

Section 4

JPA

목차

엔티티 매핑

엔티티 매핑 관련 내용

스키마 자동생성

속성

실습

실습해보기

15%

엔티티 매핑

엔티티 매핑 관련 어노테이션

엔티티 매핑 관련 어노테이션들

객체와 테이블 : @Entity, @Table

- @Entity가 있어야 JPA에서 관리, 기본생성자가 있어야 하며 final클래스, enum, interface, inner클래스 사용X
name 속성으로 엔티티 이름 지정, @Table name 속성은 테이블 이름 변경 가능(DB table명)

필드와 컬럼 : @Column

- 유니크, 길이, 이름 설정 등 속성 지정 가능 (이런 속성(이름 제외)은 ddl 생성 기능)
- nullable → 널 가능

기본 키 매핑 : @Id, @GeneratedValue

- GeneratedValue: IDENTITY(DB에 위임), SEQUENCE(시퀀스 오브젝트 사용, @SequenceGenerator 필요),
TABLE(키 생성용 테이블 사용, @TableGenerator 필요), AUTO(기본값)
- SequenceGenerator : allocationSize로 최적화 가능

연관관계 매핑 : @ManyToOne, @JoinColumn

엔티티 매핑

필드와 컬럼 매핑

컬럼 관련 어노테이션들

@Enumerated(EnumType.형태)

- enum 형태를 구현해줌

@Temporal(TemporalType.형태)

- 시간 저장 방법 지정, LocalDate, LocalDateTime 사용시 생략 가능(최신 하이버네이트 지원)

@Lob

- 큰 데이터 저장

스키마 자동생성

속성

hibernate.hbm2ddl.auto 옵션

create : drop + create

create-drop : create와 동일하지만 종료시점에 drop

update : 변경부분만 반영, 추가만 동작

validate : 정상 매핑되었는지만 확인

none : 사용하지 않음, 사실상 없는 것, 위의 것들 중 대응되지 않아 동작하지 않는 것

운영 중에는 create, create-drop, update 사용 금지

사용 추천 과정 :

개발 초기는 create or update

테스트는 update or validate

스테이징과 운영서버는 validate or none

실습

실습해보기

시퀀스 사용 실습과 DB와의 통신 시간대 확인

```
try {
    System.out.println("===프로그램 시작===");
    Member2 member = new Member2();
    System.out.println("===Member2 객체 생성 후===");
    member.setName("A");
    System.out.println("===Member2 객체 이름 설정 후===");
    em.persist(member);
    System.out.println("===영속성 컨테이너 작업 후===");
    tx.commit();
    System.out.println("===Member2 커밋 후===");
} catch (Exception e) {
    tx.rollback();
} finally {
    em.close();
}
emf.close();
}
```

Practice3 x

12월 28, 2022 6:38:38 오후 org.hibernate.engine.transaction.jta.platform.HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]

===프로그램 시작===

===Member2 객체 생성 후===

===Member2 객체 이름 설정 후===

Hibernate:

call next value for Member2_Sequence

===영속성 컨테이너 작업 후===

12월 28, 2022 6:38:38 오후 org.hibernate.id.enhanced.PooledOptimizer.HHH000201: Pooled optimizer source reported [BasicHolder[java.lang.Integer@1539e260]]

Hibernate:

/* insert DAO.Member2

*/ insert

into

Member2

(name, id)

values

(?, ?)

===Member2 커밋 후===

← 영속성 컨테이너에 넣을 경우 id가 필요하기 때문에 시퀀스에서 가져 오는 것을 볼 수 있다