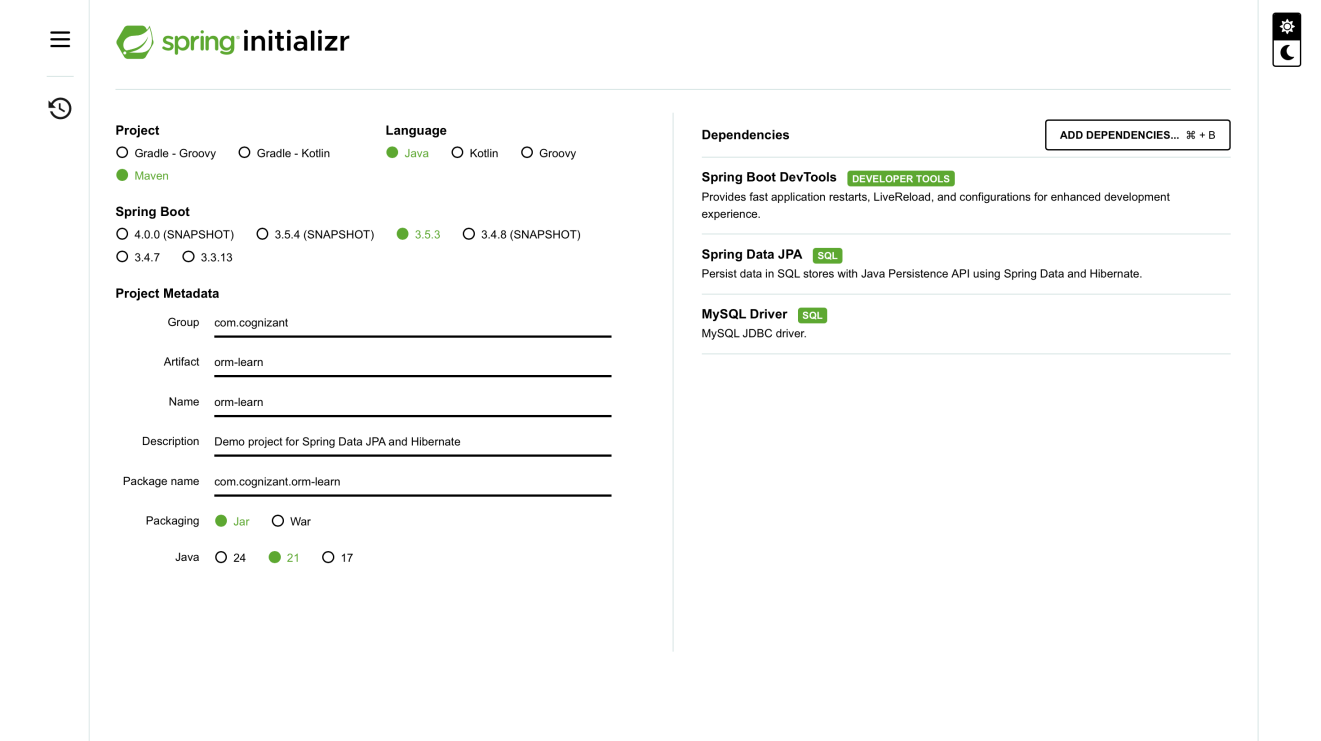
**HANDSON EXERCISES - WEEK 3**

**Skill : Spring Data JPA with Spring Boot, Hibernate**

**Exercise 1 : Spring Data JPA - Quick Example**

****

**CODE:**

**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>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.1.3</version>   
 <relativePath/>  
 </parent>  
  
 <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>  
  
 <properties>  
 <java.version>21</java.version>  
 </properties>

<dependencies>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.33</version>   
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>javax.transaction</groupId>  
 <artifactId>javax.transaction-api</artifactId>  
 <version>1.3</version>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**application.properties :**

# Logging  
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  
  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn\_fresh  
spring.datasource.username=root  
spring.datasource.password=root  
  
spring.jpa.hibernate.ddl-auto=validate  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

**com.cognizant.orm\_learn.model : Country.java :**

package com.cognizant.orm\_learn.model;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.Table;  
import jakarta.persistence.Id;  
import jakarta.persistence.Column;  
  
@Entity  
@Table(name = "country")  
public class Country {  
  
 @Id  
 @Column(name = "code")  
 private String code;  
  
 @Column(name = "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 + "]";  
 }  
}

**com.cognizant.orm\_learn.repository : CountryRepository.java :**

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

**com.cognizant.orm\_learn.service : CountryService.java :**

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

**com.cognizant.orm\_learn : OrmLearnApplication.java :**package com.cognizant.orm\_learn;  
  
import com.cognizant.orm\_learn.model.Country;  
import com.cognizant.orm\_learn.service.CountryService;  
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 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");  
 }  
}

**MYSQL : Country Table -**create table country (

code varchar(2) primary key,

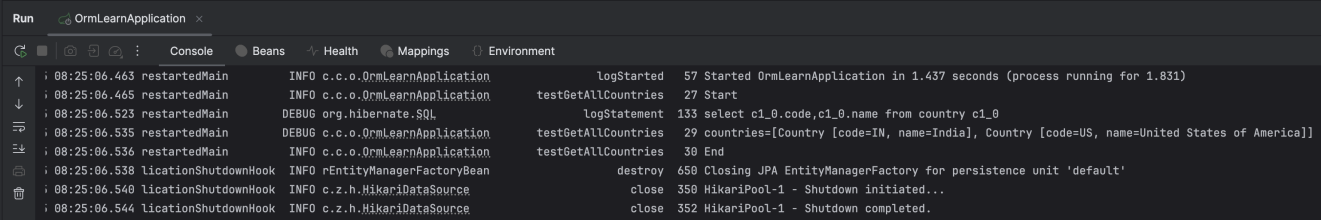
name varchar(50)

);

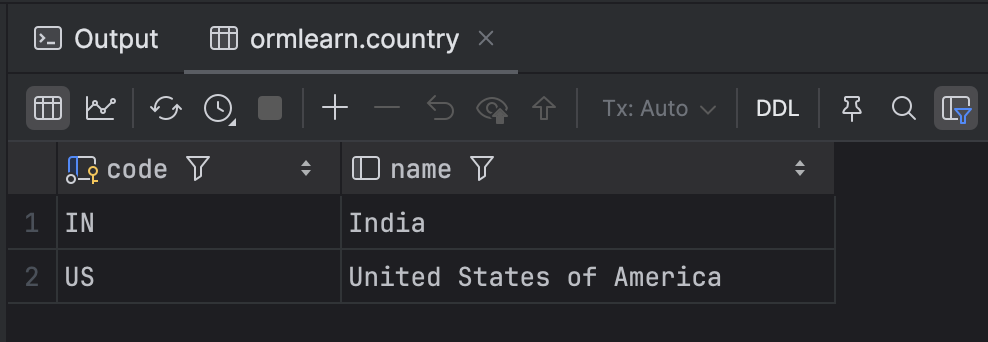
insert into country values ('IN', 'India');

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

**OUTPUT:**



SELECT \* FROM COUNTRY;



**Exercise 4 : Difference between JPA, Hibernate and Spring Data JPA**

1. **Hibernate Manual configuration - wihtout Spring Boot**

**com.hibernate : Employee.java :**

package com.hibernate;  
  
import javax.persistence.\*;  
  
@Entity  
@Table(name = "EMPLOYEE")  
public class Employee {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private int id;  
  
 private String firstName;  
 private String lastName;  
 private int salary;  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public int getSalary() {  
 return salary;  
 }  
  
 public void setSalary(int salary) {  
 this.salary = salary;  
 }  
}

**com.hibernate : ManageEmployee.java :**

package com.hibernate;  
  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
  
public class ManageEmployee {  
 public static void main(String[] args) {  
 SessionFactory factory = new Configuration()  
 .configure("hibernate.cfg.xml")  
 .addAnnotatedClass(Employee.class)  
 .buildSessionFactory();  
  
 Session session = factory.openSession();  
  
 try {  
 session.beginTransaction();  
 Employee emp = new Employee();  
 emp.setFirstName("Rafi");  
 emp.setLastName("SBM");  
 emp.setSalary(90000);  
 session.save(emp);  
 session.getTransaction().commit();  
 } finally {  
 session.close();  
 factory.close();  
 }  
 }  
}

**resources/hibernate.cfg.xml :**

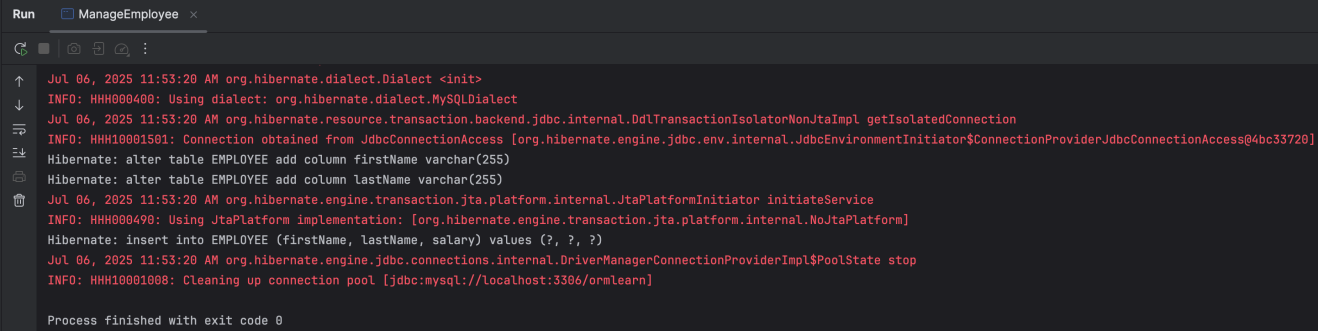
<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD//EN"  
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<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/ormlearn</property>  
 <property name="hibernate.connection.username">root</property>  
 <property name="hibernate.connection.password">root</property>  
 <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

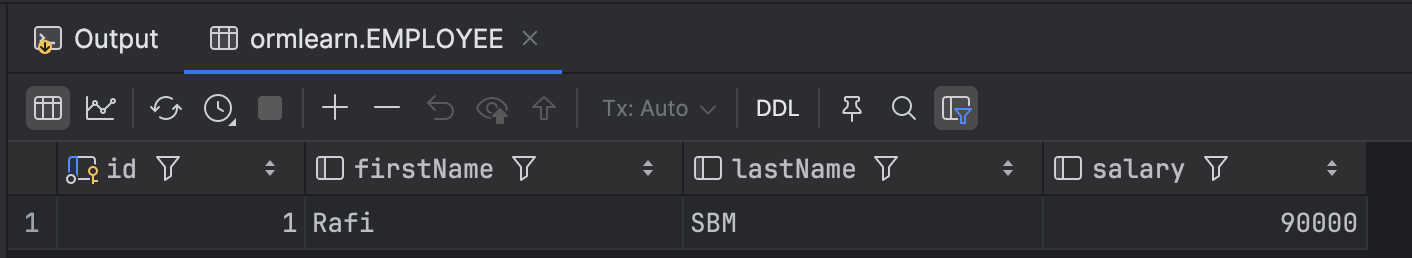
<property name="hibernate.hbm2ddl.auto">update</property>  
 <property name="show\_sql">true</property>  
 <mapping class="com.hibernate.Employee"/>

</session-factory>  
</hibernate-configuration>

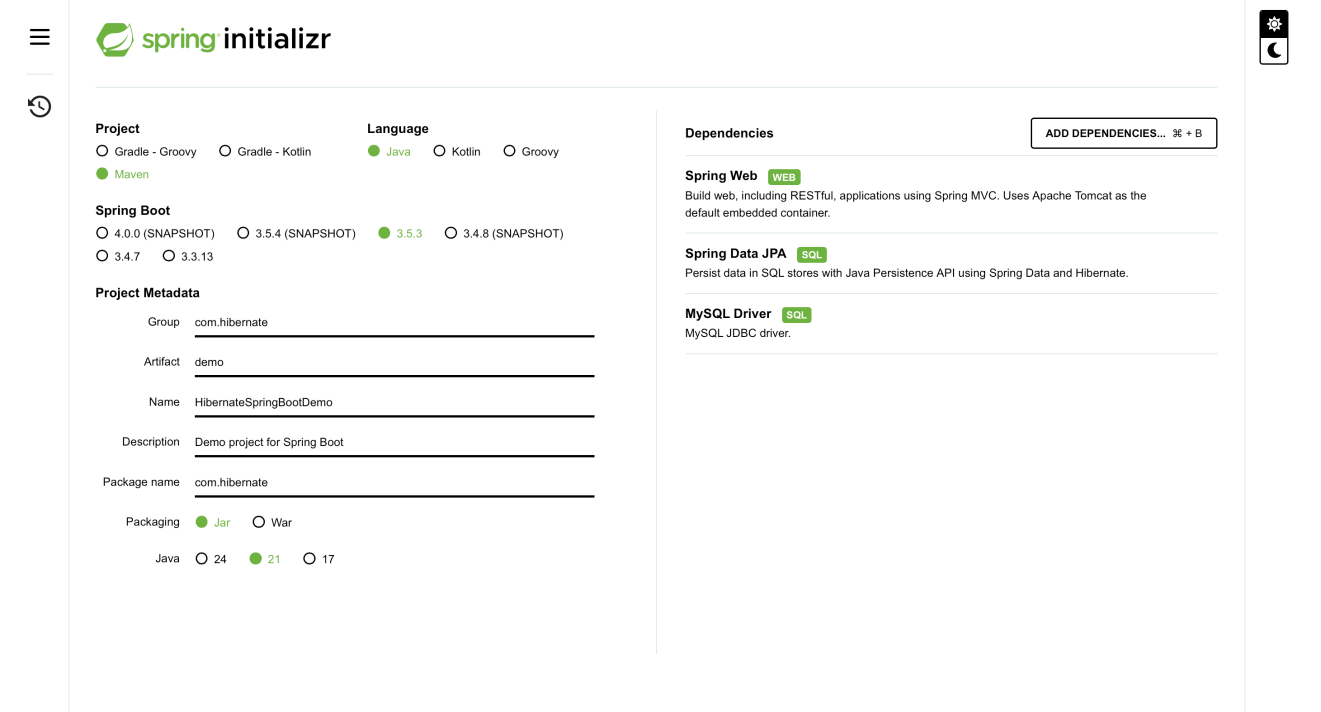
**OUTPUT :**



SELECT \* FROM EMPLOYEE;



1. **Spirng Boot + Sptring Data JPA**



**com.hibernate: Employee.java :**

package com.hibernate;  
  
import jakarta.persistence.\*;  
  
@Entity  
@Table(name = "employee")  
public class Employee {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private int id;  
  
 private String firstName;  
 private String lastName;  
 private int salary;  
  
 public int getId() {  
 return id;  
 }  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public int getSalary() {  
 return salary;  
 }  
 public void setSalary(int salary) {  
 this.salary = salary;  
 }  
}

**com.hibernate: EmployeeRepository.java :**

package com.hibernate;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {  
}

**com.hibernate: EmployeeService.java :**

package com.hibernate;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import jakarta.transaction.Transactional;  
  
@Service  
public class EmployeeService {  
  
 @Autowired  
 private EmployeeRepository employeeRepository;  
  
 @Transactional  
 public void addEmployee(Employee employee) {  
 employeeRepository.save(employee);  
 }  
}

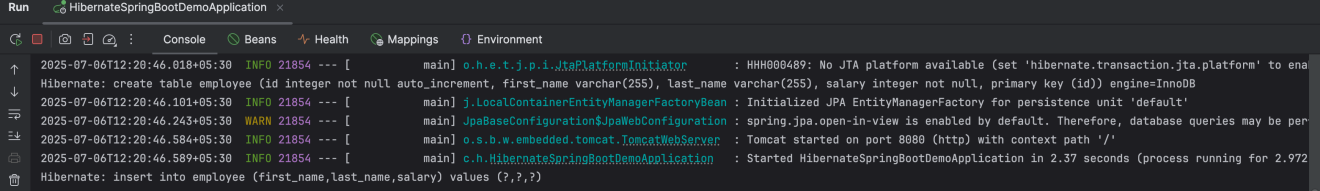
**com.hibernate: HibernateSpringBootDemoApplication.java :**

package com.hibernate;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.CommandLineRunner;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class HibernateSpringBootDemoApplication implements CommandLineRunner {  
  
 @Autowired  
 private EmployeeService employeeService;  
  
 public static void main(String[] args) {  
 SpringApplication.run(HibernateSpringBootDemoApplication.class, args);  
 }  
  
 @Override  
 public void run(String... args) throws Exception {  
 Employee emp = new Employee();  
 emp.setFirstName("Rafi");  
 emp.setLastName("SBM");  
 emp.setSalary(60000);  
 employeeService.addEmployee(emp);  
 }  
}

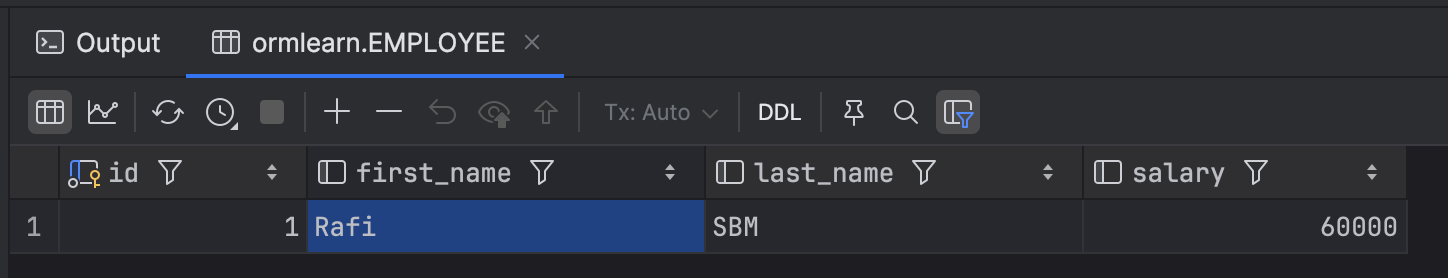
**application.properties :**

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn  
spring.datasource.username=root  
spring.datasource.password=root  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

**OUTPUT :**



SELECT \* FROM employee;



**Exercise 2 : Hibernate XML Config implementation walk through**

**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.library</groupId>  
 <artifactId>HibernateXMLDemo</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>21</maven.compiler.source>  
 <maven.compiler.target>21</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>

<dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-core</artifactId>  
 <version>5.6.15.Final</version>  
 </dependency>  
  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <version>8.2.0</version>  
 </dependency>  
  
 <dependency>  
 <groupId>jakarta.persistence</groupId>  
 <artifactId>jakarta.persistence-api</artifactId>  
 <version>2.2.3</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-simple</artifactId>  
 <version>2.0.7</version>  
 </dependency>  
 </dependencies>  
</project>

**com.hibernate.xmlconfig : Employee.java :**

package com.hibernate.xmlconfig;  
  
public class Employee {  
 private int id;  
 private String firstName;  
 private String lastName;  
 private int salary;  
  
 public int getId() {

return id;

}

public void setId(int id) {   
 this.id = id;   
 }  
  
 public String getFirstName() {   
 return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}  
  
 public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {   
 this.lastName = lastName;

}  
  
 public int getSalary() {

return salary;

}

public void setSalary(int salary) {

this.salary = salary;

}  
}

**com.hibernate.xmlconfig : ManageEmployee.java :**

package com.hibernate.xmlconfig;  
  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.hibernate.cfg.Configuration;  
  
import java.util.List;  
  
public class ManageEmployee {  
 public static void main(String[] args) {

SessionFactory factory = new Configuration().configure().buildSessionFactory();  
 Session session = factory.openSession();  
  
 Transaction tx = null;  
  
 try {  
 tx = session.beginTransaction();  
  
 Employee emp = new Employee();  
 emp.setFirstName("Rafi");  
 emp.setLastName("SBM");  
 emp.setSalary(75000);  
 session.save(emp);  
  
 List<Employee> list = session.createQuery("FROM Employee", Employee.class).list();

for (Employee e : list) {  
 System.out.println(e.getId() + ": " + e.getFirstName() + " " + e.getLastName() + " - " + e.getSalary());  
 }  
  
 tx.commit();  
 } catch (Exception e) {  
 if (tx != null) tx.rollback();  
 e.printStackTrace();  
 } finally {  
 session.close();  
 factory.close();  
 }  
 }  
}

**resources/hibernate.cfg.xml :**

<?xml version="1.0" encoding="UTF-8"?>

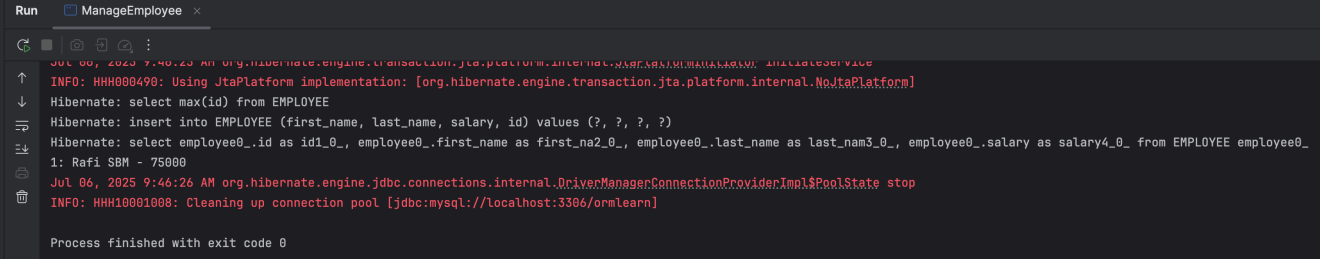
<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">  
  
<hibernate-configuration>  
 <session-factory>  
 <property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>  
 <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/ormlearn</property>  
 <property name="hibernate.connection.username">root</property>  
 <property name="hibernate.connection.password">root</property>

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>  
 <property name="hibernate.show\_sql">true</property>  
 <property name="hibernate.hbm2ddl.auto">update</property>  
  
 <mapping resource="employee.hbm.xml"/>  
 </session-factory>  
</hibernate-configuration>

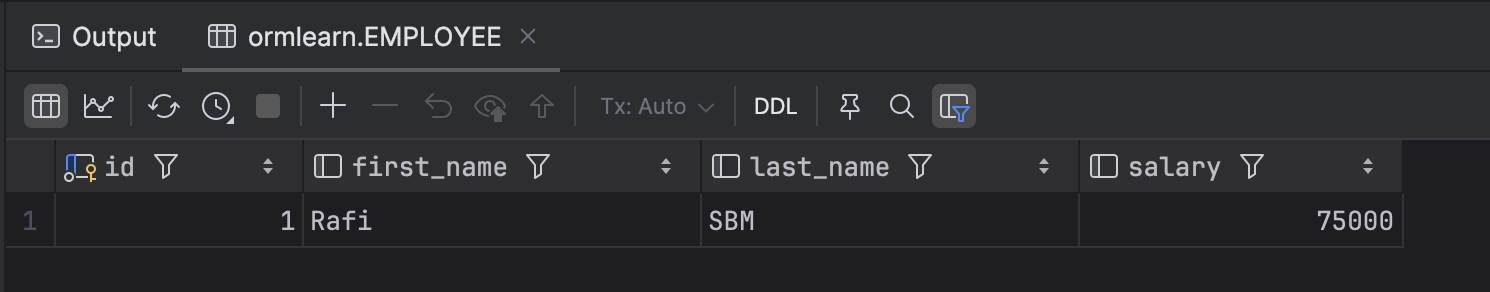
**resources/employee.hbm.xml :**

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE hibernate-mapping PUBLIC  
 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">  
  
<hibernate-mapping>  
 <class name="com.hibernate.xmlconfig.Employee" table="EMPLOYEE">  
 <id name="id" type="int" column="id">  
 <generator class="increment"/>  
 </id>  
 <property name="firstName" column="first\_name" type="string"/>  
 <property name="lastName" column="last\_name" type="string"/>  
 <property name="salary" column="salary" type="int"/>  
 </class>  
</hibernate-mapping>

**OUTPUT :**



SELECT \* FROM EMPLOYEE;



**Exercise 3 : Hibernate Annotation Config implementation walk through**

**com.hibernate.annotation : Employee.java :**

package com.hibernate.annotation;  
  
import javax.persistence.\*;  
  
@Entity  
@Table(name = "EMPLOYEE")  
public class Employee {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private int id;  
  
 @Column(name = "first\_name")  
 private String firstName;  
  
 @Column(name = "last\_name")  
 private String lastName;  
  
 @Column(name = "salary")  
 private int salary;  
  
 public int getId() {

return id;

}

public void setId(int id) {   
 this.id = id;   
 }  
  
 public String getFirstName() {   
 return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}  
  
 public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {   
 this.lastName = lastName;

}

public int getSalary() {

return salary;

}

public void setSalary(int salary) {

this.salary = salary;

}  
}

**com.hibernate.annotation : ManageEmployee.java :**

package com.hibernate.annotation;  
  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
  
public class ManageEmployee {  
 public static void main(String[] args) {  
 SessionFactory factory = new Configuration()  
 .configure("hibernate.cfg.xml")  
 .addAnnotatedClass(Employee.class)  
 .buildSessionFactory();  
  
 Session session = factory.openSession();  
  
 try {  
 session.beginTransaction();  
 Employee emp = new Employee();  
 emp.setFirstName("Rafi");  
 emp.setLastName("SBM");  
 emp.setSalary(80000);  
 session.save(emp);  
 session.getTransaction().commit();  
 System.out.println("Employee saved: " + emp);  
 }

finally {  
 session.close();  
 factory.close();  
 }  
 }  
}

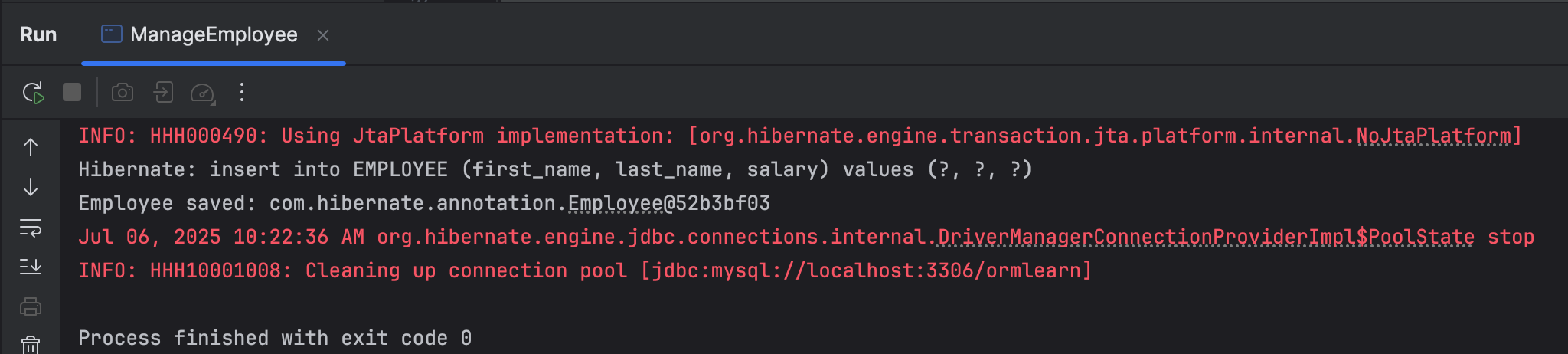
**resources/hibernate.cfg.xml :**

<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD//EN"  
 "http://www.hibernate.org/dtd/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/ormlearn</property>  
 <property name="hibernate.connection.username">root</property>  
 <property name="hibernate.connection.password">root</property>  
  
 <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>  
 <property name="hibernate.hbm2ddl.auto">update</property>  
 <property name="show\_sql">true</property>  
  
 <mapping class="com.hibernate.annotation.Employee"/>  
 </session-factory>  
</hibernate-configuration>

**OUTPUT :**



SELECT \* FROM EMPLOYEE;

