# 6. Spring Data Access with JDBC

## 6.1 Sample Data Model for Example Code

-DAO: data access object, Sprong world: repository

## 6.2 Explore JDBC Infrastructure

## 6.3 Spring JDBC Infrastructure

6.3.1 Overview and used Packages

6.3.2 Database Connections and DataSources

6.3.3 Embedded Database Support

6.3.4 Use DataSources in DAO Classes

6.3.5 Exception Handling

6.3.6 JdbcTemplate Class

6.3.7 Spring Classes that Model JDBC Operations

6.3.8 Spring Data Project: JDBC Extensions

6.3.9 Spring JDBC Testing Annotations

6.3.10 Introduce Testcontainers

6.3.11 Considerations for Using JDBC

6.3.12 Spring Boot JDBC

# 7. Spring with Hibernate

## 7.1 Sample Data Model for Example Code

## 7.2 Configure Hibernate’s SessionFactory

## 7.3 ORM Mapping Using Hibernate Annotations

## 7.4 Simple Mappings

### 7.4.1 One-to-Many Mappings

### 7.4.2 Many-to-Many Mappings

### 7.4.3 The Hibernate Session Interface

## 7.5 Query Data by Using Hibernate Query Language

### 7.5.1 Simple Querying with Lazy Fetching

7.5.2 Querying with Associations Fetching

## 7.6 Insert Data

## 7.7 Update Data

## 7.8 Delete Data

## 7.9 Execute SQL Native Queries

## 7.10 Execute Projections with Hibernate

## 7.11 Call Stored Functions with Hibernate

## 7.12 Configure Hibernate to Generate Tables from Entities

## 7.13 Annotate Methods or Fields?

## 7.14 Considerations when using Hibernate

## 7.15 Introduce jOOQ

# 8. Spring with JPA

8.1 Introduce JPA 3.1

8.1.1 Sample Data Model for Example Code

8.1.2 Configure JPA’s EntityManagerFactory

8.1.3 Use JPA Annotations for ORM Mapping

8.2 Perform Database Operations with JPA

8.2.1 Use Java Persistence Query Language to Query Data

8.2.2 Query with Untyped Results

8.2.3 Query for a Custom Result Type with a Constructor Expression

8.2.4 Insert Data

8.2.5 Update Data

8.2.6 Delete Data

8.3 Use a Native Query

8.3.1 Use a Simple native query

8.3.2 Native Querying with SQL ResultSet Mapping

8.3.3 Execute Stored Functions and Procedures

8.4 Using JPA Criteria API for a Criteria Query

# 9. Spring Transaction Management

9.1 Explore Spring Transaction Abstraction Layer

-Transaction Types

9.2 Implementations of PlatformTransactionManager

9.3 Analyze Transaction Properties

9.3.1 TransactionDefinition Interface

9.3.2 TransactionStatus Interface

9.4 Sample Data Model and Infrastructure for Example Code

9.4.1 Use Declarative Transactions

9.4.2 Roll Back Transactions

9.4.3 Use Programmatic Transactions

9.4.4 Considerations on Transaction Management

9.5 Transactional Configuration with Spring Boot

9.5.1 Transactional Tests

9.5.2 Considerations on Transaction Management

# 10. Spring Data with SQL and NoSQL Database

10.1 Introduce Spring Data JPA

10.1.1 Use Spring Data JPA Repository Abstraction for Database Operations

10.1.2 Use JpaRepository

10.1.3 Spring Data JPA with Custom Queries

10.1.4 Projection Queries

10.2 Keep track of changes on Entity Class

10.3 Keep Entity Versions by Using Hibernate Envers

10.3.1 Add Tables for Entity Versioning

10.3.2 Configure EntityManagerFactory for Entity Versioning

10.3.3 Enable Entity Versioning and History Retrieval

10.3.4 Custom Implementation for Spring Data Repositories

10.4 Spring Boot Data JPA

10.5 Spring Data with MongoDB

10.6 Considerations using Spring Data