID** (Foreign Key, refers to EmployeeID)

3. **Projects:**

- **ProjectID** (Primary Key)
- **ProjectName**
- **ProjectManagerID** (Foreign Key, refers to EmployeeID)
- **StartDate**
- **EndDate**

4. **Skills:**

- **SkillID** (Primary Key)
- **SkillName**

5. **EmployeeSkills:**

- **EmployeeID** (Primary Key, Foreign Key)
- **SkillID** (Primary Key, Foreign Key)

6. **ProjectDepartments:**

- **ProjectID** (Primary Key, Foreign Key)
- **DepartmentID** (Primary Key, Foreign Key)

7. **ProjectTeams:**

- **ProjectID** (Primary Key, Foreign Key)
- **EmployeeID** (Primary Key, Foreign Key)

ER Diagram

The ER diagram will include entities **Employees**, **Departments**, **Projects**, **Skills**, **EmployeeSkills**, **ProjectDepartments**, and **ProjectTeams** with their respective relationships and foreign keys.

ER Diagram Description:

- **Employees** works in a **Department** (1:N).
- An **Employee** can have multiple **Skills** through the **EmployeeSkills** table (M:N).
- A **Department** can participate in multiple **Projects** through the **ProjectDepartments** table (M:N).
- An **Employee** can be part of multiple **ProjectTeams** for different **Projects** (M:N).
- Each **Project** is managed by one **Employee** (Project Manager).
- Each **Department** is managed by one **Employee** (Manager).

SQL Database Schema

Here is a sample SQL schema based on the above normalization and ERD:

```
CREATE TABLE Departments (  
    DepartmentID INT PRIMARY KEY,  
    DepartmentName VARCHAR(100),  
    ManagerID INT,  
    FOREIGN KEY (ManagerID) REFERENCES Employees(EmployeeID)  
);  
  
CREATE TABLE Employees (  
    EmployeeID INT PRIMARY KEY,  
    EmployeeName VARCHAR(100),  
    DepartmentID INT,  
    Position VARCHAR(50),  
    ContactInfo VARCHAR(100),  
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)  
);  
  
CREATE TABLE Skills (  
    SkillID INT PRIMARY KEY,  
    SkillName VARCHAR(100)  
);  
  
CREATE TABLE EmployeeSkills (  
    EmployeeID INT,  
    SkillID INT,  
    PRIMARY KEY (EmployeeID, SkillID),  
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID),  
    FOREIGN KEY (SkillID) REFERENCES Skills(SkillID)  
);  
  
CREATE TABLE Projects (  
    ProjectID INT PRIMARY KEY,  
    ProjectName VARCHAR(100),
```

```
ProjectManagerID INT,  
StartDate DATE,  
EndDate DATE,  
FOREIGN KEY (ProjectManagerID) REFERENCES Employees(EmployeeID)  
);  
  
CREATE TABLE ProjectDepartments (  
    ProjectID INT,  
    DepartmentID INT,  
    PRIMARY KEY (ProjectID, DepartmentID),  
    FOREIGN KEY (ProjectID) REFERENCES Projects(ProjectID),  
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)  
);  
  
CREATE TABLE ProjectTeams (  
    ProjectID INT,  
    EmployeeID INT,  
    PRIMARY KEY (ProjectID, EmployeeID),  
    FOREIGN KEY (ProjectID) REFERENCES Projects(ProjectID),  
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)  
);
```

This schema ensures all necessary data is captured, relationships are established, and integrity constraints are enforced. The database will be able to handle the company's requirements for managing projects, departments, employees, and their skills effectively.