**Hands on 1**

**Write queries on country table using Query Methods**   
  
An application has a search text box for searching by country. When typing characters on the text box, a list of all the matching countries should be displayed. For example, if 'ou' is entered in the search box the following countries should be displayed. Write a Query Method to achieve this feature. Implement this method in CountryRepository.

**Program:**

**CountryService.java**

package com.example.orm\_learn.service;

import com.example.orm\_learn.model.Country;

import java.util.List;

public interface CountryService {

List<Country> getAllCountries();

List<Country> searchCountries(String keyword);

List<Country> searchCountriesSorted(String keyword);

List<Country> getCountriesByStartingLetter(String letter);

}

**CountryServiceImpl.java**

package com.example.orm\_learn.service;

import com.example.orm\_learn.model.Country;

import com.example.orm\_learn.repository.CountryRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryServiceImpl implements CountryService {

@Autowired

private CountryRepository countryRepository;

@Override

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

@Override

public List<Country> searchCountries(String keyword) {

return countryRepository.findByNameContainingIgnoreCase(keyword);

}

@Override

public List<Country> searchCountriesSorted(String keyword) {

return countryRepository.findByNameContainingIgnoreCaseOrderByNameAsc(keyword);

}

@Override

public List<Country> getCountriesByStartingLetter(String letter) {

return countryRepository.findByNameStartingWithIgnoreCase(letter);

}

}

**CountryRepository.java**

package com.example.orm\_learn.repository;

import com.example.orm\_learn.model.Country;

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

import java.util.List;

public interface CountryRepository extends JpaRepository<Country, String> {

// 1. Find countries where name contains a given string

List<Country> findByNameContainingIgnoreCase(String keyword);

// 2. Same as above, but sorted ascending by name

List<Country> findByNameContainingIgnoreCaseOrderByNameAsc(String keyword);

// 3. Countries starting with a given letter

List<Country> findByNameStartingWithIgnoreCase(String letter);

}

**OrmLearnApplication.java**

package com.example.orm\_learn;

import com.example.orm\_learn.model.Country;

import com.example.orm\_learn.repository.CountryRepository;

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 CountryRepository countryRepository;

public static void main(String[] args) {

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

}

@Override

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

testFindByNameContainingIgnoreCase();

testFindByNameContainingIgnoreCaseOrderByNameAsc();

testFindByNameStartingWithIgnoreCase();

}

// 1. Search country by keyword

void testFindByNameContainingIgnoreCase() {

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

List<Country> result = countryRepository.findByNameContainingIgnoreCase("ou");

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

}

// 2. Search and sort ascending

void testFindByNameContainingIgnoreCaseOrderByNameAsc() {

System.*out*.println("\n Countries containing 'ou' sorted by name:");

List<Country> result = countryRepository.findByNameContainingIgnoreCaseOrderByNameAsc("ou");

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

}

// 3. Countries starting with a letter

void testFindByNameStartingWithIgnoreCase() {

System.*out*.println("\n Countries starting with 'Z':");

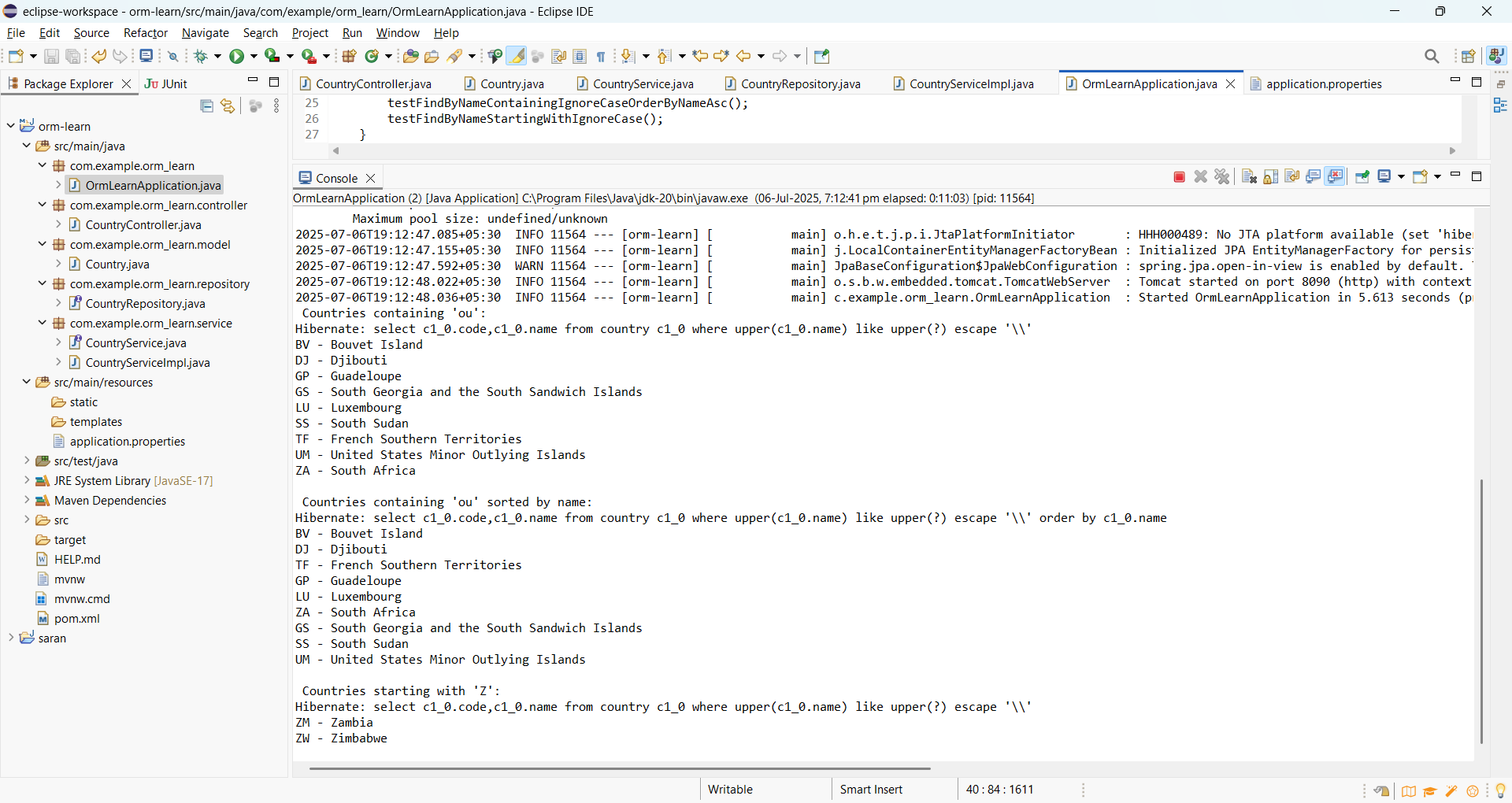
List<Country> result = countryRepository.findByNameStartingWithIgnoreCase("Z");

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

}

}

**Output:**



**Hands on 2 : Write queries on stock table using Query Methods**   
  
With one year stock data of Facebook, Google and Netflix, we need to implement Spring Data JPA Query Methods for the following scenarios:  
  
**Sample Data**  
Sample data for implementing this hands on is provided to you in the platform

**Setup stock data**

* Create a new table for storing stock details.

CREATE TABLE IF NOT EXISTS `ormlearn`.`stock` (

  `st\_id` INT NOT NULL AUTO\_INCREMENT,

  `st\_code` varchar(10),

  `st\_date` date,

  `st\_open` numeric(10,2),

  `st\_close` numeric(10,2),

  `st\_volume` numeric,

  PRIMARY KEY (`st\_id`)

);

**Create new class Stock in orm-learn project and define the required mapping annotations.**

**Stock.java**

package com.cognizant.orm\_learn.model;

import java.math.BigDecimal;

import java.sql.Date;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "stock")

public class Stock {

public Stock() {

super();

}

public Stock(int id, String code, Date date, BigDecimal open, BigDecimal close, BigDecimal volume) {

super();

this.id = id;

this.code = code;

this.date = date;

this.open = open;

this.close = close;

this.volume = volume;

}

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.*IDENTITY*)

*@Column*(name = "st\_id")

private int id;

*@Column*(name = "st\_code")

private String code;

*@Column*(name = "st\_date")

private Date date;

*@Column*(name = "st\_open")

private BigDecimal open;

*@Column*(name = "st\_close")

private BigDecimal close;

*@Column*(name = "st\_volume")

private BigDecimal volume;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

public BigDecimal getOpen() {

return open;

}

public void setOpen(BigDecimal open) {

this.open = open;

}

public BigDecimal getClose() {

return close;

}

public void setClose(BigDecimal close) {

this.close = close;

}

public BigDecimal getVolume() {

return volume;

}

public void setVolume(BigDecimal volume) {

this.volume = volume;

}

*@Override*

public String toString() {

return "Stock [id=" + id + ", code=" + code + ", date=" + date + ", open=" + open + ", close=" + close

+ ", volume=" + volume + "]";

}

}

**Create StockRepository class to write the Query Methods.**

**StockRepository.java**

package com.cognizant.orm\_learn.repository;

import java.math.BigDecimal;

import java.util.\*;

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

import com.cognizant.orm\_learn.model.Stock;

public interface StockRepository extends JpaRepository<Stock, Integer> {

List<Stock> findByCodeAndDateBetween(String code, java.util.Date start, java.util.Date end);

List<Stock> findByCodeAndCloseGreaterThan(String code, BigDecimal close);

List<Stock> findTop3ByOrderByVolumeDesc();

List<Stock> findTop3ByCodeOrderByCloseAsc(String code);

}

**Create methods in OrmLearnApplication to test by autowiring StockRepository directly.**

**OrmLearnApplication.java**

package com.cognizant.orm\_learn;

import java.math.BigDecimal;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.\*;

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.orm\_learn.model.Stock;

import com.cognizant.orm\_learn.repository.StockRepository;

*@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) {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

try {

System.*out*.println("Facebook September 2019:");

List<Stock> fbSept = stockRepository.findByCodeAndDateBetween(

"FB",

sdf.parse("2019-09-01"),

sdf.parse("2019-09-30"));

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

System.*out*.println("\nGoogle stock > 1250:");

List<Stock> googleHigh = stockRepository.findByCodeAndCloseGreaterThan("GOOGL", new BigDecimal("1250"));

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

System.*out*.println("\nTop 3 highest volume:");

List<Stock> topVolume = stockRepository.findTop3ByOrderByVolumeDesc();

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

System.*out*.println("\nNetflix lowest close:");

List<Stock> nflxLow = stockRepository.findTop3ByCodeOrderByCloseAsc("NFLX");

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

} catch (ParseException e) {

e.printStackTrace();

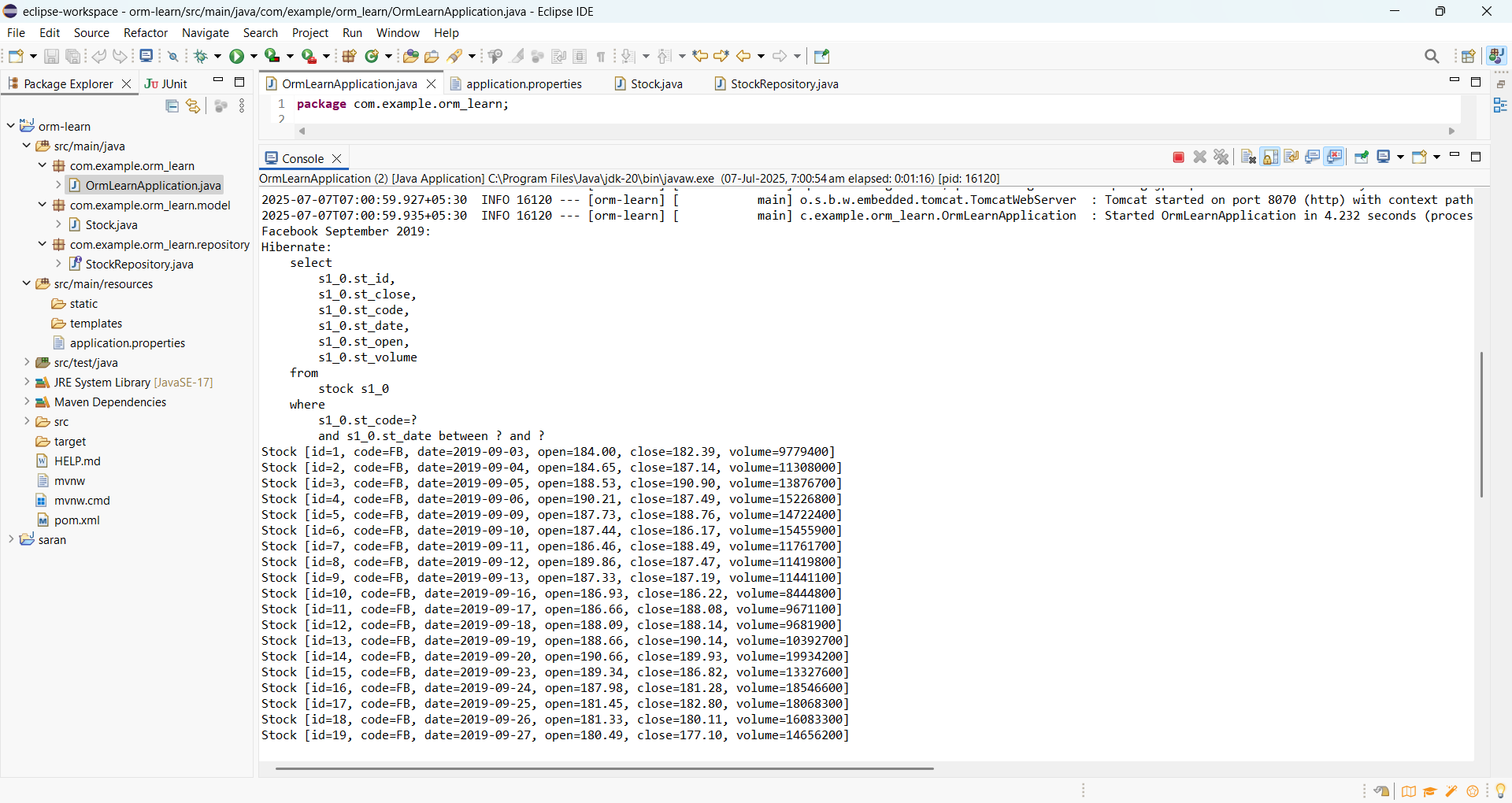
}

}

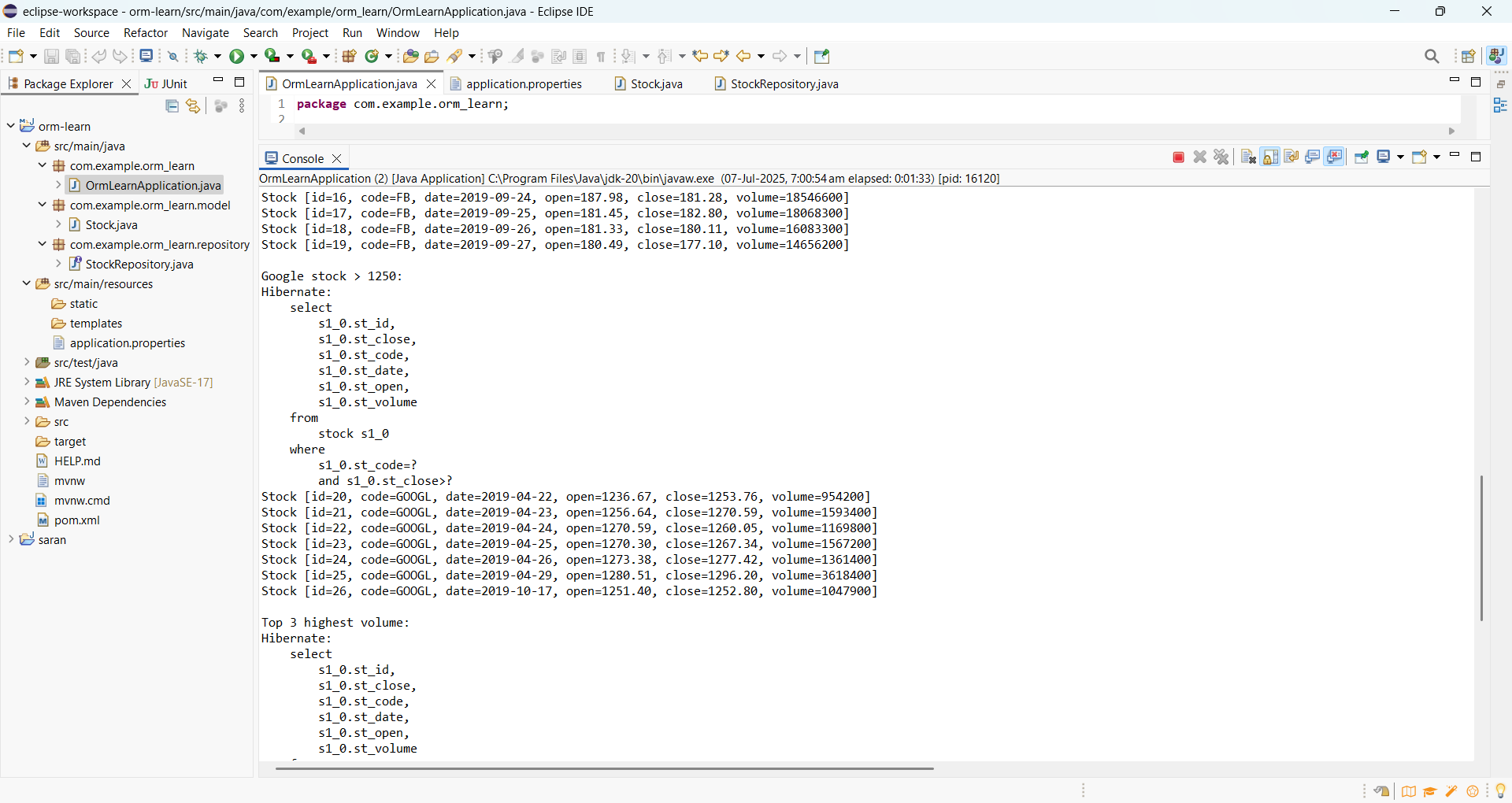
}

**Query Methods required for the following scenarios**

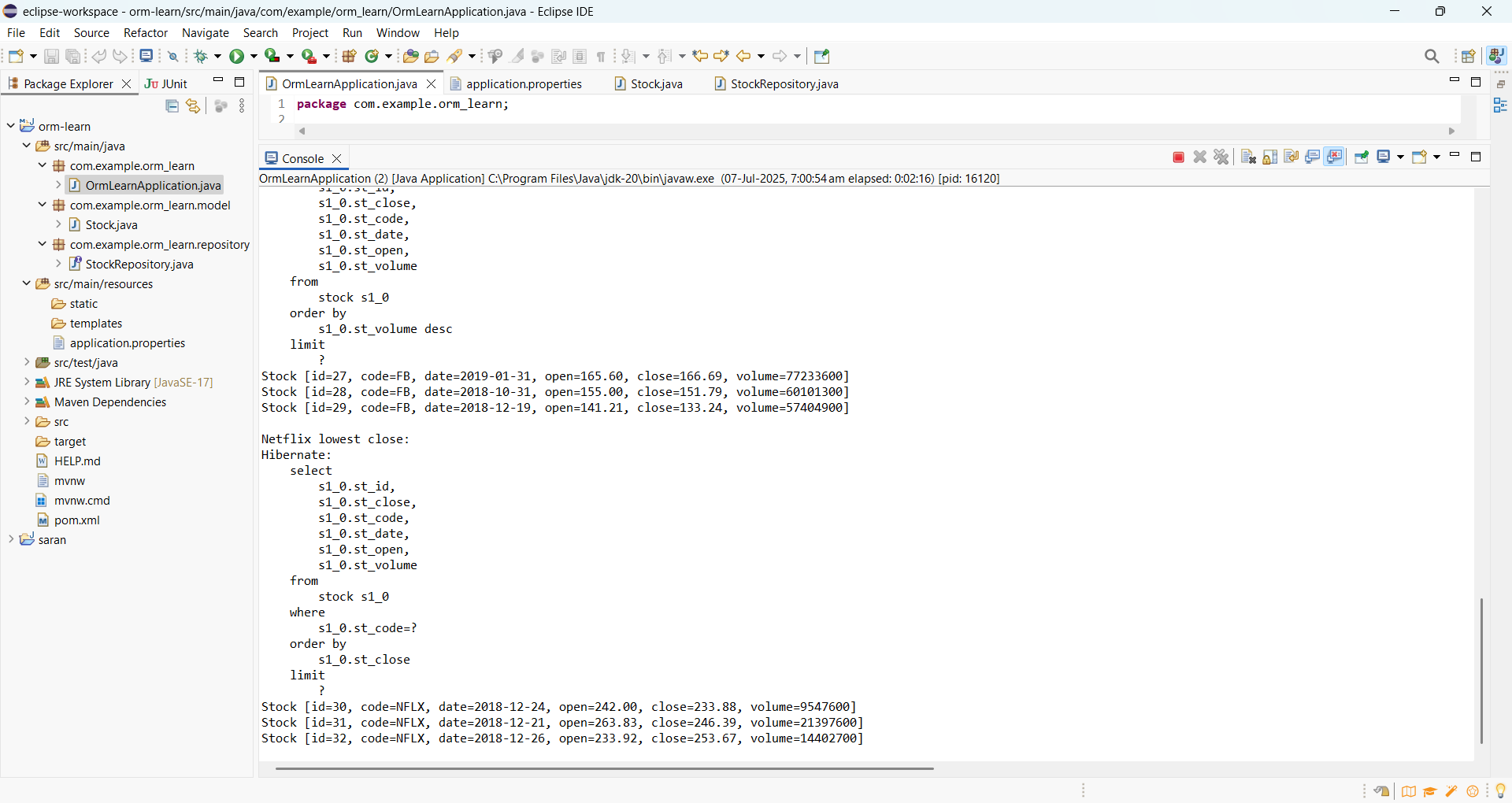
**Get all stock details of Facebook in the month of September 2019.**



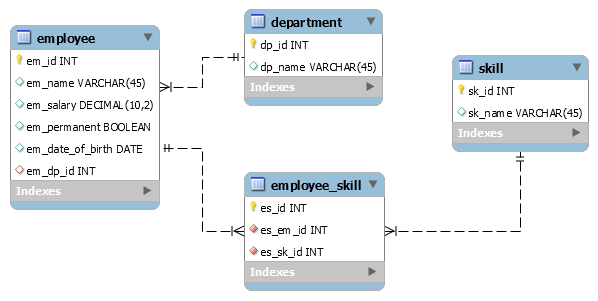
**Get all google stock details where the stock price was greater than 1250**



**Identify three dates when Netflix stocks were the lowest**



**Hands on 3 : Create payroll tables and bean mapping**   
  
To demonstrate one to many, many to one and many to many relationships in Hibernate, a schema with entities employee, department and skill will be used. In this hands on we will setup the tables and data, which forms the basis for learning the mappings in Hibernate.  
  
**Schema Structure**



**Create model classes Employee, Department and Skill in com.cognizant.orm-learn.model package**

**Employee.java**

package com.cognizant.orm\_learn.model;

import java.sql.Date;

import java.util.List;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.JoinTable;

import jakarta.persistence.ManyToMany;

import jakarta.persistence.ManyToOne;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "employee")

public class Employee {

public Employee() {

super();

}

public Employee(int id, String name, double salary, boolean permanent, Date dateOfBirth, Department department,

List<Skill> skillList) {

super();

this.id = id;

this.name = name;

this.salary = salary;

this.permanent = permanent;

this.dateOfBirth = dateOfBirth;

this.department = department;

this.skillList = skillList;

}

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

*@Column*(name = "em\_id")

private int id;

*@Column*(name = "em\_name")

private String name;

*@Column*(name = "em\_salary")

private double salary;

*@Column*(name = "em\_permanent")

private boolean permanent;

*@Column*(name = "em\_date\_of\_birth")

private Date dateOfBirth;

*@ManyToOne*

*@JoinColumn*(name = "em\_dp\_id") // Foreign Key

private Department department;

*@ManyToMany*

*@JoinTable*(name = "employee\_skill",

joinColumns = *@JoinColumn*(name = "es\_em\_id"),

inverseJoinColumns = *@JoinColumn*(name = "es\_sk\_id"))

private List<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 List<Skill> getSkillList() {

return skillList;

}

public void setSkillList(List<Skill> skillList) {

this.skillList = skillList;

}

*@Override*

public String toString() {

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

}

}

**Department.java**

package com.cognizant.orm\_learn.model;

import java.util.List;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.OneToMany;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "department")

public class Department {

public Department() {

super();

}

public Department(int id, String name, List<Employee> employeeList) {

super();

this.id = id;

this.name = name;

this.employeeList = employeeList;

}

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

*@Column*(name = "dp\_id")

private int id;

*@Column*(name = "dp\_name")

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 + "]";

}

}

**Skill.java**

package com.cognizant.orm\_learn.model;

import java.util.List;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.ManyToMany;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "skill")

public class Skill {

public Skill() {

super();

}

public Skill(int id, String name, List<Employee> employeeList) {

super();

this.id = id;

this.name = name;

this.employeeList = employeeList;

}

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

*@Column*(name = "sk\_id")

private int id;

*@Column*(name = "sk\_name")

private String name;

*@ManyToMany*(mappedBy = "skillList")

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;

}

}

**Create appropriate repository interfaces EmployeeRepository, DepartmentRepository and SkillRepository in repository package.**

**EmployeeRepository.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.Employee;

*@Repository*

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**DepartmentRepository.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.Department;

*@Repository*

public interface DepartmentRepository extends JpaRepository<Department, Integer> {

}

**SkillRepository.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.Skill;

*@Repository*

public interface SkillRepository extends JpaRepository<Skill, Integer> {

}

**EmployeeService.java**

package com.cognizant.orm\_learn.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.orm\_learn.model.Employee;

import com.cognizant.orm\_learn.repository.EmployeeRepository;

*@Service*

public class EmployeeService {

*@Autowired*

private EmployeeRepository employeeRepository;

*@Transactional*

public void displayEmployeeDetails(int id) {

Optional<Employee> optional = employeeRepository.findById(id);

if (optional.isPresent()) {

Employee emp = optional.get();

System.***out***.println("Employee: " + emp);

System.***out***.println("Department: " + emp.getDepartment());

System.***out***.println("Department Employees Count: " + emp.getDepartment().getEmployeeList().size());

System.***out***.println("Skills: " + emp.getSkillList());

} else {

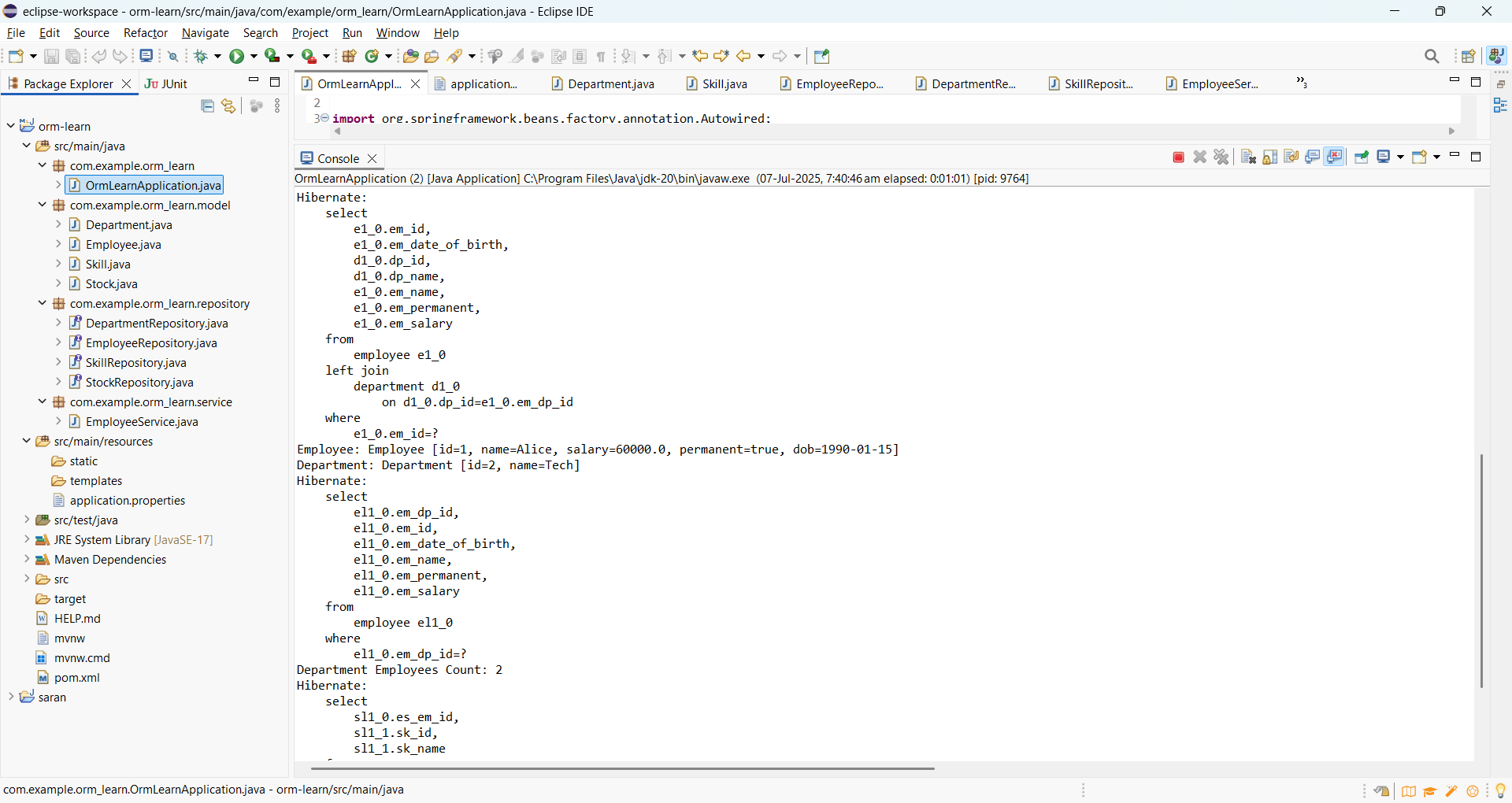
System.***out***.println("Employee not found");

}

}

}

**Output:**



**Hands on 4 : Implement many to one relationship between Employee and Department**   
  
Follow steps below to defined many to one relationship and perform persistence operations:  
  
**Preparation of Service Classes**

**Create EmployeeService, DepartmentService and SkillService defined with annotation @Service. In each of this class autowire respective repository.**

**EmployeeService.java**

package com.cognizant.orm\_learn.service;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.stereotype.Service;

import com.cognizant.orm\_learn.model.Employee;

import com.cognizant.orm\_learn.repository.EmployeeRepository;

import jakarta.transaction.Transactional;

*@Service*

public class EmployeeService {

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

*@Autowired*

private EmployeeRepository employeeRepository;

*@Transactional*

public void displayEmployeeDetails(int id) {

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

Employee employee = employeeRepository.findById(id).get();

***LOGGER***.debug("Employee: {}", employee);

***LOGGER***.debug("Department: {}", employee.getDepartment());

***LOGGER***.debug("Skills: {}", employee.getSkillList());

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

}

*@Transactional*

public Employee get(int id) {

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

return employeeRepository.findById(id).get();

}

*@Transactional*

public void save(Employee employee) {

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

employeeRepository.save(employee);

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

}

}

**DepartmentService.java**

package com.cognizant.orm\_learn.service;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.orm\_learn.model.Department;

import com.cognizant.orm\_learn.repository.DepartmentRepository;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

*@Service*

public class DepartmentService {

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

*@Autowired*

private DepartmentRepository departmentRepository;

*@Transactional*

public Department get(int id) {

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

return departmentRepository.findById(id).get();

}

*@Transactional*

public void save(Department department) {

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

departmentRepository.save(department);

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

}

}

**SkillService.java**

package com.cognizant.orm\_learn.service;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.orm\_learn.model.Skill;

import com.cognizant.orm\_learn.repository.SkillRepository;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

*@Service*

public class SkillService {

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

*@Autowired*

private SkillRepository skillRepository;

*@Transactional*

public Skill get(int id) {

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

return skillRepository.findById(id).get();

}

*@Transactional*

public void save(Skill skill) {

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

skillRepository.save(skill);

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

}

}

**Implementation of @ManyToOne mapping**

**Define department in Employee bean with @ManyToOne and @JoinTable annotation. This defines the relationship between the entities.**

**Employee.java**

package com.cognizant.orm\_learn.model;

import java.sql.Date;

import java.util.List;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.JoinTable;

import jakarta.persistence.ManyToMany;

import jakarta.persistence.ManyToOne;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "employee")

public class Employee {

public Employee() {

super();

}

public Employee(int id, String name, double salary, boolean permanent, Date dateOfBirth, Department department,

List<Skill> skillList) {

super();

this.id = id;

this.name = name;

this.salary = salary;

this.permanent = permanent;

this.dateOfBirth = dateOfBirth;

this.department = department;

this.skillList = skillList;

}

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

*@Column*(name = "em\_id")

private int id;

*@Column*(name = "em\_name")

private String name;

*@Column*(name = "em\_salary")

private double salary;

*@Column*(name = "em\_permanent")

private boolean permanent;

*@Column*(name = "em\_date\_of\_birth")

private Date dateOfBirth;

*@ManyToOne*

*@JoinColumn*(name = "em\_dp\_id")

private Department department;

*@ManyToMany*

*@JoinTable*(name = "employee\_skill",

joinColumns = *@JoinColumn*(name = "es\_em\_id"),

inverseJoinColumns = *@JoinColumn*(name = "es\_sk\_id"))

private List<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 List<Skill> getSkillList() {

return skillList;

}

public void setSkillList(List<Skill> skillList) {

this.skillList = skillList;

}

*@Override*

public String toString() {

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

}

}

**Getting Employee along with Department**

Create new method testGetEmployee() in OrmLearnApplication

**Add Employee**

Create new method testAddEmployee() in OrmLearnApplication and implement the following steps

**Update Employee**

Create new method testUpdateEmployee() in OrmLearnApplication and implement the following steps

**OrmLearnApplication.java**

package com.cognizant.orm\_learn;

import java.time.LocalDate;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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.orm\_learn.model.Department;

import com.cognizant.orm\_learn.model.Employee;

import com.cognizant.orm\_learn.service.DepartmentService;

import com.cognizant.orm\_learn.service.EmployeeService;

import com.cognizant.orm\_learn.service.SkillService;

*@SpringBootApplication*

public class OrmLearnApplication implements CommandLineRunner {

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

*@Autowired*

private EmployeeService employeeService;

*@Autowired*

private DepartmentService departmentService;

*@Autowired*

private SkillService skillService;

public static void main(String[] args) {

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

}

*@Override*

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

// Run your test methods from here

testAddEmployee();

testUpdateEmployee();

testGetEmployee();

employeeService.displayEmployeeDetails(1);

}

private void testAddEmployee() {

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

Employee employee = new Employee();

employee.setName("Bob");

employee.setSalary(70000.0);

employee.setPermanent(true);

employee.setDateOfBirth(LocalDate.*of*(1992, 5, 21));

Department department = departmentService.get(1);

employee.setDepartment(department);

employeeService.save(employee);

***LOGGER***.debug("Saved Employee: {}", employee);

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

}

private void testUpdateEmployee() {

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

Employee employee = employeeService.get(1);

Department department = departmentService.get(2); // different dept

employee.setDepartment(department);

employeeService.save(employee);

***LOGGER***.debug("Updated Employee: {}", employee);

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

}

private void testGetEmployee() {

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

Employee employee = employeeService.get(1);

***LOGGER***.debug("Employee: {}", employee);

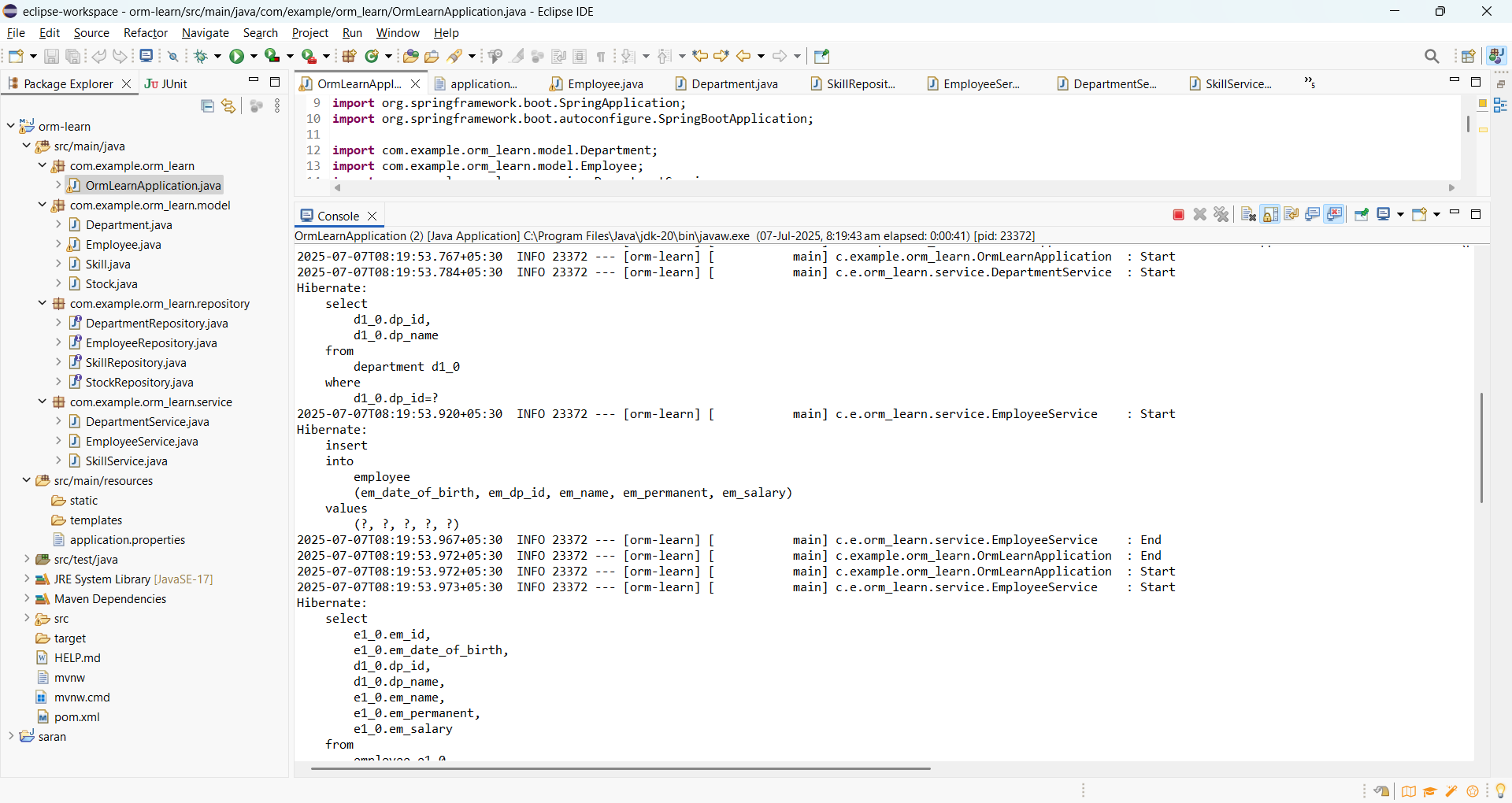
***LOGGER***.debug("Department: {}", employee.getDepartment());

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

}

}

**Output:**



**Hands on 5 : Implement one to many relationship between Employee and Department**   
  
**Department.java**

package com.cognizant.orm\_learn.model;

import java.util.List;

import java.util.Set;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.OneToMany;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "department")

public class Department {

public Department() {

super();

}

public Department(int id, String name, Set<Employee> employeeList) {

super();

this.id = id;

this.name = name;

this.employeeList = employeeList;

}

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

*@Column*(name = "dp\_id")

private int id;

*@Column*(name = "dp\_name")

private String name;

*@OneToMany*(mappedBy = "department", fetch = *FetchType*.***EAGER***)

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 void setEmployeeList(Set<Employee> employeeList) {

this.employeeList = employeeList;

}

public Set<Employee> getEmployeeList() {

return employeeList;

}

*@Override*

public String toString() {

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

}

}

**OrmLearnApplication.java**

package com.cognizant.orm\_learn;

import java.time.LocalDate;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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.orm\_learn.model.Department;

import com.cognizant.orm\_learn.model.Employee;

import com.cognizant.orm\_learn.service.DepartmentService;

import com.cognizant.orm\_learn.service.EmployeeService;

import com.cognizant.orm\_learn.service.SkillService;

*@SpringBootApplication*

public class OrmLearnApplication implements CommandLineRunner {

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

*@Autowired*

private EmployeeService employeeService;

*@Autowired*

private DepartmentService departmentService;

*@Autowired*

private SkillService skillService;

public static void main(String[] args) {

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

}

*@Override*

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

testGetDepartment(); // Comment out others like testAddEmployee()

}

private void testAddEmployee() {

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

Employee employee = new Employee();

employee.setName("Bob");

employee.setSalary(70000.0);

employee.setPermanent(true);

employee.setDateOfBirth(LocalDate.*of*(1992, 5, 21));

Department department = departmentService.get(1);

employee.setDepartment(department);

employeeService.save(employee);

***LOGGER***.debug("Saved Employee: {}", employee);

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

}

private void testUpdateEmployee() {

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

Employee employee = employeeService.get(1);

Department department = departmentService.get(2); // different dept

employee.setDepartment(department);

employeeService.save(employee);

***LOGGER***.debug("Updated Employee: {}", employee);

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

}

private void testGetEmployee() {

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

Employee employee = employeeService.get(1);

***LOGGER***.debug("Employee: {}", employee);

***LOGGER***.debug("Department: {}", employee.getDepartment());

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

}

private void testGetDepartment() {

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

Department department = departmentService.get(1); // pick department with >1 employee

***LOGGER***.debug("Department: {}", department);

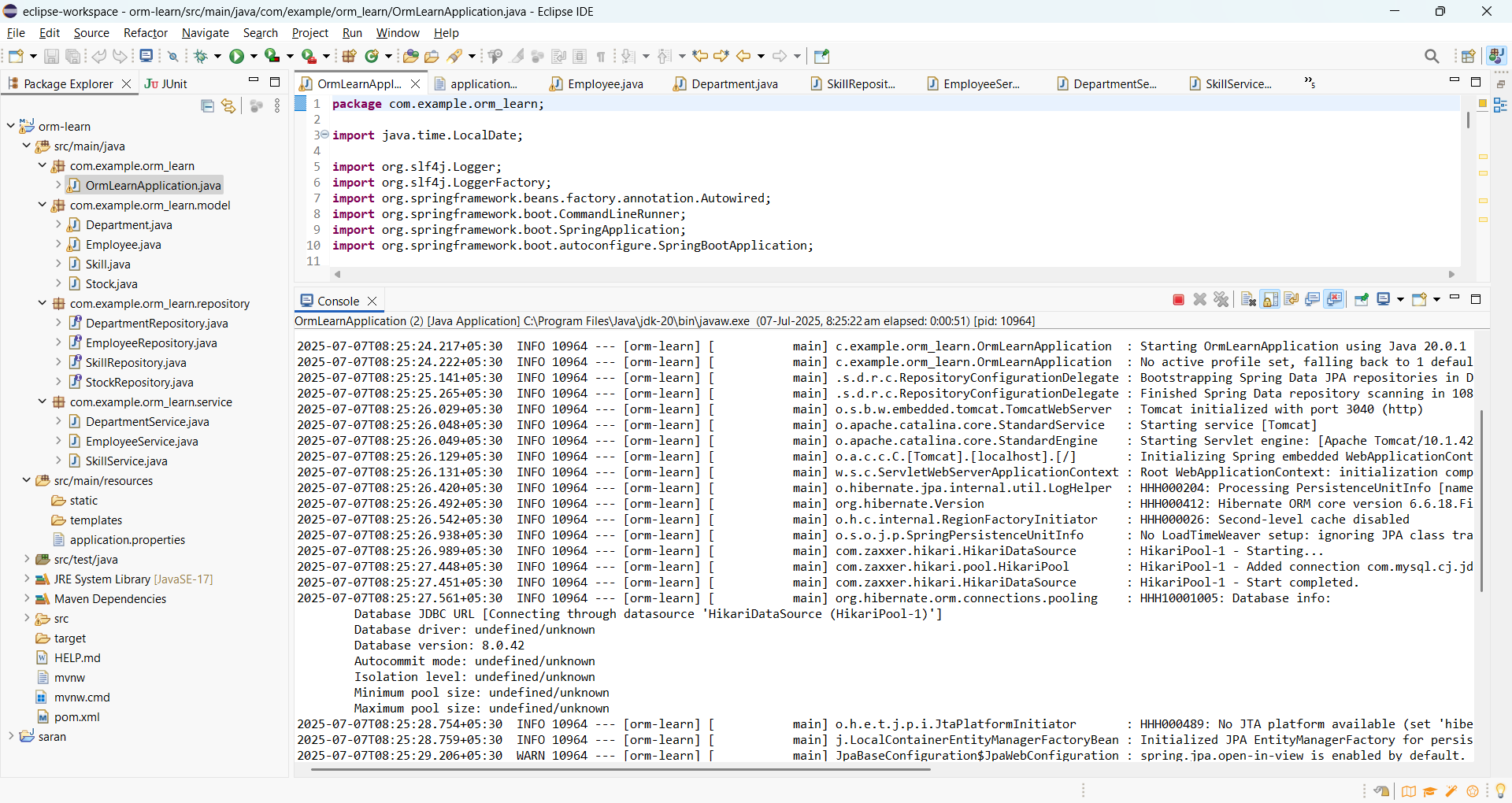
***LOGGER***.debug("Employee List: {}", department.getEmployeeList());

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

}

}

**Output:**



**Hands on 6 : Implement many to many relationship between Employee and Skill**   
  
**Employee.java**

package com.cognizant.orm\_learn.model;

import java.time.LocalDate;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.JoinTable;

import jakarta.persistence.ManyToMany;

import jakarta.persistence.ManyToOne;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "employee")

public class Employee {

public Employee() {

super();

}

public Employee(int id, String name, double salary, boolean permanent, LocalDate dateOfBirth, Department department,

Set<Skill> skillList) {

super();

this.id = id;

this.name = name;

this.salary = salary;

this.permanent = permanent;

this.dateOfBirth = dateOfBirth;

this.department = department;

this.skillList = skillList;

}

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

*@Column*(name = "em\_id")

private int id;

*@Column*(name = "em\_name")

private String name;

*@Column*(name = "em\_salary")

private double salary;

*@Column*(name = "em\_permanent")

private boolean permanent;

*@Column*(name = "em\_date\_of\_birth")

private LocalDate dateOfBirth;

*@ManyToOne*

*@JoinColumn*(name = "em\_dp\_id")

private Department department;

*@ManyToMany*(fetch = *FetchType*.***EAGER***)

*@JoinTable*(

name = "employee\_skill",

joinColumns = *@JoinColumn*(name = "es\_em\_id"),

inverseJoinColumns = *@JoinColumn*(name = "es\_sk\_id")

)

private Set<Skill> skillList;

public Set<Skill> getSkillList() {

return skillList;

}

public void setSkillList(Set<Skill> skillList) {

this.skillList = 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 LocalDate getDateOfBirth() {

return dateOfBirth;

}

public void setDateOfBirth(LocalDate dateOfBirth) {

this.dateOfBirth = dateOfBirth;

}

public Department getDepartment() {

return department;

}

public void setDepartment(Department department) {

this.department = department;

}

*@Override*

public String toString() {

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

}

}

**Department.java**

package com.cognizant.orm\_learn.model;

import java.util.List;

import java.util.Set;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.OneToMany;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "department")

public class Department {

public Department() {

super();

}

public Department(int id, String name, Set<Employee> employeeList) {

super();

this.id = id;

this.name = name;

this.employeeList = employeeList;

}

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

*@Column*(name = "dp\_id")

private int id;

*@Column*(name = "dp\_name")

private String name;

*@OneToMany*(mappedBy = "department", fetch = *FetchType*.***EAGER***)

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 void setEmployeeList(Set<Employee> employeeList) {

this.employeeList = employeeList;

}

public Set<Employee> getEmployeeList() {

return employeeList;

}

*@Override*

public String toString() {

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

}

}

**Skill.java**

package com.cognizant.orm\_learn.model;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.JoinTable;

import jakarta.persistence.ManyToMany;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "skill")

public class Skill {

public Skill() {

super();

}

public Skill(int id, String name, List<Employee> employeeList) {

super();

this.id = id;

this.name = name;

this.employeeList = employeeList;

}

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

*@Column*(name = "sk\_id")

private int id;

*@Column*(name = "sk\_name")

private String name;

*@ManyToMany*(mappedBy = "skillList")

private List<Employee> employeeList;

*@ManyToMany*(fetch = *FetchType*.***EAGER***)

*@JoinTable*(name = "employee\_skill",

joinColumns = *@JoinColumn*(name = "es\_em\_id"),

inverseJoinColumns = *@JoinColumn*(name = "es\_sk\_id"))

private Set<Skill> skillList = new HashSet<>();

public Set<Skill> getSkillList() {

return skillList;

}

public void setSkillList(Set<Skill> skillList) {

this.skillList = 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 List<Employee> getEmployeeList() {

return employeeList;

}

public void setEmployeeList(List<Employee> employeeList) {

this.employeeList = employeeList;

}

}

**OrmLearnApplication.java**

package com.cognizant.orm\_learn;

import java.time.LocalDate;

import java.util.Set;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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.orm\_learn.model.Department;

import com.cognizant.orm\_learn.model.Employee;

import com.cognizant.orm\_learn.model.Skill;

import com.cognizant.orm\_learn.service.DepartmentService;

import com.cognizant.orm\_learn.service.EmployeeService;

import com.cognizant.orm\_learn.service.SkillService;

*@SpringBootApplication*

public class OrmLearnApplication implements CommandLineRunner {

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

*@Autowired*

private EmployeeService employeeService;

*@Autowired*

private DepartmentService departmentService;

*@Autowired*

private SkillService skillService;

public static void main(String[] args) {

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

}

*@Override*

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

testAddSkillToEmployee();

}

private void testAddEmployee() {

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

Employee employee = new Employee();

employee.setName("Bob");

employee.setSalary(70000.0);

employee.setPermanent(true);

employee.setDateOfBirth(LocalDate.*of*(1992, 5, 21));

Department department = departmentService.get(1);

employee.setDepartment(department);

employeeService.save(employee);

***LOGGER***.debug("Saved Employee: {}", employee);

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

}

private void testUpdateEmployee() {

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

Employee employee = employeeService.get(1);

Department department = departmentService.get(2); // different dept

employee.setDepartment(department);

employeeService.save(employee);

***LOGGER***.debug("Updated Employee: {}", employee);

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

}

private void testGetEmployee() {

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

Employee employee = employeeService.get(1);

***LOGGER***.debug("Employee: {}", employee);

***LOGGER***.debug("Department: {}", employee.getDepartment());

***LOGGER***.debug("Skills: {}", employee.getSkillList());

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

}

private void testGetDepartment() {

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

Department department = departmentService.get(1); // pick department with >1 employee

***LOGGER***.debug("Department: {}", department);

***LOGGER***.debug("Employee List: {}", department.getEmployeeList());

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

}

private void testAddSkillToEmployee() {

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

Employee employee = employeeService.get(1); // existing employee id

Skill skill = skillService.get(1); // existing skill id

Set<Skill> skills = employee.getSkillList();

skills.add(skill);

employee.setSkillList(skills);

employeeService.save(employee);

***LOGGER***.debug("Updated Employee with new Skill: {}", employee);

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

}

}

**Output:**

