**Spring Data JPA - Task 1 Completion Summary**

**Title:** Hands-On 1 - Spring Data JPA - Quick Example

**Objective:**  
To create a Spring Boot application that connects to a MySQL database using Spring Data JPA and retrieves data from a table named country.

**1. Environment Setup**

* **Software Installed:**
  + MySQL Server 8.0
  + MySQL Workbench 8
  + Eclipse IDE for Enterprise Java Developers 2019-03 R
  + Maven 3.6.2
* **Spring Initializr Setup:**
  + Group: com.cognizant
  + Artifact: orm-learn
  + Description: Demo project for Spring Data JPA and Hibernate
  + Dependencies Selected:
    - Spring Boot DevTools
    - Spring Data JPA
    - MySQL Driver
* **Import Steps:**
  + Extracted the downloaded zip to Eclipse Workspace
  + Imported into Eclipse via: File > Import > Maven > Existing Maven Projects
* **MySQL Database Setup:**
* create schema ormlearn;
* create table country(co\_code varchar(2) primary key, co\_name varchar(50));
* insert into country values ('IN', 'India');
* insert into country values ('US', 'United States of America');

**2. Project Structure**

**src/main/java**

* Application code

**src/main/resources**

* application.properties contains DB config and logging

**src/test/java**

* Unit test placeholders

**OrmLearnApplication.java**

* Main method bootstraps the application and calls a test method

**3. application.properties Configuration**

spring.application.name=orm-learn

logging.level.org.springframework=info

logging.level.com.cognizant=debug

logging.level.org.hibernate.SQL=trace

logging.level.org.hibernate.type.descriptor.sql=trace

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger{25} %25M %4L %m%n

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

**4. Code Implementation**

**Model Class - Country.java**

@Entity

@Table(name = "country")

public class Country {

@Id

@Column(name = "co\_code")

private String code;

@Column(name = "co\_name")

private String name;

// Getters, Setters, and toString()

}

**Repository Interface - CountryRepository.java**

@Repository

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

**Service Class - CountryService.java**

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

}

**Application Class - OrmLearnApplication.java**

@SpringBootApplication

public class OrmLearnApplication {

private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);

private static CountryService countryService;

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

countryService = context.getBean(CountryService.class);

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

List<Country> countries = countryService.getAllCountries();

LOGGER.debug("countries={}", countries);

LOGGER.info("End");

}

}

**5. Testing and Verification**

* Application built using:

mvn clean package

* Application executed via OrmLearnApplication.main()
* Output validated in logs:
  + Confirmed Inside main was logged
  + Confirmed country data was retrieved from DB

**6. Project Insight by SME**

* Explained src/main/java, src/main/resources, and src/test/java
* Explained purpose of @SpringBootApplication
* Demonstrated Maven dependency tree
* Walked through pom.xml and dependencies