**Hands On 1 — Query Methods on Country table**

**Step 1: Search by partial country name (e.g., "ou")**

**Requirement**: When typing in a textbox, show a list of matching countries whose names contain a particular substring (case-insensitive).

**1. Define the Country entity**

@Entity

@Table(name = "country")

public class Country {

@Id

@Column(name = "co\_code")

private String code;

@Column(name = "co\_name")

private String name;

// Getters, setters, toString

}

**2. Create the CountryRepository interface**

public interface CountryRepository extends JpaRepository<Country, String> {

List<Country> findByNameContainingIgnoreCase(String keyword);

}

**3. Add a test method in OrmLearnApplication.java**

@Autowired

private static CountryRepository countryRepository;

private static void testFindCountriesContaining() {

LOGGER.info("Start");

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

for (Country country : countries) {

LOGGER.debug("Country: {}", country);

}

LOGGER.info("End");

}

**Step 2: Enhance the previous method to return results in ascending order**

Change the repository method to:

List<Country> findByNameContainingIgnoreCaseOrderByNameAsc(String keyword);

**Step 3: Search by countries starting with a specific alphabet**

When a user clicks an alphabet, return countries whose names **start with** that alphabet.

**Repository Method:**

List<Country> findByNameStartingWithIgnoreCase(String alphabet);

**Test Method:**

private static void testFindCountriesStartingWith() {

LOGGER.info("Start");

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

for (Country country : countries) {

LOGGER.debug("Country: {}", country);

}

LOGGER.info("End");

}