**Spring Data JPA 2 – Hands-On Assignment**

Name: Divyansh Tiwari

Superset ID: 6394658

**Hands-on 1: Query Methods on Country Table**

Goal: Implement search queries using Spring Data JPA’s method naming conventions

Steps followed:

1. Set up CountryRepository.
2. Query search name by keyword[partial name].

This is done via the following:

*List<Country> findByNameContaining(String keyword);*

1. Tested the query inside **OrmLearnApplication.java**

*private static void testQueryMethods() {*

*LOGGER.info("Start");*

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

*LOGGER.debug("Containing 'ou': {}", containing);*

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

*LOGGER.debug("Sorted containing 'ou': {}", sorted);*

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

*LOGGER.debug("Starting with Z: {}", startingWithZ);*

*LOGGER.info("End");*

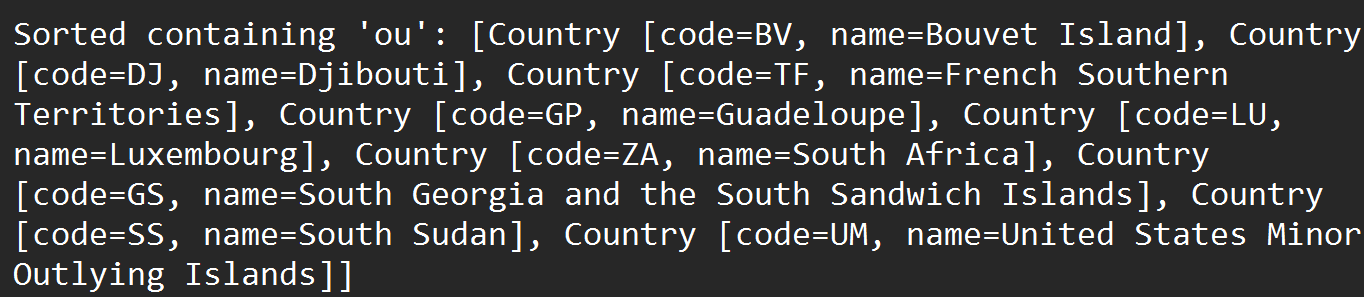
*}*

Repository Injection: *private static CountryRepository countryRepository;*

In main():

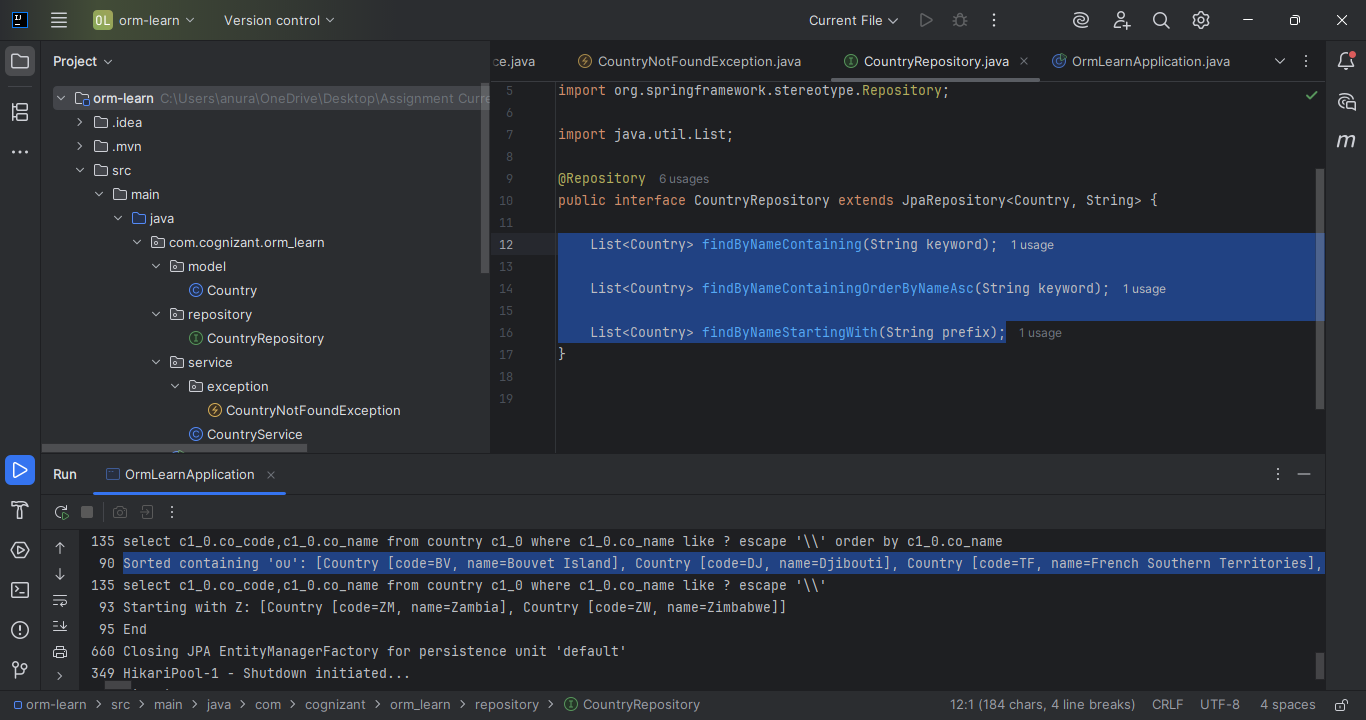
*countryRepository = context.getBean(CountryRepository.class);*

*testQueryMethods();*

OUTPUT:

Values returned by query

OUTPUT in IDE:



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

Goal: Create a table called Stock. Create a StockRepository and Implement queries.

Steps followed:

1. Created a table called stock in the SQL database.
2. Populated the table with sample data. The data is presented below.

This data was used in place of stock-data.csv because it was not available in the repository provided. This is the filler data used in place of stock-data.csv.

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

*('FB', '2019-09-01', 185.3, 186.2, 1500000),*

*('FB', '2019-09-20', 190.1, 192.8, 1750000),*

*('FB', '2019-09-30', 193.0, 194.5, 1800000),*

*('GOOGL', '2020-01-15', 1340.0, 1355.2, 1200000),*

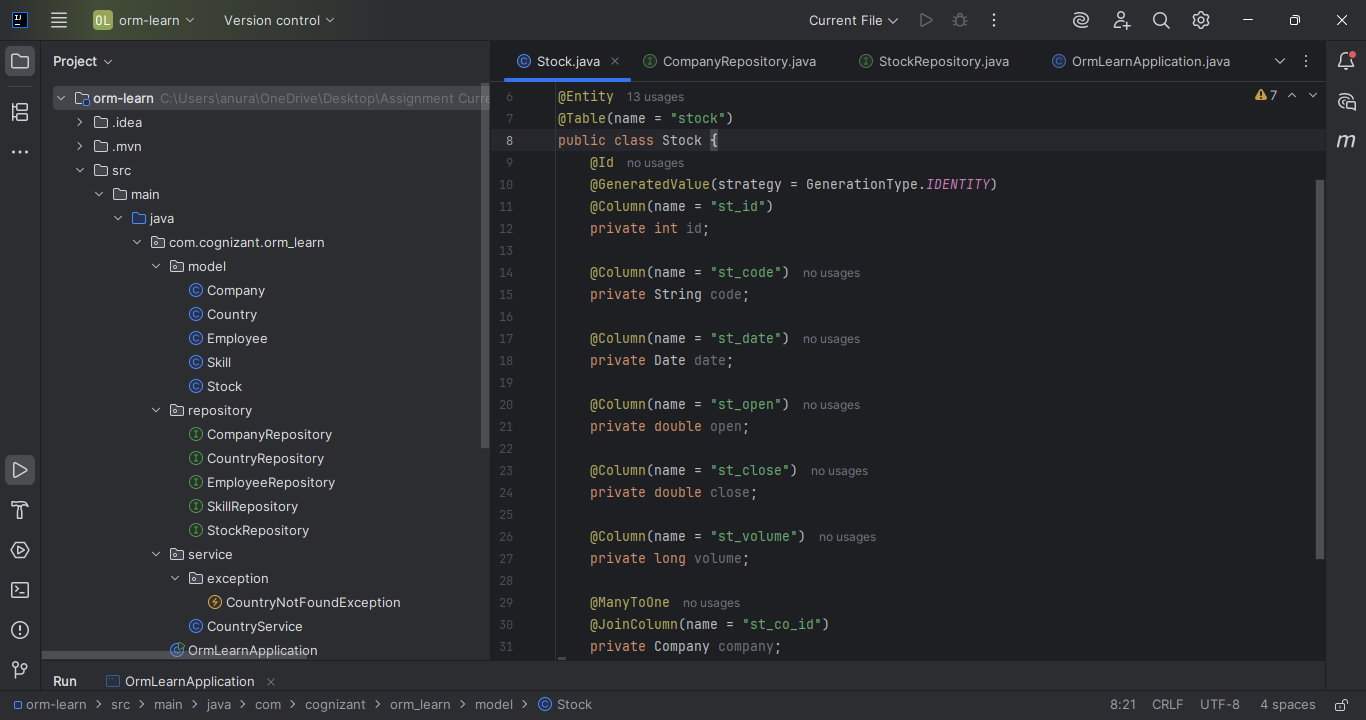
*('GOOGL', '2020-02-01', 1375.0, 1380.0, 1250000),*

*('GOOGL', '2020-02-20', 1420.1, 1410.0, 1100000),*

*('NFLX', '2019-11-10', 290.0, 280.5, 980000),*

*('NFLX', '2019-12-05', 295.0, 270.0, 960000);*

1. Created a Stock Entity called Stock.java

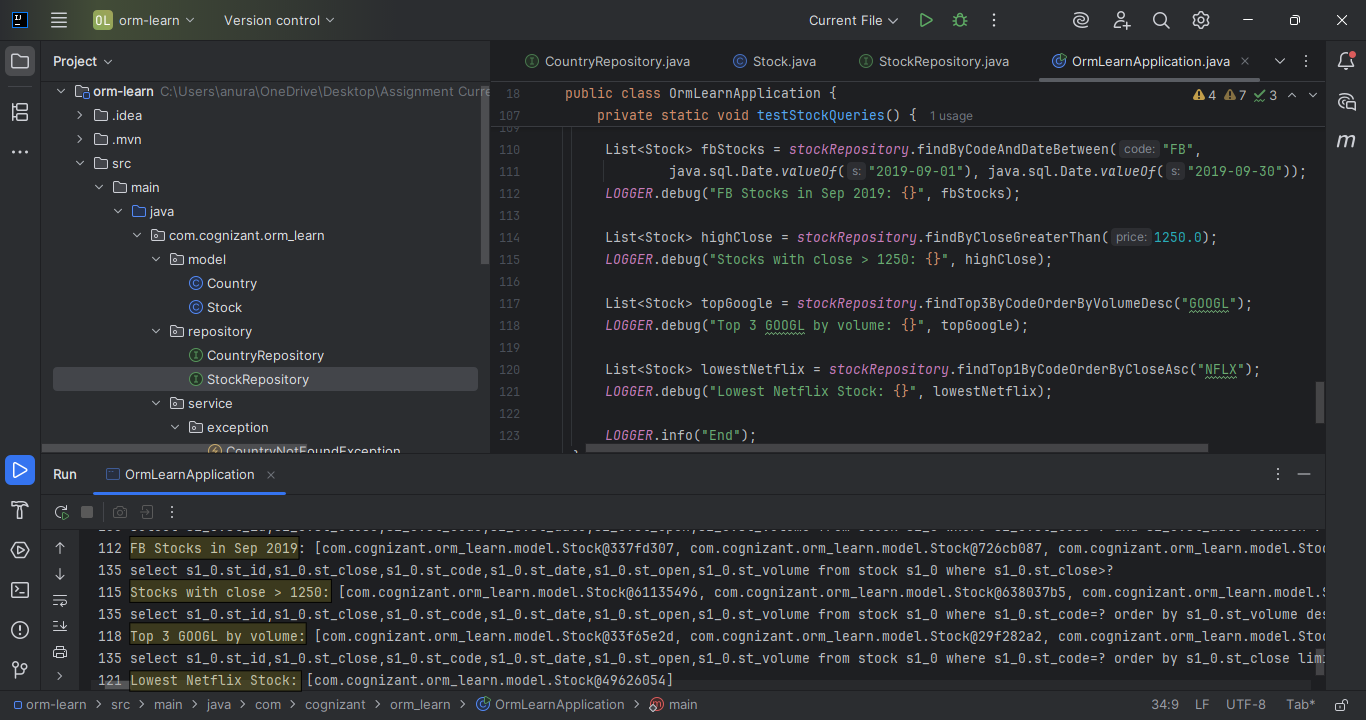


1. Created StockRepository in the repository package.
2. Tested Stock in OrmLearnApplication.java  
   *private static StockRepository stockRepository;*

In main():

*stockRepository = context.getBean(StockRepository.class);*

*testStockQueries();*

Test methods and Output:

Output is highlighted in the console for clarity.

**Hands-On 3: Creating Company and Stock Relationship (One-to-Many)**

Goal: Model a **One-to-Many relationship** between Company and Stock, where one company has many stocks, but each stock has only one company.

Steps followed:

1. Created Company table in the SQL Database.

*CREATE TABLE company (*

*co\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*co\_name VARCHAR(50)*

*);*

*INSERT INTO company (co\_name) VALUES ('Facebook'), ('Google'), ('Netflix');*

1. Modified Stock table to add Foreign key

*ALTER TABLE stock ADD COLUMN st\_co\_id INT;*

*ALTER TABLE stock ADD CONSTRAINT fk\_stock\_company*

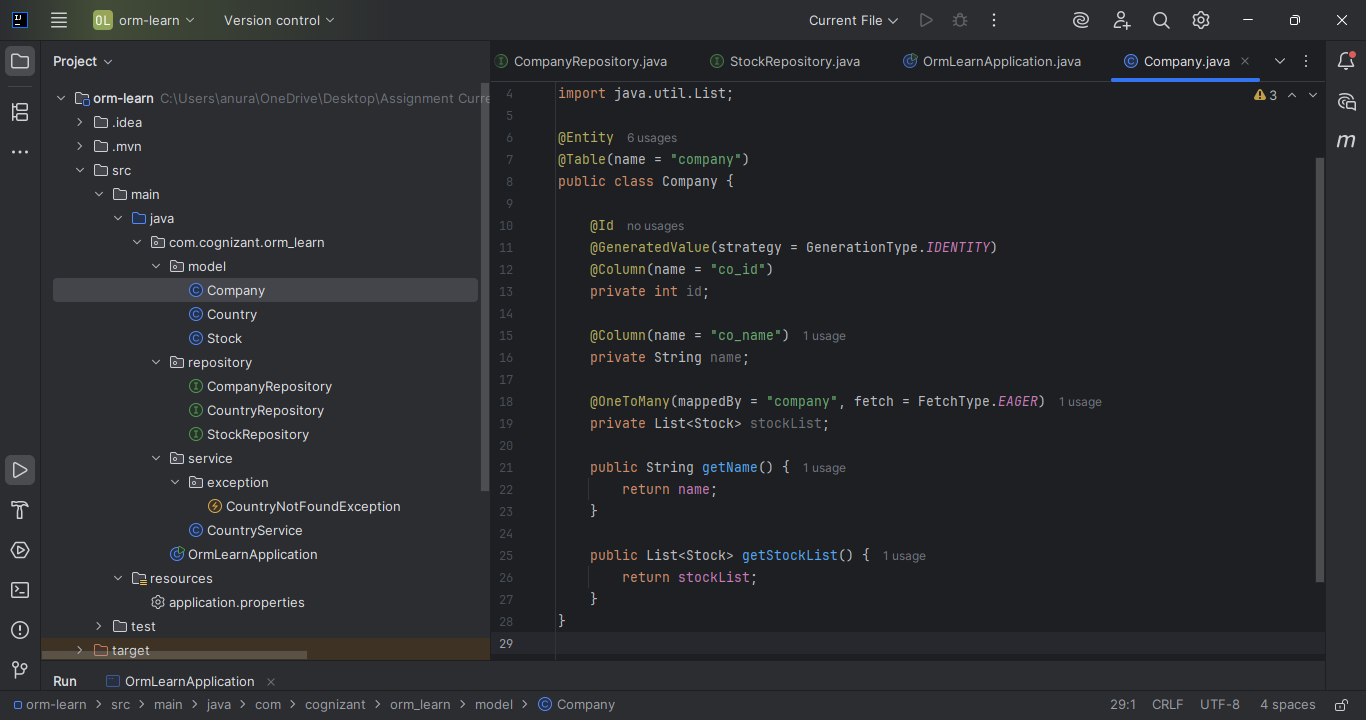
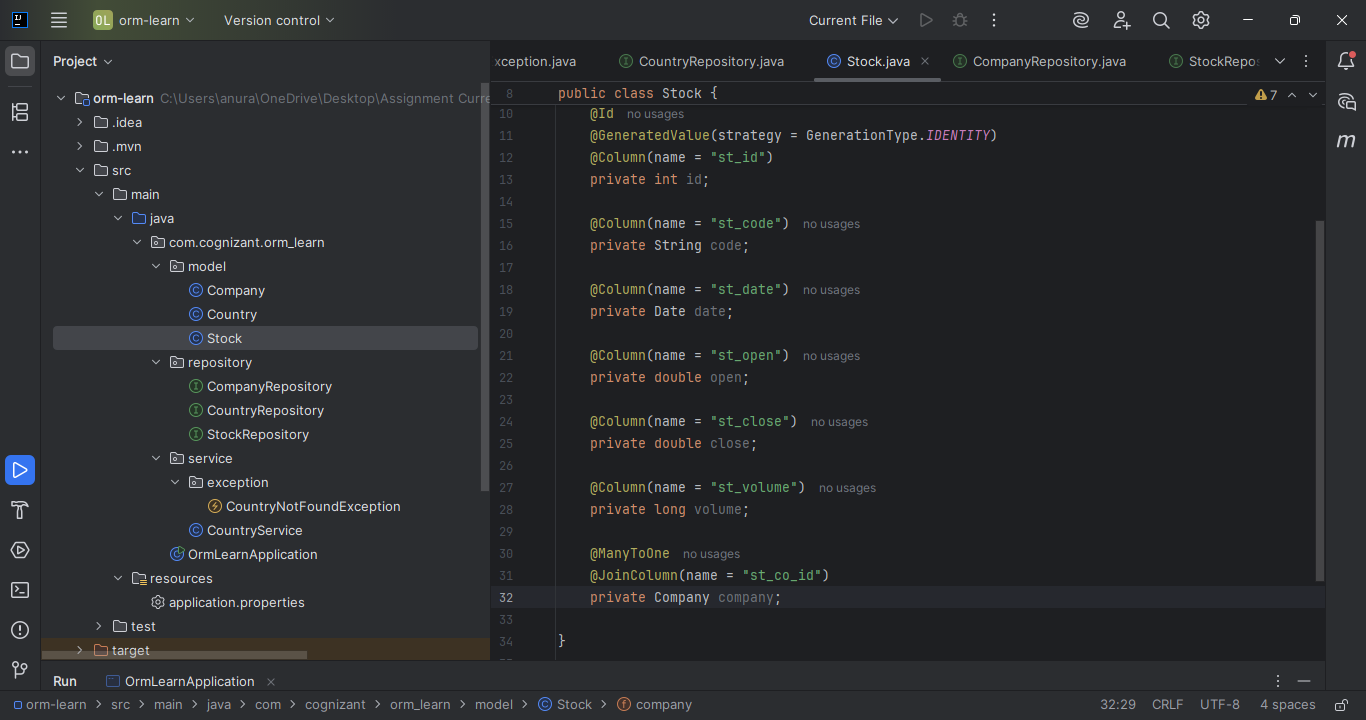
*FOREIGN KEY (st\_co\_id) REFERENCES company(co\_id);*

1. Updated Stocks table to link with Company

*UPDATE stock SET st\_co\_id = 1 WHERE st\_code = 'FB';*

*UPDATE stock SET st\_co\_id = 2 WHERE st\_code = 'GOOGL';*

*UPDATE stock SET st\_co\_id = 3 WHERE st\_code = 'NFLX';*

1. Created a Company entity inside model package
2. Updated Stocks entity to link with Company
3. Created CompanyRepository inside repository package

Inside CompanyRepository

*package com.cognizant.orm\_learn.repository;*

*import com.cognizant.orm\_learn.model.Company;*

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

*import org.springframework.stereotype.Repository;*

*@Repository*

*public interface CompanyRepository extends JpaRepository<Company, Integer> {*

*}*

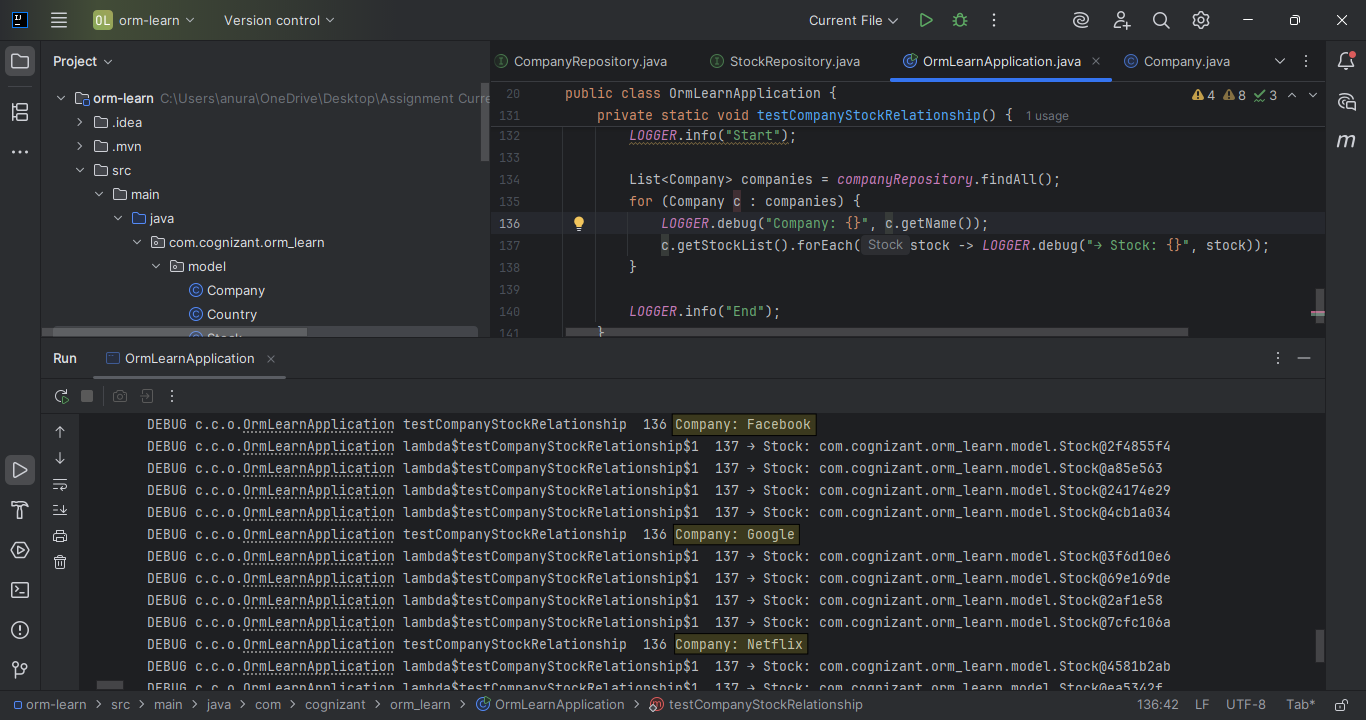
1. Added tests to OrmLearnApplication

*private static CompanyRepository companyRepository;*

In main():

*companyRepository = context.getBean(CompanyRepository.class);*

*testCompanyStockRelationship();*

TEST OUTPUT: **

Test outputs are highlighted for clarity.

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

Goal: Setting up One to many relationship between Company has many employees, and an employee belongs to one company.

Steps followed:

1. Created employee table in the SQL Database.

*CREATE TABLE employee (*

*em\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*em\_name VARCHAR(50),*

*em\_salary DOUBLE,*

*em\_permanent BOOLEAN,*

*em\_date\_of\_birth DATE,*

*em\_co\_id INT,*

*FOREIGN KEY (em\_co\_id) REFERENCES company(co\_id)*

*);*

1. Inserted Sample employees as below

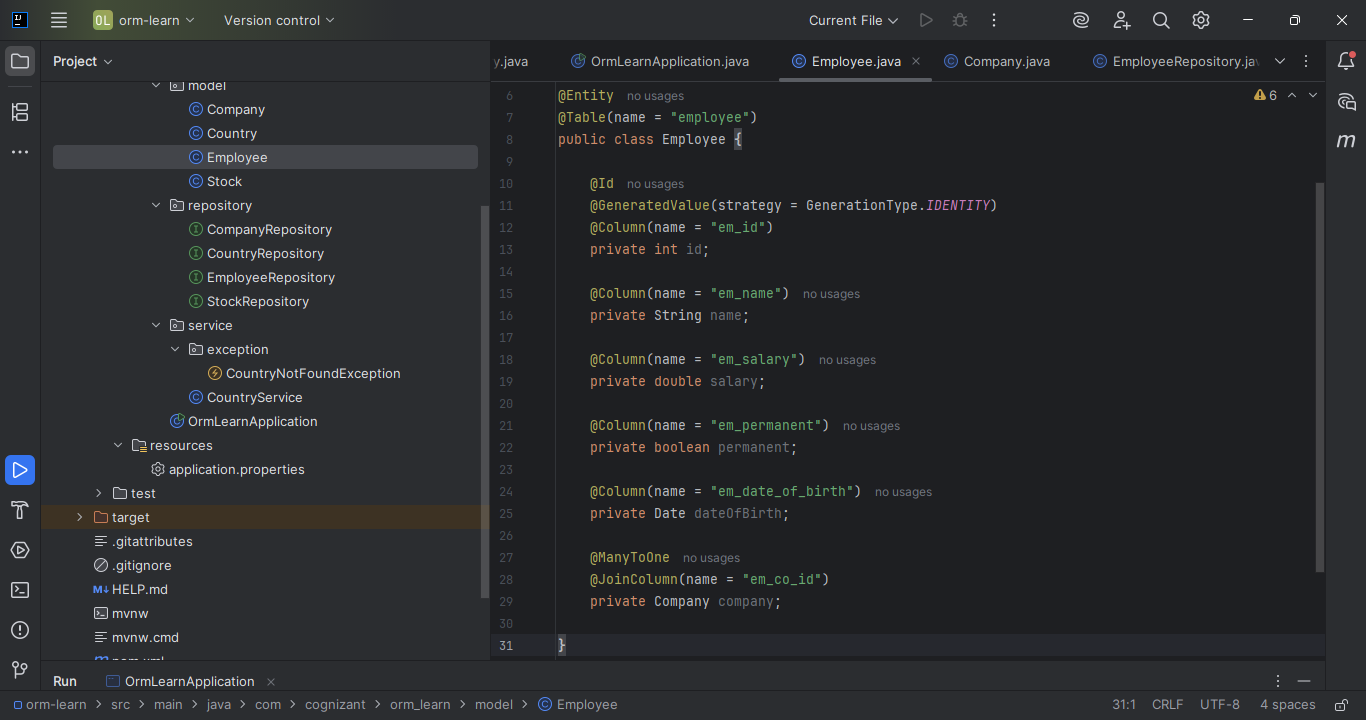
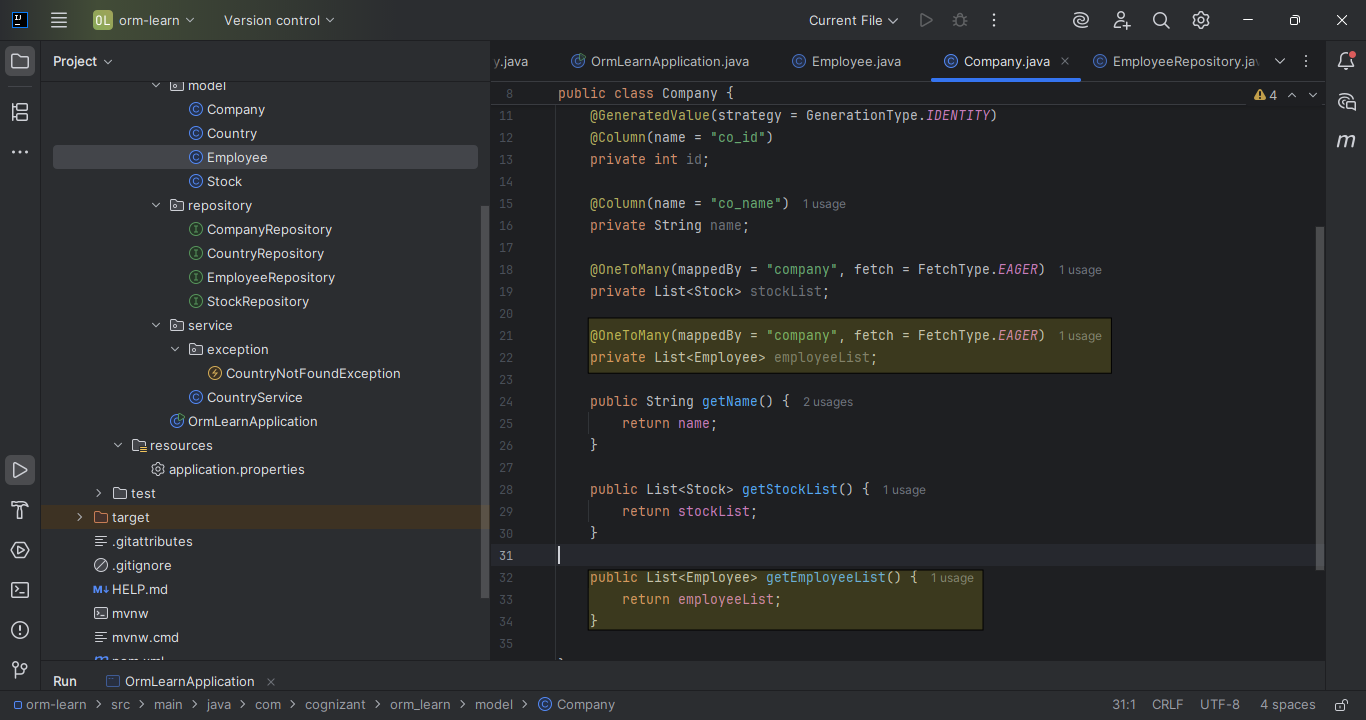
*INSERT INTO employee (em\_name, em\_salary, em\_permanent, em\_date\_of\_birth, em\_co\_id) VALUES*

*('Alice', 50000, TRUE, '1990-05-10', 1),*

*('Bob', 60000, FALSE, '1988-11-23', 1),*

*('Charlie', 70000, TRUE, '1992-01-15', 2),*

*('Dave', 55000, TRUE, '1995-09-09', 3);*

1. Created a Employee entity inside model package
2. Updated Company entity to include Employee List
3. Created EmployeeRepository inside repository package

Inside CompanyRepository

*package com.cognizant.orm\_learn.repository;*

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

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

*import org.springframework.stereotype.Repository;*

*@Repository*

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

*}*

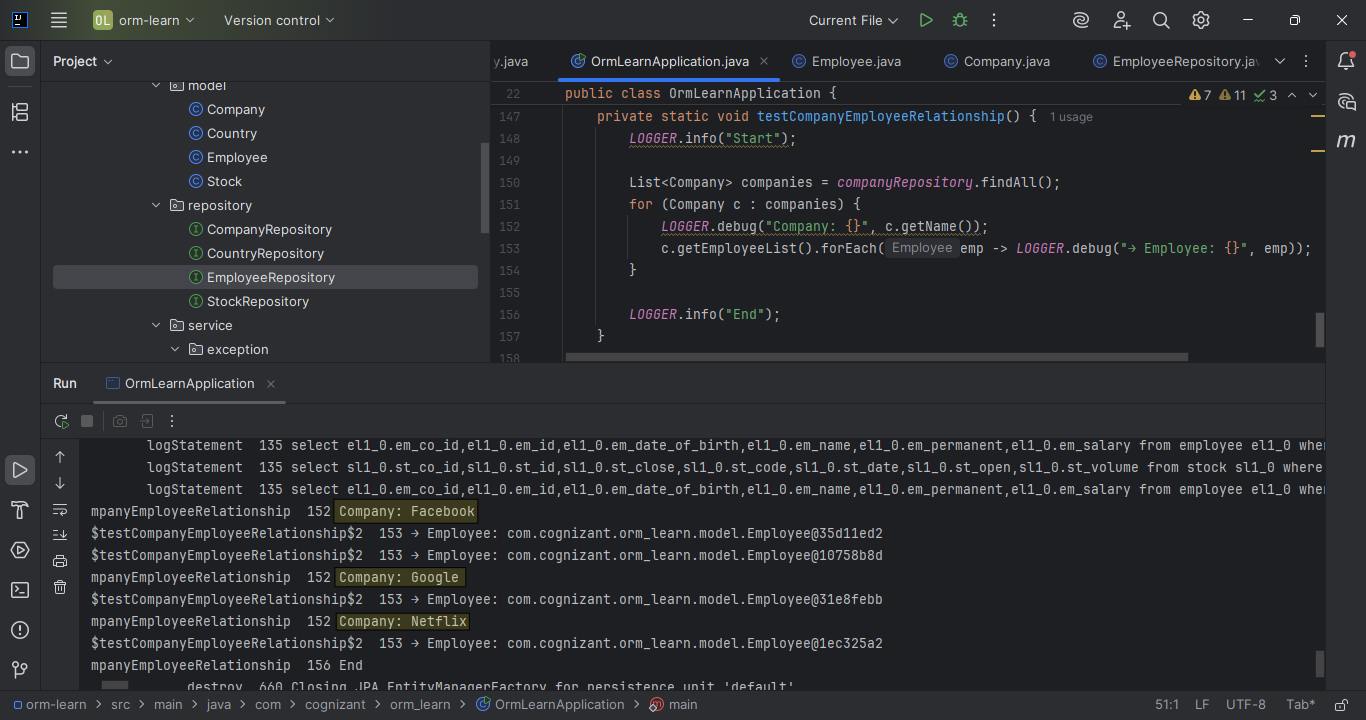
1. Added tests to OrmLearnApplication

*private static EmployeeRepository employeeRepository;*

In main():

*employeeRepository = context.getBean(EmployeeRepository.class);*

*testCompanyEmployeeRelationship();*

TEST OUTPUT: **

Test outputs are highlighted for clarity.

**Hands-On 5: Implement one to many relationship between Employee and Department**

Goal: Establishing a many to many relationship as follows, an employee can have many skills, and a skill may belong to multiple employees.

Steps followed:

1. Created skill table in the SQL Database.

*CREATE TABLE skill (*

*sk\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*sk\_name VARCHAR(50)*

1. Inserted Sample skills as below

*INSERT INTO skill (sk\_name) VALUES ('Java'), ('Python'), ('SQL'), ('Spring Boot');*

1. Created a join table employee\_skill

*CREATE TABLE employee\_skill (*

*es\_em\_id INT,*

*es\_sk\_id INT,*

*FOREIGN KEY (es\_em\_id) REFERENCES employee(em\_id),*

*FOREIGN KEY (es\_sk\_id) REFERENCES skill(sk\_id),*

*PRIMARY KEY (es\_em\_id, es\_sk\_id)*

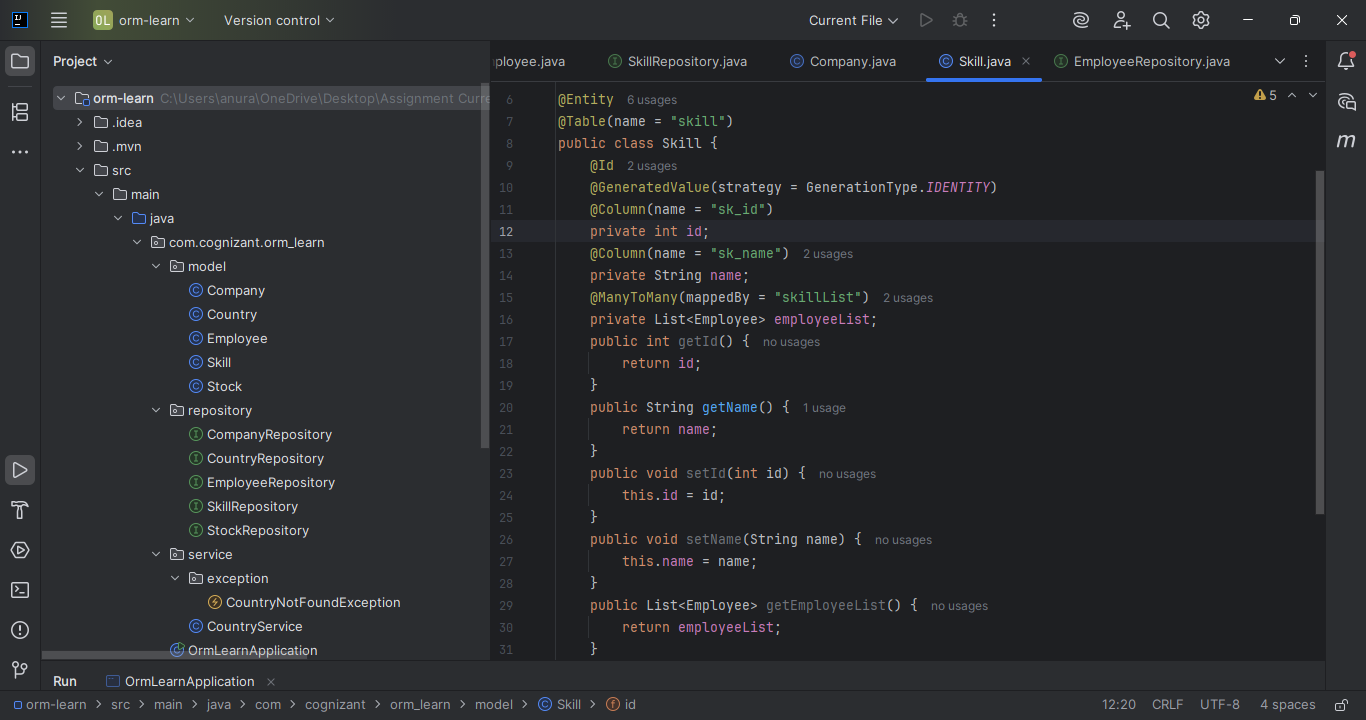
*);*

Sample employee-skill data

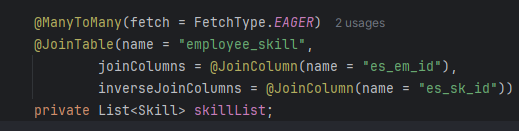
*INSERT INTO employee\_skill VALUES (1, 1), (1, 4);*

*INSERT INTO employee\_skill VALUES (2, 2);*

*INSERT INTO employee\_skill VALUES (3, 3), (3, 1);*

1. Created Skill.java in model package 
2. Update Employee to include Skill.java

Inside CompanyRepository

**

1. Created SkillRepository



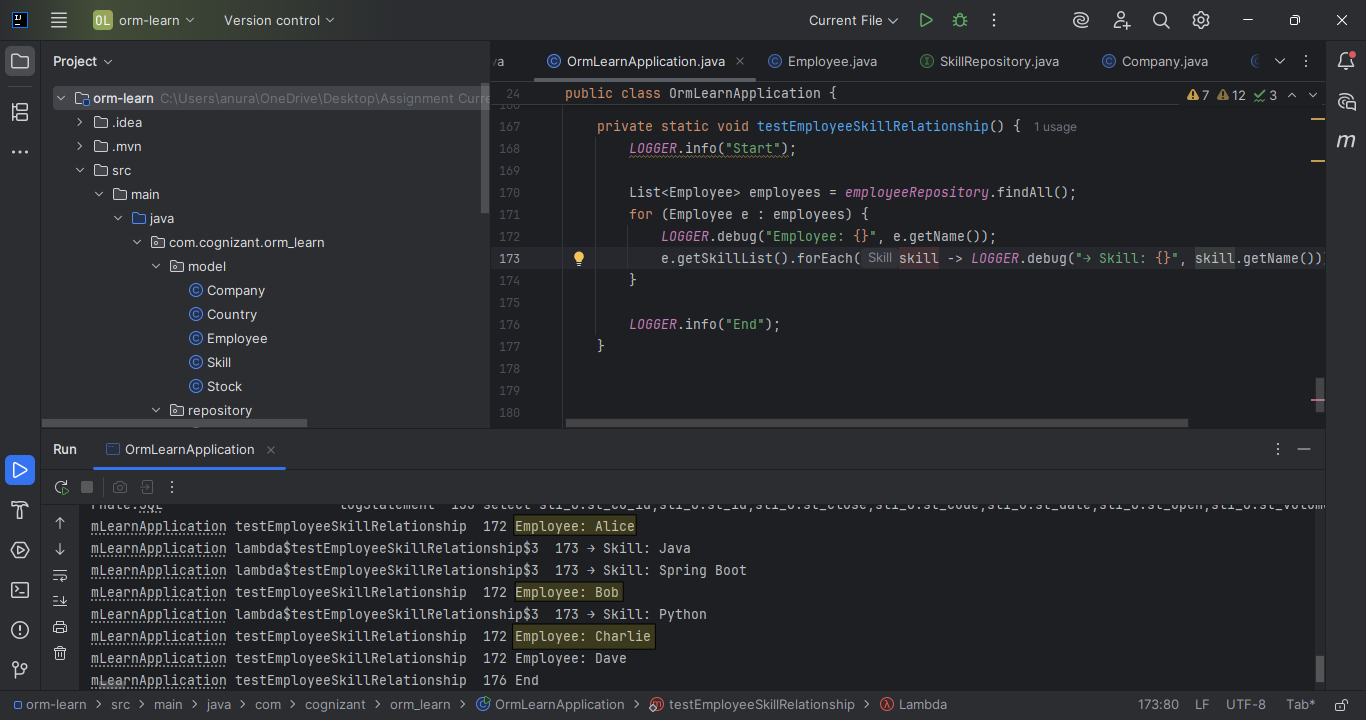
1. Added tests to OrmLearnApplication

*private static SkillRepository skillRepository;*

In main():

*skillRepository = context.getBean(SkillRepository.class);*

*testEmployeeSkillRelationship();*

TEST OUTPUT: **

Test outputs are highlighted for clarity.