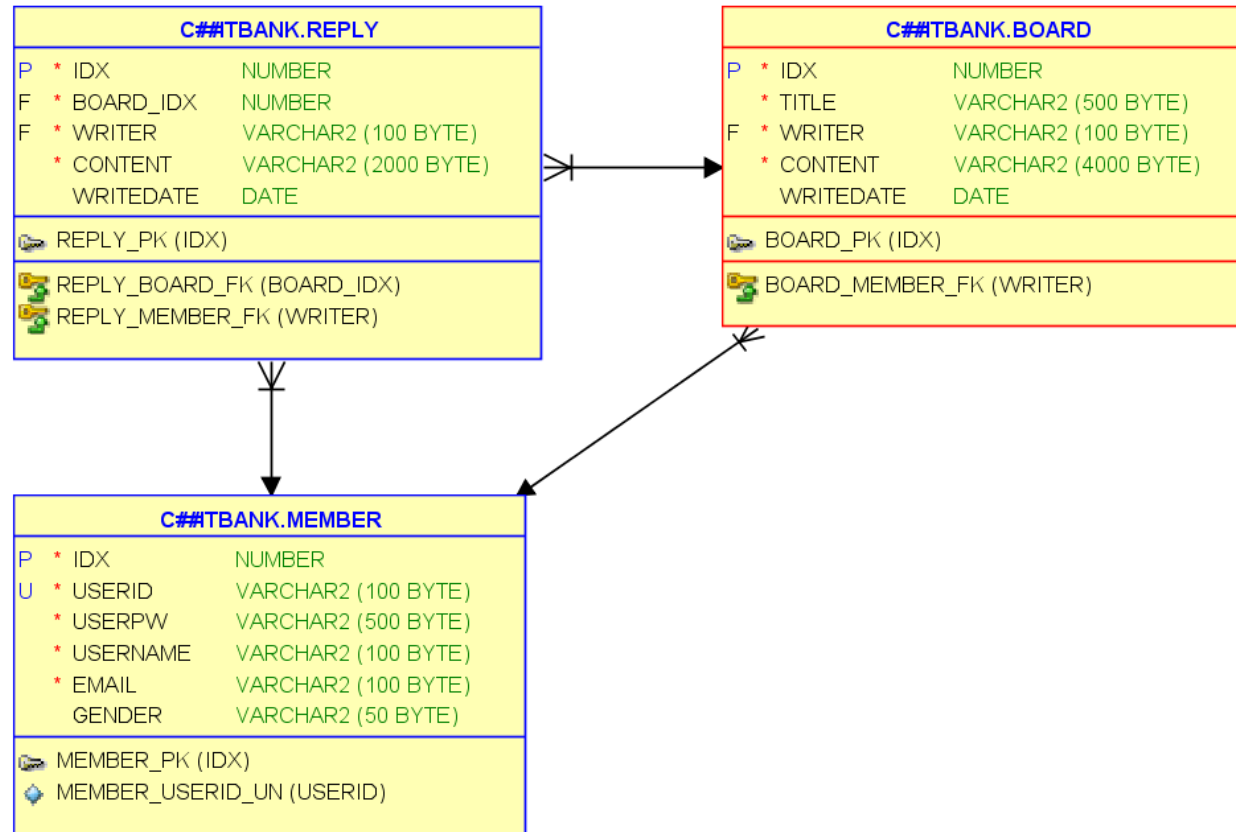


# 테이블 관계도



# 테이블 DDL

```
drop table member cascade constraints purge;
```

```
drop table board cascade constraints purge;
```

```
drop table reply cascade constraints purge;
```

```
create sequence member_seq
```

```
    start with 1
```

```
    maxvalue 99999
```

```
    increment by 1
```

```
    nocycle
```

```
    nocache;
```

```
create sequence board_seq
```

```
    start with 1
```

```
    maxvalue 99999
```

```
    increment by 1
```

```
    nocycle
```

```
    nocache;
```

```
create table member (
```

```
    idx          number          default member_seq.nextval primary key,
```

```
    userid       varchar2(100)    unique not null,
```

```
    userpw       varchar2(500)    not null,
```

```
    username     varchar2(100)    not null,
```

```
    email        varchar2(100)    not null,
```

```
    gender       varchar2(50)     check(gender in ('남성', '여성'))
```

```
);
```

```
create table board (
```

```
    idx          number          default board_seq.nextval primary key,
```

```
    title        varchar2(500)    not null,
```

```
    writer       varchar2(100)    not null,
```

```
    content      varchar2(4000)    not null,
```

```
    writeDate    date             default sysdate,
```

```
    constraint board_member_fk
```

```
    foreign key(writer)
```

```
    REFERENCES member(userid)
```

```
);
```

```
create table reply (
```

```
    idx          number          generated as identity primary key,
```

```
    board_idx    number          not null, -- 원본 게시물 번호
```

```
    writer       varchar2(100)    not null,
```

```
    content      varchar2(2000)    not null,
```

```
    writeDate    date             default sysdate,
```

```
    constraint reply_board_fk
```

```
    foreign key(board_idx)
```

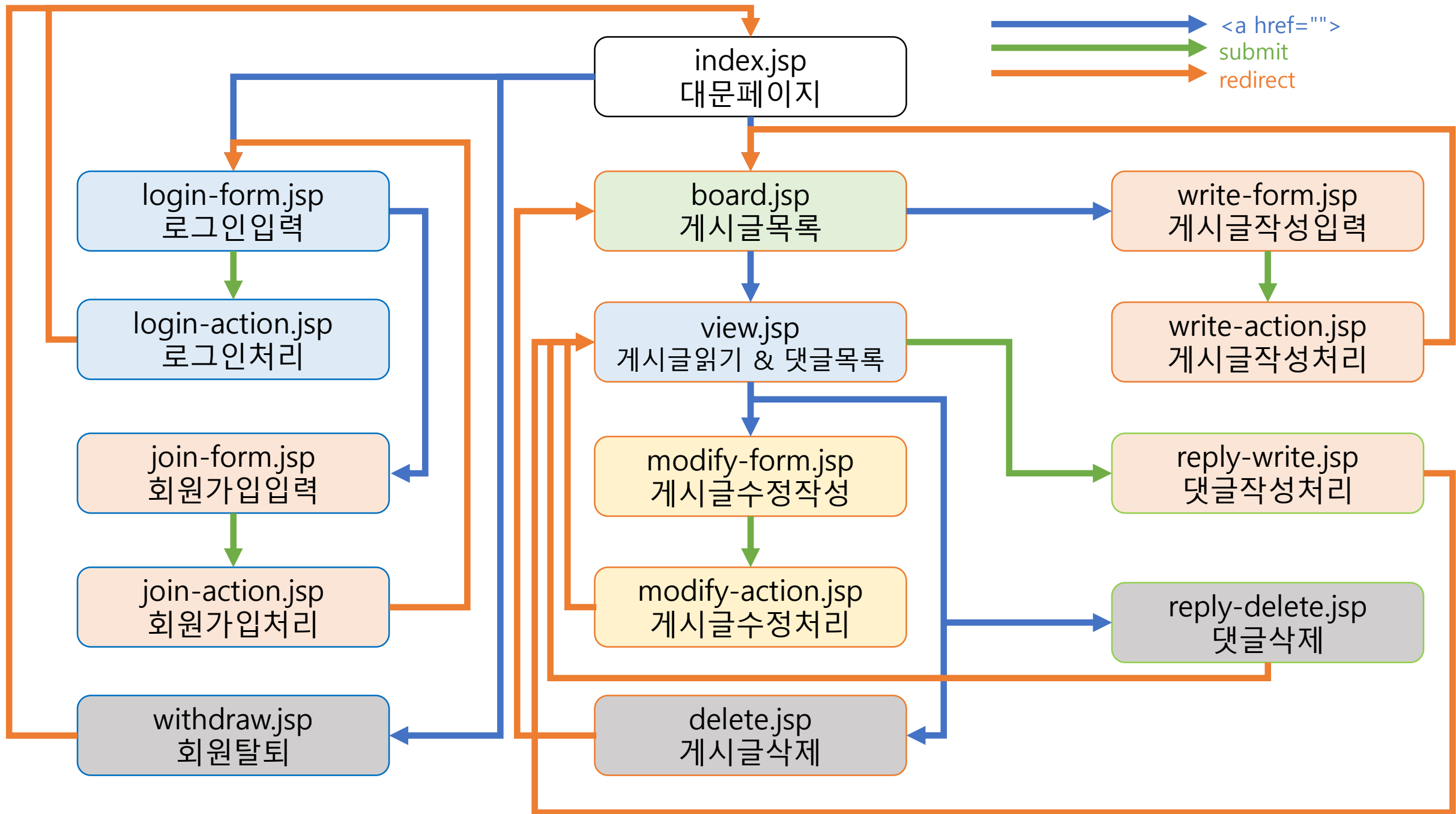
```
    REFERENCES board(idx),
```

```
    constraint reply_member_fk
```

```
    foreign key(writer)
```

```
    REFERENCES member(userid)
```

```
);
```



MEMBER

BOARD

REPLY

selectList

selectList  
게시글목록

selectList  
댓글목록

selectOne

selectOne  
로그인

selectOne  
게시글읽기

insert

insert  
회원가입

insert  
게시글작성

insert  
댓글작성

update

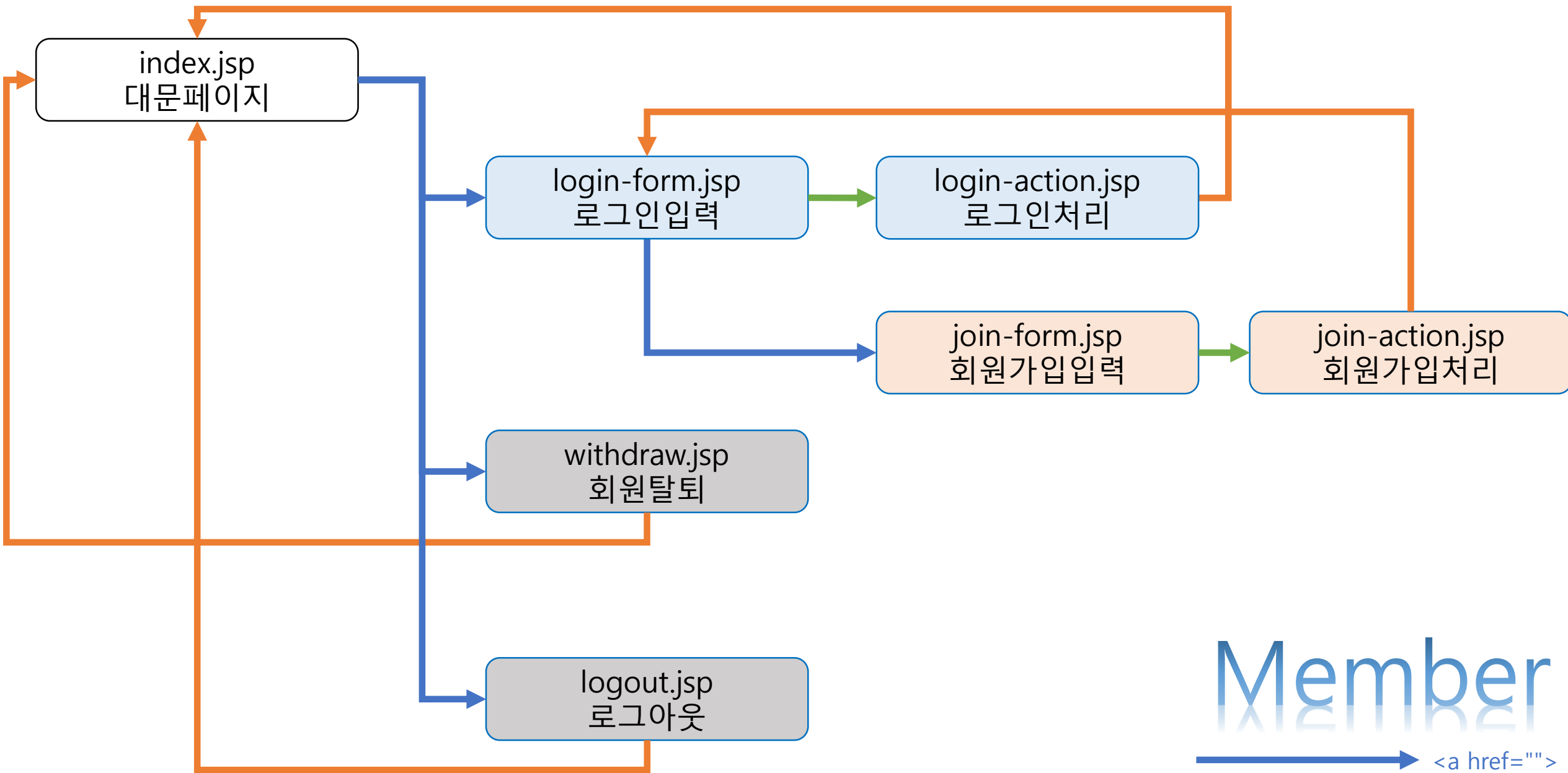
update  
게시글수정

delete

delete  
회원탈퇴

delete  
게시글삭제

delete  
댓글삭제



# Member

 `<a href="">`  
 submit  
 redirect

# member – selectOne(로그인)

- 사용자 입력값 id와 password 를 전달받아서 일치하는 계정 불러오기
- `select * from member where userid = ? and userpw = ?`
- 매개변수 타입 : MemberDTO (입력값만 들어있음)
- 반환값 타입 : MemberDTO (테이블 특정 레코드의 모든 컬럼이 들어있음)
- 로그인에 사용되므로, 반환객체를 session에 저장하기
- session에 저장했다면, 페이지 이동은 redirect로 처리하기

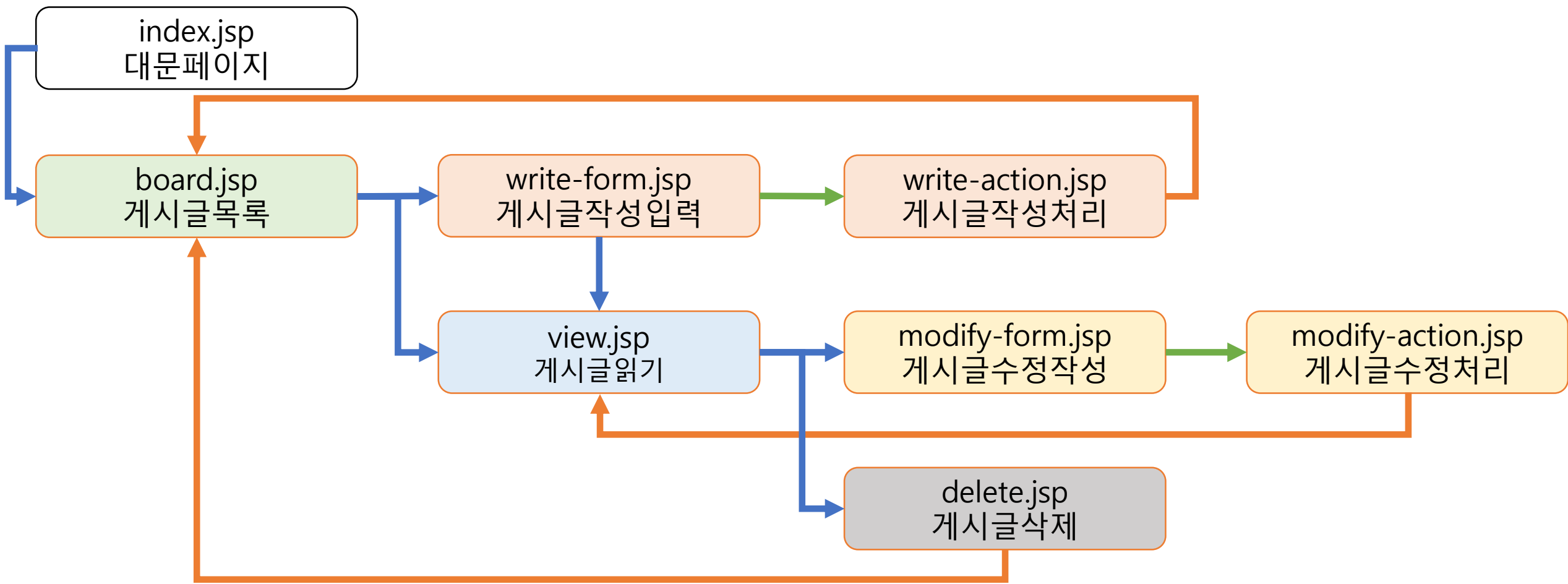
# member – insert(회원가입)

- 사용자가 입력한 모든 값을 회원 테이블에 추가하기
  - `insert into member (userid, userpw, name, gender, email) values (?, ?, ?, ?, ?)`
  - 매개변수 타입 : MemberDTO
  - 반환값 타입 : int (성공/실패, 영향을 받은 줄 수)
- 
- 반환값에 따라 서로 다른 처리를 수행하고 싶다면 <c:set>으로 받기
  - 만약 실패한다면 script를 이용하여 팝업창 경고 후 `history.back()`
  - 중복 id가 있다면 예외 (`SQLIntegrityConstraintViolationException`)
  - jsp 코드와 javascript 코드가 함께 있다면, jsp코드가 우선함에 유의




# member – delete(회원탈퇴)

- idx가 있다면 idx, 아니면 userid를 전달받아서 테이블에서 삭제하기
  - `delete from member where userid = ?`
  - 매개변수 타입 : String or MemberDTO
  - 반환값 타입 : int (성공/실패, 영향을 받은 줄 수)
- 
- 단일요소를 선택하기 위해서는 항상 primary key를 사용함
  - 로그인 상태에서 회원탈퇴를 진행해야 함
  - 회원탈퇴 시 추가로 비밀번호를 한번 더 입력받는 경우도 있음
  - 회원탈퇴가 진행되고 난 후 로그아웃 페이지로 리다이렉트하기





# Board

 `<a href="">`  
 `submit`  
 `redirect`

# board – selectList(게시글목록)

- 게시글 목록을 idx 내림차순으로 불러온다
  - `select * from board order by idx desc`
  - 매개변수 타입 : 없음
  - 반환값 타입 : List<BoardDTO>
- 
- 서브쿼리 및 조인을 이용하면 이후 댓글개수까지 가능
  - 파라미터 유무에 따라 검색기능 구현 가능
  - 페이징을 구현하려면 fetch ~ offset 쿼리를 추가로 작성해야 함

# board – selectOne(게시글읽기)

- 단일 게시글을 조회한다
  - `select * from board where idx = ?`
  - 매개변수 타입 : int
  - 반환값 타입 : BoardDTO
- 
- 로그인하지 않아도 읽기는 가능
  - 기능은 단순하나, 게시글읽기 페이지에서 다른 기능으로 연결이 많음
  - 수정 및 삭제는 idx를 꼭 전달하기
  - 댓글을 구현할 경우, 게시글과 댓글의 객체이름을 다르게 설정할 것

# board – insert(게시글작성)

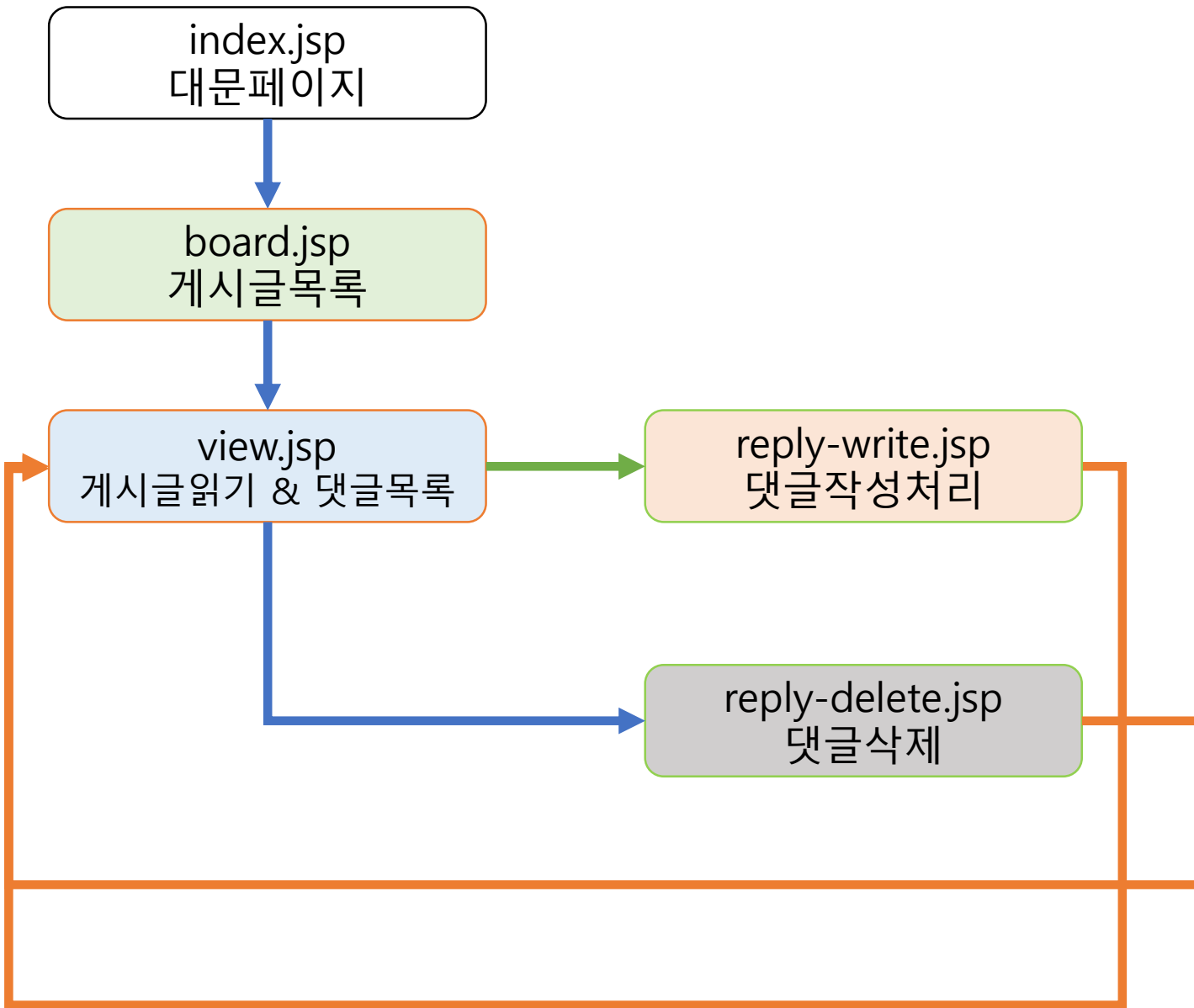
- 사용자 입력값을 board 테이블에 추가한다
  - `insert into board (title, writer, content) values (?, ?, ?)`
  - 매개변수 타입 : BoardDTO
  - 반환값 타입 : int (성공/실패, 영향을 받은 줄 수)
- 
- 로그인해야 작성 가능
  - 작성자는 직접 입력하지 말고, 세션값을 넘겨주기
  - 비회원 작성가능이라면 작성자 직접 입력하기
  - 작성이 끝나면 목록으로 넘어가기
  - 작성후 곧바로 게시글읽기로 넘어가려면 idx값을 추가로 전달해야 함

# board – delete(게시글삭제)




- idx를 전달받아서 게시글을 삭제한다
  - `delete from board where idx = ?`
  - 매개변수 타입 : int or BoardDTO (작성자 확인시)
  - 반환값 타입 : int (성공/실패, 영향을 받은 줄 수)
- 
- 로그인해야 삭제 가능
  - 세션의 로그인정보와 작성자정보가 일치해야 삭제 가능
  - 비회원 게시판이라면 작성시 글 비밀번호가 추가로 필요함
  - 삭제 후 목록으로 이동시키기

# board – update(게시글수정)

- 기존 작성된 게시글을 form에 불러와서 수정받고 데이터를 덮어씌운다
- `update board set title=?, content=? where idx=?`
- 매개변수 타입 : BoardDTO
- 반환값 타입 : int (성공/실패, 영향을 받은 줄 수)
- 로그인해야 수정 가능
- 작성자와 글번호는 수정대상이 아님
- 수정이 끝나면 게시글읽기로 넘어가기 (idx 전달 필요함)
- form에 띄울 내용은 selectOne 으로 불러오기
- form과 action을 별도 파일로 작성하는것이 쉬움



# Reply

 `<a href="">`  
 submit  
 redirect

# reply – selectList(댓글목록)

- 댓글 목록을 idx 오름차순으로 불러온다
  - `select * from reply where board_idx = ? order by idx`
  - 매개변수 타입 : int (게시글의 번호)
  - 반환값 타입 : List<ReplyDTO>
- 
- 계층형 댓글은 컬럼 및 SQL이 추가로 필요함
  - 댓글이 많은 경우, 댓글에 페이징을 적용하기도 함
  - 각 댓글마다 댓글의 idx가 부여되지만 노출시키지 않음



# reply – insert(댓글추가)

- 사용자 입력값을 reply 테이블에 추가한다
- `insert into reply (board_idx, writer, content) values (?, ?, ?)`
- 매개변수 타입 : ReplyDTO
- 반환값 타입 : int (성공/실패, 영향을 받은 줄 수)
- 댓글 추가 시, 게시글의 번호를 전달해야 추가 가능
- 로그인 해야 댓글을 작성할 수 있음
- 댓글 작성 후 게시글읽기 페이지로 돌아가야 하기 때문에 게시글 번호도 필요함

# reply – delete(댓글삭제)

- 댓글 번호를 전달받아서, 삭제한다
  - `delete from reply where idx = ?`
  - 매개변수 타입 : int or ReplyDTO
  - 반환값 타입 : int (성공/실패, 영향을 받은 줄 수)
- 
- 본인의 댓글만 삭제할 수 있어야 한다
  - 로그인 해야 댓글을 삭제할 수 있음
  - 댓글 삭제 후 게시글읽기 페이지로 돌아가야 하기 때문에 게시글 번호도 필요함