

MyBatis

동적 SQL

▶ MyBatis 동적 SQL

✓ 동적 SQL이란 ?

일반적으로 **검색 기능이나 다중 입력 처리** 등을 수행해야 할 경우 SQL을 실행하는 DAO를 여러 번 호출하거나 batch 기능을 이용하여 버퍼에 담아서 한번에 실행 시키는 방식으로 쿼리를 구현했다면, MyBatis에서는 이를 **동적으로 제어할 수 있는 구문을 제공하여 좀 더 쉽게 쿼리를 구현** 할 수 있도록 하는 기능을 지원한다.

✓ 지원 구문 종류

1. if
2. choose (when, otherwise)
3. trim (where, set)
4. foreach

▶ MyBatis 동적 SQL

✓ if 구문

- 동적 쿼리를 구현할 때 가장 기본적으로 사용되는 구문
- 특정 조건을 만족할 경우 안의 구문을 쿼리에 포함시킴

✓ if 구문 예시

```
<select id="searchBoard" resultType="arraylist">
  SELECT *
  FROM BOARD
  WHERE WRITER = 'admin'
  <if test="title != null">
    AND TITLE LIKE = #{title}
  </if>
</select>
```

▶ MyBatis 동적 SQL

✓ 다중 if 구문

- 필요로 하는 조건이 1개 이상일 시 if 구문을 여러 개 사용 가능

✓ 다중 if 구문 예시

```
<select id="searchBoard" resultType="arraylist">
  SELECT *
  FROM BOARD
  WHERE WRITER = 'admin'
  <if test="title != null">
    AND TITLE LIKE = #{title}
  </if>
  <if test="location != null">
    AND LOCATION LIKE #{location}
  </if>
</select>
```

▶ MyBatis 동적 SQL

✓ choose, when, otherwise 구문

- 자바의 if-else, switch, JSTL의 choose 구문과 유사하며 주어진 구문 중 한 가지만을 수행하고자 할 때

```
<select id="searchBoard" resultType="arraylist">
  SELECT *
  FROM BOARD
  WHERE COMMENT != ''
  <choose>
    <when test="조건식">
      ...
    </when>
    <otherwise>
      ...
    </otherwise>
  </choose>
</select>
```

▶ MyBatis 동적 SQL

✓ choose, when, otherwise 구문 예시

```
<select id="searchBoard" resultType="arraylist">
  SELECT *
  FROM BOARD
  WHERE COMMENT != ''
  <choose>
    <when test="title != null">
      AND TITLE LIKE #{title}
    </when>
    <when test="writer != null">
      AND WRITER LIKE #{writer}
    </when>
    <otherwise>
      AND FEATURED = 1
    </otherwise>
  </choose>
</select>
```

▶ MyBatis 동적 SQL

✓ 동적 SQL의 안 좋은 예..

```
<select id="searchBoard" resultType="arraylist">
  SELECT *
  FROM BOARD
  WHERE
  <if test="writer != null">
    WRITER = #{writer}
  </if>
  <if test="title != null">
    TITLE = #{title}
  </if>
</select>
```

- 결과 1 : if 조건 어느 것도 만족하지 못했을 경우

SELECT * FROM BOARD

WHERE

- 결과 2 : 두 번째 if 조건 만을 만족할 경우

SELECT * FROM BOARD

WHERE AND TITLE = 'OOO'

▶ MyBatis 동적 SQL

✓ trim, where, set 구문

- <trim> : 쿼리 구문의 특정 부분을 없앨 때 쓰인다.
- <where> : 기존 쿼리의 WHERE 절을 동적으로 구현할 때 쓰인다.
- <set> : 기존의 UPDATE SET 절을 동적으로 구현할 때 쓰인다.

▶ MyBatis 동적 SQL

✓ where 구문

- 단순히 WHERE만을 추가하지만, 만약 태그 안의 내용이 'AND'나 'OR'로 시작할 경우 'AND'나 'OR'를 제거한다.

✓ where 구문 예시

```
<select id="searchBoard" resultType="arraylist">
    SELECT *
    FROM BOARD
    <where>
        <if test="writer != null">
            writer = #{writer}
        </if>
        <if test="title != null">
            AND TITLE = #{title}
        </if>
    </where>
</select>
```

▶ MyBatis 동적 SQL

✓ trim 구문을 통해 where 구현

- 태그 안의 내용이 완성될 때 처음 시작할 단어(prefix)와 시작 시 제거해야 할 단어(prefixOverrides)를 명시하여 where와 같은 내용으로 구현 할 수 있다.

✓ trim 구문 예시

```
<select id="searchBoard" resultType="arraylist">
  SELECT *
  FROM BOARD
  <trim prefix="WHERE" prefixOverrides="AND/OR">
    <if test="writer != null">
      writer = #{writer}
    </if>
    <if test="title != null">
      AND TITLE = #{title}
    </if>
  </trim>
</select>
```

▶ MyBatis 동적 SQL

✓ set 구문

- UPDATE 하고자 하는 컬럼을 동적으로 포함시키기 위해 사용한다.
- SET 키워드를 붙이고 불필요한 콤마를 제거한다.

✓ set 구문 예시

```
<update id="updateMember">
  UPDATE MEMBER
  <set>
    <if test="userName != null">
      USER_NAME = #{userName},
    </if>
    <if test="userPwd != null">
      USER_PWD = #{userPwd},
    </if>
  </set>
</update>
```

▶ MyBatis 동적 SQL

✓ trim 구문을 통해 set 구현

- WHERE와 흡사하나 suffixOverrides 속성을 ','로 설정하여 구문의 마지막에 제거할 값을 명시한다.

✓ trim 구문 예시

```
<update id="updateMember">
    UPDATE MEMBER
    <trim prefix="SET" suffixOverrides=",">
        <if test="userName != null">
            USER_NAME = #{userName},
        </if>
        <if test="userPwd != null">
            USER_PWD = #{userPwd},
        </if>
    </trim>
</update>
```

▶ MyBatis 동적 SQL

✓ foreach 구문

- 동적 쿼리를 구현할 때 collection에 대한 반복 처리를 제공한다.

✓ foreach 구문 예시

```
<select id="searchBoadwords" resultType="arraylist">
  SELECT *
  FROM BOARD
  WHERE TITLE IN
    <foreach item="item" index="index" collection="list"
      open="(" separator="," close=")">
      #{item}
    </foreach>
</select>
```

* 결과 SQL 생성

```
SELECT *
FROM BOARD
WHERE TITLE IN('XXX', '사행성', '욕설', ...)
```

▶ MyBatis 동적 SQL - 참고

✓ foreach 태그의 속성

속 성 명	설 명
item	반복 될 때 접근 가능한 각 객체 변수
index	반복되는 횟수를 가리키는 변수, 0부터 순차적으로 증가
collection	전달받은 인자로 반복에 쓰이는 Collection 객체, List나 Array형태만 가능
open	해당 구문이 시작될 때 삽입할 문자열
separator	반복되는 객체를 나열 할 때 사용할 구분자
close	해당 구문이 종료될 때 삽입할 문자열

▶ MyBatis 동적 SQL

✓ bind 구문

- 특정 문장을 미리 생성하여 쿼리에 적용해야 할 경우 사용한다.
- '_parameter'를 통해 전달받은 값에 접근하여 구문을 생성한다.

✓ bind 구문 예시

```
<select id="searchBoard" resultType="arraylist">  
  <bind name="pattern" value="'%' + _parameter.getTitle() + '%'" />  
  SELECT *  
  FROM BOARD  
  WHERE TITLE LIKE #{pattern}  
</select>
```