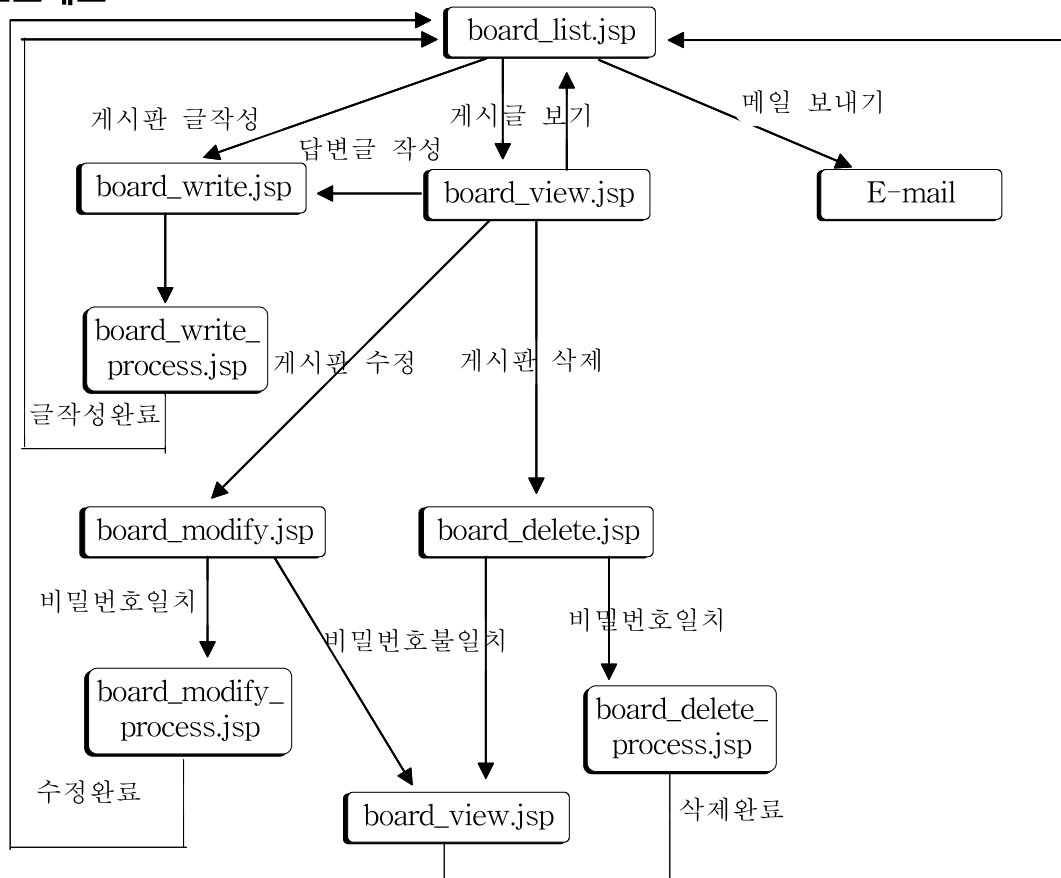


◆ 프로젝트 과제 3 ◆

1. 게시판 만들기

- 계층형 구조를 갖도록하고 검색 기능과 게시글의 타입을 갖도록 설정
- 누구나 게시물을 작성할 수 있도록 한다
- 사용자가 입력한 내용이 정상적으로 완료되었으면 게시물의 목록을 보여주는 페이지로 이동한다
- 게시물을 작성한 사용자 또는 다른 사용자가 목록에서 게시물을 선택하여 게시물의 내용을 본다
- 게시물을 작성한 작성자에 한해서 게시물의 내용을 수정 및 삭제할 수 있다
- 페이지 스킵 메소드를 사용
 - 페이지 번호는 방명록하단에 [1]번부터 [10]번까지 페이지번호를 보여주다가 더 많은 페이지가 생기면 [NEXT]와 마지막 페이지 번호가 보이도록 처리해 준다
 - 반대로 10페이지가 넘는 위치에 있다면 [1]와 [Prev]가 보이도록 한다
- 관리자는 관리자의 인증을 거친 후 모든 게시물에 대한 수정 및 삭제할 수 있다
- 검색조건 있음

2. 프로세스



3. DataBase 생성

- 데이터베이스 파일명 :
- 테이블명 : sam_board

필드명	type	null 허용	설 명
num	int		기본키로 설정, 자동증가
name	varchar(30)		글을 올린 방문객의 이름
pass	varchar(10)		저장게시판에 남긴 글을 수정 또는 삭제할 때 필요한 패스워드 저장
email	varchar(100)	yes	게시판에 글을 남긴이의 E-Mail 저장
indate	datetime		글을 올린 날짜저장
subject	varchar(100)		글의 제목을 저장
contents	text		게시판에 남기고 싶을 말을 저장
parent	int		부모글번호
indent	int		들여쓰기 기준
depth	int		답변글의 넘버를 저장하는 필드로 첫번째 답변글은 1, 두 번째 답변글은 2, n번째 답변글은 n의 값이 저장
realparent	int		바로 한 수준 위의 부모글 번호
readcnt	int		게시판에 있는 글을 몇번 읽었는지 저장
ip	varchar(20)		작성자 ip주소

4. 계층형 게시판에서 주목해야할 컬럼(parent, indent, depth, realparent)

- parent : 답변글이 최초 생기기 시작한 부모가 되는 글로써 만약 답변글이 없다면 하나의 글만 parent 값을 가지고, 답변글이 여러 있다면 이들과 관련된 글 모두 같은 parent 값을 가진다
- indent : 답변글 사이의 들여쓰기 정도를 기록하는 것으로 "board_list.jsp"에서 몇 차례 답변을 한 글 인지에 따라 들여쓰기를 할 때 사용
- realparent : 답변글이 여러 있을 때 바로 어떤 글에 대한 답변글인지 표시
- 예제

num (고유번호)	계층구조	parent (부모글번호)	depth (글 깊이)	indent (들여쓰기)	realparent
1	a	1	0	0	0
2	b	2	0	0	0
3	a.a	1	5	1	1
4	a.b	1	2	1	1
5	a.b.a	1	3	2	4
6	c	6	0	0	0
7	a.b.a.a	1	4	3	5
8	a.c	1	1	1	1
9	b.a	2	1	1	2

첫 번째(a), 두 번째(b)는 그냥 순서대로 글을 올렸지만, 세 번째(a.a)는 첫 번째(a)에 대한 첫 번째 답변글이고, 네 번째(a.b)는 첫 번째(a)에 대한 두 번째 답변글로 올린 후, 다섯 번째(a.b.a)에서 다시 네 번째(a.b)에 대한 첫 번째 답변글을 올렸다고 할 때 이들 간 답변글에 대한 계층 구조는 다음과 같다

```

if(parent != 0){ //답변글일 경우
    query = "update sam_board set depth=depth+1 where parent = ? and depth > ?";

    pstmt = conn.prepareStatement(query);
    pstmt.setInt(1,parent);
    pstmt.setInt(2,depth);
    pstmt.executeUpdate();

    depth = depth + 1;
    indent = indent + 1;
}else{ //부모글번호가 0이면 처음 작성된글
    parent = number;
    depth = 0;
    indent = 0;
}

query = "insert into sam_board (pass,name,indate,subject,contents,";
query += "parent,realparent,depth,indent,readcnt,ip) values (";
query += "?,?,?,?,?,?,?,?,?,?)";

```

- parent에 대해서는 내림차순으로, depth에 대해서는 오름차순으로 정렬되게 하며
이때 계층 구조의 들여쓰기는 indent의 숫자에 따라 처리

num (고유번호)	계층구조	parent (부모글번호)	depth (글 깊이)	indent (들여쓰기)	realparent
6	c	6	0	0	0
2	b	2	0	0	0
9	b.a	2	1	1	2
1	a	1	0	0	0
8	a.c	1	1	1	1
4	a.b	1	2	1	1
5	a.b.a	1	3	2	4
7	a.b.a.a	1	4	3	5
3	a.a	1	5	1	1

-- 전체 출력 --

```
select * from boardreply order by parent desc, depth asc
```

-- oracle 1페이지 출력(1~10) --

```
select X.* from (select rownum as rnum, A.* from
    (select * from boardreply order by parent desc, depth asc) A
    where rownum <= 10) X where X.rnum >= 1;
```

-- mysql 1페이지 출력(1~10) limit ?? 시작, 개수 --

```
select * from sam_board order by parent desc, depth asc limit 1,10
```

-- oracle 조건 검색 s_query에 조건 입력 --

```
select X.* from (select rownum as rnum, A.* from
    (select * from board order by parent desc, depth asc) A where + s_query +
    and rownum <= 10) X where + s_query + and X.rnum >= 1
```

-- mysql 조건 검색 s_query에 조건 입력 --

```
select * from sam_board "+s_query+"order by parent desc, depth asc limit ??,
```