**Primary Data Source Configuration (MySQL)**

package com.example.EmployeeManagementSystem.config;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.boot.autoconfigure.jdbc.DataSourceBuilder;

import org.springframework.boot.context.properties.ConfigurationProperties;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.Primary;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import org.springframework.orm.jpa.JpaTransactionManager;

import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;

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

import org.springframework.transaction.PlatformTransactionManager;

import org.springframework.transaction.annotation.EnableTransactionManagement;

import javax.persistence.EntityManagerFactory;

import javax.sql.DataSource;

@Configuration

@EnableTransactionManagement

@EnableJpaRepositories(

basePackages = "com.example.repository.primary",

entityManagerFactoryRef = "primaryEntityManagerFactory",

transactionManagerRef = "primaryTransactionManager"

)

public class PrimaryDataSourceConfig {

@Primary

@Bean(name = "primaryDataSource")

@ConfigurationProperties(prefix = "spring.datasource.primary")

public DataSource primaryDataSource() {

return DataSourceBuilder.create().build();

}

@Primary

@Bean(name = "primaryEntityManagerFactory")

public LocalContainerEntityManagerFactoryBean primaryEntityManagerFactory(

@Qualifier("primaryDataSource") DataSource dataSource) {

return new LocalContainerEntityManagerFactoryBean() {{

setDataSource(dataSource);

setPackagesToScan("com.example.entity.primary");

setJpaVendorAdapter(new HibernateJpaVendorAdapter());

setPersistenceUnitName("primary");

}};

}

@Primary

@Bean(name = "primaryTransactionManager")

public PlatformTransactionManager primaryTransactionManager(

@Qualifier("primaryEntityManagerFactory") EntityManagerFactory entityManagerFactory) {

return new JpaTransactionManager(entityManagerFactory);

}

}

**Secondary Data Source Configuration (PostgreSQL)**

package com.example.EmployeeManagementSystem.config;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.boot.autoconfigure.jdbc.DataSourceBuilder;

import org.springframework.boot.context.properties.ConfigurationProperties;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import org.springframework.orm.jpa.JpaTransactionManager;

import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;

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

import org.springframework.transaction.PlatformTransactionManager;

import org.springframework.transaction.annotation.EnableTransactionManagement;

import javax.persistence.EntityManagerFactory;

import javax.sql.DataSource;

@Configuration

@EnableTransactionManagement

@EnableJpaRepositories(

basePackages = "com.example.repository.secondary",

entityManagerFactoryRef = "secondaryEntityManagerFactory",

transactionManagerRef = "secondaryTransactionManager"

)

public class SecondaryDataSourceConfig {

@Bean(name = "secondaryDataSource")

@ConfigurationProperties(prefix = "spring.datasource.secondary")

public DataSource secondaryDataSource() {

return DataSourceBuilder.create().build();

}

@Bean(name = "secondaryEntityManagerFactory")

public LocalContainerEntityManagerFactoryBean secondaryEntityManagerFactory(

@Qualifier("secondaryDataSource") DataSource dataSource) {

return new LocalContainerEntityManagerFactoryBean() {{

setDataSource(dataSource);

setPackagesToScan("com.example.entity.secondary");

setJpaVendorAdapter(new HibernateJpaVendorAdapter());

setPersistenceUnitName("secondary");

}};

}

@Bean(name = "secondaryTransactionManager")

public PlatformTransactionManager secondaryTransactionManager(

@Qualifier("secondaryEntityManagerFactory") EntityManagerFactory entityManagerFactory) {

return new JpaTransactionManager(entityManagerFactory);

}

}

**EmployeeRepository :**

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

List<Employee> findByDepartment(Department department);

List<Employee> findByFirstName(String firstName);

List<Employee> findByLastName(String lastName);

}

**DepartmentRepository:**

public interface DepartmentRepository extends JpaRepository<Department, Long> {

List<Department> findByName(String name);

List<Department> findByLocation(String location);

}

**Employee.java**@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String firstName;

private String lastName;

private Department department;

// getters and setters

}

**Department.java**

@Entity

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String location;

// getters and setters

}