**EmployeeDataSourceConfig.java**

@Configuration

@EnableTransactionManagement

@EnableJpaRepositories(

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

entityManagerFactoryRef = "employeeEntityManagerFactory",

transactionManagerRef = "employeeTransactionManager"

)

public class EmployeeDataSourceConfig {

@Bean

@ConfigurationProperties("spring.datasource.employee")

public DataSourceProperties employeeDataSourceProperties() {

return new DataSourceProperties();

}

@Bean

public DataSource employeeDataSource() {

return employeeDataSourceProperties().initializeDataSourceBuilder().build();

}

@Bean

public LocalContainerEntityManagerFactoryBean employeeEntityManagerFactory(

EntityManagerFactoryBuilder builder) {

return builder

.dataSource(employeeDataSource())

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

.persistenceUnit("employee")

.build();

}

@Bean

public PlatformTransactionManager employeeTransactionManager(

@Qualifier("employeeEntityManagerFactory") EntityManagerFactory factory) {

return new JpaTransactionManager(factory);

}

}

**DepartmentDataSourceConfig.java:**

@Configuration

@EnableTransactionManagement

@EnableJpaRepositories(

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

entityManagerFactoryRef = "departmentEntityManagerFactory",

transactionManagerRef = "departmentTransactionManager"

)

public class DepartmentDataSourceConfig {

@Bean

@ConfigurationProperties("spring.datasource.department")

public DataSourceProperties departmentDataSourceProperties() {

return new DataSourceProperties();

}

@Bean

public DataSource departmentDataSource() {

return departmentDataSourceProperties().initializeDataSourceBuilder().build();

}

@Bean

public LocalContainerEntityManagerFactoryBean departmentEntityManagerFactory(

EntityManagerFactoryBuilder builder) {

return builder

.dataSource(departmentDataSource())

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

.persistenceUnit("department")

.build();

}

@Bean

public PlatformTransactionManager departmentTransactionManager(

@Qualifier("departmentEntityManagerFactory") EntityManagerFactory factory) {

return new JpaTransactionManager(factory);

}

}

**Employee.java**

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private double salary;

// Getters and Setters

}

**EmployeeRepository.java:**

public interface EmployeeRepository extends JpaRepository<Employee, Long> {}

**Department.java:**

@Entity

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

// Getters and Setters

}

**DepartmentRepository.java:**

public interface DepartmentRepository extends JpaRepository<Department, Long> {}

**EmployeeManagementApplication.java:**

@SpringBootApplication

public class EmployeeManagementApplication {

public static void main(String[] args) {

SpringApplication.run(EmployeeManagementApplication.class, args);

}

**}**

**application.properties:**

# EMPLOYEE DB

spring.datasource.employee.jdbc-url=jdbc:mysql://localhost:3306/employee\_db

spring.datasource.employee.username=root

spring.datasource.employee.password=root

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

spring.datasource.employee.hikari.pool-name=employeePool

# DEPARTMENT DB

spring.datasource.department.jdbc-url=jdbc:mysql://localhost:3306/department\_db

spring.datasource.department.username=root

spring.datasource.department.password=root

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

spring.datasource.department.hikari.pool-name=departmentPool