

폴더의 파일 열어 작업하기

요건:

특정 폴더의 파일을 모두 열어서 SQL에 있는 특정 문구로 시작하는 테이블명을 찾는다

쿼리당 하나 이상일 때는 하나로 카운트한다.

즉 쿼리하나에 하나의 API를 생성하여 타 도메인에서 현재 도메인의 테이블을 조회하는 것을 배제하여 DBMS를 분리할 수 있도록 한다.

Main 프로그램

```
package com.det;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class SocParsingWorker {

    public static String[] annotation_line = {"--", "//"};
    public static String[][] annotation_range = {

        {
            "/*", "*/", "/\\*", "\\*/"
        }

, {
            "<!--", "-->", "<!--", "-->"
        }

    };

    public static String[][] searchKeyCategoryArry = {

        { "<select", "</select>" }

, { "<insert", "</insert>" }

, { "<update", "</update>" }

, { "<delete", "</delete>" }

    };

    // 찾는 문자
    public static String findkeyString = "TB_FILEMANAGER_META";
```

```

public static void main(String[] args) throws IOException {
    // TODO Auto-generated method stub
    System.out.println(" START ");
    String fileName = "";
    List<String> docRemoveAnn = new ArrayList<String>();
    StringRemovAnno stringRemovAnno = new StringRemovAnno();
    SearchString ss = new SearchString();
    Map<String, Integer> resultMapg = new HashMap<String, Integer>(); //
SocParsingWorker.searchKeyArry.length
    Map<String, Integer> totalResultMapg = new HashMap<String, Integer>();

    /* 파일 경로에 있는 파일 가져오기 */
    File rw = new File(args[0]);

    /* 파일 경로에 있는 파일 리스트 fileList[] 에 넣기 */
    File[] fileList = rw.listFiles();

    /* fileList에 있는거 for 문 돌려서 출력 */
    for (File file : fileList) {
        if (file.isFile()) {
            fileName = file.getName();
            System.out.println("::::::::::::: PROC FILE NAME : " +
fileName + ":::::::::::::");

            // Remove Annotation
            // annotation_range
            // annotation_line
            docRemoveAnn =
stringRemovAnno.getStringRemovAnno(fileName, file.getPath());

            // 주석 제거된 문서 docRemoveAnn를 활용
            // searchKeyCategoryArray
            resultMapg = ss.getSearchCnt(docRemoveAnn);
            System.out.println("::::::::::::: File="+ fileName + " ,
Result=" + resultMapg + ":::::::::::::\n");

            // Total add count by searchKeyCategoryArray
            for (int i = 0; i <
SocParsingWorker.searchKeyCategoryArray.length; i++) {

totalResultMapg.put (SocParsingWorker.searchKeyCategoryArray[i][0].toUpperCase(),

toNlltoZero(totalResultMapg.get (SocParsingWorker.searchKeyCategoryArray[i]
[0].toUpperCase()))

+toNlltoZero(resultMapg.get (SocParsingWorker.searchKeyCategoryArray[i]
[0].toUpperCase()))

                );
            }
        }
    }
}

```



```

        removedAnnoRangeByte = removedAnnoLineByte(filename,
removedAnnoRangeByte, SocParsingWorker.annotation_line[i]);
    }

    return removedAnnoRangeByte;
}

/**
 * 영역 주식 제거
 * @param filename
 * @param fileByte
 * @return
 */
private List<String> removedAnnoRangeByte(String filename, List<String>
fileLineArry, String[] annRange) {
    System.out.println(" > Remove Anno Range : " + filename + " [" +
annRange[0] + " , " + annRange[1] + "]");
    List<String> removedAnnoRangeByte = new ArrayList<String>();

//
    System.out.println(" > ORG One Line : " + fileLineArry);

    //for ann
    boolean annFlag = false;

    for (String oneLine : fileLineArry) {
        int startAnnRag = oneLine.indexOf(annRange[0]);
        int endAnnRag = oneLine.indexOf(annRange[1]);

//
        System.out.println(" > RED One Line : " + oneLine + "
----->" + startAnnRag);

        // 한줄에 모두 있을 때
        if(startAnnRag >= 0 && endAnnRag >= 0){
            if(!annFlag) {
                // 사이를 넣어라
                String ffoneLine =
oneLine.substring(0, startAnnRag);

                removedAnnoRangeByte.add(ffoneLine);
                oneLine =
oneLine.substring(endAnnRag+annRange[1].length(), oneLine.length());
                removedAnnoRangeByte.add(oneLine);
            }

            // 시작이 있을 때
        }else if(startAnnRag >= 0) {
            if(oneLine.trim().equals(annRange[0])) {
                // skip ... 이상하네 정규 표현식에서 /* 만 있을 때 /*가 표시
됨

            }else {

```

```

oneLine =
oneLine.replaceAll(""+annRange[2]+".*([^*]|[\r\n])", "");
removedAnnoRangeByte.add(oneLine);
//
System.out.println(" > line : " + oneLine +
" [" + annRange[0] + " , " + annRange[1] + "] =====> START:" + startAnnRag );
}
annFlag = true;

// 끝이 있을 때
}else if(endAnnRag >= 0) {
oneLine = oneLine.replaceAll("."+annRange[3]+"*([^*]|
[\r\n])", "");
removedAnnoRangeByte.add(oneLine);
//
System.out.println(" > line : " + oneLine + " [" +
annRange[0] + " , " + annRange[1] + "] =====> START:" + startAnnRag );
annFlag = false;
}

// 둘다 없고 시작이 안되었을 때
if(startAnnRag < 0 && endAnnRag < 0){
if(!annFlag) {
removedAnnoRangeByte.add(oneLine);
}
}
}
//
System.out.println(" > FLT One Line : " + removedAnnoRangeByte);

//for ann
return removedAnnoRangeByte;
}

/**
 * Line 주석 제거
 * @param filename
 * @param removedAnnoRangeByte
 * @param string
 * @return
 */
private List<String> removedAnnoLineByte(String filename, List<String>
fileLineArray, String ann) {
System.out.println(" > Remove Anno Line : " + filename + " [" + ann +
" ]");
List<String> removedAnnoLineByte = new ArrayList<String>();
for (String oneLine : fileLineArray) {
int startAnnLine = oneLine.indexOf(ann);
//System.out.println(".....>>>>" + oneLine + " , " +
startAnnLine);
if( startAnnLine >= 0 ) {
oneLine = oneLine.substring(0,startAnnLine);
removedAnnoLineByte.add(oneLine);
}
}
}

```

```

        //System.out.println(" > RED One Line : " +
oneLine);
    }else {
        removedAnnoLineByte.add(oneLine);
    }
}
return removedAnnoLineByte;
}
}
}

```

문구 찾기

파일을 하나의 긴 문자열로 만들고, 카타고리로 split 배열로 나누고, 각 배열 문자열에 카테고리를 담은 문구까지 잘라서 문자열을 만든다.

즉 <select로 자르면 다음 <select까지 나누어지기 때문에 </select>가 나오는 곳까지 잘라낸다.

그리고 검색 문구를 찾는다. 하나이상 나오면 카운트

```

package com.det;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class SearchString {

    public Map<String, Integer> getSearchCnt(List<String> docRemoveAnn) {
        StringBuffer fileToOneLine = new StringBuffer();
        Map<String, Integer> resultMapg = new HashMap<String, Integer>();

        for (String onLineNoAnno : docRemoveAnn) {
            fileToOneLine.append(onLineNoAnno);
        }
//        System.out.println(fileToOneLine);

        // searchKeyArray 로 구분하여 배열로 정의
        for (int i = 0; i < SocParsingWorker.searchKeyCategoryArray.length; i+
+) {
//            System.out.println(" >> SEARCH STAR ! key=\"" +
SocParsingWorker.searchKeyArray[i][0] + "\" ... \"" + SocParsingWorker.searchKeyArray[i]
[1] + "\"");

            int searchCnt = 0;
            String[] searchArray =
fileToOneLine.toString().toUpperCase().split(SocParsingWorker.searchKeyCategoryArray[i]
[0].toUpperCase());

            for (int j = 0; j < searchArray.length; j++) {

```

```

                String s = searchArry[j];
//                System.out.println("    > search : " + s);
                if(s.indexOf(SocParsingWorker.searchKeyCategoryArry[i]
[1].toUpperCase()) >= 0) {
                    System.out.println("    >> Find! (" + j + ")" +
SocParsingWorker.searchKeyCategoryArry[i][0].toUpperCase() + "-" +
SocParsingWorker.searchKeyCategoryArry[i][1].toUpperCase()
                        + " " +
s.indexOf(SocParsingWorker.searchKeyCategoryArry[i][1].toUpperCase()) + " Col");
                    s = s.substring(0,
s.indexOf(SocParsingWorker.searchKeyCategoryArry[i][1].toUpperCase()));
                    System.out.println("        \_\_ TEXT = \"" + s.substring(0,50)
+ " ... \");
                    if(s.indexOf(SocParsingWorker.findkeyString) >= 0) {
//                        System.out.println("    >> target String : " + s);
                        searchCnt++;
                    }
                }
            }
            System.out.println("    >> ::::::::::: key=\"" +
SocParsingWorker.searchKeyCategoryArry[i][0] + "-" +
SocParsingWorker.searchKeyCategoryArry[i][1] + "\" ,
FIND(\""+SocParsingWorker.findkeyString+"\") Total Cnt = " + searchCnt);
            resultMapg.put (SocParsingWorker.searchKeyCategoryArry[i]
[0].toUpperCase(), searchCnt);
        }
        return resultMapg;
    }
}
}

```

처리 파일

개발한 프로그램의 [metaMapper.xml](#) 파일과 임의 테스트용 파일을 처리함

metaMapper.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/
mybatis-3-mapper.dtd">

<mapper namespace="com.joang.filemanager.meta.metaMapper">

<!-- ##### ===== COMMON START
===== ##### -->
<!-- ### For Paging ### -->
<sql id="pagingHeader">

```

```
SELECT A.* FROM (
SELECT row_number() over() AS rownumber, A.* FROM (
</sql>
```

```
<sql id="pagingFooter">
```

```
) A
```

```
) A WHERE A.rownumber BETWEEN #{start} AND #{last}
```

```
</sql>
```

```
<!-- ### For Paging ### -->
```

```
<!-- ##### ===== COMMON END
```

```
===== ##### -->
```

```
<!-- ##### ===== Meta List
```

```
Select ===== ##### -->
```

```
<select id="selectMetaListCnt"
```

```
parameterType="com.joang.filemanager.vo.MetaVO" resultType="int">
```

```
<![CDATA[
```

```
SELECT COUNT( DISTINCT A.META ) AS total_pages
```

```
FROM TB_FILEMANAGER_META A, TB_FILEMANAGER_META_MAPPING B,
```

```
TB_FILEMANAGER_META_FILE C, TB_USERS_MAPPING D
```

```
WHERE 1=1
```

```
AND A.IDX = B.META_IDX
```

```
AND B.FILE_IDX = C.IDX
```

```
AND C.IDX = D.FILE_IDX
```

```
]]>
```

```
AND D.USERNAME = #{userId}
```

```
AND A.DEL_GB = 'N'
```

```
AND C.DEL_GB = 'N'
```

```
<!-- 전체 -->
```

```
<if
```

```
test="positiveSearchKey != null and positiveSearchKey.size != 0 ">
```

```
<foreach collection="positiveSearchKey" item="item"
```

```
open="AND (" close=")" separator="AND">
```

```
A.META LIKE CONCAT('%',#{item},'%')
```

```
</foreach>
```

```
</if>
```

```
<if
```

```
test="negativeSearchKey != null and negativeSearchKey.size != 0 ">
```

```
AND C.IDX NOT IN (
```

```
SELECT DISTINCT B.FILE_IDX
```

```
FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
```

```
WHERE C.IDX = B.META_IDX
```

```
<foreach collection="negativeSearchKey" item="item"
```

```
open="AND (" close=")" separator="OR">
```

```
C.META LIKE CONCAT('%',#{item},'%')
```

```
</foreach>
```

```
)
```

```
</if>
```

```
<if
```

```
test="andSearchKey != null and andSearchKey.size != 0 ">
```

```
AND C.IDX IN (
```

```
SELECT DISTINCT B.FILE_IDX
```

```
FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
```

```

WHERE C.IDX = B.META_IDX
<foreach collection="andSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
</select>

<select id="selectMetaList"
parameterType="com.joang.filemanager.vo.MetaVO"
resultType="com.joang.filemanager.vo.MetaVO">
<include refid="pagingHeader"></include>
<![CDATA[
/* com.joang.filemanager.meta.metaMapper.selectMetaList */
SELECT
A.IDX,
A.META,
A.CREA_DTM,
A.CREA_ID,
COUNT( A.META ) AS COUNT_META
FROM
TB_FILEMANAGER_META A, TB_FILEMANAGER_META_MAPPING B,
TB_FILEMANAGER_META_FILE C, TB_USERS_MAPPING D
WHERE
A.DEL_GB = 'N'
AND C.DEL_GB = 'N'
AND A.IDX = B.META_IDX
AND B.FILE_IDX = C.IDX
AND C.IDX = D.FILE_IDX
]]>
AND D.USERNAME = #{userId}

<!-- 전체 -->
<if
test="positiveSearchKey != null and positiveSearchKey.size != 0 ">
<foreach collection="positiveSearchKey" item="item"
open="AND (" close=")" separator="AND">
A.META LIKE CONCAT('%',{item},%')
</foreach>
</if>
<if
test="negativeSearchKey != null and negativeSearchKey.size != 0 ">
AND C.IDX NOT IN (
SELECT DISTINCT B.FILE_IDX
FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="negativeSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
<if
test="andSearchKey != null and andSearchKey.size != 0 ">
AND C.IDX IN (
SELECT DISTINCT B.FILE_IDX

```

```

FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="andSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
<![CDATA[
GROUP BY A.IDX,
A.META,
A.CREA_DTM,
A.CREA_ID
ORDER BY A.IDX DESC
]]>
<include refid="pagingFooter"></include>
</select>

<select id="selectMetaAutocomplete"
parameterType="com.joang.filemanager.vo.MetaVO" resultType="map">
<![CDATA[
/* com.joang.filemanager.meta.metaMapper.selectMetaAutocomplete */
SELECT
A.IDX,
A.META,
A.FILE_CNT,
A.TO_CNT,
COUNT(E.FROM_IDX) AS FROM_CNT
FROM(
SELECT A.IDX,
A.META,
A.FILE_CNT,
COUNT(E.FROM_IDX) AS TO_CNT
FROM (
SELECT
A.IDX,
A.META,
COUNT( A.META ) AS FILE_CNT
FROM
TB_FILEMANAGER_META A, TB_FILEMANAGER_META_MAPPING B,
TB_FILEMANAGER_META_FILE C, TB_USERS_MAPPING D
WHERE
A.DEL_GB = 'N'
AND C.DEL_GB = 'N'
AND A.IDX = B.META_IDX
AND B.FILE_IDX = C.IDX
AND C.IDX = D.FILE_IDX
]]>
AND D.USERNAME = #{userId}
AND A.META LIKE '%'|| #{searchKey} || '%'

<![CDATA[
GROUP BY A.IDX, A.META
) A
left join TB_FILEMANAGER_META_REF E on A.IDX = E.FROM_IDX
GROUP BY A.IDX, A.META, A.FILE_CNT, E.FROM_IDX

```

```
) A
left join TB_FILEMANAGER_META_REF E on A.IDX = E.TO_IDX
GROUP BY A.IDX, A.META, A.FILE_CNT, A.TO_CNT, E.TO_IDX
```

```
ORDER BY A.IDX DESC
]]>
</select>
```

```
<select id="selectUserAutocomplete"
parameterType="com.joang.common.vo.UserVO" resultType="map">
<![CDATA[
/* com.joang.filemanager.meta.metaMapper.selectUserAutocomplete */
SELECT
tu.username
, tu.name
FROM tb_users tu
WHERE 1=1
AND ( tu.name LIKE '%||#{username}||%' OR tu.username LIKE '%||#{username}||%' )
AND tu.del_gb = 'N'
]]>
</select>
```

```
<select id="selectRefMetaList"
parameterType="com.joang.filemanager.vo.MetaVO"
resultType="com.joang.filemanager.vo.RefMetaVO">
<![CDATA[
/* com.joang.filemanager.meta.metaMapper.selectFromRefMetaList */
SELECT A.IDX, A.META, B.FROM_IDX, B.TO_IDX, B.META_REF
FROM TB_FILEMANAGER_META A,
(
SELECT TO_IDX AS IDX, FROM_IDX, B.TO_IDX, B.META_REF
FROM TB_FILEMANAGER_META A, TB_FILEMANAGER_META_REF B,
TB_FILEMANAGER_META_FILE C
WHERE
A.DEL_GB = 'N'
AND A.IDX = C.IDX
AND C.DEL_GB = 'N'
)]>
```

```
<!-- 전체 -->
<if
test="positiveSearchKey != null and positiveSearchKey.size != 0 ">
<foreach collection="positiveSearchKey" item="item"
open="AND (" close=")" separator="OR">
A.META LIKE CONCAT('%',#{item},'%')
</foreach>
</if>
<if
test="negativeSearchKey != null and negativeSearchKey.size != 0 ">
AND C.IDX NOT IN (
SELECT DISTINCT B.FILE_IDX
```

```

FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="negativeSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
<if
test="andSearchKey != null and andSearchKey.size != 0 ">
AND C.IDX IN (
SELECT DISTINCT B.FILE_IDX
FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="andSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
<![CDATA[
AND A.IDX = B.FROM_IDX
UNION
SELECT FROM_IDX AS IDX, FROM_IDX, B.TO_IDX, B.META_REF
FROM TB_FILEMANAGER_META A, TB_FILEMANAGER_META_REF B,
TB_FILEMANAGER_META_FILE C
WHERE
A.DEL_GB = 'N'
AND A.IDX = C.IDX
AND C.DEL_GB = 'N'

]]>

<!-- 전체 -->
<if
test="positiveSearchKey != null and positiveSearchKey.size != 0 ">
<foreach collection="positiveSearchKey" item="item"
open="AND (" close=")" separator="OR">
A.META LIKE CONCAT('%',{item},%')
</foreach>
</if>
<if
test="negativeSearchKey != null and negativeSearchKey.size != 0 ">
AND C.IDX NOT IN (
SELECT DISTINCT B.FILE_IDX
FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="negativeSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
<if
test="andSearchKey != null and andSearchKey.size != 0 ">
AND C.IDX IN (
SELECT DISTINCT B.FILE_IDX

```

```

FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="andSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
<![CDATA[
AND A.IDX = B.TO_IDX
) B
WHERE A.IDX = B.IDX
]]>
</select>

```

```

<select id="selectMetaValueByMetaIdx" parameterType="int"
resultType="string">
SELECT META FROM TB_FILEMANAGER_META
WHERE IDX = #{meta_idx}
</select>

```

```

<!-- ##### ===== Meta List
Select ===== ##### -->

```

```

<!-- ##### ===== File List/Detail
Select ===== ##### -->

```

```

<select id="selectFileListCnt"
parameterType="com.joang.filemanager.vo.FileVO" resultType="int">
<![CDATA[
SELECT COUNT( DISTINCT A.IDX ) AS TOTAL_PAGES
]]>

```

```

<choose>
<when test="searchKey != null and searchKey != "">
FROM TB_FILEMANAGER_META_FILE A, TB_FILEMANAGER_META_MAPPING B,
TB_FILEMANAGER_META C, TB_USERS_MAPPING D
</when>
<otherwise>
FROM
TB_FILEMANAGER_META_FILE A, TB_USERS_MAPPING D
</otherwise>
</choose>
WHERE
A.DEL_GB ='N'
AND A.IDX = D.FILE_IDX
AND D.USERNAME = #{crealId}

```

```

<choose>
<when test="searchKey != null and searchKey != "">
AND B.FILE_IDX = A.IDX
AND C.IDX = B.META_IDX

```

```

</when>
</choose>
<!-- 전체 -->
<if
test="positiveSearchKey != null and positiveSearchKey.size != 0 ">
<foreach collection="positiveSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
</if>
<if
test="negativeSearchKey != null and negativeSearchKey.size != 0 ">
AND A.IDX NOT IN (
SELECT DISTINCT B.FILE_IDX
FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="negativeSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
<if
test="andSearchKey != null and andSearchKey.size != 0 ">
AND A.IDX IN (
SELECT DISTINCT B.FILE_IDX
FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="andSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
</select>

<select id="selectFileList"
parameterType="com.joang.filemanager.vo.FileVO"
resultType="com.joang.filemanager.vo.FileVO">
<include refid="pagingHeader"></include>
<![CDATA[
/* com.joang.filemanager.meta.metaMapper.selectFileList */
SELECT
A.IDX,
E.OPN_CNT,
A.FILE_TITLE,
A.FILE_TYPE,
SUBSTRING(A.FILE_DETAIL::varchar,0,100) || '...' AS FILE_DETAIL,
A.ORIGINAL_FILE_NAME,
A.STORED_FILE_NAME,
A.FILE_SIZE,
A.CREA_DTM,
A.CREA_ID,
A.DEL_GB,
A.DOWN_YN,
A.STAR_LEVEL
]]>

```

```

<choose>
<when test="searchKey != null and searchKey != "" ">
, C.META AS SEARCH_KEY
FROM
TB_FILEMANAGER_META_FILE A, TB_FILEMANAGER_META_MAPPING B,
TB_FILEMANAGER_META C, TB_USERS_MAPPING D
</when>
<otherwise>
FROM
TB_FILEMANAGER_META_FILE A, TB_USERS_MAPPING D
</otherwise>
</choose>
, (SELECT COUNT(IDX) AS OPN_CNT, FILE_IDX FROM TB_USERS_MAPPING TUM GROUP BY
FILE_IDX ) E

    <![CDATA[
        WHERE
            A.DEL_GB ='N'
            AND A.IDX = D.FILE_IDX
            AND A.IDX = E.FILE_IDX
    ]]>
AND D.USERNAME = #{creald}

<choose>
<when test="searchKey != null and searchKey != "" ">
AND B.FILE_IDX = A.IDX
AND C.IDX = B.META_IDX
</when>
</choose>
<!-- 전체 -->
<if
test="positiveSearchKey != null and positiveSearchKey.size != 0 ">
<foreach collection="positiveSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',#{item},'%')
</foreach>
</if>
<if
test="negativeSearchKey != null and negativeSearchKey.size != 0 ">
AND A.IDX NOT IN (
SELECT DISTINCT B.FILE_IDX
FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="negativeSearchKey" item="item"
open="AND (" close=")" separator="OR">
C.META LIKE CONCAT('%',#{item},'%')
</foreach>
)
</if>
<if
test="andSearchKey != null and andSearchKey.size != 0 ">
AND A.IDX IN (
SELECT DISTINCT B.FILE_IDX
FROM TB_FILEMANAGER_META_MAPPING B, TB_FILEMANAGER_META C
WHERE C.IDX = B.META_IDX
<foreach collection="andSearchKey" item="item"
open="AND (" close=")" separator="OR">

```

```

C.META LIKE CONCAT('%',{item},%')
</foreach>
)
</if>
<choose>
<when test="searchKey != null and searchKey != "" ">
<![CDATA[
ORDER BY CONCAT(A.STAR_LEVEL, A.IDX)
]]>
</when>
<otherwise>
  <![CDATA[
ORDER BY A.IDX
  ]]>
</otherwise>
</choose>
<choose>
<when test="orderBy == 'ASC'">
  <![CDATA[
ASC
  ]]>
</when>
<otherwise>
  <![CDATA[
DESC
  ]]>
</otherwise>
</choose>
<include refid="pagingFooter"></include>
</select>

<!-- file detail -->
<select id="selectFileDetail" parameterType="int"
resultType="com.joang.filemanager.vo.FileVO">
  <![CDATA[
/* com.joang.filemanager.meta.metaMapper.selectFileDetail */
SELECT
IDX,
E.OPN_CNT,
FILE_TITLE,
FILE_TYPE,
FILE_DETAIL,
ORIGINAL_FILE_NAME,
STORED_FILE_NAME,
FILE_SIZE,
CREA_DTM,
CREA_ID,
DEL_GB,
DOWN_YN,
ERROR,
STAR_LEVEL
FROM TB_FILEMANAGER_META_FILE, (SELECT COUNT(IDX) AS OPN_CNT, FILE_IDX FROM
TB_USERS_MAPPING TUM GROUP BY FILE_IDX ) E
WHERE IDX = #{idx}
AND E.FILE_IDX = #{idx}
AND DEL_GB = 'N'

```

```
]]>
</select>
```

```
<select id="selectFileShareUserListByFile" parameterType="int"
resultType="com.joang.filemanager.vo.FileShareUserVO">
  <![CDATA[
    SELECT
    B.USERNAME,
    C.NAME,
    B.FILE_IDX,
    B.CREA_DTM,
    B.CREA_ID
    FROM TB_FILEMANAGER_META_FILE A, TB_USERS_MAPPING B, TB_USERS C
    WHERE 1=1
    AND A.IDX = #{idx}
    AND A.IDX = B.FILE_IDX
    AND C.USERNAME = B.USERNAME
  ]]>
</select>
```

```
<select id="getTargetMetaldx" resultType="string">
SELECT COALESCE(MAX(
IDX ), 0) FROM TB_FILEMANAGER_META_FILE
</select>
```

```
<!-- ##### ===== File List/Detail
Select =====>
```

```
<!-- ##### ===== Meta INSERT/UPDATE
Select =====>
```

```
<select id="checkMeta" parameterType="String"
resultType="hashmap">
<![CDATA[
  SELECT
  idx
  FROM
  TB_FILEMANAGER_META
  WHERE
  DEL_GB ='N'
  AND TRIM(META) = TRIM(#{meta})
]]>
</select>
```

```
<insert id="insertMeta" parameterType="hashmap">
<![CDATA[
  INSERT INTO TB_FILEMANAGER_META
  (
  USER_ID, META, CREA_DTM, CREA_ID, DEL_GB
  )
  VALUES
  (
  #{userId},
  #{meta},
```

```

        NOW(),
        #{userId},
        'N'
    )
]]>
<selectKey keyProperty="meta_idx" resultType="int"
order="AFTER">
SELECT COALESCE(MAX( IDX ), 0) AS META_IDX FROM TB_FILEMANAGER_META
</selectKey>
</insert>

<insert id="insertMetaMapping" parameterType="hashmap">
<selectKey keyProperty="idx" resultType="int" order="BEFORE">
SELECT COALESCE(MAX( IDX ), 0) FROM TB_FILEMANAGER_META_FILE
</selectKey>
<![CDATA[
    INSERT INTO TB_FILEMANAGER_META_MAPPING
    (
        USER_ID, META_IDX, FILE_IDX, CREA_DTM, CREA_ID, DEL_GB
    )
    VALUES
    (
        #{userId},
        #{meta_idx}::INTEGER,
]]>
<choose>
<when test="file_idx != null">
<!-- 변경되어야할 파일의 file_idx -->
#{file_idx}::INTEGER,
</when>
<otherwise>
<!-- 새 파일 업로드인 경우 idx -->
#{idx}::INTEGER,
</otherwise>
</choose>
<![CDATA[
    NOW(),
    #{userId},
    'N'
    )
]]>
</insert>

<insert id="insertMetaExistMapping" parameterType="hashmap">
<![CDATA[
    INSERT INTO TB_FILEMANAGER_META_MAPPING
    (
        USER_ID, META_IDX, FILE_IDX, CREA_DTM, CREA_ID, DEL_GB
    )
    VALUES
    (
        #{userId},
        #{meta_idx}::INTEGER,
        #{idx}::INTEGER,
        NOW(),
        #{userId},
        'N'
    )
]]>

```

```
)  
]]>  
</insert>
```

```
<insert id="insertShareUser" parameterType="com.joang.filemanager.vo.FileShareUserVO">  
  <![CDATA[  
    INSERT INTO TB_USERS_MAPPING  
    (  
      USERNAME, FILE_IDX, CREA_DTM, CREA_ID  
    )  
    VALUES  
    (  
      #{username},  
      #{file_idx}::INTEGER,  
      NOW(),  
      #{crea_id}  
    )  
  ]]>  
</insert>
```

```
<!-- Meta Merge -->  
<insert id="updateMergeMetaMapping" parameterType="hashmap">  
/*  
updateMergeMetaMapping */  
UPDATE TB_FILEMANAGER_META_MAPPING SET  
META_IDX = #{targetMeta}::INTEGER WHERE META_IDX =  
#{sourceMeta}::INTEGER;  
</insert>
```

```
<insert id="updateMergeMeta" parameterType="hashmap">  
UPDATE  
TB_FILEMANAGER_META SET  
DEL_GB = 'Y' WHERE IDX = #{sourceMeta}::INTEGER;  
</insert>
```

```
<!-- Delete Meta Detail -->  
<insert id="deleteMetaDetail" parameterType="int">  
DELETE FROM  
TB_FILEMANAGER_META  
WHERE IDX = #{meta_idx};  
</insert>
```

```
<!-- ##### ===== Meta INSERT/UPDATE  
Select ===== ##### -->
```

```
<!-- ##### ===== File INSERT/UPDATE  
Select ===== ##### -->
```

```
<insert id="insertFile" parameterType="hashmap">  
  <![CDATA[  
    INSERT INTO TB_FILEMANAGER_META_FILE
```

```

(
  FILE_TITLE,
  FILE_DETAIL,
  ORIGINAL_FILE_NAME,
  STORED_FILE_NAME,
  FILE_SIZE,
  FILE_TYPE,
  CREA_DTM,
  CREA_ID,
  DOWN_YN,
  STAR_LEVEL
)
VALUES
(
  #{file_title},
  #{file_detail},
  #{original_file_name},
  #{stored_file_name},
  #{file_size},
  #{file_type},
  NOW(),
  #{crea_id},
  #{down_yn},
  #{star_level}::INTEGER
)
]]>
</insert>

<insert id="updateFile" parameterType="hashmap">
  <![CDATA[
UPDATE TB_FILEMANAGER_META_FILE SET
  ]]>
  <if test="file_title != null and file_title != "">
FILE_TITLE = #{file_title} ,
  </if>
  <if test="file_type != null and file_type != "">
FILE_TYPE = #{file_type} ,
  </if>
  <if test="file_detail != null and file_detail != "">
FILE_DETAIL = #{file_detail} ,
  </if>
  <if test="crea_id != null and crea_id != "">
CREA_ID = #{crea_id} ,
  </if>
  <if test="star_level != null and star_level != "">
STAR_LEVEL = #{star_level}::INTEGER ,
  </if>
  <![CDATA[
    CREA_DTM = NOW()
WHERE IDX = #{idx}::INTEGER;
  ]]>
</insert>

<delete id="deleteShareUsers" parameterType="int">
  <![CDATA[
DELETE FROM TB_USERS_MAPPING
WHERE FILE_IDX = #{file_idx}

```

```

]]>
</delete>

<select id="selectMetaListByFile" parameterType="int"
resultType="com.joang.filemanager.vo.MetaVO">
  <![CDATA[
    SELECT
    A.IDX,
    A.USER_ID,
    A.META,
    A.CREA_DTM,
    A.CREA_ID,
    A.DEL_GB
    FROM TB_FILEMANAGER_META A, TB_FILEMANAGER_META_MAPPING B,
    TB_FILEMANAGER_META_FILE C
    WHERE A.IDX = B.META_IDX
    AND B.FILE_IDX = C.IDX
    AND C.DEL_GB = 'N'
    AND C.IDX = #{idx}
  ]]>
</select>

<!-- update File Detail -->
<insert id="updateFileDetail" parameterType="hashmap">
UPDATE
TB_FILEMANAGER_META_FILE SET
DEL_GB = 'Y' WHERE IDX = #{idx}::INTEGER;
</insert>

<!-- File Mapping update -->
<insert id="updateFileMetaMapping" parameterType="hashmap">
UPDATE
TB_FILEMANAGER_META_MAPPING SET
FILE_IDX = #{targetMeta}::INTEGER WHERE FILE_IDX = #{sourceMeta}::INTEGER;
</insert>

<!-- Delete File detail -->
<delete id="deleteFileDetail" parameterType="int">
<![CDATA[
  DELETE
  FROM
    TB_FILEMANAGER_META_FILE
  WHERE
    IDX = #{file_idx}::INTEGER;
  ]]>
</delete>

<update id="updateYouTuBeDownloadYn" parameterType="int">
/*
updateYouTuBeDownloadYn */
UPDATE TB_FILEMANAGER_META_FILE SET
DOWN_YN = 'N'
WHERE
IDX = #{file_idx};
</update>

```

```
<!-- ##### ===== File INSERT/UPDATE
Select ===== ##### -->
```

```
<!-- ##### ===== MApping Management
===== ##### -->
```

```
<!-- ##### Meta Ref ##### -->
```

```
<select id="selectMetaRefList"
parameterType="com.joang.filemanager.vo.MetaRefVO"
resultType="com.joang.filemanager.vo.MetaRefVO">
  <![CDATA[
    SELECT
    A.FROM_IDX,
    A.TO_IDX,
    A.META_REF,
    A.CREA_DTM,
    A.CREA_ID
    FROM TB_FILEMANAGER_META_REF A, TB_FILEMANAGER_META_MAPPING B,
    TB_FILEMANAGER_META_FILE C, TB_USERS_MAPPING D
    WHERE 1=1
    ]]>
  <if test="fromIdx != null and fromIdx != '' and fromIdx != 0">
  <![CDATA[
    AND A.FROM_IDX = #{fromIdx}
  ]]>
  </if>
  <if test="toIdx != null and toIdx != '' and toIdx != 0">
  <![CDATA[
    AND A.TO_IDX = #{toIdx}
  ]]>
  </if>
  <![CDATA[
    AND A.FROM_IDX = B.META_IDX
    AND B.FILE_IDX = C.IDX
    AND A.DEL_GB = 'N'
    AND C.IDX = D.FILE_IDX
    AND D.USERNAME = #{crea_id}
  ]]>
  UNION
  SELECT
  A.FROM_IDX,
  A.TO_IDX,
  A.META_REF,
  A.CREA_DTM,
  A.CREA_ID
  FROM TB_FILEMANAGER_META_REF A, TB_FILEMANAGER_META_MAPPING B,
  TB_FILEMANAGER_META_FILE C, TB_USERS_MAPPING D
  WHERE 1=1
  ]]>
  <if test="fromIdx != null and fromIdx != '' and fromIdx != 0">
  <![CDATA[
    AND A.FROM_IDX = #{fromIdx}
  ]]>
  </if>
```

```

<if test="toldx != null and toldx != '' and toldx != 0">
<![CDATA[
AND A.TO_IDX = #{toldx}
]]>
</if>
    <![CDATA[
        AND A.TO_IDX = B.META_IDX
        AND B.FILE_IDX = C.IDX
AND A.DEL_GB = 'N'
AND C.IDX = D.FILE_IDX
AND D.USERNAME = #{crea_id}
    ]]>
</select>

<!-- Check mapping -->
<select id="checkMetaMapping" parameterType="hashmap"
resultType="hashmap">
<![CDATA[
    SELECT
meta_idx
FROM
    TB_FILEMANAGER_META_MAPPING
    WHERE
        DEL_GB = 'N'
        AND META_IDX = #{meta_idx}::INTEGER
AND FILE_IDX = #{file_idx}::INTEGER
    ]]>
</select>

<!-- Delete meta mapping -->
<delete id="deleteMetaMapping" parameterType="hashmap">
<![CDATA[
    DELETE
FROM
    TB_FILEMANAGER_META_MAPPING
    WHERE
        META_IDX = #{meta_idx}::INTEGER
AND FILE_IDX = #{file_idx}::INTEGER
    ]]>
</delete>

<select id="selectMappingListByMetalDx" parameterType="int"
resultType="com.joang.filemanager.vo.MetaVO">
    <![CDATA[
        SELECT
IDX,
META_IDX,
FILE_IDX
FROM TB_FILEMANAGER_META_MAPPING
WHERE META_IDX = #{idx}
    ]]>
</select>

<!-- Delete Mapping -->
<delete id="deleteMetaMappingByFileIdx" parameterType="int">
<![CDATA[

```

```
DELETE
FROM
    TB_FILEMANAGER_META_MAPPING
WHERE
    FILE_IDX = #{file_idx}
]]>
</delete>
```

```
<insert id="insertMetaRef"
parameterType="com.joang.filemanager.vo.MetaRefVO">
<![CDATA[
INSERT INTO TB_FILEMANAGER_META_REF (FROM_IDX, TO_IDX, META_REF, CREA_DTM,
CREA_ID, UPDT_DTM, DEL_GB) VALUES
(
    #{fromIdx}, #{toIdx}, #{metaRef}, NOW(), #{creaId}, NOW(), 'N'
)
]]>
</insert>
```

```
<insert id="updateMetaRef"
parameterType="com.joang.filemanager.vo.MetaRefVO">
<![CDATA[
UPDATE TB_FILEMANAGER_META_REF SET
META_REF = #{metaRef},
UPDT_DTM = NOW()
WHERE
FROM_IDX = #{fromIdx}
AND TO_IDX = #{toIdx}
AND CREA_ID = #{crea_id}
]]>
</insert>
```

```
<insert id="deleteMetaRef"
parameterType="com.joang.filemanager.vo.MetaRefVO">
<![CDATA[
DELETE FROM TB_FILEMANAGER_META_REF
WHERE
FROM_IDX = #{fromIdx}
AND TO_IDX = #{toIdx}
AND CREA_ID = #{crea_id}
]]>
</insert>
```

```
<update id="updateMergeMetaFromRef" parameterType="hashmap">
/*
updateMergeMetaRef */
UPDATE TB_FILEMANAGER_META_REF SET
FROM_IDX = #{targetFromMeta}::INTEGER
WHERE
FROM_IDX = #{sourceFromMeta}::INTEGER AND TO_IDX = #{sourceToMeta}::INTEGER
AND CREA_ID = #{crea_id}
</update>
```

```
<update id="updateMergeMetaToRef" parameterType="hashmap">
```

```

/*
updateMergeMetaRef */
UPDATE TB_FILEMANAGER_META_REF SET
TO_IDX = #{targetToMeta}::INTEGER
WHERE
TO_IDX = #{sourceToMeta}::INTEGER AND FROM_IDX = #{sourceFromMeta}::INTEGER
AND CREA_ID = #{crea_id}
</update>

<delete id="daleteMergeMetaRef" parameterType="hashmap">
/*
daleteMergeMetaFromRef */
DELETE
FROM
TB_FILEMANAGER_META_REF
WHERE
FROM_IDX = #{sourceFromMeta}::INTEGER AND TO_IDX = #{sourceToMeta}::INTEGER
AND CREA_ID = #{crea_id}
</delete>

<delete id="deleteMetaMappingFROMRef" parameterType="int">
/*
deleteMetaMappingFROMRef */
DELETE
FROM
TB_FILEMANAGER_META_REF
WHERE
FROM_IDX = #{deleteMeta}
</delete>

<delete id="deleteMetaMappingTORef" parameterType="int">
/*
deleteMetaMappingTORef */
DELETE
FROM
TB_FILEMANAGER_META_REF
WHERE
TO_IDX = #{deleteMeta}
</delete>

<!-- ##### ===== Mapping Management
===== ##### -->

</mapper>

```

test.txt

```

test1
test2 // hyunsu
hhhh /* hello 0

hello

mmmm */ lllll

```

```

/*
여기
*/

123456 <!-- test
  hdewhfd
  wehfiuheri
  --> dfjrwevniearlivsefh

1234 <!-- hhyubn --> 555555555555
hhhh -- nfejeewfhlwjka

test /* hahahah */
<!--
fwef -->
우리 대한민국 만세 // 류재환
FINE

```

처리 결과

처리결과

```

START
::: PROC FILE NAME : metaMapper.xml:::
> fileName Path : C:\HYUNSU\eclipse-showcase-
workspace\SocParsing\parsing\metaMapper.xml
> Remove Anno Range : metaMapper.xml [/* , */]
> Remove Anno Range : metaMapper.xml [<!-- , -->]
> Remove Anno Line : metaMapper.xml [--]
> Remove Anno Line : metaMapper.xml [//]
>> Find! (1)<SELECT->/SELECT> 1415 Col
\_ TEXT = " ID="SELECTMETALISTCNT" PARAMETERTYPE="COM.JOANG. ... "
>> Find! (2)<SELECT->/SELECT> 1760 Col
\_ TEXT = " ID="SELECTMETALIST" PARAMETERTYPE="COM.JOANG.FIL ... "
>> Find! (3)<SELECT->/SELECT> 1199 Col
\_ TEXT = " ID="SELECTMETAAUTOCOMPLETE" PARAMETERTYPE="COM.J ... "
>> Find! (4)<SELECT->/SELECT> 331 Col
\_ TEXT = " ID="SELECTUSERAUTOCOMPLETE" PARAMETERTYPE="COM.J ... "
>> Find! (5)<SELECT->/SELECT> 2723 Col
\_ TEXT = " ID="SELECTREFMETALIST" PARAMETERTYPE="COM.JOANG. ... "
>> Find! (6)<SELECT->/SELECT> 136 Col
\_ TEXT = " ID="SELECTMETAVALUEBYMETAIDX" PARAMETERTYPE="INT" ... "
>> Find! (7)<SELECT->/SELECT> 1610 Col
\_ TEXT = " ID="SELECTFILELISTCNT" PARAMETERTYPE="COM.JOANG. ... "
>> Find! (8)<SELECT->/SELECT> 2618 Col
\_ TEXT = " ID="SELECTFILELIST" PARAMETERTYPE="COM.JOANG.FIL ... "
>> Find! (9)<SELECT->/SELECT> 536 Col
\_ TEXT = " ID="SELECTFILEDETAIL" PARAMETERTYPE="INT" RESULT ... "
>> Find! (10)<SELECT->/SELECT> 384 Col
\_ TEXT = " ID="SELECTFILESHAREUSERLISTBYFILE" PARAMETERTYPE= ... "
>> Find! (11)<SELECT->/SELECT> 107 Col
\_ TEXT = " ID="GETTARGETMETAIDX" RESULTTYPE="STRING"> SELEC ... "

```

```

>> Find! (12)<SELECT-~/SELECT> 228 Col
\_ TEXT = " ID="CHECKMETA" PARAMETERTYPE="STRING" RESULTTYPE ... "
>> Find! (15)<SELECT-~/SELECT> 399 Col
\_ TEXT = " ID="SELECTMETALISTBYFILE" PARAMETERTYPE="INT" RE ... "
>> Find! (16)<SELECT-~/SELECT> 1426 Col
\_ TEXT = " ID="SELECTMETAREFLIST" PARAMETERTYPE="COM.JOANG. ... "
>> Find! (17)<SELECT-~/SELECT> 305 Col
\_ TEXT = " ID="CHECKMETAMAPPING" PARAMETERTYPE="HASHMAP" RE ... "
>> Find! (18)<SELECT-~/SELECT> 241 Col
\_ TEXT = " ID="SELECTMAPPINGLISTBYMETAIDX" PARAMETERTYPE="IN ... "
>> :::::::::: key="<select-~/select>" , FIND("TB_FILEMANAGER_META") Total Cnt = 15
>> Find! (1)<INSERT-~/INSERT> 453 Col
\_ TEXT = " ID="INSERTMETA" PARAMETERTYPE="HASHMAP"> <![CD ... "
>> Find! (2)<INSERT-~/INSERT> 648 Col
\_ TEXT = " ID="INSERTMETAMAPPING" PARAMETERTYPE="HASHMAP"> ... "
>> Find! (3)<INSERT-~/INSERT> 375 Col
\_ TEXT = " ID="INSERTMETAEXISTMAPPING" PARAMETERTYPE="HASHMA ... "
>> Find! (4)<INSERT-~/INSERT> 332 Col
\_ TEXT = " ID="INSERTSHAREUSER" PARAMETERTYPE="COM.JOANG.FIL ... "
>> Find! (5)<INSERT-~/INSERT> 171 Col
\_ TEXT = " ID="UPDATERGEMETAMAPPING" PARAMETERTYPE="HASHMA ... "
>> Find! (6)<INSERT-~/INSERT> 130 Col
\_ TEXT = " ID="UPDATERGEMETA" PARAMETERTYPE="HASHMAP"> UP ... "
>> Find! (7)<INSERT-~/INSERT> 104 Col
\_ TEXT = " ID="DELETEMETADETAIL" PARAMETERTYPE="INT"> DELET ... "
>> Find! (8)<INSERT-~/INSERT> 659 Col
\_ TEXT = " ID="INSERTFILE" PARAMETERTYPE="HASHMAP"> <![C ... "
>> Find! (9)<INSERT-~/INSERT> 635 Col
\_ TEXT = " ID="UPDATEFILE" PARAMETERTYPE="HASHMAP"> <![C ... "
>> Find! (10)<INSERT-~/INSERT> 129 Col
\_ TEXT = " ID="UPDATEFILEDETAIL" PARAMETERTYPE="HASHMAP"> U ... "
>> Find! (11)<INSERT-~/INSERT> 170 Col
\_ TEXT = " ID="UPDATEFILEMETAMAPPING" PARAMETERTYPE="HASHMAP ... "
>> Find! (12)<INSERT-~/INSERT> 270 Col
\_ TEXT = " ID="INSERTMETAREF" PARAMETERTYPE="COM.JOANG.FILE ... "
>> Find! (13)<INSERT-~/INSERT> 256 Col
\_ TEXT = " ID="UPDATERGEMETAREF" PARAMETERTYPE="COM.JOANG.FILE ... "
>> Find! (14)<INSERT-~/INSERT> 210 Col
\_ TEXT = " ID="DELETEMETAREF" PARAMETERTYPE="COM.JOANG.FILE ... "
>> :::::::::: key="<insert-~/insert>" , FIND("TB_FILEMANAGER_META") Total Cnt = 13
>> Find! (1)<UPDATE-~/UPDATE> 130 Col
\_ TEXT = " ID="UPDATEYOUTUBEDOWNLOADYN" PARAMETERTYPE="INT"> ... "
>> Find! (2)<UPDATE-~/UPDATE> 239 Col
\_ TEXT = " ID="UPDATERGEMETAFROMREF" PARAMETERTYPE="HASHMA ... "
>> Find! (3)<UPDATE-~/UPDATE> 233 Col
\_ TEXT = " ID="UPDATERGEMETATOREF" PARAMETERTYPE="HASHMAP" ... "
>> :::::::::: key="<update-~/update>" , FIND("TB_FILEMANAGER_META") Total Cnt = 3
>> Find! (1)<DELETE-~/DELETE> 122 Col
\_ TEXT = " ID="DELETESHAREUSERS" PARAMETERTYPE="INT"> <![ ... "
>> Find! (2)<DELETE-~/DELETE> 190 Col
\_ TEXT = " ID="DELETEFILEDETAIL" PARAMETERTYPE="INT"> <![CDA ... "
>> Find! (3)<DELETE-~/DELETE> 240 Col
\_ TEXT = " ID="DELETETAMAPPING" PARAMETERTYPE="HASHMAP"> < ... "
>> Find! (4)<DELETE-~/DELETE> 198 Col
\_ TEXT = " ID="DELETETAMAPPINGBYFILEIDX" PARAMETERTYPE="IN ... "
>> Find! (5)<DELETE-~/DELETE> 199 Col
\_ TEXT = " ID="DALETETEMERGEMETAREF" PARAMETERTYPE="HASHMAP"> ... "

```

```
>> Find! (6)<DELETE-~/DELETE> 124 Col
\_ TEXT = " ID="DELETOMETAMAPPINGFROMREF" PARAMETERTYPE="INT" ... "
>> Find! (7)<DELETE-~/DELETE> 120 Col
\_ TEXT = " ID="DELETOMETAMAPPINGTOREF" PARAMETERTYPE="INT"> ... "
>> :::::::::: key="<delete-~/delete>" , FIND("TB_FILEMANAGER_META") Total Cnt = 6
:::::::::::: File=metaMapper.xml , Result={<SELECT=15, <DELETE=6, <INSERT=13,
<UPDATE=3}:::::::::::::

:::::::::::: PROC FILE NAME : test.txt::::::::::::
> fileName Path : C:\HYUNSU\eclipse-showcase-workspace\SocParsing\parsing\test.txt
> Remove Anno Range : test.txt [/* , */]
> Remove Anno Range : test.txt [<!-- , -->]
> Remove Anno Line : test.txt [--]
> Remove Anno Line : test.txt [//]
>> :::::::::: key="<select-~/select>" , FIND("TB_FILEMANAGER_META") Total Cnt = 0
>> :::::::::: key="<insert-~/insert>" , FIND("TB_FILEMANAGER_META") Total Cnt = 0
>> :::::::::: key="<update-~/update>" , FIND("TB_FILEMANAGER_META") Total Cnt = 0
>> :::::::::: key="<delete-~/delete>" , FIND("TB_FILEMANAGER_META") Total Cnt = 0
:::::::::::: File=test.txt , Result={<SELECT=0, <DELETE=0, <INSERT=0, <UPDATE=0}:::::::::::::

>>>>>>>>> All File Find ["TB_FILEMANAGER_META"] Result={<SELECT=15, <DELETE=6,
<INSERT=13, <UPDATE=3}
```

🔄Revision #6

★Created 2025-02-09 15:30:21 UTC by Hyeon Su Ryu

✎Updated 2025-02-09 16:22:59 UTC by Hyeon Su Ryu