

JPA로 게시판 만들기

1 JPA의 개요와 설정

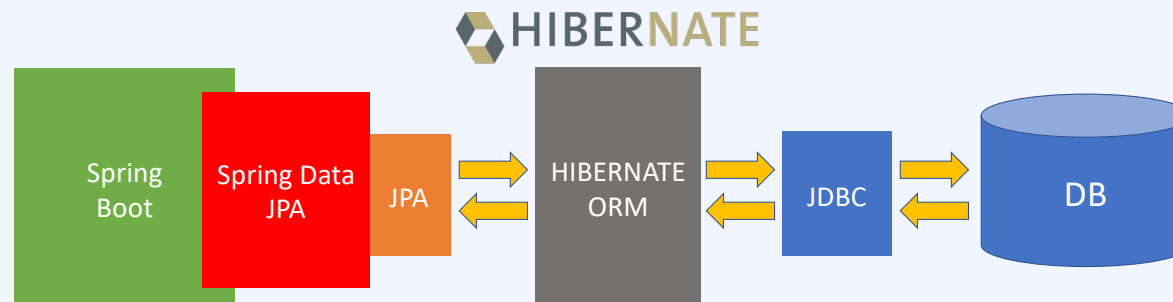
JPA의 개요와 설정

1. JPA(Java Persistence API)란?

ORM(Object-Relational Mapping)을 위한 Java 표준 API

JPA 구현체 : HIBERNATE , OpenJpa, EclipseLink, DataNucleus, ...

Persistence(영속성) : 애플리케이션 종료 후에도 객체(데이터)가 유지되는 것



JPA의 개요와 설정

3. 객체 모델과 관계형 모델의 비교

항목	객체 모델(object model)	관계형 모델(relational model)
구조	나누기 쉬움	나누기 어려움
상속	지원	지원 안함
비교	동일 비교(==)와 동등 비교(equals())	동일 비교(==)
조회	그래프(객체간의 참조 연결)	테이블(2차원 데이터)
관계	단방향(단방향x2로 양방향처리)	양방향(한쪽 테이블에만 FK지정)

4.

JPA로
게시판 만들기

JPA의 개요와 설정

4. SQL Mapper(MyBatis) vs. ORM framework(HIBERNATE)

4.

JPA로
게시판 만들기

```
public class BoardDto {
    private Integer bno;
    private String title;
    private String content;
    private String writer;
    private int view_cnt;
    private int comment_cnt;
    private Date reg_date;

    // 생성자, getter & setter, toString()
    // ...
}
```



MyBatis

```
<mapper namespace="com.fastcampus.ch4.dao.BoardMapper">
  <select id="select" parameterType="int" resultType="BoardDto">
    SELECT bno, title, content, writer
      , view_cnt, comment_cnt, reg_date
    FROM board
    WHERE bno = #{bno}
  </select>
</mapper>
```

```
@Entity
class Board {
    @Id @GeneratedValue
    private Long bno;
    private String title;
    private String content;
    private String writer;
    private Long viewCnt;
    private Long commentCnt;
    private
    //...
```

HIBERNATE
ORM

```
create table board (
    bno int auto_increment primary key,
    title varchar(45) not null,
    content text not null,
    writer varchar(30) not null,
    view_cnt int default 0 null,
    comment_cnt int default 0 null,
    reg_date datetime null
);
```

BOARD					

Hibernate: insert into user (email, in_date, name, password, up_date, id) values (?, ?, ?, ?, ?, ?)

JPA의 개요와 설정

5. Spring Data와 Spring Data JPA

4.

JPA로
게시판 만들기

Spring Data – 저장소 종류가 달라도 일관된 데이터 처리 방법을 제공

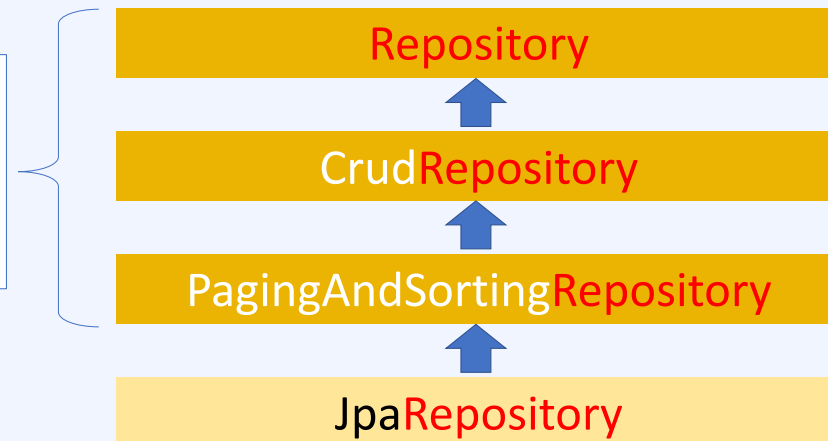
Spring Data JPA – JPA를 위한 저장소(JpaRepository)와 관련 기능을 제공



Spring Data

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.

[참고] <https://spring.io/projects/>



JPA로 게시판 만들기

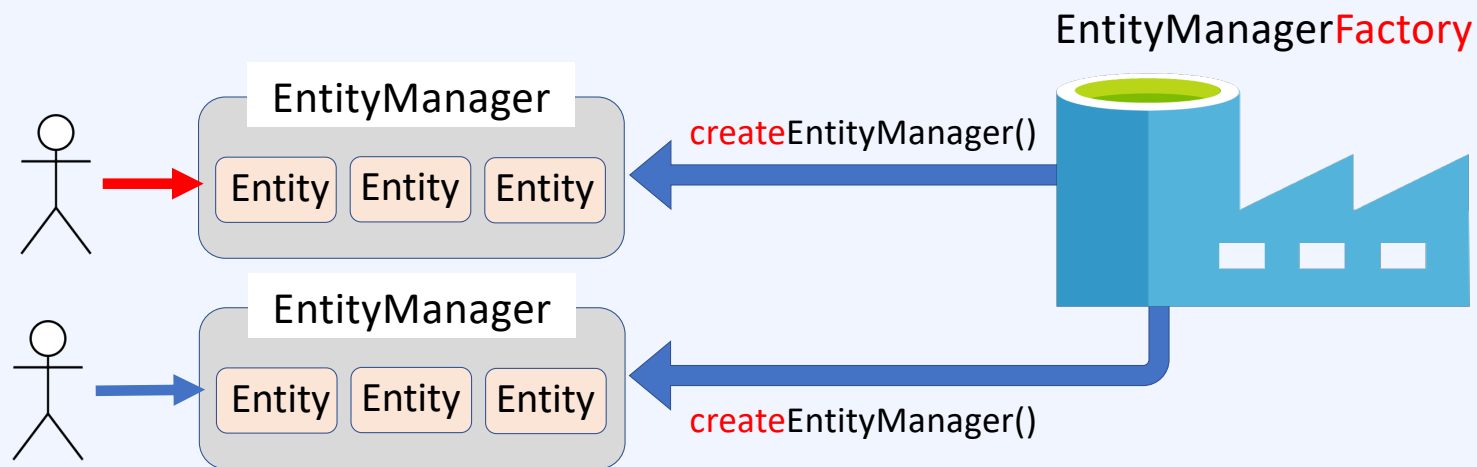
2 EntityManager와 EntityManagerFactory

EntityManager와 EntityManagerFactory

1. EntityManager와 EntityManagerFactory

EntityManagerFactory : EntityManager를 생성. 애플리케이션에 하나

EntityManager : Entity를 저장 관리. 사용자당 하나



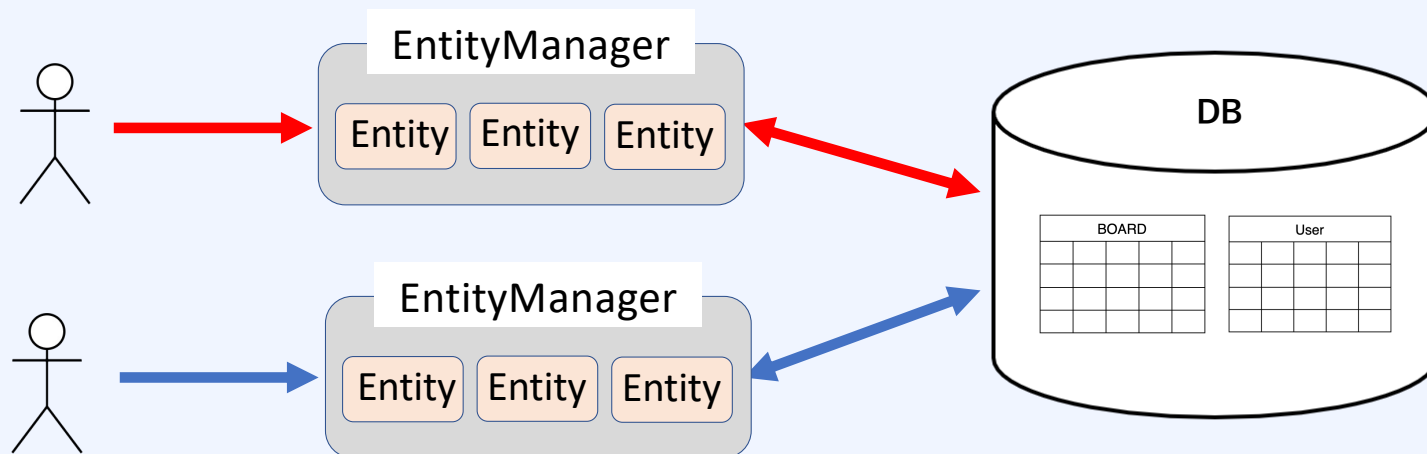
EntityManager와 EntityManagerFactory

1. EntityManager와 EntityManagerFactory

4.

JPA로
게시판 만들기

사용자는 직접 DB에 명령내리는 대신, EntityManager로만 작업
저장은 persist(), 조회는 find(), 삭제는 remove(), 변경은 Entity의 setter를 이용



EntityManager와 EntityManagerFactory

2. Entity클래스의 작성

Entity클래스 : DB테이블의 한 행(row)을 정의한 것

Entity클래스를 작성하고 @Entity를 붙인다.

Entity클래스에서 키(PK)로 사용할 속성에 @Id를 붙인다.

```
@Entity
public class User {
    @Id
    private String id;
    private String password;
    private String name;
    private String email;
    @Temporal(value=TemporalType.TIMESTAMP)
    private Date inDate; // 입력일
    @Temporal(value=TemporalType.TIMESTAMP)
    private Date upDate; // 변경일
```



```
create table user (
    id varchar(255) not null,
    email varchar(255),
    in_date datetime(6),
    name varchar(255),
    password varchar(255),
    up_date datetime(6),
    primary key (id)
) engine=InnoDB
```

4.

JPA로
게시판 만들기

EntityManager와 EntityManagerFactory

3. Entity클래스의 작성을 위한 애너테이션

4.

JPA로
게시판 만들기

애너테이션	설 명(기본값)
@Entity	Entity클래스에 사용
@Id	DB테이블의 PK로 사용할 속성에 사용
@Table	DB테이블에 대한 세부 정보 지정에 사용 name 테이블 이름을 직접 지정 catalog DB 카탈로그 지정 schema DB스키마 지정 uniqueConstraint 유일값 제약조건 지정
@Column	name 컬럼 이름(속성이름) unique 유일 값 제약조건 추가(false) nullable null입력 허용(true) insertable insert문에 포함여부(true) updatable update문에 포함여부(true) columnDefinition 직접 컬럼 속성 기술 length 문자열의 길이(255) precision 실수의 정밀도(0) scale 실수의 소수점 자리수(0)

애너테이션	설 명
@Enumerated	enum타입인 속성에 사용
@Temporal	java.util.Date 타입의 날짜 매핑에 사용 TemporalType.DATE 날짜만 TemporalType.TIME 시간만 TemporalType.TIMESTAMP 날짜&시간
@Transient	테이블의 컬럼에서 제외할 속성에 사용
@Generated	자동 번호 생성을 적용할 속성에 사용 strategy 자동 번호 생성 전략을 선택 generator 이미 생성된 키 생성기를 참조 GenerationType.TABLE DB테이블 사용 GenerationType.SEQUENCE 시퀀스(Oracle) 사용 GenerationType.IDENTITY IDENTITY(MySQL)사용 GenerationType.AUTO DB에 따라 자동선택

EntityManager와 EntityManagerFactory

4. EntityTransaction

Hibernate: insert into user (email, in_date, name, password, up_date, id) values (?, ?, ?, ?, ?, ?)

id	email	in_date	name	password	up_date
aaa	aaa@aaa.com	2023-05-06 08:27:25.885000	Lee	1234	2023-05-06 08:27:25.885000

```
EntityManager entityManager = entityManagerFactory.createEntityManager();  
EntityTransaction tx = entityManager.getTransaction(); // 트랜잭션 얻기
```

```
User user = new User(); // User객체를 생성하고 초  
user.setId("aaa");  
user.setPassword("1234");  
// ...
```

```
tx.begin(); // 트랜잭션을 시작  
entityManager.persist(user); // User객체를 저장  
tx.commit(); // 트랜잭션을 종료(작업 내용을 DB에 반영)
```

```
try {  
    tx.begin(); // 트랜잭션을 시작  
    entityManager.persist(user); // User객체를 저장  
    tx.commit(); // 트랜잭션을 종료(작업 내용을 DB에 반영)  
} catch (Exception e) {  
    tx.rollback(); // 트랜잭션을 종료(작업 취소)  
} finally {  
    entityManager.close();  
}
```

EntityManager와 EntityManagerFactory

5. Transaction이란?

4.

JPA로
게시판 만들기

더이상 나눌 수 없는 작업의 단위.

계좌 이체의 경우, 출금과 입금이 하나의 tx로 묶여야 됨.

'모'아니면 '도'. 출금과 입금이 모두 성공하지 않으면 실패



EntityManager와 EntityManagerFactory

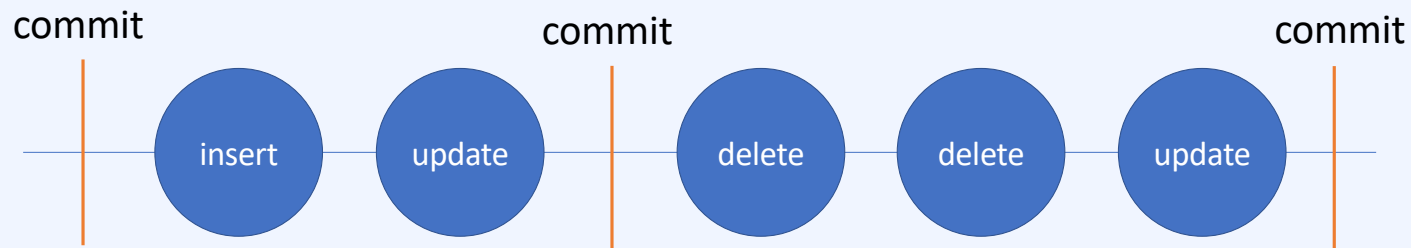
6. 커밋(commit)과 롤백(rollback)

4.

JPA로
게시판 만들기

커밋(commit) – 작업 내용을 DB에 영구적으로 저장

롤백(rollback) – 최근 변경사항을 취소(마지막 커밋으로 복귀)



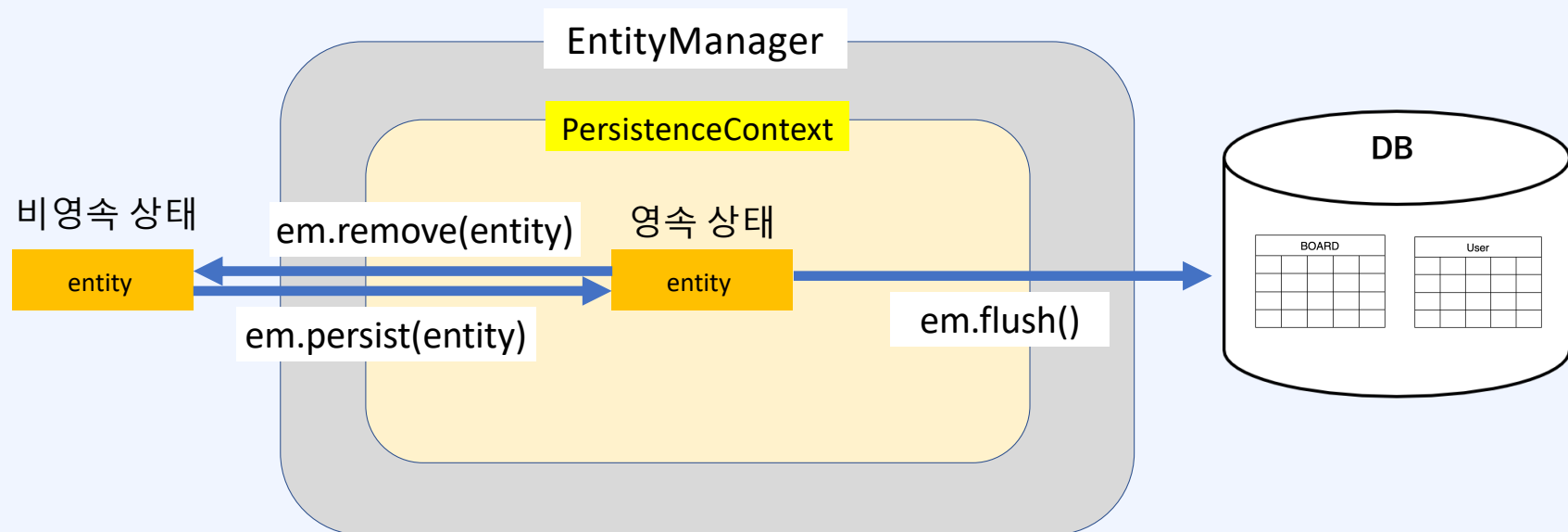
EntityManager와 EntityManagerFactory

7. PersistenceContext

4.

JPA로
게시판 만들기

Entity를 저장하는 공간. `persist(entity)`는 entity를 영속 상태로 변경
캐시, 변경 감지, 지연 로딩, 트랜잭션 지원



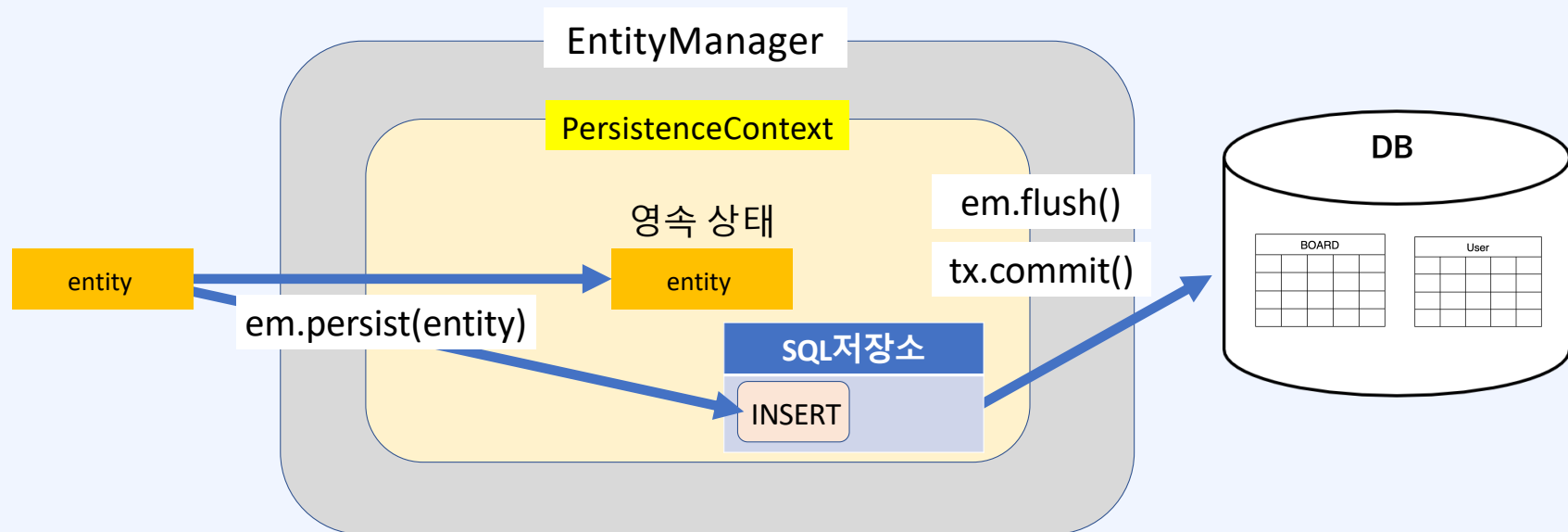
EntityManager와 EntityManagerFactory

8. flush()와 tx.commit()

4.

JPA로
게시판 만들기

persist()는 Entity를 영속 상태로 바꾸고, SQL저장소에 INSERT문을 저장
flush()는 SQL저장소에 누적된 SQL을 DB로 전달. tx.rollback() 가능
tx.commit()은 작업 내용을 DB에 영구 반영(flush() 자동 호출). tx.rollback() 불가



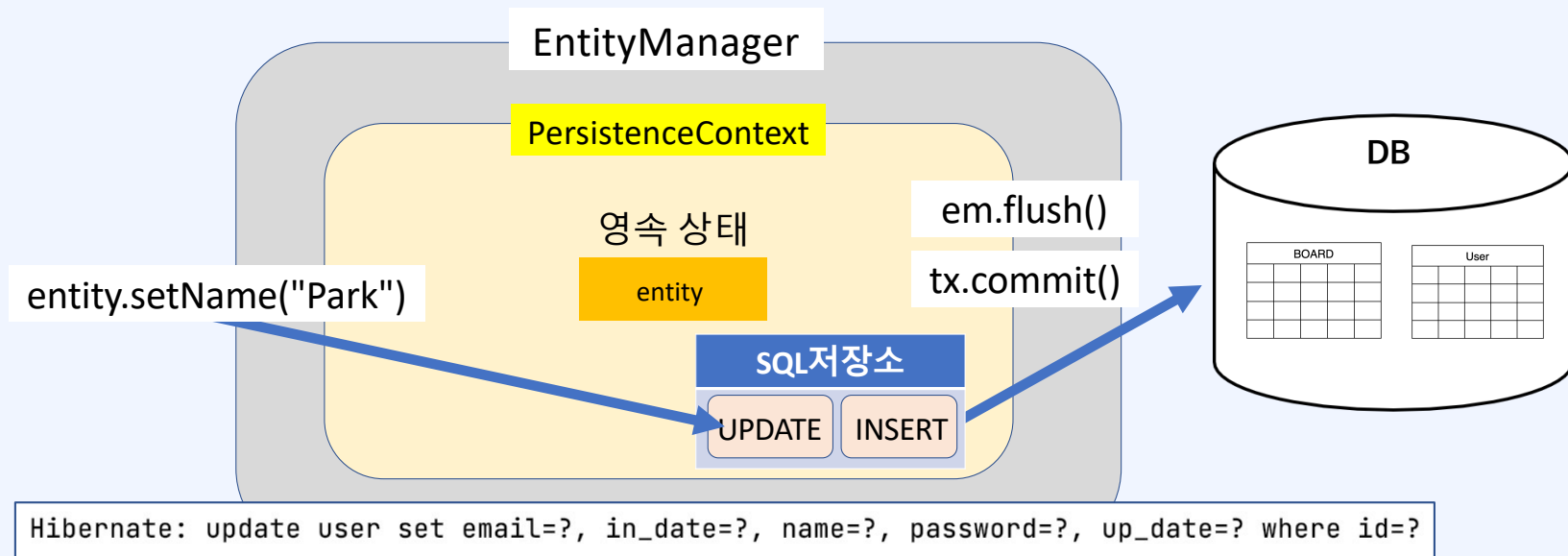
EntityManager와 EntityManagerFactory

8. flush()와 tx.commit()

4.

JPA로
게시판 만들기

persist()는 Entity를 영속 상태로 바꾸고, SQL저장소에 INSERT문을 저장
flush()는 SQL저장소에 누적된 SQL을 DB로 전달. tx.rollback() 가능
tx.commit()은 작업 내용을 DB에 영구 반영(flush() 자동 호출). tx.rollback() 불가



EntityManager와 EntityManagerFactory

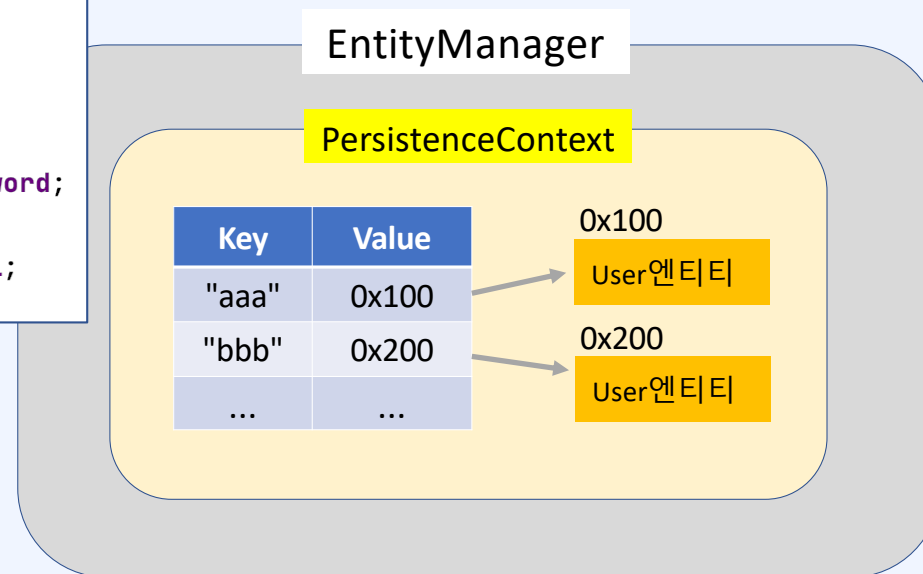
9. PersistenceContext의 캐시

4.

JPA로
게시판 만들기

실제 Entity가 저장되는 공간(Map). Entity의 @Id컬럼을 Key로 사용
캐시에서 Entity를 먼저 조회. 캐시에 없을 때만 DB에서 조회.

```
@Entity
public class User {
    @Id
    private String id;
    private String password;
    private String name;
    private String email;
}
```



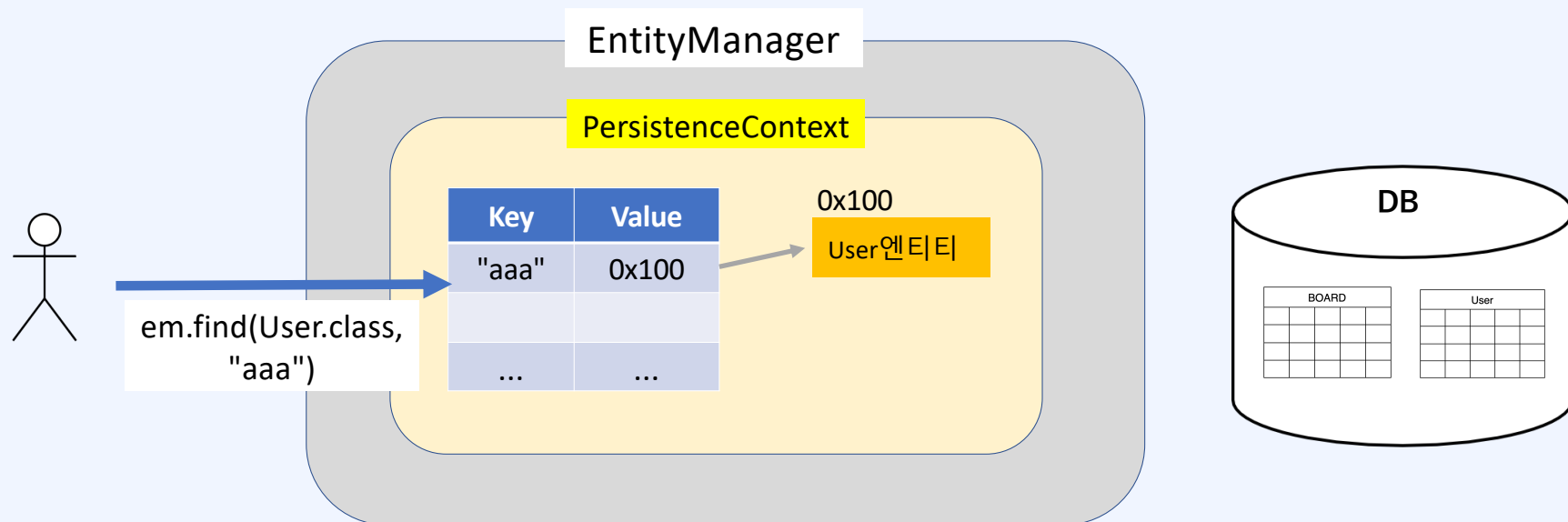
EntityManager와 EntityManagerFactory

9. PersistenceContext의 캐시

4.

JPA로
게시판 만들기

Entity가 저장되는 공간(Map). Entity의 @Id컬럼을 Key로 사용
캐시에서 Entity를 먼저 조회. 캐시에 없을 때만 DB에서 조회.



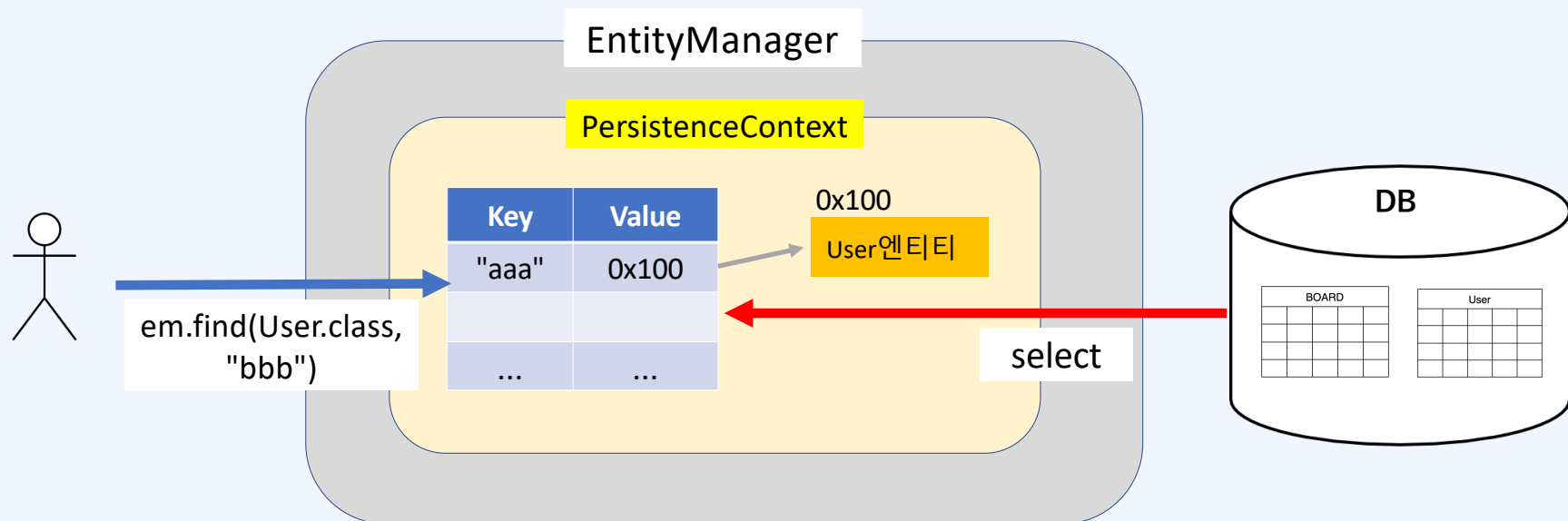
EntityManager와 EntityManagerFactory

9. PersistenceContext의 캐시

4.

JPA로
게시판 만들기

Entity가 저장되는 공간(Map). Entity의 @Id컬럼을 Key로 사용
캐시에서 Entity를 먼저 조회. 캐시에 없을 때만 DB에서 조회.



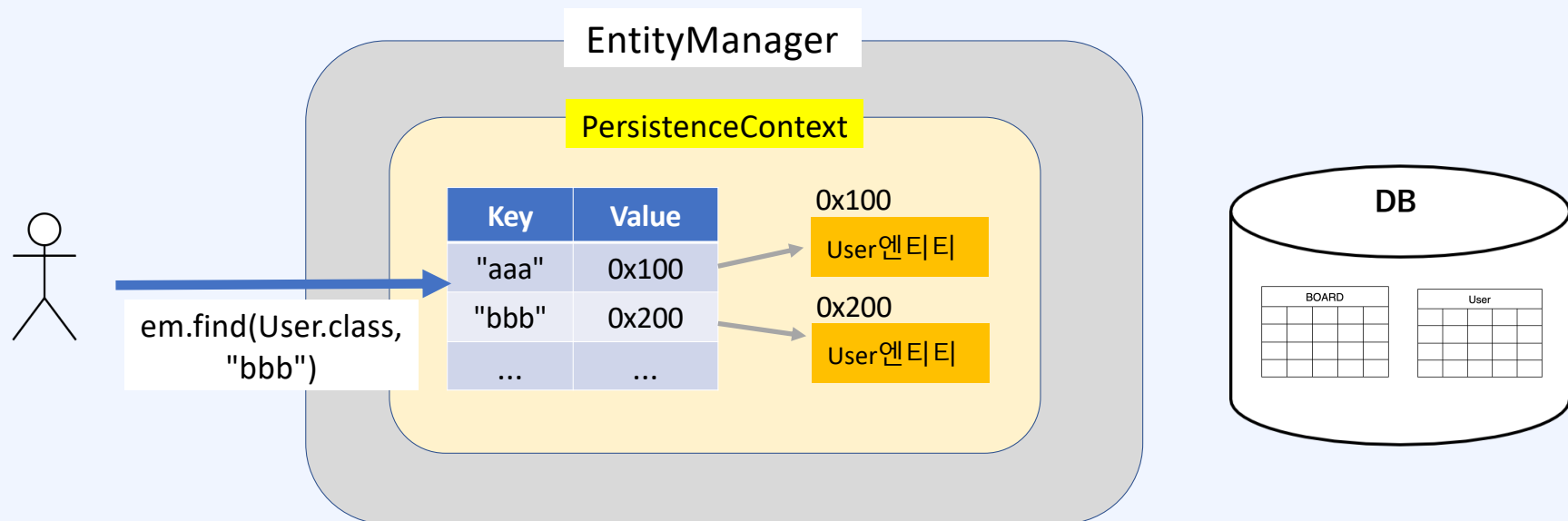
EntityManager와 EntityManagerFactory

9. PersistenceContext의 캐시

4.

JPA로
게시판 만들기

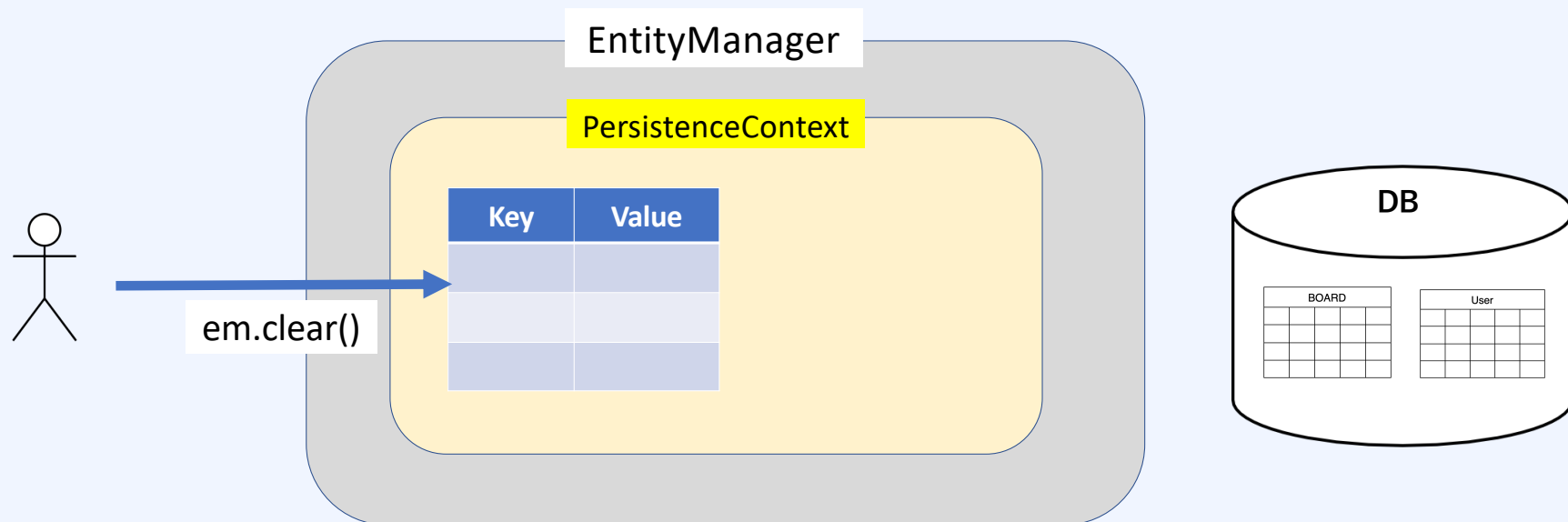
Entity가 저장되는 공간(Map). Entity의 @Id컬럼을 Key로 사용
캐시에서 Entity를 먼저 조회. 캐시에 없을 때만 DB에서 조회.



EntityManager와 EntityManagerFactory

9. PersistenceContext의 캐시

Entity가 저장되는 공간(Map). Entity의 @Id컬럼을 Key로 사용
캐시에서 Entity를 먼저 조회. 캐시에 없을 때만 DB에서 조회.



4.

JPA로
게시판 만들기