

7장 Spring Data JPA



목차

- 1. JPA 개요
- 2. Spring Data JPA 개요
- 3. JPA Entity와 Entity Manager
- 4. JPA 연관관계
- 5. QueryDSL

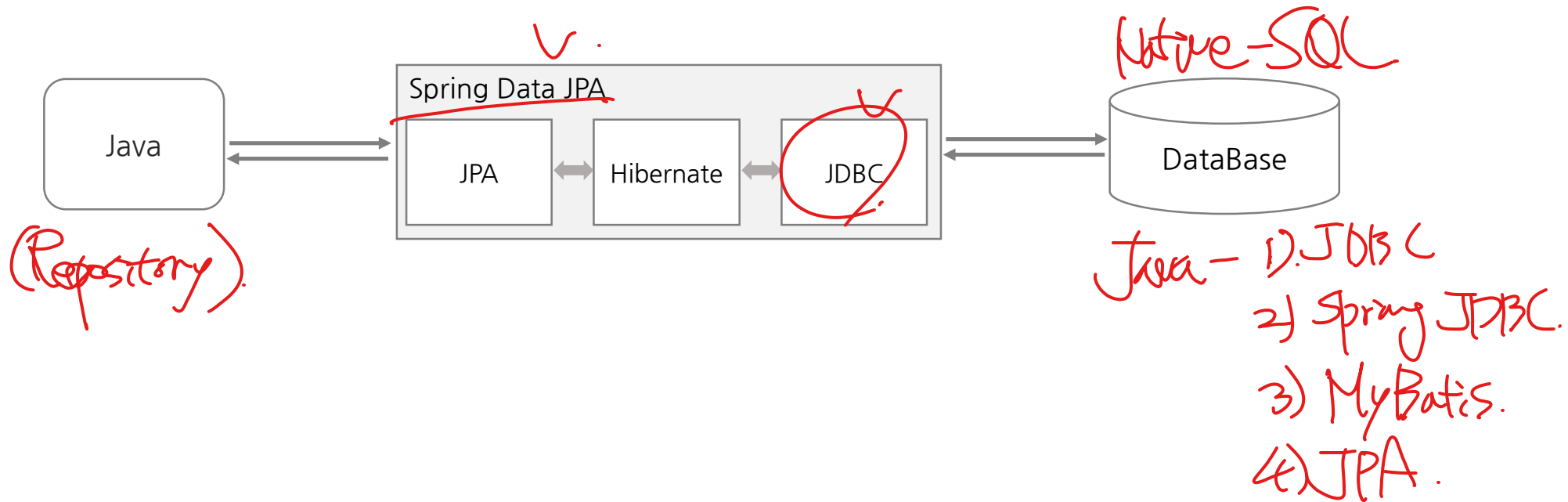
1. JPA 개요

Persistence : 영속성, 데이터를 영구 저장

JPA (Java Persistence API)는 자바 ORM 기술에 대한 API 표준으로 Java를 이용해서 데이터를 유지관리 할 수 있는 기술

Hibernate는 JPA 인터페이스를 구현한 구현체이자 자바용 ORM 프레임워크

JPA는 복잡한 SQL 실행이 어렵다는 것과 JPA 기술을 애플리케이션 개발에 제대로 활용하기 위해서는 많은 SQL 개발 경험 필요



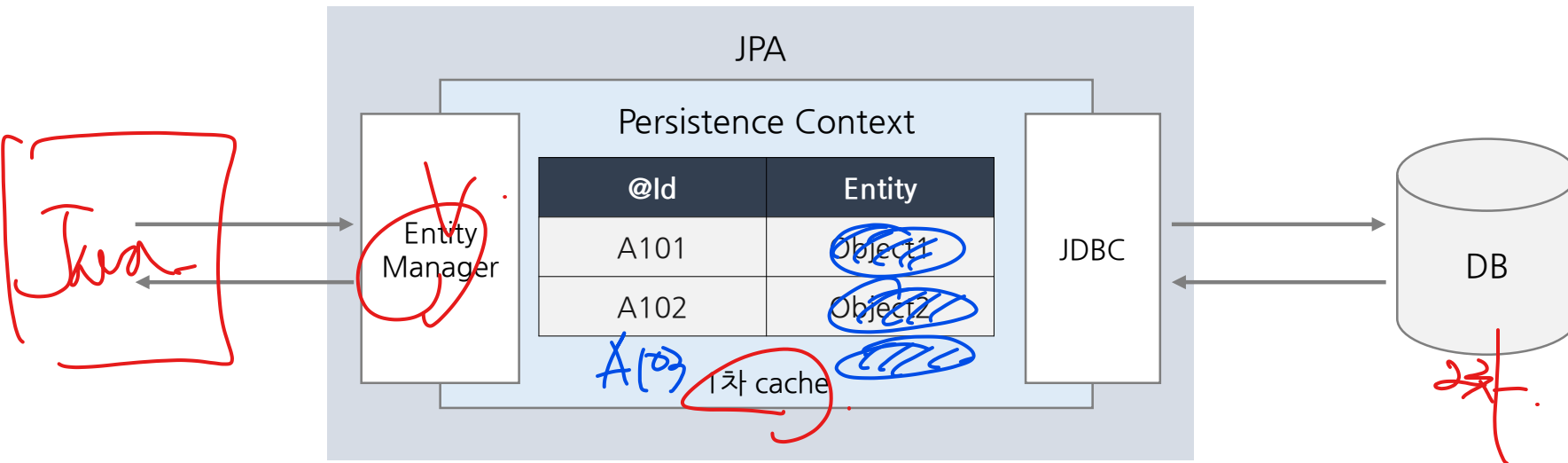
2. Spring Data JPA 개요

- Spring Data JPA는 Spring에서 JPA 기술을 간편하고 편리하게 사용하기 위한 프레임워크
- Spring Data JPA의 핵심은 JPA Repository 인터페이스를 제공하여 데이터베이스와 상호작용(CRUD)
- 엔티티 Entity는 데이터베이스 테이블과 매핑되는 자바 객체로서 엔티티를 통해 데이터베이스와 상호작용

| 어노테이션 | 설명 |
|------------------------|---|
| <u>@Entity</u> | 엔티티 클래스를 정의할 때 사용하는 JPA 어노테이션 |
| <u>@Table</u> | 엔티티 클래스가 매핑될 데이터베이스 테이블의 정보를 명시하는 어노테이션 |
| <u>@Id</u> | 엔티티 클래스의 기본키 Primary Key를 나타내는 필드에 부여하는 어노테이션 |
| <u>@GeneratedValue</u> | 엔티티의 기본 키를 자동으로 생성하기 위해 사용되는 어노테이션 |
| <u>@Column</u> | 엔티티 클래스의 필드와 데이터베이스의 컬럼을 매핑에 사용되는 어노테이션 |
| <u>@Query</u> | JPQL이나 네이티브 SQL 직접 작성할 때 사용되는 어노테이션 |
| <u>@Transactional</u> | 트랜잭션을 처리하기 위한 어노테이션 |

3. Entity와 Entity Manager

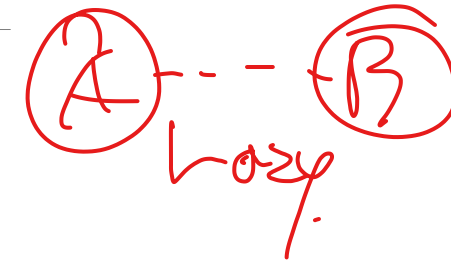
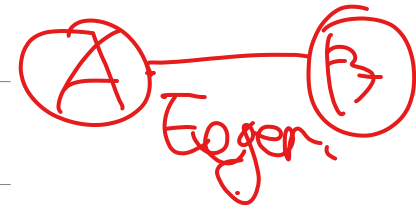
- 영속성 컨텍스트(Persistence Context)는 Entity를 관리하고 저장하는 저장 공간 (비지리)
- Entity Manager는 Entity를 관리하는 객체로 Entity를 영속성 컨텍스트 1차 캐시에 저장하고 접근 제어
- 영속성 컨텍스트의 캐시, 쓰기 지연, 변경 감지, 지연 로딩의 특징으로 데이터베이스의 성능을 극대화 ✓



4. JPA 연관관계

- JPA 연관관계는 Entity 간 서로 논리적인 의미를 갖고 서로 참조하는 관계로 방향성과 다중성을 고려하여 관계를 설정
- FetchType은 하나의 엔티티를 조회할 때 연관된 엔티티를 효율적으로 로딩하여 성능을 최적화하기 위한 지연전략
- 엔티티 간의 연관관계를 올바르게 설정하고 관리함으로써 데이터베이스의 일관성을 유지하고 객체지향적인 모델링을 설계

| 관계 | 어노테이션 | 패치전략 | 설명 |
|-----|-------------|-----------------|-----------------------------------|
| 1:1 | @OneToOne | FetchType.EAGER | 1:1 관계 설정, 연관된 엔티티를 즉시 조회 |
| N:1 | @ManyToOne | FetchType.EAGER | N:1 관계 설정, 연관된 엔티티를 즉시 조회 |
| 1:N | @OneToMany | FetchType.LAZY | 1:N 관계 설정, 연관된 엔티티를 사용되는 시점에 조회 |
| N:M | @ManyToMany | FetchType.EAGER | N:M 관계 설정, 1:N와 N:1 관계로 나눠서 관계 설정 |



5. QueryDSL

- QueryDSL Domain Specific Language 은 Java로 타입 안전한 쿼리 작성 및 데이터베이스와의 상호작용을 도와주는 프레임워크
- QueryDSL은 JPA와 함께 사용되며, 복잡한 SQL 쿼리나 JPQL을 객체지향적으로 작성할 수 있도록 지원
- QueryDSL은 JPA 엔티티 기반의 Q도메인 클래스를 생성해 동적으로 쿼리 작성

| 주요 구성 요소 | 설명 |
|-------------------|--|
| Q도메인 클래스 | <ul style="list-style-type: none">• QueryDSL의 핵심 구성 요소로 쿼리 작성 시 타입 안전성을 제공• 엔티티 클래스를 기반으로 자동 생성되는 메타 모델 클래스 |
| JPAQueryFactory | <ul style="list-style-type: none">• JPA 기반 쿼리 작성 시 사용하는 클래스• 쿼리 작성 및 실행을 위한 메서드를 제공 |
| BooleanExpression | <ul style="list-style-type: none">• 조건절을 표현하는데 사용되는 클래스• 복잡한 where 조건을 만들 때 사용 |
| Projection | <ul style="list-style-type: none">• 쿼리 결과를 특정 DTO로 매핑할 때 사용• 결과 집합을 원하는 형태의 객체로 변환하는데 사용 |