**HANDS ON 1**

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

**Application.properties:**

spring.application.name=orm\_learn

spring.datasource.url=jdbc:h2:mem:orm\_learn

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

spring.jpa.hibernate.ddl-auto=create

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

server.port=8080

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

**Country.java:**

spring.application.name=orm\_learn

spring.datasource.url=jdbc:h2:mem:orm\_learn

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

spring.jpa.hibernate.ddl-auto=create

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

server.port=8080

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

**CountryRepository.java:**

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

**import** java.util.List;

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

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

**public** **interface** CountryRepository **extends** JpaRepository<Country, String> {

List<Country> findByNameContainingIgnoreCase(String keyword);

List<Country> findByNameContainingIgnoreCaseOrderByNameAsc(String keyword);

List<Country> findByNameStartingWithIgnoreCase(String prefix);

}

**CountryService.java:**

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

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

**import** com.cognizant.orm\_learn.repository.CountryRepository;

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

**import** org.springframework.stereotype.Service;

**import** jakarta.transaction.Transactional;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

**public** Country findCountryByCode(String code) {

**return** countryRepository.findById(code).orElse(**null**);

}

@Transactional

**public** **void** addCountry(Country country) {

countryRepository.save(country);

}

}

**OrmLearnApplication.java:**

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

**import** java.util.List;

**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.Country;

**import** com.cognizant.orm\_learn.repository.CountryRepository;

@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 {

countryRepository.save(**new** Country("IN", "India"));

countryRepository.save(**new** Country("US", "United States"));

countryRepository.save(**new** Country("FR", "France"));

countryRepository.save(**new** Country("ZA", "South Africa"));

countryRepository.save(**new** Country("AU", "Australia"));

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

List<Country> countriesWithOu = countryRepository.findByNameContainingIgnoreCase("In");

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

System.***out***.println("\nCountries containing 'F' sorted by name:");

List<Country> countriesWithOuSorted = countryRepository.findByNameContainingIgnoreCaseOrderByNameAsc("F");

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

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

List<Country> countriesStartWithZ = countryRepository.findByNameStartingWithIgnoreCase("A");

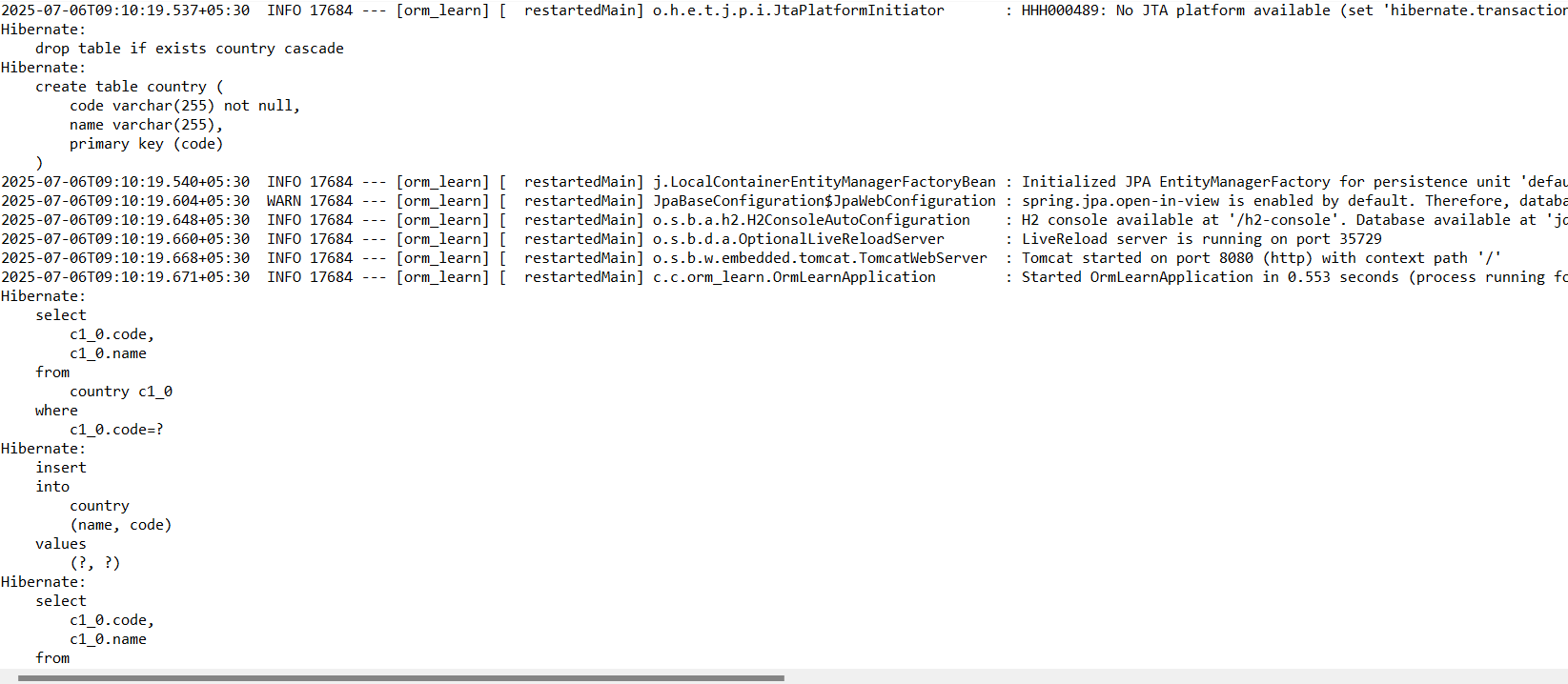
countriesStartWithZ.forEach(c -> System.***out***.println(c.getCode() + " - " + c.getName()));

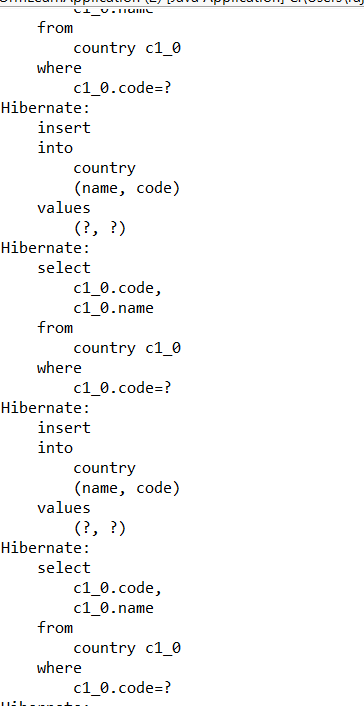
}

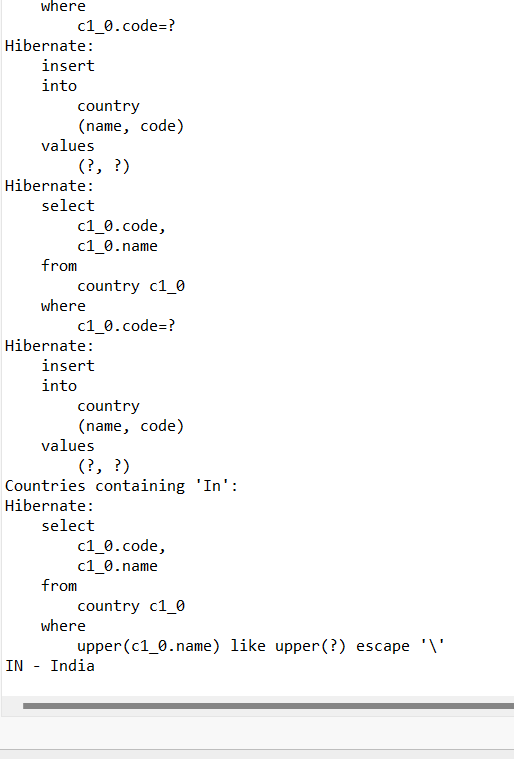
}

**OUTPUT:**











IN <http://localhost:8080/h2-console>:

