**Exercise 2: Employee Management System - Creating Entities**

**Business Scenario:**

Define JPA entities for Employee and Department with appropriate relationships.

**Instructions:**

1. **Creating JPA Entities:**
   * Define **Employee** entity with fields: **id, name, email, department**.
   * Define **Department** entity with fields: **id, name**.
2. **Mapping Entities to Database Tables:**
   * Use annotations like **@Entity, @Table, @Id, @GeneratedValue**, etc.
   * Define one-to-many relationship between **Department** and **Employee**.

**Department.java 🡪**

package com.code.employee.entity;

import java.util.List;

import jakarta.persistence.\*;

import lombok.Getter;

import lombok.Setter;

@Entity

@Table(name="departments")

@Getter

@Setter

public class Department {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    @Column(nullable=false)

    private String name;

    // creating a relationship with Employee One to Many

    // in one department there will be many employees

    @OneToMany(mappedBy = "department", cascade = CascadeType.ALL)

    private List<Employee> employees;

}

**Employee.java 🡪**

package com.code.employee.entity;

import jakarta.persistence.\*;

import lombok.Getter;

import lombok.Setter;

@Entity

@Table

@Getter

@Setter

public class Employee {

    @Id

    @GeneratedValue(strategy = GenerationType.AUTO)

    private int id;

    @Column(name="name", nullable=false)

    private String name;

    @Column(name="email", nullable=false, unique=true)

    private String email;

     //relationship with department one employee can work in one department

    //under one department there will be many employees

    @ManyToOne

    //creating a foreign key department id ref to the pk of department

    @JoinColumn(name = "department\_id", nullable = false)

    private Department department;

}