**Exercise 10: Employee Management System - Hibernate-Specific Features**

**Employee**

package com.employee.management.system.entities;

import lombok.Data;

import javax.persistence.\*;

import org.hibernate.annotations.BatchSize;

import org.hibernate.annotations.Fetch;

import org.hibernate.annotations.FetchMode;

@Data

@Entity

@Table(name = "employees")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String email;

@ManyToOne

@JoinColumn(name = "department\_id")

@Fetch(FetchMode.JOIN)

@BatchSize(size = 10)

private Department department;

}

**HibernateConfig**

package com.employee.management.system.config;

import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;

import org.springframework.stereotype.Component;

import javax.persistence.EntityManagerFactory;

@Component

public class HibernateConfig extends HibernateJpaVendorAdapter {

@Override

public void postProcessEntityManagerFactory(EntityManagerFactory entityManagerFactory) {

super.postProcessEntityManagerFactory(entityManagerFactory);

entityManagerFactory.getJpaPropertyMap().put("hibernate.dialect", "org.hibernate.dialect.H2Dialect");

entityManagerFactory.getJpaPropertyMap().put("hibernate.hbm2ddl.auto", "update");

entityManagerFactory.getJpaPropertyMap().put("hibernate.jdbc.batch\_size", 10);

}

}

**EmployeeRepository**

package com.employee.management.system.repositories;

import com.employee.management.system.entities.Employee;

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

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

@Query("SELECT e FROM Employee e")

List<Employee> findAllWithBatchProcessing();

}

In the above code, I have leveraged Hibernate-specific features to enhance the application's performance and capabilities.

I have used Hibernate-specific annotations like @BatchSize and @Fetch to customize entity mappings and optimize batch processing.

I have configured Hibernate dialect and properties for optimal performance using HibernateConfig.

I have implemented batch processing with Hibernate for bulk operations using EmployeeRepository.

Note that I have used @BatchSize to specify the batch size for batch processing, and @Fetch to customize the fetching strategy for associations.