

□ Study3_1 들어가기 전 생각해보기

- Connection은 해당 DB 접속이라 그렇다 치고....
- DAO(persistence Layer)에서 Db사용 불편하지 않았나요?
 - StringBuffer : sb.append
 - pstmt.setString으로 하나하나 '?' 처리
 - rs.getString으로 하나하나 setting
- 이를 해결하기 위한 persistence Framework가 등장했는데...

□ ORM (Object-Relation Mapping)

- 객체 관계 매핑
- 객체와 RDB를 별개로 설계하고 ORM이
중간에서 매핑해주는 역할
- SQL쿼리문 작성x
 - 쿼리문 작성 안하니까
Mysql,oracle 상관없이 다 적용 가능

□ JPA (JAVA Persistence API)

- JAVA에서 ORM의 기술 API 표준명세
- ORM을 사용하기 위한 인터페이스모음
- 구현체 : Hibernate, EclipseLink,
DataNucleus 등

□ SQLMapper

- SQL문을 이용하여 RDB에 접근,
- 쿼리문 결과를 VO로 변환
1ROW=>1VO
- RDB에 따라 SQL 문법이 다르기 때문에 특정
RDB에 종속적이다.

□ Mybaits

- MyBatis는 자바에서 SQL Mapper를
지원해주는 Framework이다.
- 자세한 설명은 mybatis홈페이지 참고

	ORM	SQLMapper
장점	1) RDB에 종류와 관계없이 사용 가능하다. 2) 기본적인 CRUD 제공과 페이징 처리 등 상당 부분 구현되어 있어 비즈니스 로직에 집중할 수 있다. 3) 테이블 생성, 변경 등 엔티티 관리가 간편하다. 4) 쿼리에 집중할 필요 없어 빠른 개발이 가능하다. - SQL을 몰라도 된다는 뜻은 아니다. JPA는 SQL을 잘할수록 훨씬 더 잘 사용할 수 있다.	1) JPA에 비해 쉽다. : SQL 쿼리를 그대로 사용하기에 복잡한 Join, 튜닝 등을 좀더 수월하게 작성 가능하다. 2) SQL의 세부적인 내용 변경 시 좀 더 간편하다. 3) 동적 쿼리 사용 시 JPA보다 간편하게 구현 가능하다.
단점	1) 어렵다. : 단방향, 양방향, 임베디드 관계 등 이해해야할 내용이 많으며, 연관관계 이해 없이 잘못 코딩 했을 시 성능상의 문제와 동작이 원하대로 되지 않는 일이 발생한다.	1) 데이터 베이스 설정 변경 시 수정할 부분이 너무 많다. (Oracle의 페이징 쿼리를 MySQL에서 사용 불가능!) 2) Mapper작성부터 인터페이스 설계까지 JPA보다 많은 설계와 파일, 로직이 필요하다. 3) 특정 DB에 종속적이다.

한국은 SQLMapper우세, 특히 정부기관 그래서 저희는 SQLMapper인 mybatis 사용
 (정부와 별로 관계없는 대부분의 사기업들은 ORM 사용 추세)
 세계적으론 ORM 우세 (거의 95%)