2. Spring Data Jpa Hands On

Demonstrate implementation of Query Methods feature of Spring Data JPA

CountryRepository.java

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

   // Search by containing text
   List<Country> findByNameContaining(String keyword);

   // Search by containing text and order by name
   List<Country> findByNameContainingOrderByNameAsc(String keyword);

   // Filter with starting text
   List<Country> findByNameStartingWith(String prefix);
}
```

StockRepository.java

```
public interface StockRepository extends JpaRepository<Stock, Integer> {

   // Get Facebook stock details for September 2019
   List<Stock> findByCodeAndDateBetween(String code, LocalDate start, LocalDate end);

   // Get Google stocks with close price greater than 1250
   List<Stock> findByCodeAndCloseGreaterThan(String code, BigDecimal price);

   // Top 3 by volume
   List<Stock> findTop3ByOrderByVolumeDesc();

   // Bottom 3 Netflix close prices
   List<Stock> findTop3ByCodeOrderByCloseAsc(String code);
}
```

Demonstrate implementation of O/R Mapping

1. @ManyToOne and @JoinColumn

```
@ManyToOne
@JoinColumn(name = "em_dp_id")
private Department department;
```

2. @OneToMany, mappedBy, FetchType.LAZY, FetchType.EAGER

```
@OneToMany(mappedBy = "department", fetch = FetchType.EAGER)
private Set<Employee> employeeList;
```

3. @ManyToMany and @JoinTable

Employee.java

```
@ManyToMany(fetch = FetchType.EAGER)
@JoinTable(name = "employee_skill",
   joinColumns = @JoinColumn(name = "es_em_id"),
   inverseJoinColumns = @JoinColumn(name = "es_sk_id"))
private Set<Skill> skillList;
```

Skill.java

```
@ManyToMany(mappedBy = "skillList")
private Set<Employee> employeeList;
```

Service Layer Example (EmployeeService.java)

```
@Service
public class EmployeeService {
   @Autowired
   private EmployeeRepository employeeRepository;

   @Transactional
   public Employee get(int id) {
      return employeeRepository.findById(id).get();
   }
 @Transactional
   public void save(Employee employee) {
      employeeRepository.save(employee);
   }
}
```