application.properties

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

spring.datasource.username=root

spring.datasource.password=rootpassword

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

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

spring.jpa.show-sql=true

application-dev.properties

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

spring.datasource.username=sa

spring.datasource.password=

spring.datasource.driver-class-name=org.h2.Driver

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

application-prod.properties

spring.datasource.url=jdbc:mysql://prod-db-server:3306/employeedb

spring.datasource.username=produser

spring.datasource.password=prodpassword

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

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

application.properties

spring.profiles.active=dev

--spring.profiles.active

java -jar your-app.jar --spring.profiles.active=prod

Employee Data Source Configuration

@Configuration

@EnableTransactionManagement

@EnableJpaRepositories(

basePackages = "com.example.employeemanagementsystem.repository.employee",

entityManagerFactoryRef = "employeeEntityManagerFactory",

transactionManagerRef = "employeeTransactionManager"

)

public class EmployeeDataSourceConfig {

@Primary

@Bean(name = "employeeDataSource")

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

public DataSource dataSource() {

return DataSourceBuilder.create().build();

}

@Primary

@Bean(name = "employeeEntityManagerFactory")

public LocalContainerEntityManagerFactoryBean entityManagerFactory(

EntityManagerFactoryBuilder builder,

@Qualifier("employeeDataSource") DataSource dataSource) {

return builder

.dataSource(dataSource)

.packages("com.example.employeemanagementsystem.entity.employee")

.persistenceUnit("employee")

.build();

}

@Primary

@Bean(name = "employeeTransactionManager")

public PlatformTransactionManager transactionManager(

@Qualifier("employeeEntityManagerFactory") EntityManagerFactory entityManagerFactory) {

return new JpaTransactionManager(entityManagerFactory);

}

}

Department Data Source Configuration

@Configuration

@EnableTransactionManagement

@EnableJpaRepositories(

basePackages = "com.example.employeemanagementsystem.repository.department",

entityManagerFactoryRef = "departmentEntityManagerFactory",

transactionManagerRef = "departmentTransactionManager"

)

public class DepartmentDataSourceConfig {

@Bean(name = "departmentDataSource")

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

public DataSource dataSource() {

return DataSourceBuilder.create().build();

}

@Bean(name = "departmentEntityManagerFactory")

public LocalContainerEntityManagerFactoryBean entityManagerFactory(

EntityManagerFactoryBuilder builder,

@Qualifier("departmentDataSource") DataSource dataSource) {

return builder

.dataSource(dataSource)

.packages("com.example.employeemanagementsystem.entity.department")

.persistenceUnit("department")

.build();

}

@Bean(name = "departmentTransactionManager")

public PlatformTransactionManager transactionManager(

@Qualifier("departmentEntityManagerFactory") EntityManagerFactory entityManagerFactory) {

return new JpaTransactionManager(entityManagerFactory);

}

}

application.propertiesspring.datasource.employee.url=jdbc:mysql://localhost:3306/employeedb

spring.datasource.employee.username=employeeuser

spring.datasource.employee.password=employeepassword

spring.datasource.employee.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

spring.datasource.department.url=jdbc:mysql://localhost:3306/departmentdb

spring.datasource.department.username=departmentuser

spring.datasource.department.password=departmentpassword

spring.datasource.department.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect