14. 컬렉션과 부가기능

...

Blur

컬렉션

- Java의 Collection, List, Set, Map를 JPA에 대응하는 래퍼컬렉션를 사용하여 보장

표 14.1 하이버네이트 내장 컬렉션과 특징

컬렉션 인터페이스	내장 컬렉션	중복 허용	순서 보관
Collection, List	PersistenceBag	O	X
Set	PersistenceSet	X	X
List + @OrderColumn	PersistentList	O	O

- Set - Equal(), hashcode() 메소드로 중복 체크

컬렉션

List + @OrderColumn - 순서 저장(Column 추가)

[BOARD]

ID	TITLE	CONTENT
1	제목1	내용1

[COMMENT]

ID	COMMENT	COMMENTS_ID (FK)	POSITION @OrderColumn
1	댓글1	1	0
2	댓글2	1	1
3	댓글3	1	2
4	댓글4	1	3

⇒ @OrderBy 이용해서 순서 보장(모든 컬렉션 사용 가능)

Converter

- 엔티티를 변환시켜 데이터베이스에 저장

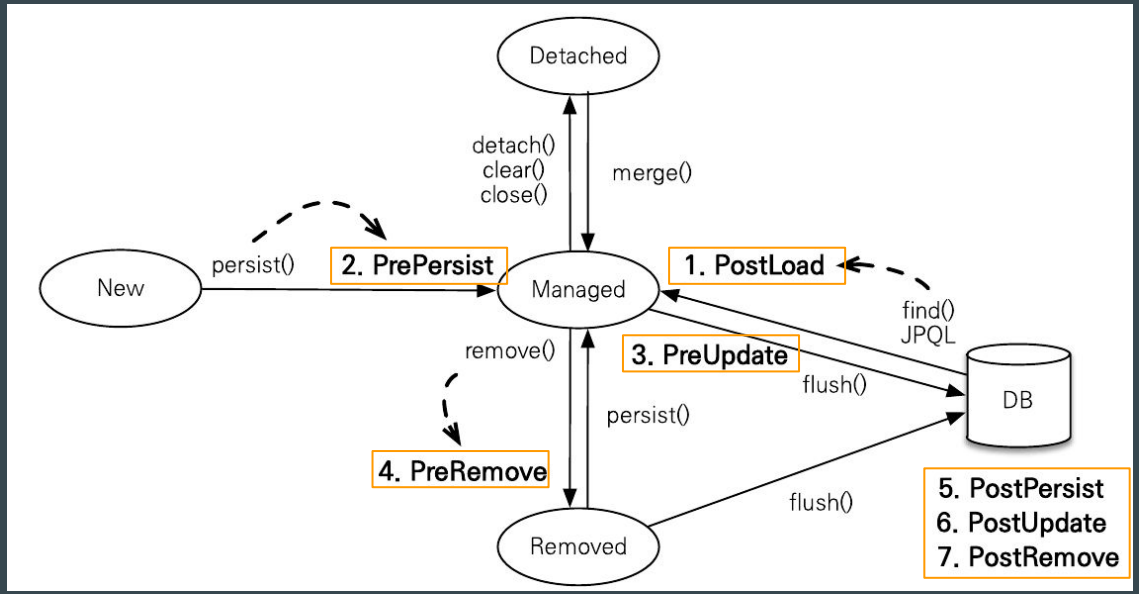
  ⇒ Ex) boolean -> Y/N

표 14.2 @Convert 속성 정리

속성	기능	기본값
converter	사용할 컨버터를 지정한다.	
attributeName	컨버터를 적용할 필드를 지정한다.	
disableConversion	글로벌 컨버터나 상속 받은 컨버터를 사용하지 않는다.	false

리스너

- 엔티티 생명주기에 따른 이벤트 처리



엔티티 그래프

- FetchType.EAGER - 연관된 엔티티 함께 조회

  ⇒ 같은 JPQL을 중복해서 작성하는 경우 발생

1. Named 엔티티 그래프(정적 그래프 탐색)
2. 동적 그래프 탐색

엔티티 그래프

- Named Graph: attributeNodes로 함께 조회할 속성 선택

```
@NamedEntityGraph(name = "Order.withMember", attributeNodes = {
    @NamedAttributeNode("member")
})
@Entity
@Table(name = "ORDERS")
public class Order {
```

- 동적 그래프

```
EntityGraph<Order> graph = em.createEntityGraph(Order.class);
graph.addAttributeNodes("member");
```

엔티티 그래프

- Entity는 Root에서 시작
- 이미 로딩된 엔티티는 그래프가 적용되지 않는다

Thank you