**EXERCISE 9**

*SPRING BOOT AUTO CONFIGURATION*# application.properties

# H2 Database configuration (default)

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=password

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

EXTERNALIZE CONFIGURATION

# application-dev.properties

spring.datasource.url=jdbc:h2:mem:devdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=password

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

# application-prod.properties

spring.datasource.url=jdbc:mysql://localhost:3306/employeedb

spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver

spring.datasource.username=root

spring.datasource.password=securepassword

spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect

spring.jpa.show-sql=false

spring.jpa.hibernate.ddl-auto=validate

*MANAGE MULTI DATA SOURCES*

# application.properties

# Primary Data Source (H2)

spring.datasource.primary.url=jdbc:h2:mem:primarydb

spring.datasource.primary.driverClassName=org.h2.Driver

spring.datasource.primary.username=sa

spring.datasource.primary.password=password

spring.jpa.primary.database-platform=org.hibernate.dialect.H2Dialect

# Secondary Data Source (MySQL)

spring.datasource.secondary.url=jdbc:mysql://localhost:3306/secondarydb

spring.datasource.secondary.driverClassName=com.mysql.cj.jdbc.Driver

spring.datasource.secondary.username=root

spring.datasource.secondary.password=securepassword

spring.jpa.secondary.database-platform=org.hibernate.dialect.MySQLDialect

*CONFIGURE MULTI SOURCES*

package com.example.EmployeeManagementSystem.config;

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

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

import org.springframework.boot.jdbc.DataSourceBuilder;

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 javax.sql.DataSource;

import java.util.HashMap;

@Configuration

public class DataSourceConfig {

@Primary

@Bean(name = "primaryDataSource")

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

public DataSource primaryDataSource() {

return DataSourceBuilder.create().build();

}

@Bean(name = "secondaryDataSource")

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

public DataSource secondaryDataSource() {

return DataSourceBuilder.create().build();

}

@Primary

@Bean(name = "primaryEntityManagerFactory")

public LocalContainerEntityManagerFactoryBean primaryEntityManagerFactory(

@Qualifier("primaryDataSource") DataSource dataSource) {

LocalContainerEntityManagerFactoryBean em = new LocalContainerEntityManagerFactoryBean();

em.setDataSource(dataSource);

em.setPackagesToScan(new String[]{"com.example.EmployeeManagementSystem.model.primary"});

em.setJpaVendorAdapter(new HibernateJpaVendorAdapter());

HashMap<String, Object> properties = new HashMap<>();

properties.put("hibernate.dialect", "org.hibernate.dialect.H2Dialect");

em.setJpaPropertyMap(properties);

return em;

}

@Bean(name = "secondaryEntityManagerFactory")

public LocalContainerEntityManagerFactoryBean secondaryEntityManagerFactory(

@Qualifier("secondaryDataSource") DataSource dataSource) {

LocalContainerEntityManagerFactoryBean em = new LocalContainerEntityManagerFactoryBean();

em.setDataSource(dataSource);

em.setPackagesToScan(new String[]{"com.example.EmployeeManagementSystem.model.secondary"});

em.setJpaVendorAdapter(new HibernateJpaVendorAdapter());

HashMap<String, Object> properties = new HashMap<>();

properties.put("hibernate.dialect", "org.hibernate.dialect.MySQLDialect");

em.setJpaPropertyMap(properties);

return em;

}

@Primary

@Bean(name = "primaryTransactionManager")

public JpaTransactionManager primaryTransactionManager(

@Qualifier("primaryEntityManagerFactory") LocalContainerEntityManagerFactoryBean primaryEntityManagerFactory) {

return new JpaTransactionManager(primaryEntityManagerFactory.getObject());

}

@Bean(name = "secondaryTransactionManager")

public JpaTransactionManager secondaryTransactionManager(

@Qualifier("secondaryEntityManagerFactory") LocalContainerEntityManagerFactoryBean secondaryEntityManagerFactory) {

return new JpaTransactionManager(secondaryEntityManagerFactory.getObject());

}

}

*ENABLING JPA REPOSITORIES*

package com.example.EmployeeManagementSystem.config;

import org.springframework.context.annotation.Configuration;

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

@Configuration

@EnableJpaRepositories(

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

entityManagerFactoryRef = "primaryEntityManagerFactory",

transactionManagerRef = "primaryTransactionManager"

)

public class PrimaryDataSourceConfig {

}

@Configuration

@EnableJpaRepositories(

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

entityManagerFactoryRef = "secondaryEntityManagerFactory",

transactionManagerRef = "secondaryTransactionManager"

)

public class SecondaryDataSourceConfig {

}