**Hands-On 1: Write queries on country table using Query Methods** :

**Country.java:**

package com.ormlearn.countryquerydemo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

*@Entity*

public class Country {

*@Id*

private String code;

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;

}

}

**CountryRepository.java:**

package com.ormlearn.countryquerydemo.repository;

import java.util.List;

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

import com.ormlearn.countryquerydemo.model.Country;

public interface CountryRepository extends JpaRepository<Country, String> {

List<Country> findByNameContaining(String text);

List<Country> findByNameContainingOrderByNameAsc(String text);

List<Country> findByNameStartingWith(String prefix);

}

**CountryquerydemoApplication.java:**

package com.ormlearn.countryquerydemo;

import com.ormlearn.countryquerydemo.model.Country;

import com.ormlearn.countryquerydemo.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 CountryquerydemoApplication implements CommandLineRunner {

*@Autowired*

private CountryRepository countryRepository;

public static void main(String[] args) {

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

}

*@Override*

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

System.***out***.println("=== Adding Sample Data ===");

addSampleCountries();

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

List<Country> list1 = countryRepository.findByNameContaining("ou");

list1.forEach(c -> System.***out***.println(c.getCode() + " - " + c.getName()));

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

List<Country> list2 = countryRepository.findByNameContainingOrderByNameAsc("ou");

list2.forEach(c -> System.***out***.println(c.getCode() + " - " + c.getName()));

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

List<Country> list3 = countryRepository.findByNameStartingWith("Z");

list3.forEach(c -> System.***out***.println(c.getCode() + " - " + c.getName()));

}

private void addSampleCountries() {

Country c1 = new Country(); c1.setCode("BV"); c1.setName("Bouvet Island");

Country c2 = new Country(); c2.setCode("DJ"); c2.setName("Djibouti");

Country c3 = new Country(); c3.setCode("GP"); c3.setName("Guadeloupe");

Country c4 = new Country(); c4.setCode("GS"); c4.setName("South Georgia and the South Sandwich Islands");

Country c5 = new Country(); c5.setCode("LU"); c5.setName("Luxembourg");

Country c6 = new Country(); c6.setCode("SS"); c6.setName("South Sudan");

Country c7 = new Country(); c7.setCode("TF"); c7.setName("French Southern Territories");

Country c8 = new Country(); c8.setCode("UM"); c8.setName("United States Minor Outlying Islands");

Country c9 = new Country(); c9.setCode("ZA"); c9.setName("South Africa");

Country c10 = new Country(); c10.setCode("ZM"); c10.setName("Zambia");

Country c11 = new Country(); c11.setCode("ZW"); c11.setName("Zimbabwe");

countryRepository.save(c1);

countryRepository.save(c2);

countryRepository.save(c3);

countryRepository.save(c4);

countryRepository.save(c5);

countryRepository.save(c6);

countryRepository.save(c7);

countryRepository.save(c8);

countryRepository.save(c9);

countryRepository.save(c10);

countryRepository.save(c11);

}}

**application.properties:**

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

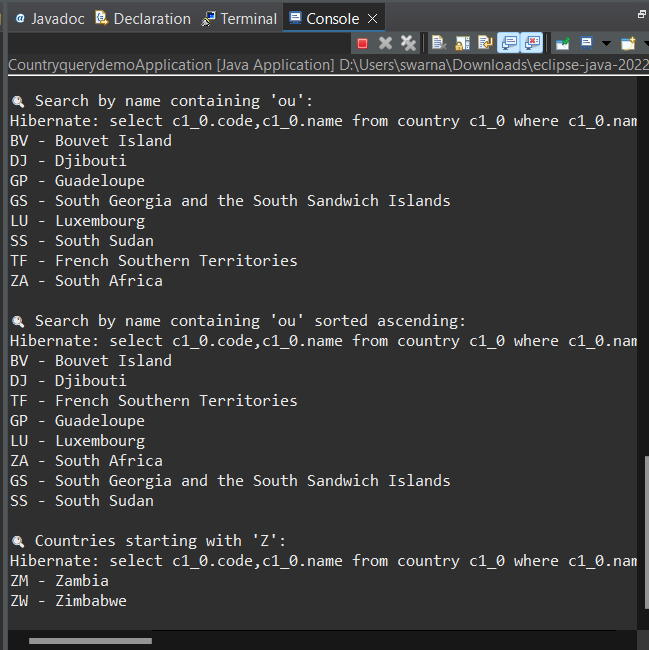
spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

spring.h2.console.enabled=true

spring.jpa.show-sql=true



**Hands-On: 2 Write queries on stock table using Query**

**StockquerydemoApplication.java:**

package com.ormlearn.stockquerydemo;

import com.ormlearn.stockquerydemo.model.Stock;

import com.ormlearn.stockquerydemo.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.math.BigDecimal;

import java.time.LocalDate;

import java.util.List;

*@SpringBootApplication*

public class StockquerydemoApplication implements CommandLineRunner {

*@Autowired*

private StockRepository stockRepository;

public static void main(String[] args) {

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

}

*@Override*

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

System.***out***.println("\n Facebook stocks in Sep 2019:");

List<Stock> fbStocks = stockRepository.findByCodeAndDateBetween(

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

fbStocks.forEach(this::printStock);

System.***out***.println("\nGoogle stocks where close > 1250:");

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

googlStocks.forEach(this::printStock);

System.***out***.println("\n Top 3 volume stocks:");

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

topVolume.forEach(this::printStock);

System.***out***.println("\n Netflix lowest 3 close:");

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

lowNetflix.forEach(this::printStock);

}

private void printStock(Stock stock) {

System.***out***.printf("%-6s | %-10s | %-8s | %-8s | %d\n",

stock.getCode(), stock.getDate(), stock.getOpen(), stock.getClose(), stock.getVolume());

}

}

**Stock.java:**

package com.ormlearn.stockquerydemo.model;

import jakarta.persistence.\*;

import java.math.BigDecimal;

import java.time.LocalDate;

*@Entity*

*@Table*(name = "stock")

public class Stock {

*@Id*

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

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

private int id;

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

private String code;

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

private LocalDate date;

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

private BigDecimal open;

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

private BigDecimal close;

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

private long 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 LocalDate getDate() { return date; }

public void setDate(LocalDate 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 long getVolume() { return volume; }

public void setVolume(long volume) { this.volume = volume; }

}

**StockRepository.java:**

package com.ormlearn.stockquerydemo.repository;

import com.ormlearn.stockquerydemo.model.Stock;

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

import java.math.BigDecimal;

import java.time.LocalDate;

import java.util.List;

public interface StockRepository extends JpaRepository<Stock, Integer> {

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

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

List<Stock> findTop3ByOrderByVolumeDesc();

List<Stock> findTop3ByCodeOrderByCloseAsc(String code);

}

**application.properties:**

spring.datasource.url=jdbc:h2:mem:ormlearn

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.h2.console.enabled=true

spring.jpa.hibernate.ddl-auto=update

**schema.sql:**

CREATE TABLE stock (

st\_code VARCHAR(10),

st\_date DATE,

st\_open DOUBLE,

st\_close DOUBLE,

st\_volume BIGINT

);

**data.sql:**

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

('GOOGL', '2019-04-29', 1280.51, 1296.20, 3618400),

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

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

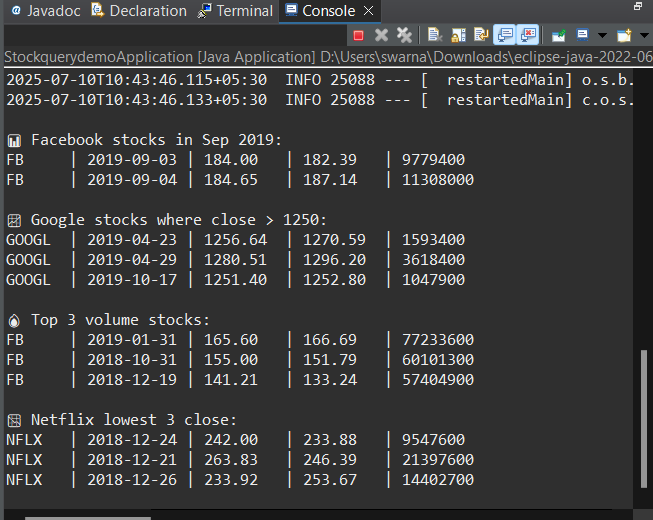
('FB', '2018-10-31', 155.00, 151.79, 60101300),

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

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

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

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



**Hands-On 3: Create payroll tables and bean mapping** :

**PayrollApplication.java:**

package com.example.payroll;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.domain.EntityScan;

import org.springframework.context.annotation.ComponentScan;

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

*@SpringBootApplication*

*@ComponentScan*(basePackages = "com.example.payroll")

*@EntityScan*("com.example.payroll.model")

*@EnableJpaRepositories*("com.example.payroll.repository")

public class PayrollApplication {

public static void main(String[] args) {

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

}

}

**Department.java:**

package com.example.payroll.model;

import jakarta.persistence.\*;

*@Entity*

*@Table*(name = "department")

public class Department {

*@Id*

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

private int id;

*@Column*(name = "name")

private String name;

public int getId() {

return id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

*@Override*

public String toString() {

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

}}

**Employee.java:**

package com.example.payroll.model;

import jakarta.persistence.\*;

import java.util.\*;

*@Entity*

*@Table*(name = "employee")

public class Employee {

*@Id*

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

private int id;

*@Column*(name = "name")

private String name;

*@Column*(name = "salary")

private double salary;

*@Column*(name = "permanent")

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 List<Skill> skills = new ArrayList<>();

public int getId() {

return 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> getSkills() {

return skills;

}

public void setSkills(List<Skill> skills) {

this.skills = skills;

}

*@Override*

public String toString() {

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

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

}}

**Skill.java:**

package com.example.payroll.model;

import jakarta.persistence.\*;

*@Entity*

*@Table*(name = "skill")

public class Skill {

*@Id*

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

private int id;

*@Column*(name = "name")

private String name;

public int getId() {

return id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

*@Override*

public String toString() {

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

}}

**DepartmentRepository.java:**

package com.example.payroll.model;

import jakarta.persistence.\*;

*@Entity*

*@Table*(name = "skill")

public class Skill {

*@Id*

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

private int id;

*@Column*(name = "name")

private String name;

public int getId() {

return id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

*@Override*

public String toString() {

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

}}

**EmployeeRepository.java:**

package com.example.payroll.repository;

import com.example.payroll.model.Employee;

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**SkillRepository.java:**

package com.example.payroll.repository;

import com.example.payroll.model.Skill;

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

public interface SkillRepository extends JpaRepository<Skill, Integer> {

}

**DataLoader.java:**

package com.example.payroll.repository;

import com.example.payroll.model.Skill;

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

public interface SkillRepository extends JpaRepository<Skill, Integer> {

}

**Schema.sql:**

CREATE TABLE department (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50)

);

CREATE TABLE employee (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

salary DOUBLE,

permanent BOOLEAN,

date\_of\_birth DATE,

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES department(id)

);

CREATE TABLE skill (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100)

);

CREATE TABLE employee\_skill (

employee\_id INT,

skill\_id INT,

PRIMARY KEY (employee\_id, skill\_id),

FOREIGN KEY (employee\_id) REFERENCES employee(id),

FOREIGN KEY (skill\_id) REFERENCES skill(id)

);

**data.sql:**

INSERT INTO department (name) VALUES ('HR'), ('Finance');

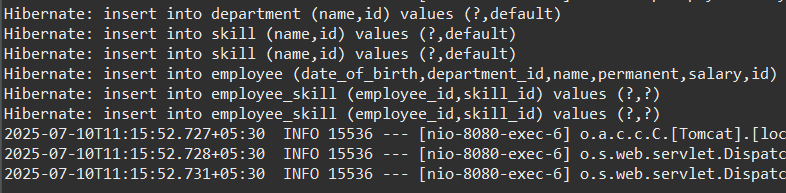
INSERT INTO employee (name, salary, permanent, date\_of\_birth, department\_id)

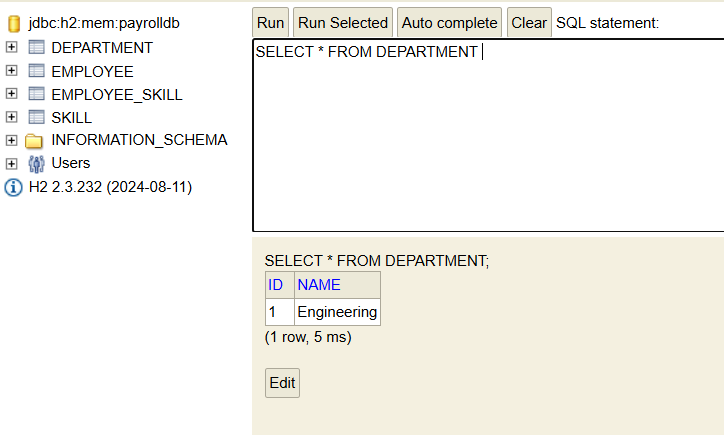
VALUES ('Alice', 60000.0, true, '1990-03-15', 1),

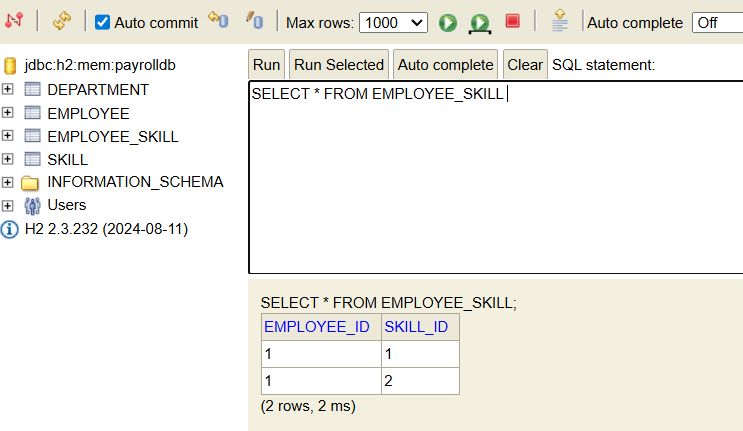
('Bob', 45000.0, false, '1992-08-25', 2);

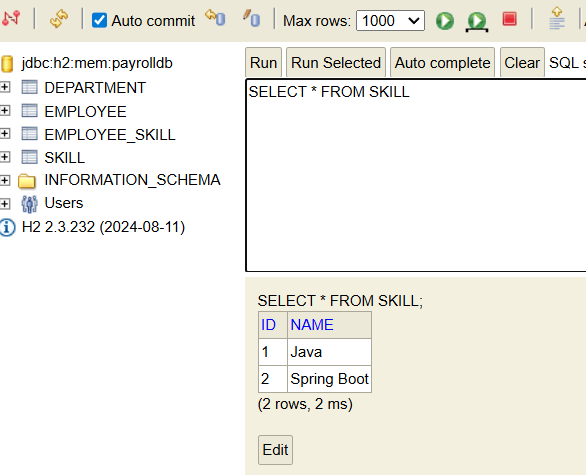
INSERT INTO skill (name) VALUES ('Java'), ('Python');

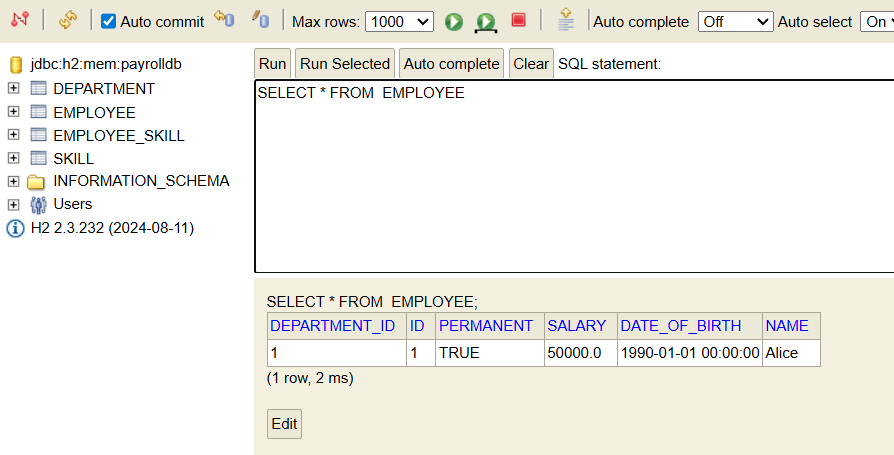
INSERT INTO employee\_skill (employee\_id, skill\_id) VALUES (1, 1), (2, 2);











**Hands-On 4: Implement many to one relationship between Employee and Department** :

**PayrollAppplication.java:**

package com.example.payroll;

import com.example.payroll.model.Department;

import com.example.payroll.model.Employee;

import com.example.payroll.service.DepartmentService;

import com.example.payroll.service.EmployeeService;

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 java.text.SimpleDateFormat;

*@SpringBootApplication*

public class PayrollApplication implements CommandLineRunner {

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

*@Autowired*

private DepartmentService departmentService;

*@Autowired*

private EmployeeService employeeService;

public static void main(String[] args) {

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

}

*@Override*

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

testAddEmployee();

testGetEmployee();

}

private void testAddEmployee() throws Exception {

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

Department department = departmentService.get(1);

Employee employee = new Employee();

employee.setName("Alice");

employee.setSalary(50000);

employee.setPermanent(true);

employee.setDateOfBirth(sdf.parse("1990-01-01"));

employee.setDepartment(department);

employeeService.save(employee);

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

}

private void testGetEmployee() {

Employee employee = employeeService.get(1);

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

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

}

}

Employee.java:

package com.example.payroll.model;

import jakarta.persistence.\*;

import java.util.Date;

import java.util.List;

import java.util.ArrayList;

*@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")

*@Temporal*(*TemporalType*.***DATE***)

private Date dateOfBirth;

*@ManyToOne*

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

private Department department;

*@ManyToMany*

*@JoinTable*(

name = "employee\_skill",

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

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

)

private List<Skill> skills = new ArrayList<>();

public List<Skill> getSkills() {

return skills;

}

public void setSkills(List<Skill> skills) {

this.skills = skills;

}

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; }

}

**Department.java:**

package com.example.payroll.model;

import jakarta.persistence.\*;

*@Entity*

*@Table*(name = "department")

public class Department {

*@Id*

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

private int id;

*@Column*

private String name;

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; }

}

**DepartmentRepository.java:**

package com.example.payroll.repository;

import com.example.payroll.model.Department;

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

public interface DepartmentRepository extends JpaRepository<Department, Integer> {}

**EmployeeRepository.java:**

package com.example.payroll.repository;

import com.example.payroll.model.Employee;

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

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

**DepartmentService.java:**

package com.example.payroll.service;

import com.example.payroll.model.Department;

import com.example.payroll.repository.DepartmentRepository;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

*@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).orElse(null);

}

*@Transactional*

public void save(Department department) {

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

departmentRepository.save(department);

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

}}

**EmployeeService.java:**

package com.example.payroll.service;

import com.example.payroll.model.Employee;

import com.example.payroll.repository.EmployeeRepository;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

*@Service*

public class EmployeeService {

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

*@Autowired*

private EmployeeRepository employeeRepository;

*@Transactional*

public Employee get(int id) {

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

return employeeRepository.findById(id).orElse(null);

}

*@Transactional*

public void save(Employee employee) {

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

employeeRepository.save(employee);

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

}

}

**application.properties:**

spring.datasource.url=jdbc:h2:mem:payrolldb

spring.datasource.driverClassName=org.h2.Driver

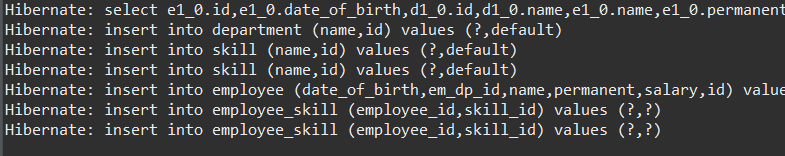
spring.datasource.username=sa

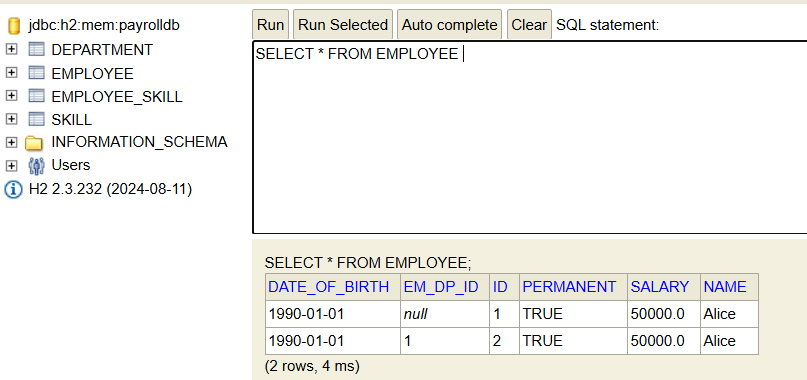
spring.datasource.password=

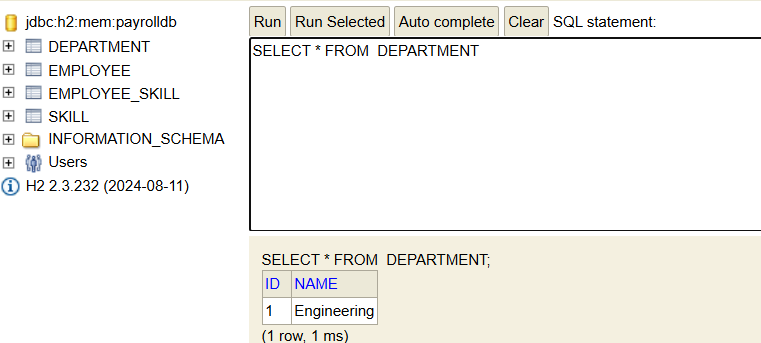
spring.jpa.hibernate.ddl-auto=create

spring.h2.console.enabled=true

spring.jpa.show-sql=true







**Hands on 5: Implement one to many relationship between Employee and Department** & **Hands on 6:** **Implement many to many relationship between Employee and Skill** :

**PayrollApplication.java:**

package com.example.payroll;

import com.example.payroll.model.Department;

import com.example.payroll.model.Employee;

import com.example.payroll.model.Skill;

import com.example.payroll.service.DepartmentService;

import com.example.payroll.service.EmployeeService;

import com.example.payroll.service.SkillService;

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 java.text.SimpleDateFormat;

import java.util.HashSet;

*@SpringBootApplication*

public class PayrollApplication implements CommandLineRunner {

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

*@Autowired*

private DepartmentService departmentService;

*@Autowired*

private EmployeeService employeeService;

*@Autowired*

private SkillService skillService;

public static void main(String[] args) {

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

}

*@Override*

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

testAddEmployee();

testAddSkillToEmployee();

}

private void testAddEmployee() throws Exception {

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

Department department = departmentService.get(1);

Employee employee = new Employee();

employee.setName("Alice");

employee.setSalary(50000);

employee.setPermanent(true);

employee.setDateOfBirth(sdf.parse("1990-01-01"));

employee.setDepartment(department);

employeeService.save(employee);

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

}

private void testGetDepartment() {

Department department = departmentService.get(1);

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

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

}

private void testAddSkillToEmployee() {

Employee employee = employeeService.get(1);

if (employee == null) {

***LOGGER***.error("Employee with ID 1 not found!");

return;

}

Skill skill = skillService.get(1);

if (skill == null) {

***LOGGER***.error("Skill with ID 1 not found!");

Return;

}

if (employee.getSkillList() == null) {

employee.setSkillList(new HashSet<>());

}

employee.getSkillList().add(skill);

employeeService.save(employee);

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

}

private void testGetEmployee() {

Employee employee = employeeService.get(1);

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

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

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

}}

**Department.java:**

package com.example.payroll.model;

import jakarta.persistence.\*;

import java.util.Set;

*@Entity*

*@Table*(name = "department")

public class Department {

*@Id*

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

private int id;

*@Column*

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 Set<Employee> getEmployeeList() { return employeeList; }

public void setEmployeeList(Set<Employee> employeeList) { this.employeeList = employeeList; }

}

**Employee.java:**

package com.example.payroll.model;

import jakarta.persistence.\*;

import java.util.Date;

import java.util.Set;

import java.util.HashSet;

*@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")

*@Temporal*(*TemporalType*.***DATE***)

private Date 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 = 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 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; }

}

**Skill.java:**

package com.example.payroll.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; }}

**DataLoader.java:**

package com.example.payroll.runner;

import com.example.payroll.model.\*;

import com.example.payroll.repository.\*;

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

import java.text.SimpleDateFormat;

import java.util.\*

*@Component*

public class DataLoader implements CommandLineRunner {

private final EmployeeRepository employeeRepository;

private final DepartmentRepository departmentRepository;

private final SkillRepository skillRepository;

public DataLoader(EmployeeRepository e, DepartmentRepository d, SkillRepository s) {

this.employeeRepository = e;

this.departmentRepository = d;

this.skillRepository = s;

}

*@Override*

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

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

Department d1 = new Department();

d1.setName("Engineering");

departmentRepository.save(d1);

Skill s1 = new Skill();

s1.setName("Java");

Skill s2 = new Skill();

s2.setName("Spring Boot");

skillRepository.saveAll(Arrays.*asList*(s1, s2));

Employee e1 = new Employee();

e1.setName("Alice");

e1.setSalary(50000);

e1.setPermanent(true);

e1.setDateOfBirth(sdf.parse("1990-01-01"));

e1.setDepartment(d1);

e1.getSkillList().addAll(Arrays.*asList*(s1, s2));

employeeRepository.save(e1);

}}

**DepartmentService.java:**

package com.example.payroll.service;

import com.example.payroll.model.Department;

import com.example.payroll.repository.DepartmentRepository;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

*@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).orElse(null);

}

*@Transactional*

public void save(Department department) {

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

departmentRepository.save(department);

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

}}

**EmployeeService.java:**

package com.example.payroll.service;

import com.example.payroll.model.Employee;

import com.example.payroll.repository.EmployeeRepository;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

*@Service*

public class EmployeeService {

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

*@Autowired*

private EmployeeRepository employeeRepository;

*@Transactional*

public Employee get(int id) {

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

return employeeRepository.findById(id).orElse(null);

}

*@Transactional*

public void save(Employee employee) {

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

employeeRepository.save(employee);

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

}}

**SkillService.java:**

package com.example.payroll.service;

import com.example.payroll.model.Skill;

import com.example.payroll.repository.SkillRepository;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

*@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).orElse(null);

}

*@Transactional*

public void save(Skill skill) {

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

skillRepository.save(skill);

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

}}

**DepartmentRepository.java:**

package com.example.payroll.repository;

import com.example.payroll.model.Department;

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

public interface DepartmentRepository extends JpaRepository<Department, Integer> {}

**EmployeeRepository.java:**

package com.example.payroll.repository;

import com.example.payroll.model.Employee;

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

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

**SkillRepository.java:**

package com.example.payroll.repository;

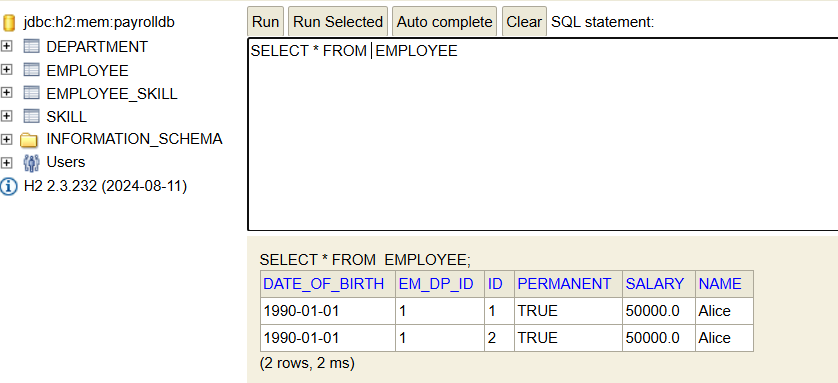
import com.example.payroll.model.Skill;

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

public interface SkillRepository extends JpaRepository<Skill, Integer> {

}

One to Many:



Many to Many:

