Exercise 3: Creating Repositories

Understanding Spring Data Repositories

Spring Data JPA provides a repository abstraction layer that significantly simplifies data access layer development. By extending the `JpaRepository` interface, you get a set of methods for basic CRUD operations out of the box. Additionally, Spring Data JPA can automatically generate query methods based on method names.

Creating Repositories

Java

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

public interface EmployeeRepository extends JpaRepository<Employee,
Long> {
    // Custom query methods can be added here
}

public interface DepartmentRepository extends JpaRepository<Department,
Long> {
    // Custom query methods can be added here
}
```

Explanation of the Code

- **Inheritance:** Both `EmployeeRepository` and `DepartmentRepository` extend `JpaRepository`.
- **Generic Types:** The first generic type is the entity class (Employee or Department), and the second is the type of the ID (Long in this case).

- **CRUD Operations:** By extending `JpaRepository`, you automatically get methods like `save`, `findById`, `findAll`, `deleteById`, etc.

Derived Query Methods

Spring Data JPA supports creating custom query methods based on method names. For example:

Java

```
public interface EmployeeRepository extends JpaRepository<Employee, Long>
{
    List<Employee> findByDepartmentName(String departmentName);
    List<Employee> findByNameStartingWith(String namePrefix);
}
```

- `findByDepartmentName`: This method will retrieve a list of employees whose department name matches the given `departmentName`.
- `findByNameStartingWith`: This method will retrieve a list of employees whose name starts with the given `namePrefix`.

Spring Data JPA will automatically generate the corresponding JPQL query based on the method name.

Key Points:

- Spring Data JPA significantly reduces boilerplate code for data access.
- Derived query methods provide a convenient way to create custom queries.
- You can also use `@Query` annotation for more complex queries.