

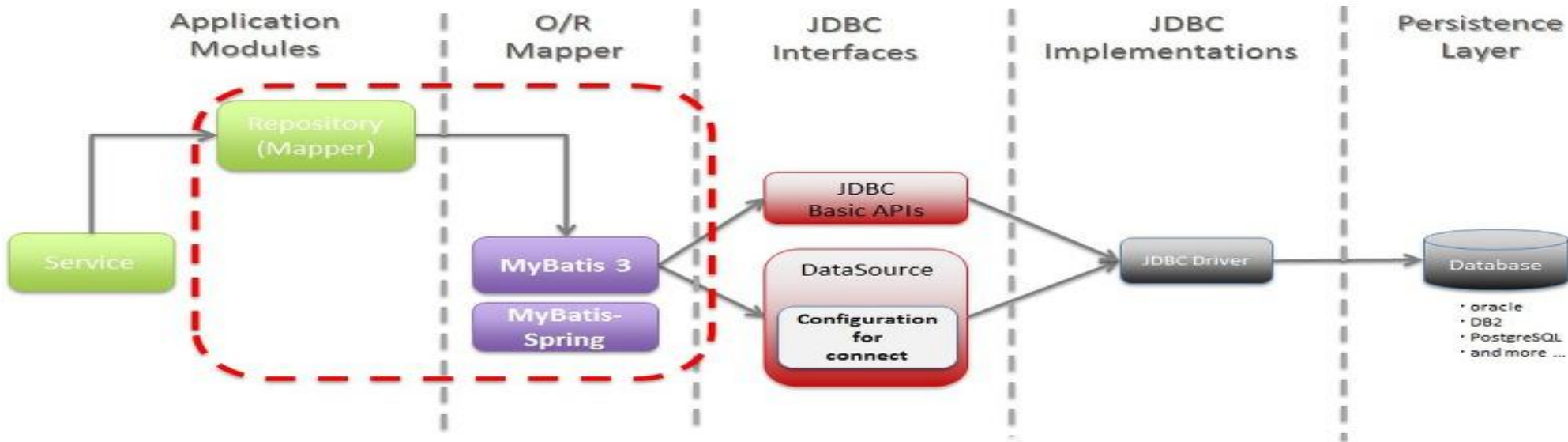


# Spring Boot

소프트웨어융합공학과  
웹서버 프로그래밍  
허우행

# MyBatis

- ▶ <http://mybatis.org/spring/>
- ▶ 복잡한 쿼리나 다이나믹 쿼리 적용
- ▶ 프로그램 코드와 SQL 쿼리의 분리로 코드의 간결성 및 유지보수성 향상
- ▶ `resultType`, `resultClass` 등 Vo를 사용하지 않고 조회결과를 사용자 정의 DTO, MAP 등으로 맵핑
- ▶ 빠른 개발이 가능하여 생산성 향상





# where 문



```
<select id="selectPatternMasterList" resultType="com.qhedge.online.stock.dto.PatternMasterDto">
  SELECT pattern_id,
    account_id,
    DATE_FORMAT(creation_date, '%Y-%m-%d') creation_date,
    pattern_type
  FROM pattern_master
  <where>
    <if test="fromDate != null">
      and open_date >= STR_TO_DATE(CONCAT(LEFT('${fromDate}', 7), '-01'), '%Y-%m-%d')
    </if>
    <if test="toDate != null">
      and open_date <= LAST_DAY(STR_TO_DATE(CONCAT(LEFT('${toDate}', 7), '-01'), '%Y-%m-%d'))
    </if>
  </where>
</select>
```



# set 문



```
<update id="updateAuthorIfNecessary">  
  update Author  
  <set>  
    <if test="username != null">username=#{username},</if>  
    <if test="password != null">password=#{password},</if>  
    <if test="email != null">email=#{email},</if>  
    <if test="bio != null">bio=#{bio}</if>  
  </set>  
  where id=#{id}  
</update>
```





# foreach 문



```
<select id="selectPostIn" resultType="domain.blog.Post">  
  SELECT *  
  FROM POST P  
  WHERE ID in  
  <foreach item="item" index="index" collection="list"  
    open="(" separator="," close=")">  
    #{item}  
  </foreach>  
</select>
```



# Java 코드로 작성

```
public String selectPersonSql() {  
    return new SQL()  
        .SELECT("P.ID", "A.USERNAME", "A.PASSWORD", "P.FULL_NAME", "D.DEPARTMENT_NAME", "C.COMPANY_NAME")  
        .FROM("PERSON P", "ACCOUNT A")  
        .INNER_JOIN("DEPARTMENT D on D.ID = P.DEPARTMENT_ID", "COMPANY C on D.COMPANY_ID = C.ID")  
        .WHERE("P.ID = A.ID", "P.FULL_NAME like #{name}")  
        .ORDER_BY("P.ID", "P.FULL_NAME")  
        .toString();  
}
```

```
public String insertPersonSql() {  
    return new SQL()  
        .INSERT_INTO("PERSON")  
        .INTO_COLUMNS("ID", "FULL_NAME")  
        .INTO_VALUES("#{id}", "#{fullName}")  
        .toString();  
}
```