package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.Employee;

import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Find employees by department name

List<Employee> findByDepartmentName(String departmentName);

// Find employees with a name containing a specific string

List<Employee> findByNameContaining(String keyword);

// Find employees by email and name

Employee findByEmailAndName(String email, String name);

}

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.Employee;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Custom query to find employees with a specific email domain

@Query("SELECT e FROM Employee e WHERE e.email LIKE %:domain")

List<Employee> findByEmailDomain(@Param("domain") String domain);

// Custom query to find employees with names starting with a specific prefix

@Query("SELECT e FROM Employee e WHERE e.name LIKE :prefix%")

List<Employee> findByNameStartingWith(@Param("prefix") String prefix);

// Custom query to find employees belonging to a department by ID

@Query("SELECT e FROM Employee e WHERE e.department.id = :departmentId")

List<Employee> findByDepartmentId(@Param("departmentId") Long departmentId);

}

package com.example.employeemanagementsystem;

import jakarta.persistence.\*;

@Entity

@Table(name = "employees")

@NamedQueries({

@NamedQuery(

name = "Employee.findByDepartmentName",

query = "SELECT e FROM Employee e WHERE e.department.name = :departmentName"

),

@NamedQuery(

name = "Employee.findByNameContaining",

query = "SELECT e FROM Employee e WHERE e.name LIKE %:keyword%"

)

})

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@Column(nullable = false)

private String name;

@Column(nullable = false, unique = true)

private String email;

@ManyToOne(fetch = FetchType.LAZY)

@JoinColumn(name = "department\_id")

private Department department;

// Getters and Setters

}

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.Employee;

import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Use named query to find employees by department name

List<Employee> findByDepartmentName(String departmentName);

// Use named query to find employees with names containing a specific keyword

List<Employee> findByNameContaining(String keyword);

}