

초격차 패키지 Online.

# JPA 기본기

## PART1 | ORM, JPA, JPQL 개요

기본기 정리하고 넘어가기

## PART2 | 기존의 기술들 - iBATIS, MyBatis, JdbcTemplate

JPA 를 사용하기 전에 널리 사용하던 기술들

## PART3 | Hibernate vs. Spring Data JPA

둘이 무슨 차이에요?

## PART4 | in memory 테스트 DB - H2

날아가도 안전한 테스트 DB를 손쉽게 만드는 방법

# JPA 기본기

## 1 ORM, JPA, JPQL 개요

## ORM, JPA, JPQL 개요

# 1.

ORM, JPA, JPQL  
개요

### Spring Data JPA 를 시작하기 전에..

정확한 개념을 세우고 들어가보자

- ORM?
- JPA?
- JPQL?

## ORM

# 1.

ORM, JPA, JPQL  
개요

## Object Relational Mapping

객체 지향 언어를 이용하여, 서로 호환되지 않는 타입 간의 데이터를 변환하는 기술

- 좁은 의미: DB(RDBMS) 테이블 데이터를 (자바) 객체와 매핑하는 기술
- 효과: RDBMS를 객체 지향 DB로 가상화하는 것
- ORM 으로 얻고자 하는 것
  - DB의 추상화: 특정 DB에 종속된 표현(ex: SQL)이나 구현이 사라지고, DB 변경에 좀 더 유연해짐
  - 객체의 이점을 활용: 객체간 참조, type-safety
  - 관심사 분리: DB 동작에 관한 코드 작성의 반복을 최소화하고 비즈니스 로직에 집중

## Jakarta(Java) Persistence API

자바에서 ORM 기술을 사용해 RDBMS를 다루기 위한 인터페이스 표준 명세

- API + JPQL + metadata (+ Criteria API)
- 기본적으로 관계형 데이터베이스의 영속성(persistence)만을 규정
  - JPA 구현체 중에 다른 유형의 데이터베이스 모델을 지원하는 경우가 있지만, 원래 JPA 스펙과는 무관
- 이름의 변화
  - Java Persistence API -> Jakarta Persistence API
  - 2017년 9월, 오라클이 Java EE 를 이클립스 재단으로 이관 -> 상표권 문제로 이름을 변경
  - Spring Boot: 2.2 부터 Jakarta EE 로 의존성이 변경됨
    - 현재: JPA 2.2.3 (패키지명은 아직 javax.persistence.\*)
    - 미래: JPA 3.0 이 도입되면 패키지명이 완전히 jakarta.persistence.\* 로 변경될 전망

## JPA: Persistence

# 1.

ORM, JPA, JPQL  
개요

### Persistence (영속성)

프로세스가 만든 시스템의 상태가 종료된 후에도 사라지지 않는 특성

- 구현 방법: 시스템의 상태를 데이터 저장소에 데이터로 저장한다.
- 사라지는 데이터 - 주기억장치(휘발성 스토리지)에 저장된 데이터
  - 프로세스 메모리 안의 데이터 (변수, 상수, 객체, 함수 등)
- 사라지지 않는 데이터 - 보조기억장치(비휘발성 스토리지)에 저장된 데이터
  - 하드디스크, SSD 에 기록된 데이터 (파일, 데이터베이스 등)
- 영속성 프레임워크: 영속성을 관리하는 부분을 persistence layer 로 추상화하고, 이를 전담하는 프레임워크에게 관리를 위임
- JPA 에서 persistence 란: 프로세스가 DB로부터 읽거나 DB에 저장한 정보의 특성

## JPQL

# 1.

ORM, JPA, JPQL  
개요

## Jakarta(Java) Persistence Query Language

플랫폼으로부터 독립적인 객체 지향 쿼리 언어

- JPA 표준의 일부로 정의됨
- RDBMS의 엔티티(Entity)를 다루는 쿼리를 만드는데 사용
- SQL의 영향을 받아서 형식이 매우 유사

## JPQL

# 1.

ORM, JPA, JPQL  
개요

### JPQL vs. SQL(MySQL)

```
SELECT DISTINCT a
FROM Author a
INNER JOIN a.books b
WHERE b.publisher.name = 'XYZ Press'
AND a.lastName IS NULL
;
```

```
SELECT DISTINCT a.*
FROM author a
INNER JOIN book b on b.id = a.book_id
INNER JOIN publisher p on p.id = b.publisher_id
WHERE p.name = 'XYZ Press'
AND a.last_name IS NULL
;
```



## JPQL

# 1.

ORM, JPA, JPQL  
개요

- SQL 과 JPQL 은 다른 언어이다.
  - SQL: 표준 ANSI SQL 을 기준으로 만든, 특정 DB에 종속적인 언어
  - JPQL: 특정 DB에 종속적인 언어가 아님
- JPA 프레임워크를 사용한다면
  - 특별한 요구사항이 있지 않은 한, JPQL 을 몰라도 된다
  - JPQL을 직접 코드에서 사용하고 있다면, 반드시 필요했던 일인지 검토하기

## Reference

# 1.

ORM, JPA, JPQL  
개요

- [https://en.wikipedia.org/wiki/Object%E2%80%93relational\\_mapping](https://en.wikipedia.org/wiki/Object%E2%80%93relational_mapping)
- <https://docs.oracle.com/javaee/7/tutorial/persistence-intro.htm>
- [https://en.wikipedia.org/wiki/Persistence\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Persistence_(computer_science))
- [https://en.wikipedia.org/wiki/Jakarta\\_Persistence\\_Query\\_Language](https://en.wikipedia.org/wiki/Jakarta_Persistence_Query_Language)