

JPA Entity Mapping

- 객체와 테이블 매핑: @Entity @Table
 - 기본생성자 필수 (public or protected 생성자)
 - final class, enum, interface, inner class 사용 금지
 - 필드 final 금지
 - 필드와 컬럼 매핑: @Column
 - 기본키 매핑: @Id
 - 연관관계 매핑: @ManyToOne @JoinColumn
-

- hibernate ddl 가급적 사용하지 않는게 좋다.
 - create: drop 하고 create
 - create-drop: 종료시점 drop
 - update: 변경된 부분만 반영
 - validate: entity와 table 정상 매핑 확인
 - none: 사용안함
-

- @Column
 - name: 필드와 매핑할 테이블 컬럼명
 - insertable, updatable: 등록, 변경 가능여부
 - nullable: not null 제약조건
 - unique
 - ◆ 잘안씀. @Table 에서 (uniqueConstraints = "") 설정으로 함
 - columnDefinition: 직접 컬럼정보 입력
 - length: 문자길이 String만
 - precision, scale: BigDecimal 타입에서 사용
-

- @Enumerated
 - EnumType.ORDINAL: enum 순서를 데이터베이스에 저장 (default) - 지양
 - EnumType.STRING: enum 이름을 데이터베이스에 저장
-

- 날짜는 LocalDateTime java type으로
-

- 기본키 매핑
 - @ID
 - ◆ @GeneratedValue: 자동 생성

- ◇ IDENTITY: 데이터베이스에 위임 MySQL
- ◇ SEQUENCE: 데이터베이스 시퀀스 오브젝트 사용 ORACLE @SequenceGenerator 필요

```
@Entity
@SequenceGenerator(
    name = "MEMBER_SEQ_GENERATOR",
    sequenceName = "MEMBER_SEQ", //매핑할 데이터베이스 시퀀스 이름
    initialValue = 1, allocationSize = 1)
public class Member {

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE,
        generator = "MEMBER_SEQ_GENERATOR")
    private Long id;
```

- 주의: allocationSize 기본값 = 50
- name: 식별자 생성기 이름 필수
- sequenceName: 데이터베이스 등록되어 있는 시퀀스 이름
- initialValue: DDL 생성시에만 사용됨 처음 1 시작하는 수를 지정
- allocationSize: 시퀀스 한 번 호출에 증가하는 수(성능 최적화에 사용됨)
 - 데이터베이스 시퀀스 값 하나씩 증가하도록 설정되어 있으면 이 값을 반드시 1로 설정해야한다
- catalog, schema: 데이터베이스 catalog, schema 이름
- ◇ TABLE: 키 생성용 테이블 사용 모든 DB에서 사용 @TableGenerator 필요
 - 키 생성 전용 테이블을 하나 만들어 데이터베이스 시퀀스를 흉내내는 전략
 - 모든 DB에 적용가능
 - 단점은 성능

@TableGenerator - 속성

속성	설명	기본값
name	식별자 생성기 이름	필수
table	키생성 테이블명	hibernate_sequences
pkColumnName	시퀀스 컬럼명	sequence_name
valueColumnName	시퀀스 값 컬럼명	next_val
pkColumnValue	키로 사용할 값 이름	엔티티 이름
initialValue	초기 값, 마지막으로 생성된 값이 기준이다.	0
allocationSize	시퀀스 한 번 호출에 증가하는 수(성능 최적화에 사용됨)	50
catalog, schema	데이터베이스 catalog, schema 이름	
uniqueConstraints(DDL)	유니크 제약 조건을 지정할 수 있다.	

- ◇ AUTO: 방언에 따라 자동지정
- 권장하는 식별자 전략
 - ◆ 기본 키 제약 조건: null 아니고 유일, 변하면 안된다.

- ◆ 미래까지 이 조건을 만족하는 자연키는 찾기 어렵다. 대리키(대체키)를 사용
 - ◆ 권장: Long형 + 대체키 + 키 생성전략 사용
 - ◇ auto increment
 - ◇ sequence
 - ◇ uuid
 - ◇ random 값 조합
-

IDENTITY 전략

- 기본 키 생성 데이터베이스에 위임
- JPA는 보통 트랜잭션 커밋 시점에 insert sql 실행
- AUTO_INCREMENT 는 INSERT SQL 실행 한 이후에 ID를 알 수 있음
- 영속성 컨텍스트에서 @Id 값을 알아야 하는데 DB에 저장 후에 알 수 있다.
- 그래서 persist 되는 시점에 insert 하고 DB에서 식별자를 조회한다.

#dev/jpa/tutorial