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

**Business Scenario:**

Leverage Hibernate-specific features to enhance your application's performance and capabilities.

**Instructions:**

1. **Hibernate-Specific Annotations:**
   * Use Hibernate-specific annotations to customize entity mappings.
2. **Configuring Hibernate Dialect and Properties:**
   * Configure Hibernate dialect and properties for optimal performance.
3. **Batch Processing:**
   * Implement batch processing with Hibernate for bulk operations.

**application.properties.java🡪**

spring.application.name=EmployeeProjectManagement

# Primary Data Source (Employee DB)

spring.datasource.primary.jdbc-url=jdbc:h2:mem:employee\_db;

spring.datasource.primary.username=sa

spring.datasource.primary.password=password

# Secondary Data Source (Department DB)

spring.datasource.secondary.jdbc-url=jdbc:h2:mem:department\_db

spring.datasource.secondary.username=sa

spring.datasource.secondary.password=password

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.transaction.jta.platform=org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform

spring.h2.console.enabled=true

spring.jpa.generate-ddl=true

spring.jpa.properties.hibernate.show\_sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.H2Dialect

#we add the following property to application.properties to see some statistics:

spring.jpa.properties.hibernate.generate\_statistics=true

#The first property tells Hibernate to collect inserts in batches of four.

#The order\_inserts property tells Hibernate to take the time to group inserts by entity, creating larger batches.

spring.jpa.properties.hibernate.jdbc.batch\_size=4

spring.jpa.properties.hibernate.order\_inserts=true

server.port=8185