**Superset Id:** **6376594**

**Spring Data JPA with Hibernate(Hands-On)**

Exercise-1:Spring Data JPA - Quick Example

Project Name:orm-learn

OrmLearnApplication.java Code:

**package** com.cognizant.ormlearn;

**import** java.util.List;

**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.model.Country;

**import** com.cognizant.ormlearn.service.CountryService;

@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);

***LOGGER***.info("Inside main");

*countryService* = context.getBean(CountryService.**class**); // Get the service bean

*testGetAllCountries*(); // Call the test method

}

**private** **static** **void** testGetAllCountries() {

***LOGGER***.info("Start");

List<Country> countries = *countryService*.getAllCountries(); // Call the service

System.***out***.println("OUTPUT: " + countries);

// Print list

***LOGGER***.info("End");

}

}

Country.java Code:

**package** com.cognizant.ormlearn.model;

**import** jakarta.persistence.Column;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.Id;

**import** jakarta.persistence.Table;

@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 + "]";

}

}

CountryRepository.java Code:

**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> {

}

CountryService.java Code:

**package** com.cognizant.ormlearn.service;

**import** java.util.List;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.cognizant.ormlearn.model.Country;

**import** com.cognizant.ormlearn.repository.CountryRepository;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

**public** List<Country> getAllCountries() {

**return** countryRepository.findAll();

}

}

application.properties Code:

# Spring logs

logging.level.org.springframework=info

logging.level.com.cognizant=debug

# Hibernate SQL output logs

logging.level.org.hibernate.SQL=trace

logging.level.org.hibernate.type.descriptor.sql=trace

# Console log format

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger**{25}** %25M %4L %m%n

# DB connection config

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=@Charu23

# Hibernate

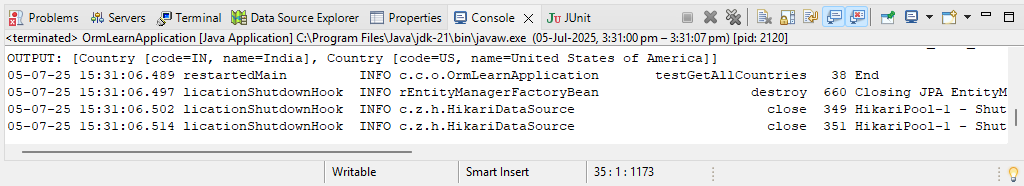
spring.jpa.hibernate.ddl-auto=validate

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

Expected Output:

OUTPUT: [Country [code=IN, name=India], Country [code=US, name=United States of America]]

Output:



**Exercise-2:Difference between JPA, Hibernate and Spring Data JPA**

Project Name:orm-learn

OrmLearnApplication.java Code:

**package** com.cognizant.ormlearn;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** com.cognizant.ormlearn.model.Employee;

**import** com.cognizant.ormlearn.service.EmployeeService;

@SpringBootApplication

**public** **class** OrmLearnApplication **implements** CommandLineRunner {

@Autowired

**private** EmployeeService employeeService;

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(OrmLearnApplication.**class**, args);

}

@Override

**public** **void** run(String... args) **throws** Exception {

System.***out***.println("Inside main");

Employee emp = **new** Employee();

emp.setId(1);

emp.setName("John Doe");

emp.setSalary(50000.00);

employeeService.addEmployee(emp);

System.***out***.println("Employee added successfully!");

}

}

Employee.java Code:

**package** com.cognizant.ormlearn.model;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.Id;

**import** jakarta.persistence.Table;

@Entity

@Table(name = "employee")

**public** **class** Employee {

@Id

**private** **int** id;

**private** String name;

**private** **double** salary;

// Getters and Setters

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getSalary() {

**return** salary;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

}

@Override

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";

}

}

EmployeeRepository.java Code:

**package** com.cognizant.ormlearn.repository;

**import** org.springframework.data.jpa.repository.JpaRepository;

**import** com.cognizant.ormlearn.model.Employee;

**public** **interface** EmployeeRepository **extends** JpaRepository<Employee, Integer> {

// No code needed here, JpaRepository gives CRUD operations

}

EmployeeService.java Code:

**package** com.cognizant.ormlearn.service;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.cognizant.ormlearn.model.Employee;

**import** com.cognizant.ormlearn.repository.EmployeeRepository;

@Service

**public** **class** EmployeeService {

@Autowired

**private** EmployeeRepository employeeRepository;

@Transactional

**public** **void** addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

application.properties Code:

# Spring logs

logging.level.org.springframework=info

logging.level.com.cognizant=debug

# Hibernate SQL output logs

logging.level.org.hibernate.SQL=trace

logging.level.org.hibernate.type.descriptor.sql=trace

# Console log format

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger**{25}** %25M %4L %m%n

# DB connection config

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=@Charu23

# Hibernate

spring.jpa.hibernate.ddl-auto=update

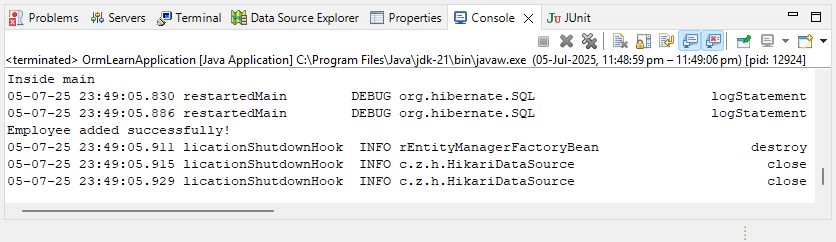
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

Expected Output:

Inside main

Employee added successfully!

Output:



Additional Exercises

**Exercise-5:Implement services for managing Country**

Project Name:orm-learn

OrmLearnApplication.java Code:

**package** com.cognizant.ormlearn;

**import** com.cognizant.ormlearn.model.Country;

**import** com.cognizant.ormlearn.service.CountryService;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** java.util.List;

@SpringBootApplication

**public** **class** OrmLearnApplication **implements** CommandLineRunner {

@Autowired

**private** CountryService countryService;

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(OrmLearnApplication.**class**, args);

}

@Override

**public** **void** run(String... args) **throws** Exception {

System.***out***.println("Inside main");

// Find by code

System.***out***.println("Country: " + countryService.getCountry("IN"));

// Add new country

Country c = **new** Country();

c.setCode("ZZ");

c.setName("Zootopia");

countryService.addCountry(c);

// Update country

c.setName("Zootopia Updated");

countryService.updateCountry(c);

// Search by name

List<Country> matches = countryService.searchCountries("land");

matches.forEach(System.***out***::println);

// Delete country

countryService.deleteCountry("ZZ");

System.***out***.println("Operations completed.");

}

}

Country.java Code:

**package** com.cognizant.ormlearn.model;

**import** jakarta.persistence.\*;

@Entity

@Table(name = "country")

**public** **class** Country {

@Id

@Column(name = "co\_code")

**private** String code;

@Column(name = "co\_name")

**private** String name;

// Getters and Setters

**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;

}

// Optional: toString() for logging/debug

@Override

**public** String toString() {

**return** "Country [code=" + code + ", name=" + name + "]";

}

}

CountryRepository.java Code:

**package** com.cognizant.ormlearn.repository;

**import** java.util.List;

**import** org.springframework.data.jpa.repository.JpaRepository;

**import** com.cognizant.ormlearn.model.Country;

**public** **interface** CountryRepository **extends** JpaRepository<Country, String> {

List<Country> findByNameContaining(String keyword);

}

CountryService.java Code:

**package** com.cognizant.ormlearn.service;

**import** java.util.List;

**import** java.util.Optional;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.cognizant.ormlearn.model.Country;

**import** com.cognizant.ormlearn.repository.CountryRepository;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

**public** Country getCountry(String code) **throws** Exception {

Optional<Country> result = countryRepository.findById(code);

**if** (result.isPresent()) {

**return** result.get();

} **else** {

**throw** **new** Exception("Country not found with code: " + code);

}

}

**public** **void** addCountry(Country country) {

countryRepository.save(country);

}

**public** **void** updateCountry(Country country) **throws** Exception {

**if** (countryRepository.existsById(country.getCode())) {

countryRepository.save(country);

} **else** {

**throw** **new** Exception("Cannot update - Country code not found: " + country.getCode());

}

}

**public** **void** deleteCountry(String code) **throws** Exception {

**if** (countryRepository.existsById(code)) {

countryRepository.deleteById(code);

} **else** {

**throw** **new** Exception("Cannot delete - Country code not found: " + code);

}

}

**public** List<Country> searchCountries(String keyword) {

**return** countryRepository.findByNameContaining(keyword);

}

}

application.properties Code:

# --------------------------------------

# Spring Logging Configuration

# --------------------------------------

logging.level.org.springframework=info

logging.level.com.cognizant=debug

# Hibernate SQL Logging (Optional for debugging)

logging.level.org.hibernate.SQL=trace

logging.level.org.hibernate.type.descriptor.sql=trace

# Console Log Format

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger**{25}** %25M %4L %m%n

# --------------------------------------

# MySQL DB Connection

# --------------------------------------

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

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn?useSSL=false&allowPublicKeyRetrieval=true

spring.datasource.username=root

spring.datasource.password=@Charu23

# --------------------------------------

# Hibernate JPA Configuration

# --------------------------------------

spring.jpa.hibernate.ddl-auto=validate

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

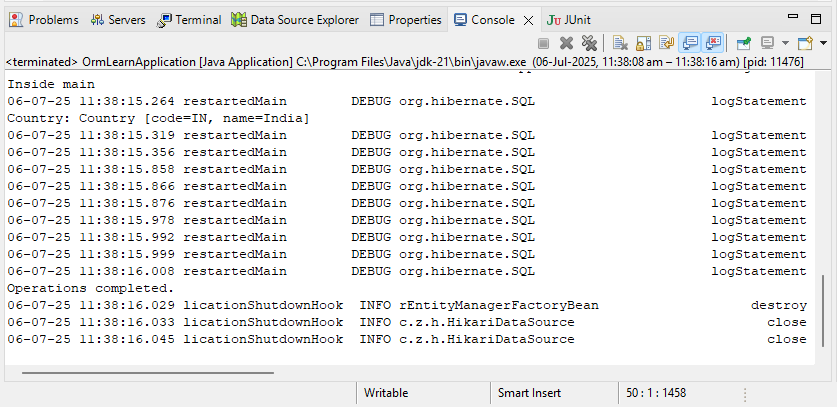
Excepted Output:

Inside main

Country: Country [code=IN, name=India]

Operations completed.

Output:



**Exercise-6:Find a country based on country code**

Project Name:orm-learn

OrmLearnApplication.java Code:

**package** com.cognizant.ormlearn;

**import** com.cognizant.ormlearn.model.Country;

**import** com.cognizant.springlearn.service.CountryService;

**import** com.cognizant.springlearn.service.exception.CountryNotFoundException;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@SpringBootApplication

@ComponentScan(basePackages = {"com.cognizant.ormlearn", "com.cognizant.springlearn"})

@EnableJpaRepositories(basePackages = "com.cognizant.springlearn.repository")

**public** **class** OrmLearnApplication {

**private** **static** **final** Logger ***LOGGER*** = LoggerFactory.*getLogger*(OrmLearnApplication.**class**);

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(OrmLearnApplication.**class**, args);

}

@Bean

**public** CommandLineRunner testRunner(CountryService countryService) {

**return** args -> {

***LOGGER***.info("Start");

Country country = countryService.findCountryByCode("IN");

System.***out***.println("Country: " + country);

***LOGGER***.info("End");

};

}

}

CountryRepository.java Code:

**package** com.cognizant.springlearn.repository;

**import** org.springframework.data.jpa.repository.JpaRepository;

**import** com.cognizant.ormlearn.model.Country;

**public** **interface** CountryRepository **extends** JpaRepository<Country, String> {

}

CountryService.java Code:

**package** com.cognizant.springlearn.service;

**import** java.util.Optional;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.cognizant.ormlearn.model.Country; // ✅ use your correct path here

**import** com.cognizant.springlearn.repository.CountryRepository;

**import** com.cognizant.springlearn.service.exception.CountryNotFoundException;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

@Transactional

**public** Country findCountryByCode(String code) **throws** CountryNotFoundException {

Optional<Country> result = countryRepository.findById(code);

**if** (!result.isPresent()) {

**throw** **new** CountryNotFoundException("Country not found: " + code);

}

**return** result.get();

}

}

CountryNotFoundException.java Code:

**package** com.cognizant.springlearn.service.exception;

**public** **class** CountryNotFoundException **extends** Exception {

**public** CountryNotFoundException(String message) {

**super**(message);

}

}

application.properties Code:

# 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

# Database config

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=@Charu23

# Hibernate

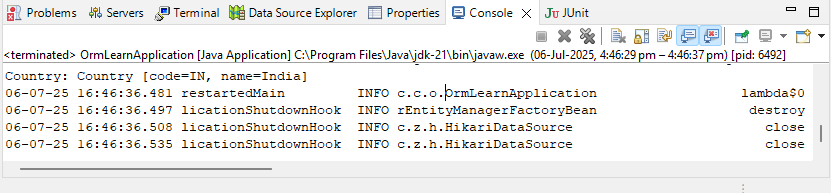
spring.jpa.hibernate.ddl-auto=validate

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

Excepted Output:

Country: Country [code=IN, name=India]

Output:



**Exercise-7:Add a new country**

Project Name:orm-learn

OrmLearnApplication.java Code:

**package** com.cognizant.ormlearn;

**import** com.cognizant.ormlearn.model.Country;

**import** com.cognizant.springlearn.service.CountryService;

**import** com.cognizant.springlearn.service.exception.CountryNotFoundException;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.data.jpa.repository.config.EnableJpaRepositories;

**import** jakarta.transaction.Transactional;

@SpringBootApplication

@ComponentScan(basePackages = {"com.cognizant.ormlearn", "com.cognizant.springlearn"})

@EnableJpaRepositories(basePackages = "com.cognizant.springlearn.repository")

**public** **class** OrmLearnApplication {

**private** **static** **final** Logger ***LOGGER*** = LoggerFactory.*getLogger*(OrmLearnApplication.**class**);

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(OrmLearnApplication.**class**, args);

}

// New method for Hands-on 7: Add a new country

@Bean

**public** CommandLineRunner testAddCountry(CountryService countryService) {

**return** args -> {

***LOGGER***.info("Start - testAddCountry");

// Step 1: Create a new country instance

Country newCountry = **new** Country();

newCountry.setCode("XY");

newCountry.setName("Testland");

// Step 2: Add country using service

countryService.addCountry(newCountry);

// Step 3: Fetch and display added country

Country fetchedCountry = countryService.findCountryByCode("XY");

System.***out***.println("Added Country: " + fetchedCountry);

***LOGGER***.info("End - testAddCountry");

};

}

// Optionally comment this if you already had a previous runner

/\*

@Bean

public CommandLineRunner testRunner(CountryService countryService) {

return args -> {

LOGGER.info("Start");

Country country = countryService.findCountryByCode("IN");

System.out.println("Country: " + country);

LOGGER.info("End");

};

}

\*/

}

CountryService.java Code:

**package** com.cognizant.springlearn.service;

**import** com.cognizant.ormlearn.model.Country;

**import** com.cognizant.springlearn.repository.CountryRepository;

**import** com.cognizant.springlearn.service.exception.CountryNotFoundException;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** java.util.Optional;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

**public** Country findCountryByCode(String code) **throws** CountryNotFoundException {

Optional<Country> result = countryRepository.findById(code);

**if** (result.isEmpty()) {

**throw** **new** CountryNotFoundException("Country not found with code: " + code);

}

**return** result.get();

}

@Transactional

**public** **void** addCountry(Country country) {

countryRepository.save(country);

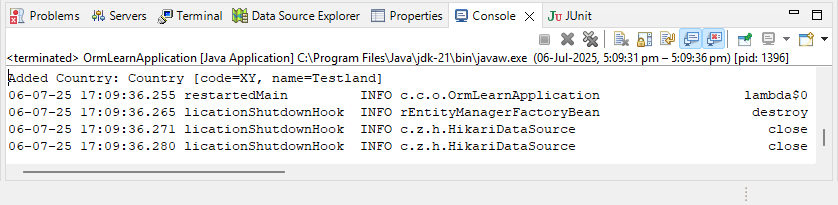
}

}

Excepted Output:

Added Country: Country [code=XY, name=Testland]

Output:



**Exercise-5:Get all employees using Native Query**

Project Name:orm-learn

OrmLearnApplication.java Code:

**package** com.cognizant.ormlearn;

**import** com.cognizant.ormlearn.model.Country;

**import** com.cognizant.ormlearn.model.Employee;

**import** com.cognizant.springlearn.service.CountryService;

**import** com.cognizant.springlearn.service.exception.CountryNotFoundException;

**import** com.cognizant.ormlearn.service.EmployeeService;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.data.jpa.repository.config.EnableJpaRepositories;

**import** java.util.List;

@SpringBootApplication

@ComponentScan(basePackages = {"com.cognizant.ormlearn", "com.cognizant.springlearn"})

@EnableJpaRepositories(basePackages = {"com.cognizant.ormlearn.repository", "com.cognizant.springlearn.repository"})

**public** **class** OrmLearnApplication {

**private** **static** **final** Logger ***LOGGER*** = LoggerFactory.*getLogger*(OrmLearnApplication.**class**);

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(OrmLearnApplication.**class**, args);

}

// Uncomment this if testing CountryService add/find

/\*

@Bean

public CommandLineRunner testCountryRunner(CountryService countryService) {

return args -> {

LOGGER.info("Start Country Add Test");

Country newCountry = new Country();

newCountry.setCode("JP");

newCountry.setName("Japan");

countryService.addCountry(newCountry);

Country country = countryService.findCountryByCode("JP");

System.out.println("Country: " + country);

LOGGER.info("End Country Add Test");

};

}

\*/

// For Hands-on 5: Get all employees using Native Query

@Bean

**public** CommandLineRunner testEmployeeRunner(EmployeeService employeeService) {

**return** args -> {

***LOGGER***.info("Start Employee Native Query Test");

System.***out***.println("All Employees (Native Query):");

List<Employee> employees = employeeService.getAllEmployeesNative();

employees.forEach(System.***out***::println);

***LOGGER***.info("End Employee Native Query Test");

};

}

}

Employee.java Code:

**package** com.cognizant.ormlearn.model;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.Id;

**import** jakarta.persistence.Table;

@Entity

@Table(name = "employee")

**public** **class** Employee {

@Id

**private** **int** id;

**private** String name;

**private** **double** salary;

**private** String permanent;

**private** String dateOfBirth;

// Getters and Setters

**public** **int** getId() { **return** id; }

**public** **void** setId(**int** id) { **this**.id = id; }

**public** String getName() { **return** name; }

**public** **void** setName(String name) { **this**.name = name; }

**public** **double** getSalary() { **return** salary; }

**public** **void** setSalary(**double** salary) { **this**.salary = salary; }

**public** String getPermanent() { **return** permanent; }

**public** **void** setPermanent(String permanent) { **this**.permanent = permanent; }

**public** String getDateOfBirth() { **return** dateOfBirth; }

**public** **void** setDateOfBirth(String dateOfBirth) { **this**.dateOfBirth = dateOfBirth; }

@Override

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", salary=" + salary + ", permanent=" + permanent + ", dob=" + dateOfBirth + "]";

}

}

EmployeeRepository.java Code:

**package** com.cognizant.ormlearn.repository;

**import** com.cognizant.ormlearn.model.Employee;

**import** org.springframework.data.jpa.repository.JpaRepository;

**import** org.springframework.data.jpa.repository.Query;

**import** java.util.List;

**public** **interface** EmployeeRepository **extends** JpaRepository<Employee, Integer> {

@Query(value = "SELECT \* FROM employee", nativeQuery = **true**)

List<Employee> getAllEmployeesNative();

}

EmployeeService.java Code:

**package** com.cognizant.ormlearn.service;

**import** com.cognizant.ormlearn.model.Employee;

**import** com.cognizant.ormlearn.repository.EmployeeRepository;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** java.util.List;

@Service

**public** **class** EmployeeService {

@Autowired

**private** EmployeeRepository employeeRepository;

**public** List<Employee> getAllEmployeesNative() {

**return** employeeRepository.getAllEmployeesNative();

}

}

application.properties Code:

# Database connection properties

spring.datasource.url=jdbc:mysql://localhost:3306/orm\_learn

spring.datasource.username=root

spring.datasource.password=@Charu23

# JPA & Hibernate settings

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

# MySQL dialect (optional in Hibernate 6+, but still safe to add)

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

# Prevent timezone issues (optional, but recommended)

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

spring.datasource.hikari.connection-timeout=20000

spring.datasource.hikari.maximum-pool-size=5

Expected Output:

All Employees (Native Query):

Hibernate: SELECT \* FROM employee

Employee [id=1, name=John, salary=50000.0, permanent=true, dob=1990-01-01]

Employee [id=2, name=Alice, salary=60000.0, permanent=false, dob=1992-05-20]

Employee [id=3, name=Bob, salary=55000.0, permanent=true, dob=1988-11-15]

Output:

