

## 2. 애플리케이션 개발

우선 xml 파일이랑 h2 설정부터...스브....

멤버 클래스에서 @Entity를 써놓으면 jpa를 사용하는 애구나 라고 인식을 해서 이걸 관리해야겠다고 생각함.

@Table, @Column 등으로 매핑 가능.

애플리케이션 개발 시 EntityManagerFactory라는 공장을 생성해서 EntityManager를 찍어낼 수 있다. EntityManagerFactory emf = Persistence.createEntityManagerFactory("hello");// 애플리케이션 로딩 시점에 딱 하나만 만들고 실제 뭔가 트랜잭션 단위가 오면 예로 고객이 어떤 행위를 하고 나가는 것 그럴 때 마다 디비 커넥션을 얻어서 쿼리를 날리고 종료되는 일반적인 단위를 할때마다 만들어주어야함

java는 트랜잭션이 진짜 중요함!

EntityManager entityManager = emf.createEntityManager(); //정말 쉽게 말하면 DB 커넥션 하나 받았다고 생각하면 됨.

//code

entityManager.close();

emf.close();

code 부분에 실제 동작하는 코드를 작성하게 된다. 디비에 데이터를 저장한다던가 불러온다던가..

그리고 실제 애플리케이션이 완전히 끝나면 EntityManagerFactory를 닫아줘야된다.

```
<property name="hibernate.show_sql" value="true"/>
<property name="hibernate.format_sql" value="true"/>
<property name="hibernate.use_sql_comments" value="true"/>
```

Hibernate:

```
/* insert hellojpa.Member
*/ insert
into
    Member
    (name, id)
values
    (?, ?)
```

쿼리문 작성 후 h2 db 보면 이 화면

실행Run Selected자동 완성지우기S

SELECT \* FROM MEMBER

SELECT \* FROM MEMBER;

ID	NAME
1	HelloA
2	HelloB

(2 행, 4 ms)

정석 코드는 try catch finally를 써야함

em.close(); //애가 db 커넥션을 물고 작동하므로 닫아주어야 함.

엔티티 매니저 팩토리는 하나만 생성해서 애플리케이션 전체에서 공유

엔티티 매니저는 스레드간에 공유X (사용하고 버려야 한다). 고객의 요청이 올 때 마다 썼다가 버렸다가 함...

**JPA**의 모든 데이터 변경은 트랜잭션 안에서 실행 → 결국 **RDB**는 데이터 변경 자체를 트랜잭션 안에서 처리하도록 하고있음. 트랜잭션 안걸어도 **db**자체에서 처리함

## JPQL

```
List<Member> result = em.createQuery("select m from Member",  
Member.class).getResultList();
```

jpql이 일반 sql과 다른 점은 절대로 테이블을 대상으로 쿼리를 짜지 않는다. MEMBER가 아닌 Member 인 이유가 그것임. 객체(엔티티)를 대상으로 짤다!!!1

객체를 대상으로 짜는 객체지향 쿼리. 방언을 설정하면 그 방언대로 알아서 바뀌줌.

```
<property name="javax.persistence.jdbc.url" value="jdbc:h2:tcp://localhost/~/test2"/>  
<property name="hibernate.dialect" value="org.hibernate.dialect.H2Dialect"/>  
<!-- 옵션 -->
```

방언 설정.

## 총정리

- jpa는 항상 entitymanagerfactory를 만들어야한다. 이게 db 테이블 하나하나에 묶여서 돌아감
- hello가 설정파일의 설정 읽어와서 만든다.
- entitymanager는 고객의 요청이 올 때마다 열어서 db작업을 해야하면 이걸 통해서
- jpa의 모든 데이터 변경은 트랜잭션 안에서!
- 데이터 변경이 있을 때는 무조건 트랜잭션 안에서
- commit을 꼭 해줘야 변경이 반영이 됨.
- em.close(), emf.close()를 꼭 해줘야 커넥션 풀링같은 리소스 릴리즈가 가능해진다.

#JPA