**Write queries on country table using Query Methods**

**Country.java**

package com.cognizant.ormlearn;

import java.util.List;

import org.springframework.boot.CommandLineRunner;

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.repository.CountryRepository;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

private final CountryRepository countryRepository;

public OrmLearnApplication(CountryRepository countryRepository) {

this.countryRepository = countryRepository;

}

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String[] args) {

System.out.println("Countries containing 'ou':");

List<Country> countries = countryRepository.findByNameContaining("ou");

countries.forEach(c -> System.out.println(c.getCode() + " " + c.getName()));

System.out.println("\nCountries containing 'ou' in ascending order:");

countries = countryRepository.findByNameContainingOrderByNameAsc("ou");

countries.forEach(c -> System.out.println(c.getCode() + " " + c.getName()));

System.out.println("\nCountries starting with 'Z':");

countries = countryRepository.findByNameStartingWith("Z");

countries.forEach(c -> System.out.println(c.getCode() + " " + c.getName()));

}

}

**CountryRepository.java**

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);

List<Country> findByNameContainingOrderByNameAsc(String keyword);

List<Country> findByNameStartingWith(String prefix);

}

**ormApplication.java**

package com.cognizant.ormlearn;

import java.util.List;

import org.springframework.boot.CommandLineRunner;

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.repository.CountryRepository;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

private final CountryRepository countryRepository;

public OrmLearnApplication(CountryRepository countryRepository) {

this.countryRepository = countryRepository;

}

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) {

System.out.println("Countries containing 'ou':");

List<Country> countries = countryRepository.findByNameContaining("ou");

countries.forEach(c -> System.out.println(c.getCode() + " " + c.getName()));

System.out.println("\nCountries containing 'ou' in ascending order:");

countries = countryRepository.findByNameContainingOrderByNameAsc("ou");

countries.forEach(c -> System.out.println(c.getCode() + " " + c.getName()));

System.out.println("\nCountries starting with 'Z':");

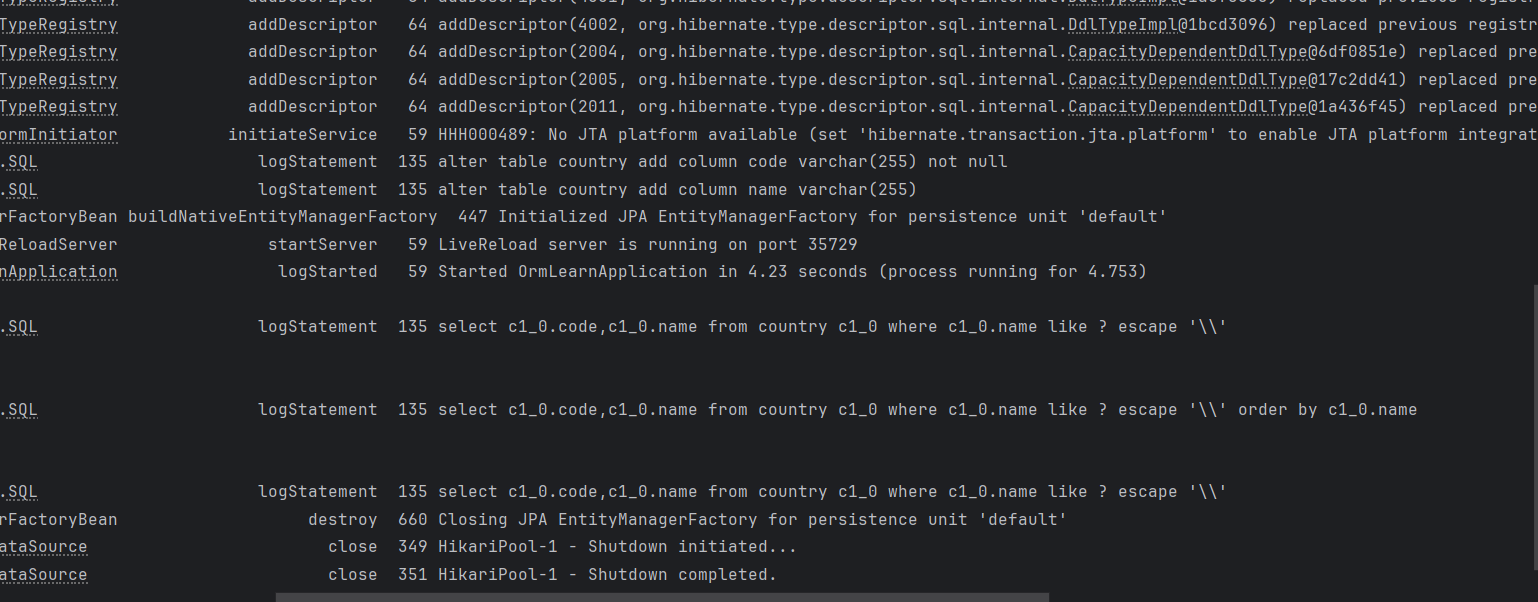
countries = countryRepository.findByNameStartingWith("Z");

countries.forEach(c -> System.out.println(c.getCode() + " " + c.getName()));

}

}

**Output:**

****

**Write queries on stock table using Query Methods** 

CREATE TABLE IF NOT EXISTS stock (

st\_id INT NOT NULL AUTO\_INCREMENT,

st\_code VARCHAR(10),

st\_date DATE,

st\_open DECIMAL(10,2),

st\_close DECIMAL(10,2),

st\_volume BIGINT,

PRIMARY KEY (st\_id)

);

INSERT INTO stock (st\_code, st\_date, st\_open, st\_close, st\_volume) VALUES

('FB', '2019-09-03', 184.00, 182.39, 9779400),

('FB', '2019-09-04', 184.65, 187.14, 11308000),

('GOOGL', '2019-04-22', 1236.67, 1253.76, 954200),

('GOOGL', '2019-04-23', 1256.64, 1270.59, 1593400),

('NFLX', '2018-12-21', 263.83, 246.39, 21397600),

('NFLX', '2018-12-24', 242.00, 233.88, 9547600),

('NFLX', '2018-12-26', 233.92, 253.67, 14402700),

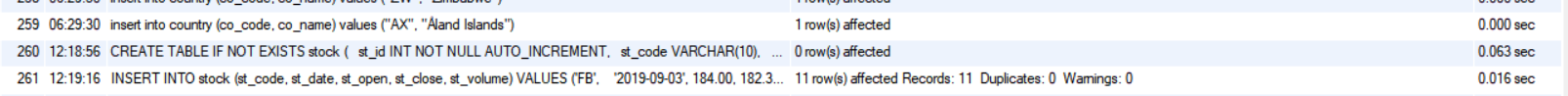
('FB', '2019-01-31', 165.60, 166.69, 77233600),

('FB', '2018-10-31', 155.00, 151.79, 60101300),

('FB', '2018-12-19', 141.21, 133.24, 57404900),

('GOOGL', '2019-10-17', 1251.40, 1252.80, 1047900);

**Output:**

****

**Stock.java**

package com.example.ormlearn.model;

import jakarta.persistence.\*;

import java.time.LocalDate;

@Entity

@Table(name = "stock")

public class Stock {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int stId;

private String stCode;

private LocalDate stDate;

private double stOpen;

private double stClose;

private long stVolume;

}

**StockRepository.java**

package com.example.ormlearn.repository;

import com.example.ormlearn.model.Stock;

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

import java.time.LocalDate;

import java.util.List;

public interface StockRepository extends JpaRepository<Stock, Integer> {

List<Stock> findByStCodeAndStDateBetween(String code, LocalDate start, LocalDate end);

List<Stock> findByStCodeAndStCloseGreaterThan(String code, double price);

List<Stock> findTop3ByOrderByStVolumeDesc();

List<Stock> findTop3ByStCodeOrderByStCloseAsc(String code);

}

**Ormapplication.java**

package com.example.ormlearn;

import com.example.ormlearn.model.Stock;

import com.example.ormlearn.repository.StockRepository;

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.time.LocalDate;

import java.util.List;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private StockRepository stockRepository;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String[] args) {

System.out.println("FB Stocks - Sept 2019:");

List<Stock> fbStocks = stockRepository.findByStCodeAndStDateBetween(

"FB", LocalDate.of(2019, 9, 1), LocalDate.of(2019, 9, 30));

fbStocks.forEach(System.out::println);

System.out.println("Google Stocks > 1250:");

stockRepository.findByStCodeAndStCloseGreaterThan("GOOGL", 1250)

.forEach(System.out::println);

System.out.println("Top 3 by Volume:");

stockRepository.findTop3ByOrderByStVolumeDesc()

.forEach(System.out::println);

System.out.println("Netflix lowest prices:");

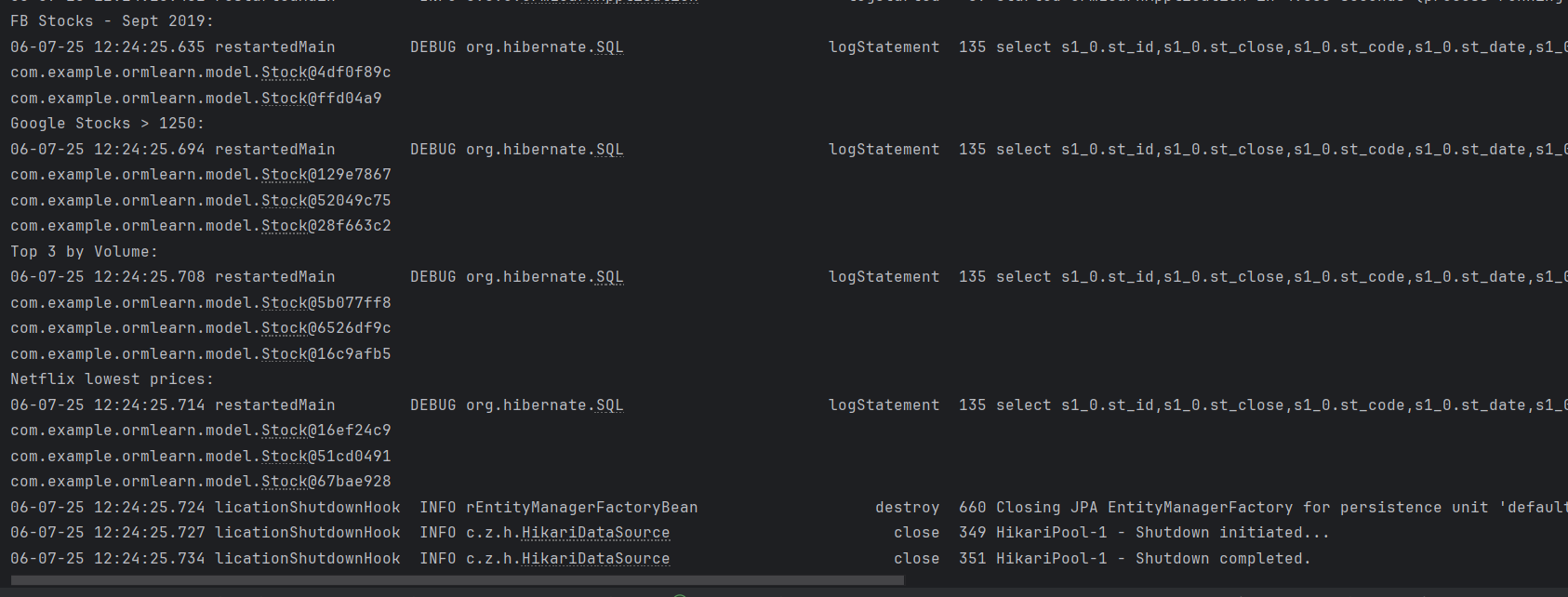
stockRepository.findTop3ByStCodeOrderByStCloseAsc("NFLX")

.forEach(System.out::println);

}

}

**Output:**

****

**Create payroll tables and bean mapping** 

**Department.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.List;

@Entity

@Table(name = "department")

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column

private String name;

@OneToMany(mappedBy = "department")

private List<Employee> employeeList;

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 List<Employee> getEmployeeList() {

return employeeList;

}

public void setEmployeeList(List<Employee> employeeList) {

this.employeeList = employeeList;

}

@Override

public String toString() {

return "Department [id=" + id + ", name=" + name + "]";

}

}

**Employee.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.sql.Date;

import java.util.Set;

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column

private String name;

@Column

private double salary;

@Column

private boolean permanent;

@Column(name = "date\_of\_birth")

private Date dateOfBirth;

@ManyToOne

@JoinColumn(name = "department\_id")

private Department department;

@ManyToMany

@JoinTable(

name = "employee\_skill",

joinColumns = @JoinColumn(name = "employee\_id"),

inverseJoinColumns = @JoinColumn(name = "skill\_id")

)

private Set<Skill> skillList;

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 boolean isPermanent() {

return permanent;

}

public void setPermanent(boolean permanent) {

this.permanent = permanent;

}

public Date getDateOfBirth() {

return dateOfBirth;

}

public void setDateOfBirth(Date dateOfBirth) {

this.dateOfBirth = dateOfBirth;

}

public Department getDepartment() {

return department;

}

public void setDepartment(Department department) {

this.department = department;

}

public Set<Skill> getSkillList() {

return skillList;

}

public void setSkillList(Set<Skill> skillList) {

this.skillList = skillList;

}

@Override

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", salary=" + salary +

", permanent=" + permanent + ", dateOfBirth=" + dateOfBirth + "]";

}

}

**Skill.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.Set;

@Entity

@Table(name = "skill")

public class Skill {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column

private String name;

@ManyToMany(mappedBy = "skillList")

private Set<Employee> employeeList;

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 Set<Employee> getEmployeeList() {

return employeeList;

}

public void setEmployeeList(Set<Employee> employeeList) {

this.employeeList = employeeList;

}

@Override

public String toString() {

return "Skill [id=" + id + ", name=" + name + "]";

}

}

**DepartmentRepository.java**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Department;

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

public interface DepartmentRepository extends JpaRepository<Department, Integer> {

}

**EmployeeRepository.java**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Employee;

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**SkillRepository.java**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Skill;

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

import org.springframework.stereotype.Repository;

@Repository

public interface SkillRepository extends JpaRepository<Skill, Integer> {

}

**Ormapplication.java**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Department;

import com.cognizant.ormlearn.model.Employee;

import com.cognizant.ormlearn.model.Skill;

import com.cognizant.ormlearn.repository.DepartmentRepository;

import com.cognizant.ormlearn.repository.EmployeeRepository;

import com.cognizant.ormlearn.repository.SkillRepository;

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.sql.Date;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private EmployeeRepository employeeRepository;

@Autowired

private DepartmentRepository departmentRepository;

@Autowired

private SkillRepository skillRepository;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) {

Department department = new Department();

department.setName("Technology");

departmentRepository.save(department);

Employee employee = new Employee();

employee.setName("Alice Johnson");

employee.setSalary(75000.00);

employee.setPermanent(true);

employee.setDateOfBirth(Date.valueOf("1992-08-10"));

employee.setDepartment(department);

employeeRepository.save(employee);

Skill javaSkill = new Skill();

javaSkill.setName("Java");

Skill sqlSkill = new Skill();

sqlSkill.setName("SQL");

skillRepository.saveAll(List.of(javaSkill, sqlSkill));

Set<Skill> skills = new HashSet<>();

skills.add(javaSkill);

skills.add(sqlSkill);

employee.setSkillList(skills);

employeeRepository.save(employee);

System.out.println("All Employees:");

employeeRepository.findAll().forEach(System.out::println);

System.out.println("\nAll Departments:");

departmentRepository.findAll().forEach(System.out::println);

System.out.println("\nAll Skills:");

skillRepository.findAll().forEach(System.out::println);

}

}

**Output:**

