



210726 Mybatis

1교시

HomeWork확인 Mybatis를 사용하여
selectOne 기능 만들기

insert문 만들기

```
//글 작성
public int insert(MDBoardDto dto) {
    SqlSession session = null;
    int res = 0;

    try {
        session = getSqlSessionFactory().openSession(false);
        res = session.insert("com.my.multi.insert",dto);

        if(res>0) {
            session.commit();
        }
    } catch (Exception e) {

        e.printStackTrace();
    }finally {
        session.close();
    }

    return res;
}
```

- openSession(false)를 함으로 Auto Commit을 false처리 한 후 try~catch문 안에 조건문을 사용하여 commit 처리를 해주었다.

Mapper.xml 로 이동

```
<insert parameterType="MBoardDto" id="insert" >
    INSERT INTO MBOARD
    VALUES(MBOARDSEQ.NEXTVAL,#{seq},#{writer},#{title},#{content},SYSDATE)
</insert>
```

- parameterType은 넘어오는 값의 타입이 무엇인지 지정해주고 id값을 넣어준다.
- id는 dao에서 Mapper.xml의 namespace+"id"로 사용할 예정이다

-
- 다음은 update 방법이다

MBoardDao_old

```
//글 수정
public int update(MBoardDto dto) {
    SqlSession session = null;
    int res = 0;

    try {
        session = getSqlSessionFactory().openSession(false);
        res = session.update(namespace+"update",dto);

        if(res>0) {
            session.commit();
        }
    } catch (Exception e) {

        e.printStackTrace();
    }finally {
        session.close();
    }
    return res;
}
```

- 동일하게 session 객체를 만들어주고
- session객체를 사용하여 namespace + "update"를 실행할거다를 명시한후 dto는 현재 selectOne된 게시글의 정보를 매개변수를 이용하여 넘겨받고 있다.
- parameterType의경우 config.xml에서 Alias 를 통하여 별칭을 지정해주어서 MBoardDto로 사용하고있으며 별칭을 사용하지 않는다면 full package명을 작성하여도 된다. com.multi.dto.MBoardDto처럼

2교시

MBoardDao_old 로 이동

```
//글 삭제
public int delete(int seq) {
    SqlSession session = null;
    int res = 0;

    try {
        session = getSqlSessionFactory().openSession(false);
        res = session.delete("com.my.multi.delete", seq);

        if(res>0) {
            session.commit();
        }
    } catch (Exception e) {

        e.printStackTrace();
    }finally {
        session.close();
    }

    return res;
}
```

int seq 를 받아오는 메소드이다.

이때에도 namespace를 사용하여도 되지만

full package명을 작성하여 처리하였다.

multi_mapper.xml로 이동

```
<delete id="delete" parameterType="int">
    DELETE
    FROM MDBOARD
    WHERE SEQ = #{seq}
</delete>
```

MultiDelete checkBox사용

- hashMap을 사용하여 처리해보자.

Dao에서

```
//다중 삭제
public int MultiDelete(String[] seq) {
    int count = 0;
    Map<String,String[]> map = new HashMap<String,String[]>();
    map.put("seq", seq);

    SqlSession session = null;

    try {
        session = getSessionFactory().openSession(false);
        count = session.delete(namespace + "muldel",map);

        if(count == seq.length) {
            session.commit();
        }
    } catch (Exception e) {

        e.printStackTrace();
    }finally {
        session.close();
    }

    return count;
}
```

- count라는 변수를 생성해준후
- Map객체생성
- session Delete 실행시에 map을 넘겨준다.
- if(count==seq.length)는 몇개를 삭제하였고 삭제된 결과의 갯수가 같다면!

multi_mapper.xml

```
<delete id="muldel" parameterType="map">
    DELETE FROM MDBOARD
    WHERE SEQ IN
    <foreach collection="seq" item="seq" open="(" separator="," close=")">
        #{seq}
    </foreach>

</delete>
```

WHERE절에 IN

- WHERE 절 내에서 특정값 여러개를 선택하는 SQL 연산자;

을 사용한후 foreach(반복문)을 사용

collection = 선택된게시글 seq

item = 게시글의 변수 seq

(0,0,0,0,0)라는 형태로 쿼리문이 생성되어야 하기 때문에

open, separator, close 를 사용하였다

#{seq}는 foreach를 통하여 한개씩 넣어준다.

결과)

- WHERE SEQ IN (#{seq},#{seq},#{seq}#{seq}) 라는 형태로 들어가게 된다.

- 이 처럼 동적 sql을 사용할때 참고할 수 있는 링크

<https://mybatis.org/mybatis-3/ko/dynamic-sql.html>

