Demonstrate Writing Hibernate Query Language and Native Query

Java Entity Class: Employee.java

package com.example.querydemo.entity;  
  
import jakarta.persistence.\*;  
  
@Entity  
@Table(name = "employees")  
public class Employee {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private Long id;  
  
 private String name;  
 private String department;  
 private Double salary;  
  
 public String getName() {

return name;

}

public String getDepartment() {

return department;

}

public Double getSalary() {

return salary;

}

}

Repository Interface: EmployeeRepository.java

package com.example.querydemo.repository;  
  
import com.example.querydemo.entity.Employee;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
import java.util.List;  
  
public interface EmployeeRepository extends JpaRepository<Employee, Long> {  
  
 @Query("SELECT e FROM Employee e WHERE e.salary > :minSalary")  
 List<Employee> findEmployeesWithHighSalary(@Param("minSalary") double salary);  
  
 @Query("SELECT AVG(e.salary) FROM Employee e")  
 Double findAverageSalary();  
  
 @Query(value = "SELECT \* FROM employees WHERE department = :dept", nativeQuery = true)  
 List<Employee> findByDepartment(@Param("dept") String department);  
}

REST Controller: EmployeeController.java

package com.example.querydemo.controller;  
  
import com.example.querydemo.entity.Employee;  
import com.example.querydemo.repository.EmployeeRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/employees")  
public class EmployeeController {  
  
 @Autowired  
 private EmployeeRepository repository;  
  
 @PostMapping  
 public Employee save(@RequestBody Employee emp) {  
 return repository.save(emp);  
 }  
  
 @GetMapping("/highsalary/{salary}")  
 public List<Employee> getHighSalary(@PathVariable double salary) {  
 return repository.findEmployeesWithHighSalary(salary);  
 }  
  
 @GetMapping("/average-salary")  
 public Double getAverageSalary() {  
 return repository.findAverageSalary();  
 }  
  
 @GetMapping("/department/{dept}")  
 public List<Employee> getByDepartment(@PathVariable String dept) {  
 return repository.findByDepartment(dept);  
 }  
}

Spring Boot Application: QueryDemoApplication.java

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

Configuration: application.properties

spring.datasource.url=jdbc:mysql://localhost:3306/testdb  
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.MySQL8Dialect

SQL Script to Insert Data

CREATE DATABASE testdb;  
  
USE testdb;  
  
INSERT INTO employees(name, department, salary) VALUES  
('John Doe', 'IT', 80000),  
('Jane Smith', 'HR', 50000),  
('Mark Wilson', 'IT', 95000),  
('Emily Johnson', 'Finance', 70000);

**Output:**









