**COGNIZANT DIGI NURTURE 4.0**

**WEEK 3 MANDATORY HANDS ON BY (6421405)**

**Spring Data Jpa:**

**Spring Data JPA - Quick Example**

* **Group:** com.cognizant
* **Artifact:** orm-learn
* **Description:** Demo project for Spring Data JPA and Hibernate

**Dependencies:**

* Spring Boot DevTools
* Spring Data JPA
* MySQL Driver

**MySql Workbench:**

create schema ormlearn;

// Create country table & data

use ormlearn;

create table country (

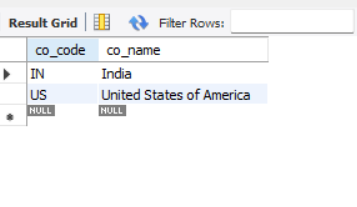
co\_code varchar(2) primary key,

co\_name varchar(50)

);

insert into country values ('IN', 'India');

insert into country values ('US', 'United States of America');



**Application.properties:**

* Place the code in application.properties

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn  
spring.datasource.username=root  
spring.datasource.password=uthay  
  
  
spring.jpa.hibernate.ddl-auto=validate  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect  
  
  
logging.level.org.springframework=info  
logging.level.com.cognizant=debug  
logging.level.org.hibernate.SQL=trace  
logging.level.org.hibernate.type.descriptor.sql=trace  
logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger{25} %25M %4L %m%n

**Java Class:**

src/main/java/com/cognizant/ormlearn

Package : model

Class : Country

**Country.java:**

package com.cognizant.ormlearn.model;  
  
import jakarta.persistence.\*;  
import jakarta.persistence.Column;  
import jakarta.persistence.Entity;  
@Entity  
@Table(name = "country")  
public class Country {  
 @Id  
 @Column(name = "co\_code")  
 private String code;  
 @Column(name = "co\_name")  
 private String name;  
 public String getCode() {  
 return code;  
 }  
 public void setCode(String code) {  
 this.code = code;  
 }  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

Package : Repository

Class : CountryRepository

**CountryRepository.java:**

package com.cognizant.ormlearn.repository;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
import com.cognizant.ormlearn.model.Country;  
  
@Repository  
public interface CountryRepository extends JpaRepository<Country, String> {  
}

Package : Service

Class : CountryService

**CountryService.java:**

package com.cognizant.ormlearn.service;  
  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
  
import com.cognizant.ormlearn.model.Country;  
import com.cognizant.ormlearn.repository.CountryRepository;  
  
@Service  
public class CountryService {  
 @Autowired  
 private CountryRepository countryRepository;  
  
 @Transactional  
 public List<Country> getAllCountries() {  
 return countryRepository.findAll();  
 }  
}

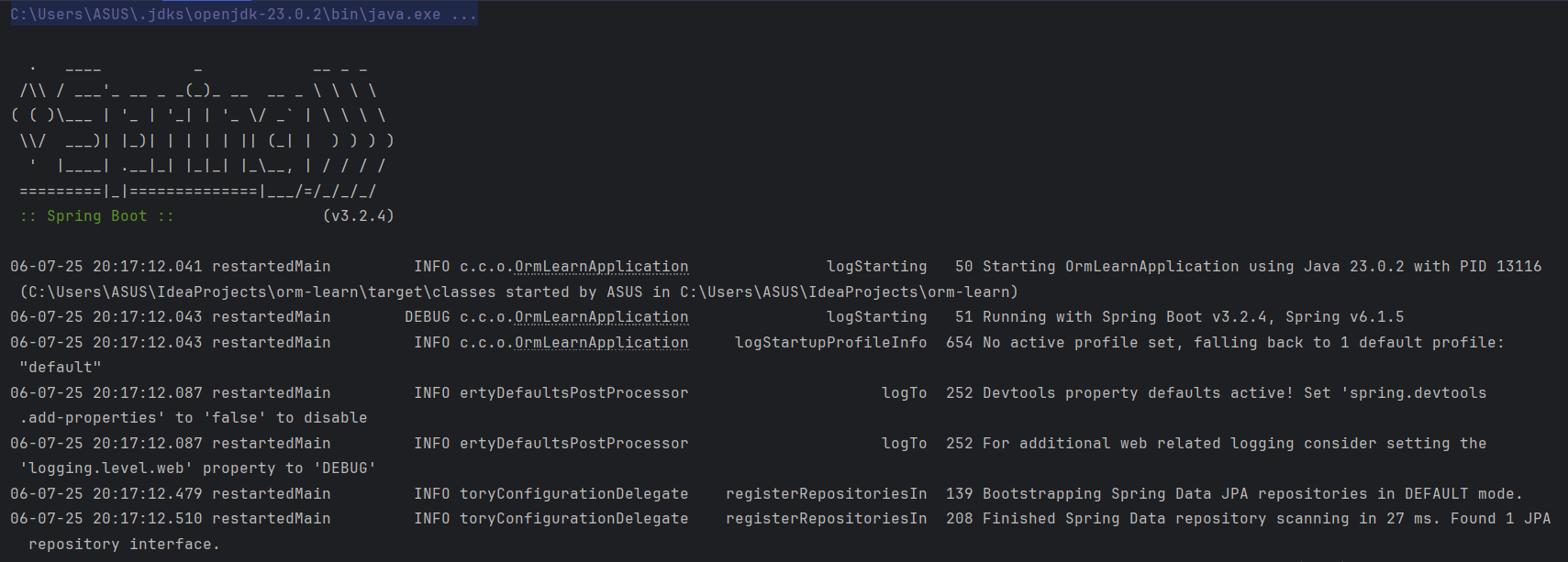
**OrmlearnApplication:**

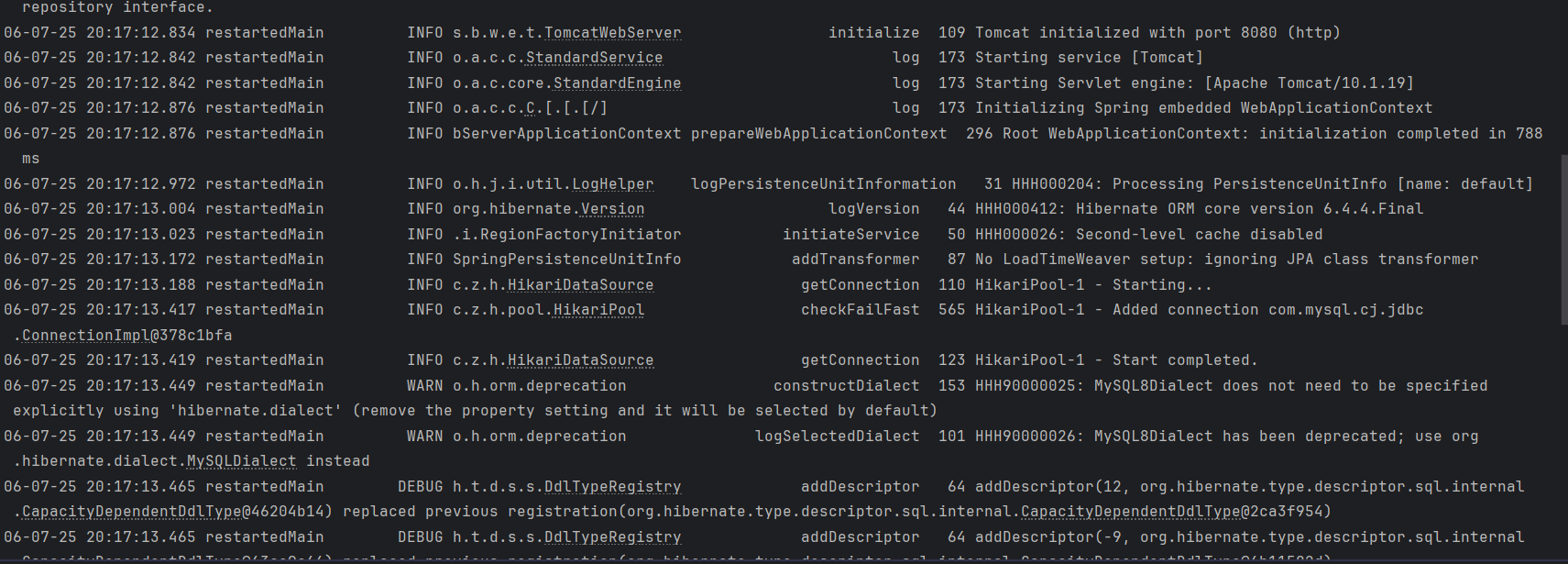
package com.cognizant.ormlearn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ApplicationContext;  
  
import com.cognizant.ormlearn.service.CountryService;  
import com.cognizant.ormlearn.model.Country;  
  
import java.util.List;  
  
@SpringBootApplication  
public class OrmLearnApplication {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);  
 private static CountryService countryService;  
  
 public static void main(String[] args) {  
 ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);  
 countryService = context.getBean(CountryService.class);  
 testGetAllCountries();  
 }  
  
 private static void testGetAllCountries() {  
 LOGGER.info("Start");  
 List<Country> countries = countryService.getAllCountries();  
 LOGGER.debug("countries={}", countries);  
 LOGGER.info("End");  
 }  
}

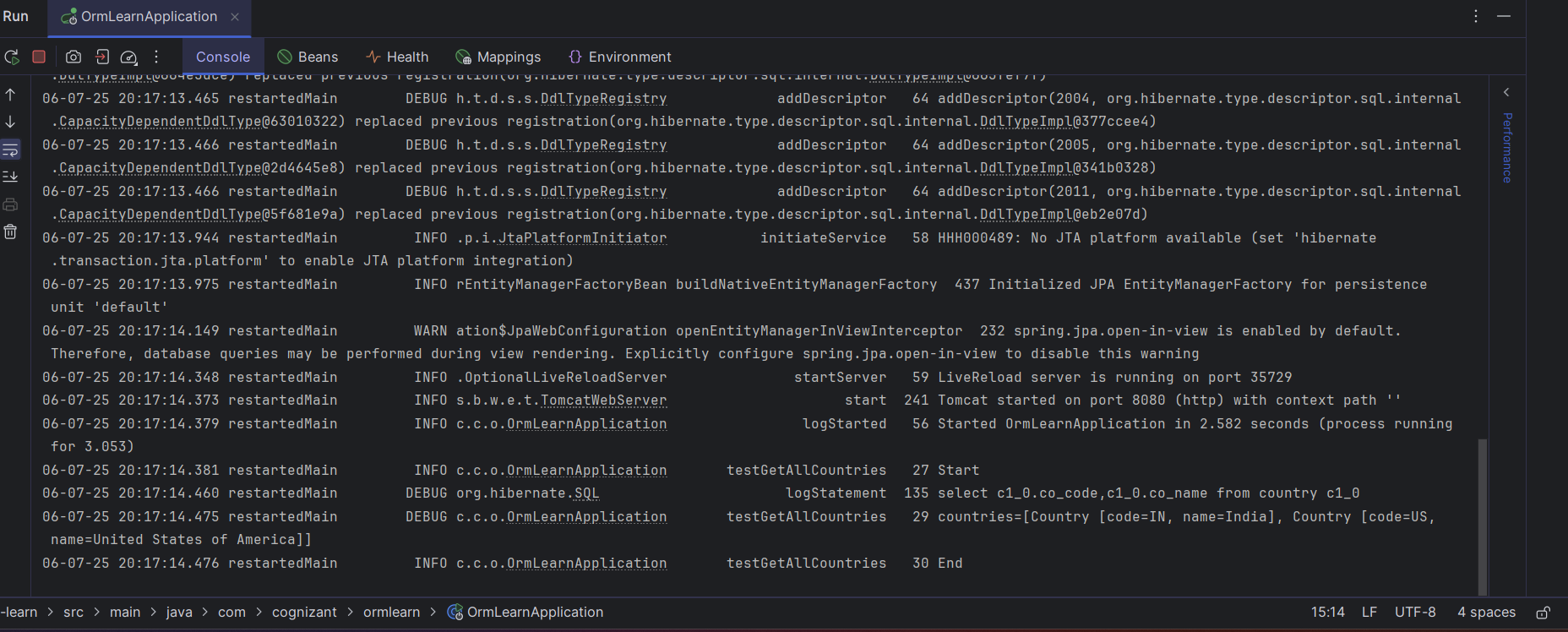
**Pom.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>orm-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>orm-learn</name>  
 <description>Demo project for Spring Data JPA and Hibernate</description>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.4</version>  
 <relativePath/>  
 </parent>  
  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
  
 <dependencies>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
  
  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <version>8.3.0</version>  
 <scope>runtime</scope>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <version>1.18.30</version>  
 <scope>provided</scope>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>jakarta.persistence</groupId>  
 <artifactId>jakarta.persistence-api</artifactId>  
 <version>3.1.0</version>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**OUTPUT:**







**Difference between JPA, Hibernate and Spring Data JPA** :

* **Group**: com.example
* **Artifact**: employee-demo

**Pom.xml:**

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.example</groupId>  
 <artifactId>spring-data-jpa-example</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <name>spring-data-jpa-example</name>  
 <description>Demo project for Spring Data JPA with MySQL</description>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.4</version>  
 <relativePath/>  
 </parent>  
  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
  
 <dependencies>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
  
  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springdoc</groupId>  
 <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>  
 <version>2.5.0</version>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <version>3.2.4</version>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**JAVA CLASS:**

Package : Model

Class : Employee

**Employee.java**

package com.example.employeedemo.model;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.GeneratedValue;  
import jakarta.persistence.GenerationType;  
import jakarta.persistence.Id;  
  
@Entity  
public class Employee {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private Integer id;  
 private String name;  
 private String department;  
  
 public Employee() {  
 }  
  
 public Employee(String name, String department) {  
 this.name = name;  
 this.department = department;  
 }  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getDepartment() {  
 return department;  
 }  
  
 public void setDepartment(String department) {  
 this.department = department;  
 }  
}

Package : controller

Class : EmployeeController

**EmployeeController.java**

package com.example.employeedemo.controller;  
  
import com.example.employeedemo.model.Employee;  
import com.example.employeedemo.service.EmployeeService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
@RequestMapping("/employees")  
public class EmployeeController {  
  
 @Autowired  
 private EmployeeService employeeService;  
  
 @PostMapping("/add")  
 public String createEmployee(@RequestBody Employee employee) {  
 employeeService.addEmployee(employee);  
 return "Employee created successfully!";  
  
 }  
  
}

Package : hibernate

Class : HibernateEmployeeService

**HibernateEmployeeService.java**

package com.example.employeedemo.hibernate;  
  
import com.example.employeedemo.model.Employee;  
import org.hibernate.HibernateException;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
public class HibernateEmployeeService {  
  
 public Integer addEmployee(Employee employee) {  
 Session session = HibernateUtil.getSessionFactory().openSession();  
 Transaction tx = null;  
 Integer employeeId = null;  
  
 try {  
 tx = session.beginTransaction();  
 employeeId = (Integer) session.save(employee);  
 tx.commit();  
 } catch (HibernateException e) {  
 if (tx != null) tx.rollback();  
 e.printStackTrace();  
 } finally {  
 session.close();  
 }  
 return employeeId;  
 }  
}

**HibernateUtil.java:**

package com.example.employeedemo.hibernate;  
  
import com.example.employeedemo.model.Employee;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
  
public class HibernateUtil {  
  
 private static final SessionFactory sessionFactory;  
  
 static {  
 try {  
 sessionFactory = new Configuration()  
 .configure("hibernate.cfg.xml")  
 .addAnnotatedClass(Employee.class)  
 .buildSessionFactory();  
 } catch (Throwable ex) {  
 throw new ExceptionInInitializerError(ex);  
 }  
 }  
  
 public static SessionFactory getSessionFactory() {  
 return sessionFactory;  
 }  
}

Package : repository

Class : EmployeeRepository

**EmployeeRepository.java**

package com.example.employeedemo.hibernate;  
  
import com.example.employeedemo.model.Employee;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
  
public class HibernateUtil {  
  
 private static final SessionFactory sessionFactory;  
  
 static {  
 try {  
 sessionFactory = new Configuration()  
 .configure("hibernate.cfg.xml")  
 .addAnnotatedClass(Employee.class)  
 .buildSessionFactory();  
 } catch (Throwable ex) {  
 throw new ExceptionInInitializerError(ex);  
 }  
 }  
  
 public static SessionFactory getSessionFactory() {  
 return sessionFactory;  
 }  
}

Package : service

Class : EmployeeDemoApplication

**EmployeeDemoApplication.java**

package com.example.employeedemo;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class EmployeeDemoApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(EmployeeDemoApplication.class, args);  
 }  
  
}

**Application.properties:**

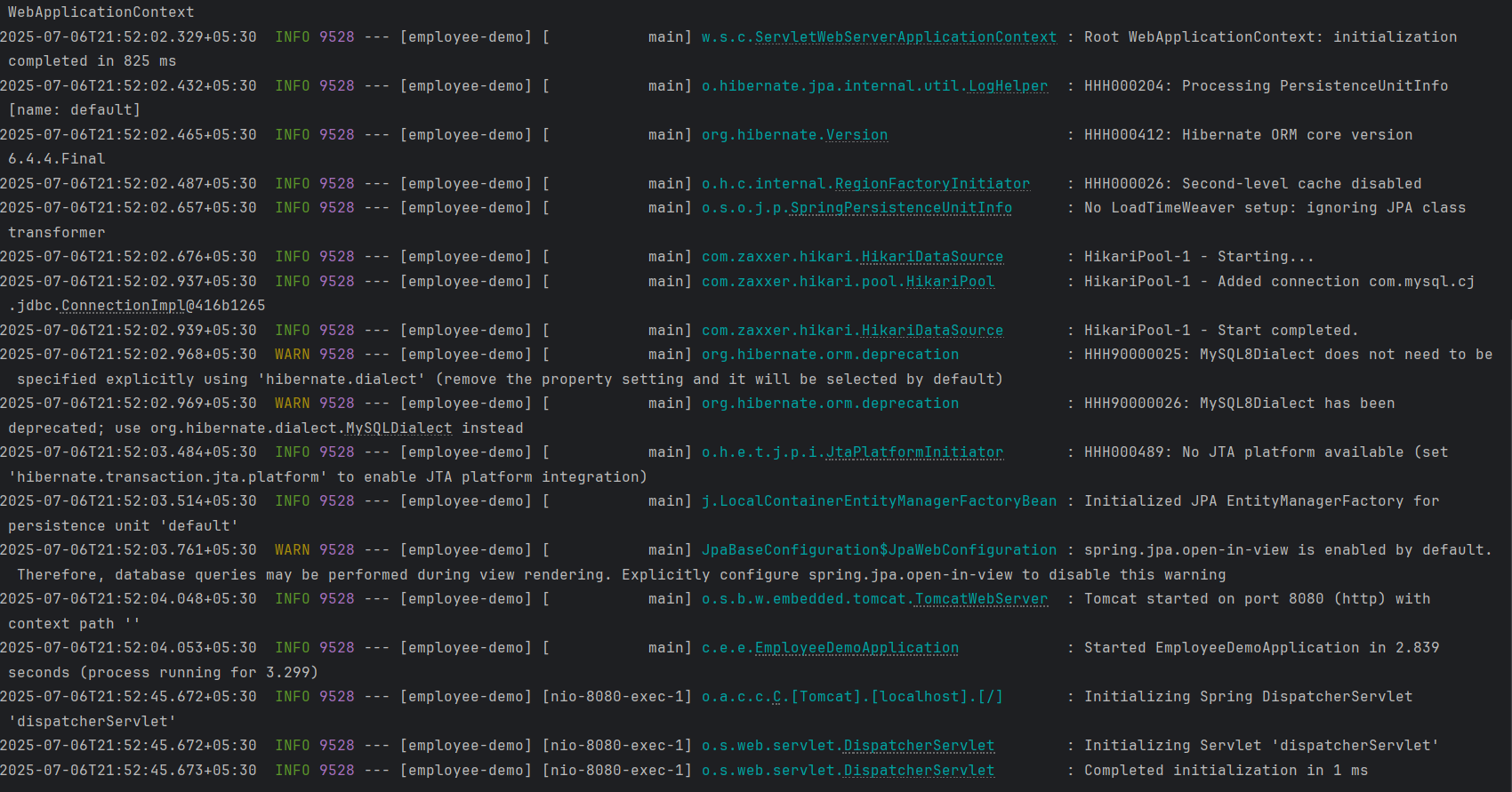
spring.application.name=employee-demo  
spring.datasource.url=jdbc:mysql://localhost:3306/employee\_db  
spring.datasource.username=root  
spring.datasource.password=uthay  
  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

**hibernate.cfg.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD//EN"  
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">  
<hibernate-configuration>  
 <session-factory>  
 <property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>  
 <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/employee\_db</property>  
 <property name="hibernate.connection.username">root</property>  
 <property name="hibernate.connection.password">your\_password</property>  
 <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>  
 <property name="hibernate.hbm2ddl.auto">update</property>  
 <property name="show\_sql">true</property>  
 </session-factory>  
</hibernate-configuration>

**OUTPUT:**

****

****