

데이터베이스

연동

최희선

2025.02.01

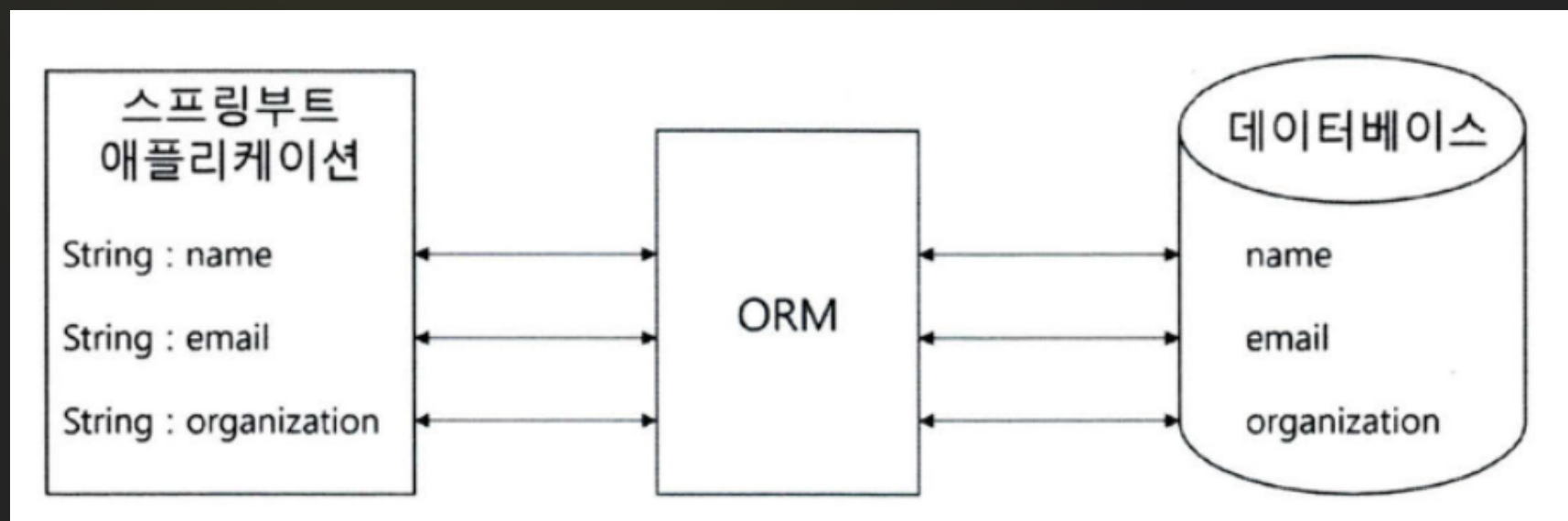
CONTENTS

- chapter 1 ORM
- chapter 2 JPA
- chapter 3 영속성 컨텍스트

☆ { ORM, JPA }

ORM(object relational mapping)은 객체 관계 매핑을 의미함.
객체지향 언어에서 객체는 클래스를 의미하는데, 클래스는 데이터 베이스의 테이블과 *매핑
하기 위해 만들어진 것이 아니기 때문에 *RDB 테이블과 불일치가 존재함.

ORM은 이 둘의 불일치와 제약사항을 해결하는 역할을 함



tip)

- '매핑'이란?
어떤 값을 다른 값에 대응 시키는 과정.
- 'RDB'란?
관계형 데이터베이스로, 데이터를 테이블 형태로 구조화한
데이터 베이스임

{ ORM 장점 }

1. ORM을 사용하면서 데이터베이스 쿼리를 객체지향적으로 조작 가능
2. 재사용 및 유지보수가 편리함
3. 데이터베이스에 대한 종속성이 줄어들음.

tip)'쿼리'란? 데이터베이스에 정보를 요청하는 것

{ ORM 단점 }

1. ORM만으로 온전한 서비스를 구현하는데 한계가 있음.
(복잡한 서비스의 경우 직접 쿼리를 구현하지 않고 코드로 구현하기 어려움)
2. 애플리케이션의 객체 관점과 데이터베이스의 관계 관점의 불일치 발생

{ ORM, JPA }

JPA는 ORM 기술 표준으로 채택된 인터페이스 모음임.

ORM이 큰 개념이라면 JPA는 더 구체화된 스펙을 포함함.

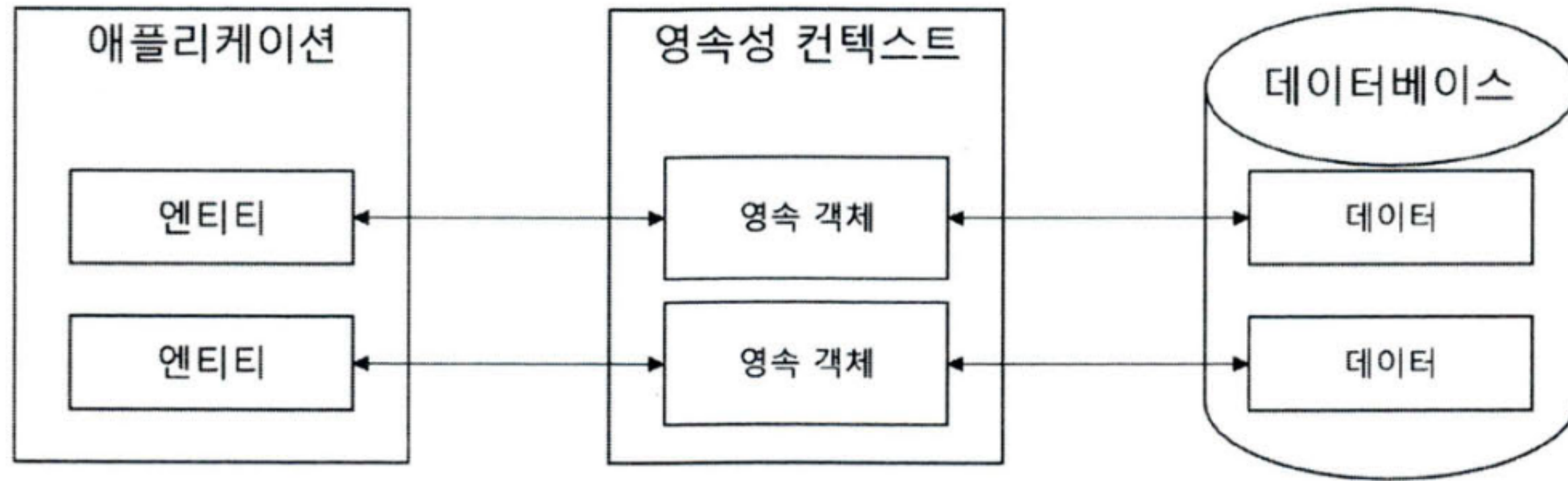
{ 영속성 컨텍스트 }

영속성 컨텍스트는 애플리케이션과 데이터베이스 사이에서 엔티티와 레코드의 **괴리를 해소하는** 기능과 **객체를 보관**하는 기능을 수행함.

엔티티 객체가 영속성 컨텍스트에 들어오면 JPA는 엔티티 객체의 매핑 정보를 데이터 베이스에 반영하는 작업을 수행함.

→ 엔티티 객체가 영속성 컨텍스트에 들어와 JPA의 관리 대상이 되는 시점부터 해당 객체를 **영속 객체**라고 부름.

{ 영속성 컨텍스트 }



{THANK YOU}