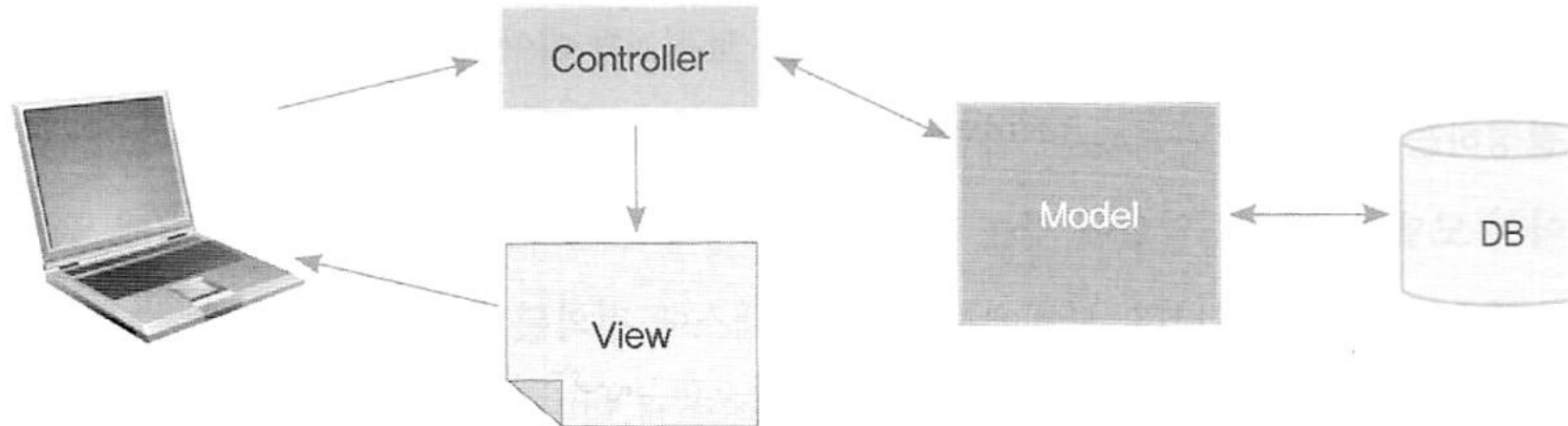


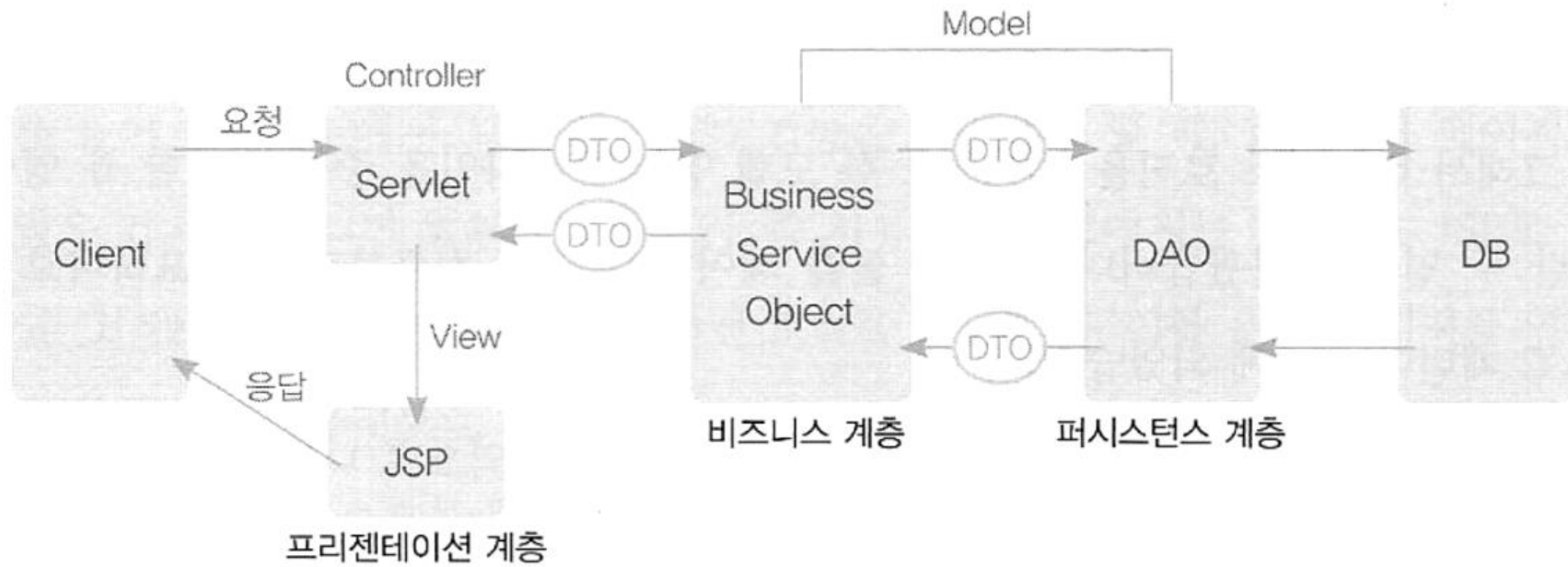
MVC 패턴(모델2)을 사용한 게시판

❖ MVC 패턴



구성요소	설명
모델	주로 데이터베이스와 연동하는 비즈니스 로직을 통해 얻어온 데이터를 저장하는 용도로 사용한다.
뷰	사용자와의 인터페이스로 주로 JSP 페이지를 사용한다.
컨트롤	모델과 뷰 사이의 흐름을 제어하는 역할을 한다. 클라이언트의 요청을 받아 이를 수행하기 위한 모델 영역의 어떤 비즈니스 로직을 수행할지를 결정하고 이를 처리한 후에 결과를 보여주기 위한 뷰를 선택한다. 주로 서블릿을 사용한다.

❖ MVC 패턴



❖ MVC 패턴의 컨트롤러 : 서블릿

- 웹 브라우저의 요청 받기
- 요청 분석
- 모델, 비즈니스 로직 구성
- 처리 결과 모델 가공
- 뷰로 모델 전달

❖ MVC 패턴의 뷰 : JSP

- 컨트롤러로부터 받은 모델 화면 출력

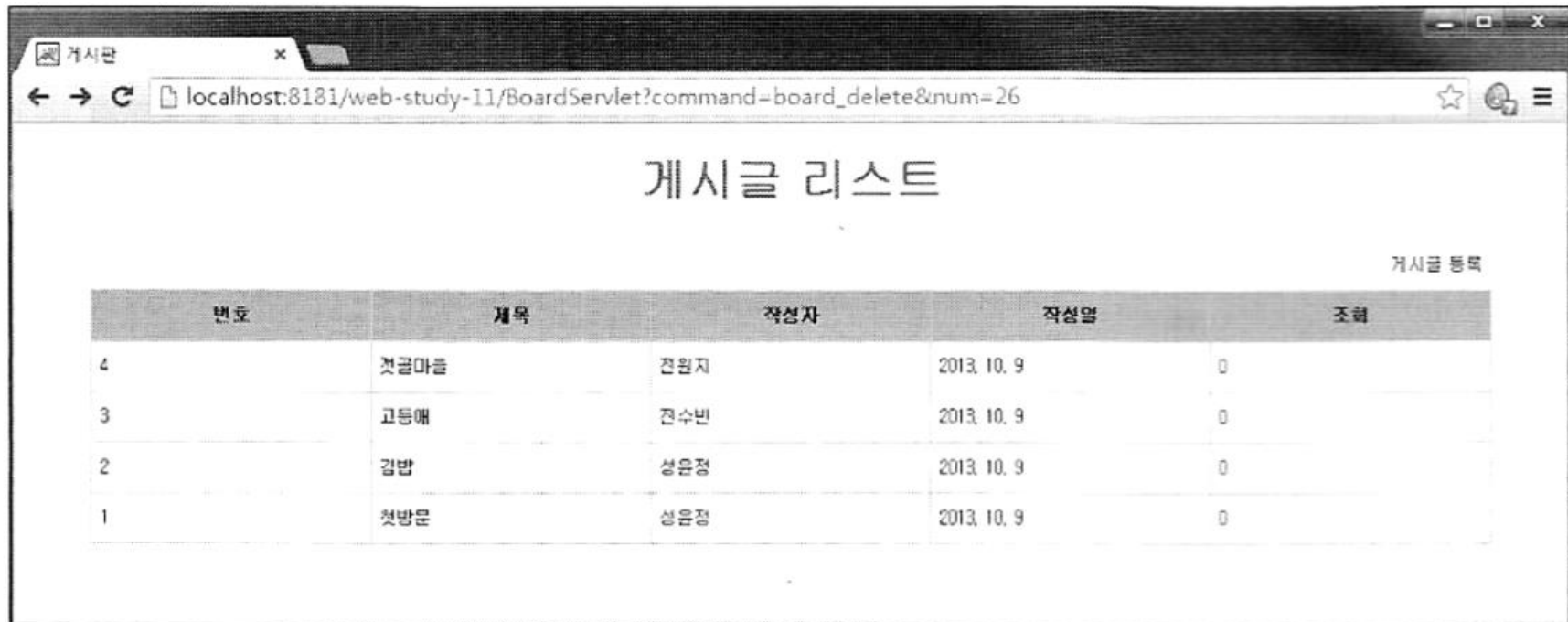
❖ MVC 패턴의 모델

- 비즈니스 로직 처리
- 처리 결과 데이터 저장

게시판 - 모델2 기반의 간단한 MVC 패턴 구현하기

❖ 게시판 스토리보드

○ 게시글 리스트



게시글 리스트

게시글 등록

번호	제목	작성자	작성일	조회
4	첫글마름	전원지	2013. 10. 9	0
3	고등애	전수빈	2013. 10. 9	0
2	김밥	성은정	2013. 10. 9	0
1	첫방문	성은정	2013. 10. 9	0

게시판 - 모델2 기반의 간단한 MVC 패턴 구현하기

❖ 게시판 스토리보드

- 게시글 등록

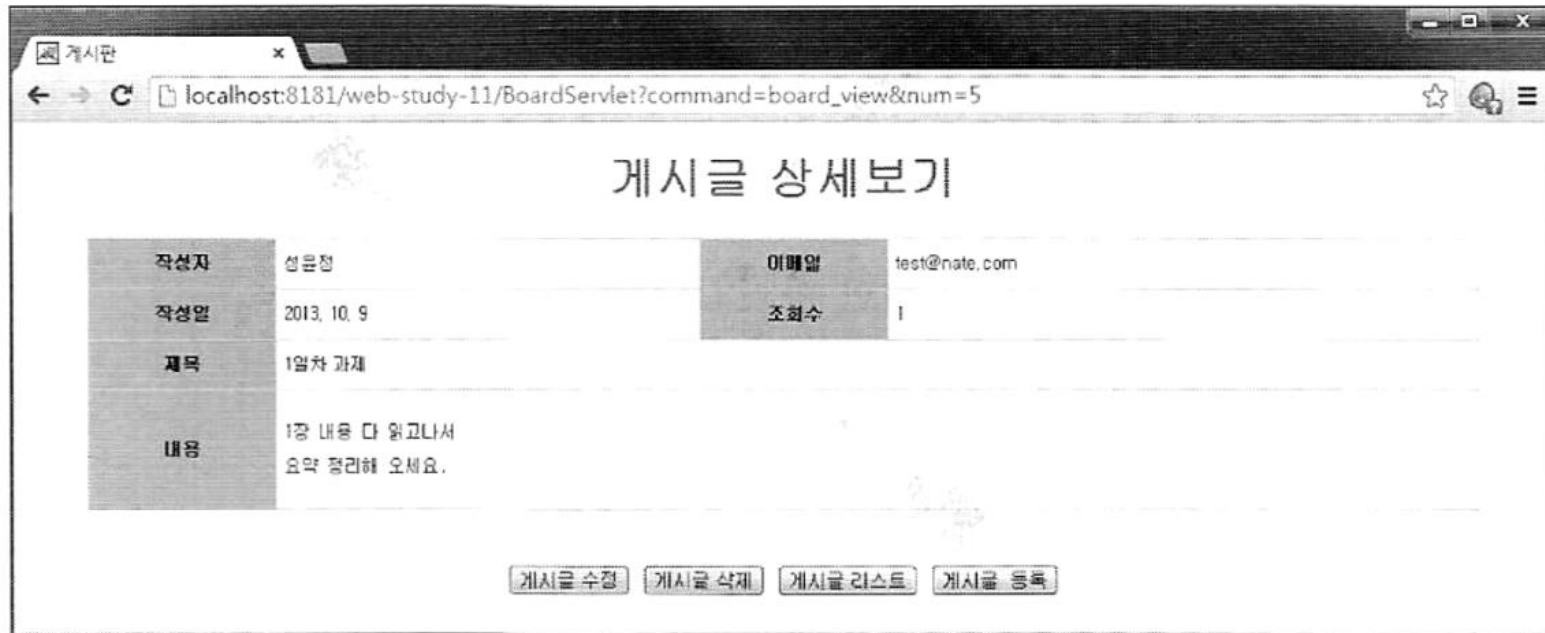
The screenshot shows a web browser window with the title '게시판' (Board). The address bar displays 'localhost:8181/web-study-11/BoardServlet?command=board_write_form'. The main heading is '게시글 등록' (Post Registration). The form is organized into a table-like structure with labels on the left and input fields on the right.

작성자	<input type="text" value="성은정"/> * 필수
비밀번호	<input type="password" value="****"/> * 필수 (게시물 수정 삭제시 필요합니다.)
이메일	<input type="text" value="test@nate.com"/>
제목	<input type="text" value="1주차 과제"/> * 필수
내용	<div><p>1장 내용 다 읽고나서 요약 정리해 오세요.</p></div>

At the bottom of the form, there are three buttons: '등록' (Register), '다시 작성' (Write Again), and '목록' (List).

게시판 - 모델2 기반의 간단한 MVC 패턴 구현하기

- ❖ 게시판 스토리보드
 - 게시글 상세 보기

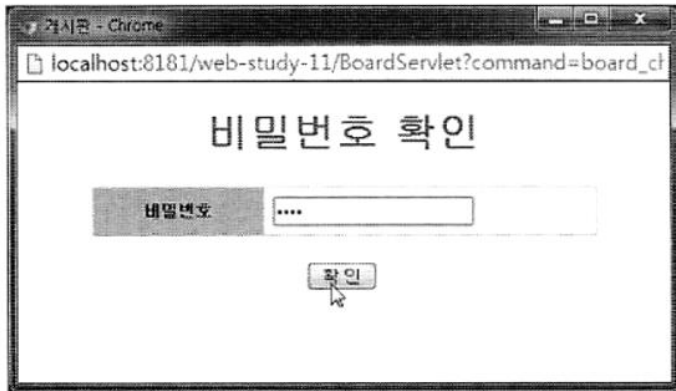


게시판 - 모델2 기반의 간단한 MVC 패턴 구현하기

❖ 게시판 스토리보드

○ 게시물 수정하기

▼ 게시물 비밀번호 입력 화면

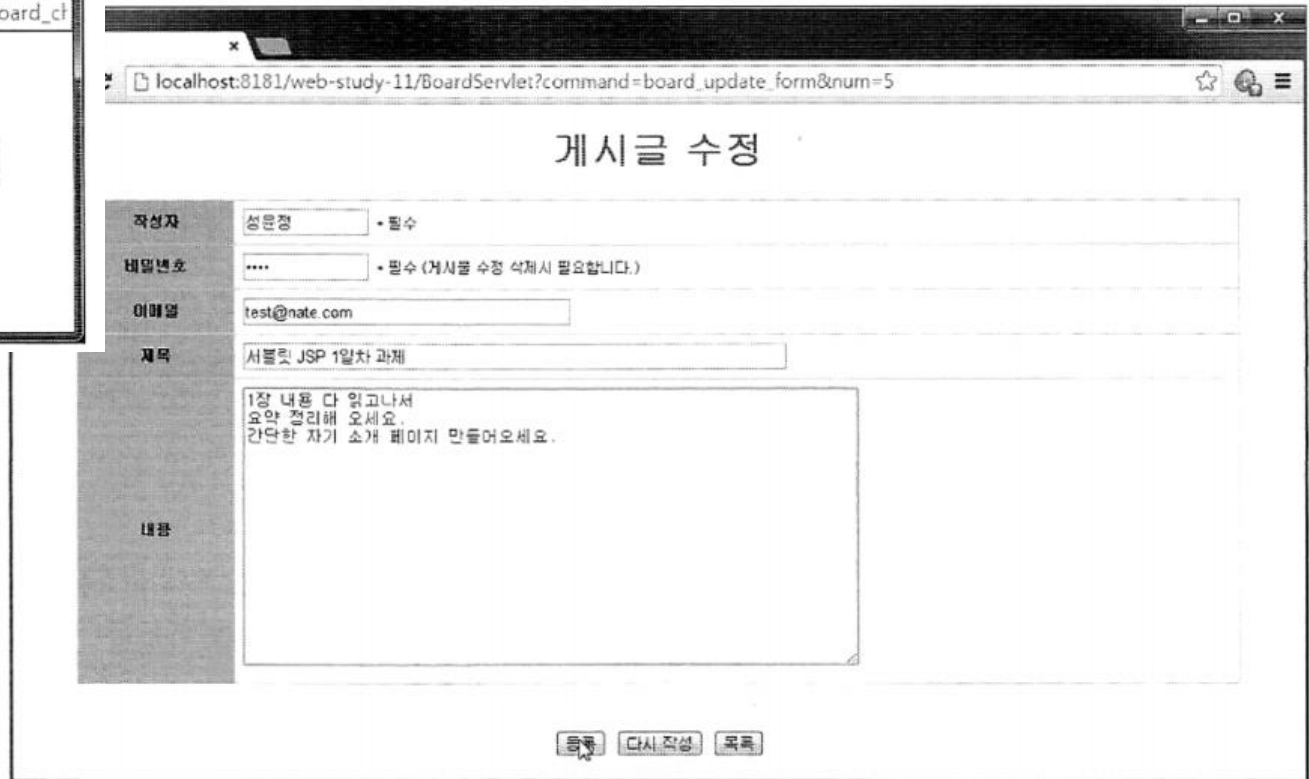


게시판 - Chrome

localhost:8181/web-study-11/BoardServlet?command=board_ct

비밀번호 확인

비밀번호



localhost:8181/web-study-11/BoardServlet?command=board_update_form&num=5

게시글 수정

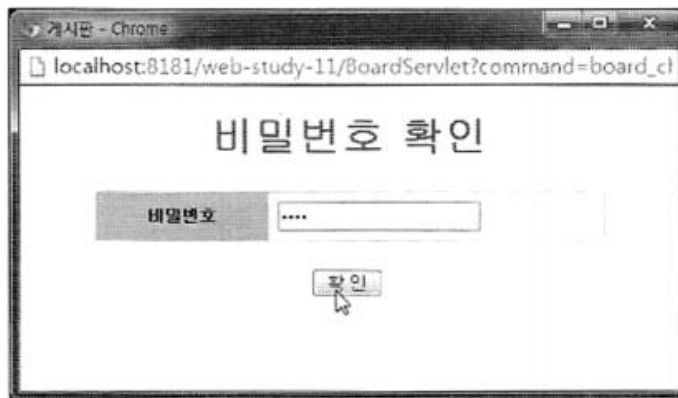
작성자	성은정 • 필수
비밀번호 • 필수 (게시물 수정 삭제시 필요합니다.)
이메일	test@nate.com
제목	서플릿 JSP 1주차 과제
내용	<div>1장 내용 다 읽고나서 요약 정리해 오세요. 간단한 자기 소개 페이지 만들어주세요.</div>

게시판 - 모델2 기반의 간단한 MVC 패턴 구현하기

❖ 게시판 스토리보드

- 게시글 삭제

▼ 게시글 비밀번호 입력 화면



❖ 게시판 파일

○ JSP

[JSP 페이지(위치 : WebContent\Wboard)]

파일	설명
boardList.jsp	게시글 리스트 페이지
boardWrite.jsp	게시글 등록 페이지
boardView.jsp	게시글 상세보기 페이지
boardUpdate.jsp	게시글 수정 페이지
boardCheckPass.jsp	게시글을 수정 혹은 삭제할 수 있는 자격이 있는지 비밀번호 확인을 위한 비밀번호 입력 페이지
checkSuccess.jsp	게시글 수정과 삭제를 위한 페이지

❖ 게시판 파일

○ 서블릿

[컨트롤러 역할을 하는 서블릿 클래스 파일(위치 : srcWcomWsaeyanWcontroller)]

파일	설명
BoardServlet.java	요청을 받아서 요청에 해당하는 Model과 View를 호출하는 역할을 한다.
ActionFactory.java	커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스가 있어야 한다.

❖ 게시판 파일

○ 모델

[모델 역할을 하는 클래스 파일(위치 : src\com\saeyan\controller.action)]

파일	설명	요청 command URL 패턴
Action.java	명령 처리 클래스의 슈퍼 클래스로 사용할 인터페이스	
BoardListActoin.java	게시글 전체 정보를 데이터베이스에서 얻어온다.	board_list
BoardWriteFormAction.java	새로운 게시글 정보를 입력받기 위한 게시글 등록 페이지로 이동하도록 한다.	board_write_form
BoardWriteActoin.java	입력한 게시글 정보를 데이터베이스에 추가한다.	board_write
BoardViewActoin.java	게시글 번호에 해당되는 게시글을 데이터베이스에서 찾아 게시글의 상세 정보를 보여준다.	board_view

❖ 게시판 파일

○ 모델

파일	설명	요청 command URL 패턴
BoardCheckPassFormAction.java	게시글을 수정 혹은 삭제할 수 있는 자격이 있는지를 위해서 비밀번호 확인을 위해서 비밀번호 입력 페이지로 이동하도록 한다.	board_check_pass_form
BoardCheckPassAction.java	게시글 비밀번호가 일치하는지 비교한다.	board_check_pass
BoardUpdateFormAction.java	게시글 수정 페이지로 이동하도록 한다.	board_update_form
BoardUpdateActoin.java	입력한 정보로 데이터베이스에 게시글 정보를 수정한다.	board_update
BoardDeleteActoin.java	데이터베이스에 게시글 정보를 삭제한다.	board_delete

❖ 유효성 체크 자바스크립트

파일	설명
board.js	폼에 입력된 정보가 올바른지 판단하는 자바스크립트

❖ VO

[게시글 정보를 저장할 VO 클래스 (위치: src\Wcom\Wsaeyan\Wdto)]

파일	설명
BoardVO.java	게시글 정보를 저장하는 클래스

❖ DAO

[데이터베이스의 board 테이블을 처리할 DAO 클래스 (위치: src\Wcom\Wsaeyan\Wdao)]

파일	설명
BoardDAO.java	데이터베이스 테이블과 연동해서 작업하는 데이터베이스 처리 클래스

❖ 데이터베이스 준비

[board 테이블 구조]

컬럼명	자료형	제약조건	DEFAULT 값	설명
num	NUMBER(5)	PRIMARY KEY		글번호
pass	VARCHAR2(30)			비밀번호
name	VARCHAR2(30)			작성자
email	VARCHAR2(30)			이메일
title	VARCHAR2(50)			제목
content	VARCHAR2(1000)			내용
readcount	NUMBER(4)		0	조회수
writedate	DATE		SYSDATE	작성일

❖ 테이블 생성

```
CREATE TABLE board(  
    num          NUMBER(5)          PRIMARY KEY,  
    pass         VARCHAR2(30),  
    name         VARCHAR2(30),  
    email        VARCHAR2(30),  
    title        VARCHAR2(50),  
    content      VARCHAR2(1000),  
    readcount    NUMBER(4)          DEFAULT 0,  
    writedate    DATE                DEFAULT SYSDATE  
);
```

❖ 시퀀스 생성

```
create sequence board_seq start with 1 increment by 1;
```

❖ 샘플 데이터 삽입

NUM	PASS	NAME	EMAIL	TITLE	CONTENT	READCOUNT	WRITEDAT
1	1234	성윤정	pinksung@nate.com	첫방문	반갑습니다.	0	13/11/26
2	1234	성윤정	pinksung@nate.com	김밥	맛있어요.	0	13/11/26
3	3333	전수빈	raccon@nate.com	고등애	일식입니다.	0	13/11/26
4	1111	전원지	one@nate.com	갯골마을	돼지삼겹살이 맛있습니다.	0	13/11/26

❖ 샘플 데이터 삽입

```
INSERT INTO board(num, name, email, pass, title, content)
VALUES(board_seq.NEXTVAL, '성윤정', 'pinksung@nate.com', '1234', '첫방문', '반갑습니다.');
```

```
INSERT INTO board(num, name, email, pass, title, content)
VALUES(board_seq.NEXTVAL, '성윤정', 'pinksung@nate.com', '1234', '김밥', '맛있어요.');
```

```
INSERT INTO board(num, name, email, pass, title, content)
VALUES(board_seq.NEXTVAL, '전수빈', 'raccon@nate.com', '3333', '고등애', '일식입니다.');
```

```
INSERT INTO board(num, name, email, pass, title, content)
VALUES(board_seq.NEXTVAL, '전원지', 'one@nate.com', '1111', '갯골마을', '돼지삼겹살이 맛있습니다.');
```

```
commit;
```

Angular 어플리케이션

❖ Todo 뷰

```
app
├── index.html
└── views                // 부분 뷰(Partial View)
    ├── todo.html
```

```
<div ng-controller="ToDoCtrl">
  <div class="page-header">
    ...
  </div>
  <div class="panel">
    ...
  </div>
</div>
```