

## 1. 게시판 작성원리

\* 게시판 \*

- 1) b\_ref (reply family)->윗글과 답변글을 묶어준다.
- 2) b\_step(reply step)->몇번째단계의 답변인지를 나타내는 컬럼  
->같은 그룹내에서도 답변글을 구분
- 3) b\_level(reply level)=> 글의 정렬순서를 지정해주는 컬럼  
(들여쓰기)

	ref	step	level
aaa	0	0	0
_aaa-1	0	1	1
_aaa-2	0	2	2
_aaa-3	0	3	2
bbb	1	0	0
ccc	2	0	0

\* 게시판의 특성 \*

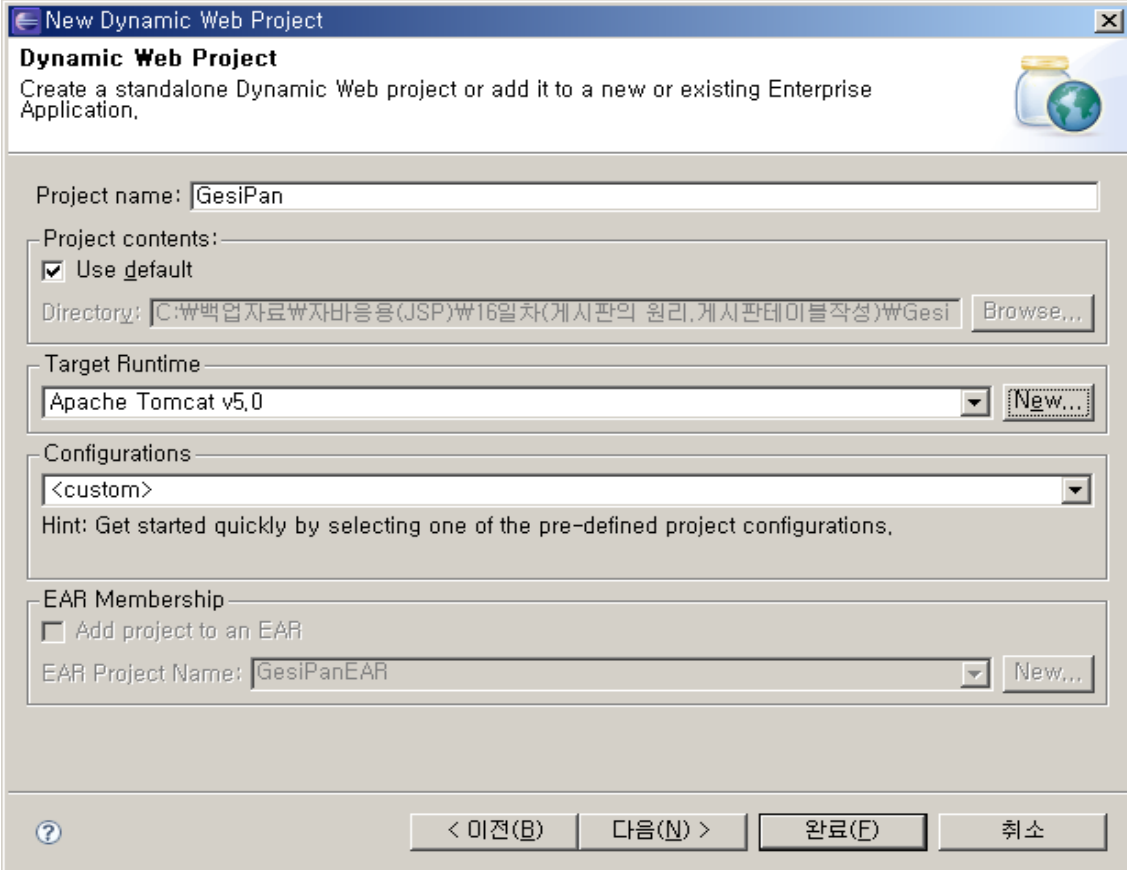
- 1) 맨위의 글이 ->가장 최근에 작성된 글  
=>DB저장순서 => Web상에서의 순서가 틀림.   ccc 2  
  bbb 1  
  aaa 0

=>답변글이 들어가면 DB에 저장의 어떻게 될까?

aaa 0->1

bbb 1=>0

게시판작성 프로젝트명:GesiPan



**New Dynamic Web Project**

**Dynamic Web Project**  
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: GesiPan

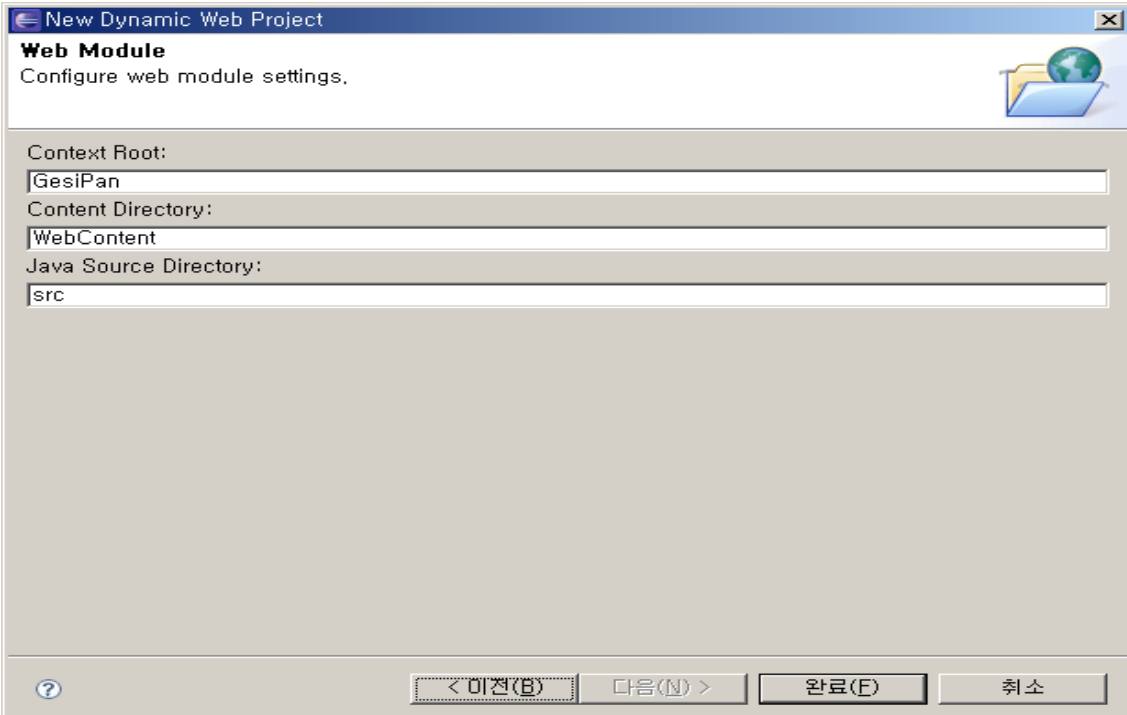
Project contents:  
☒ Use default  
Directory: C:\백업자료\자바응용(JSP)\16일차(게시판의 원리, 게시판테이블작성)\Gesi Browse...

Target Runtime  
Apache Tomcat v5.0 New...

Configurations  
<custom>  
Hint: Get started quickly by selecting one of the pre-defined project configurations.

EAR Membership  
☐ Add project to an EAR  
EAR Project Name: GesiPanEAR New...

< 이전(B) 다음(N) > 완료(F) 취소



**New Dynamic Web Project**

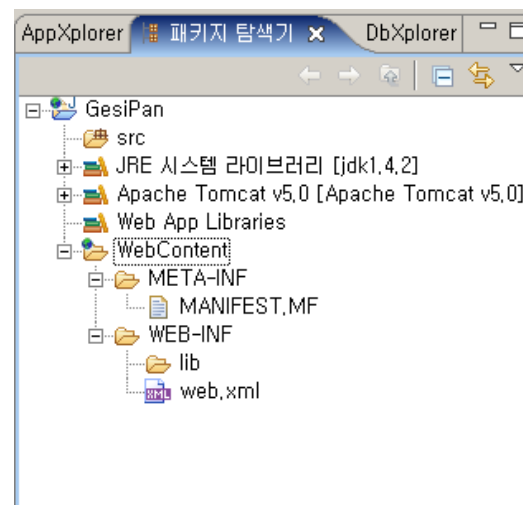
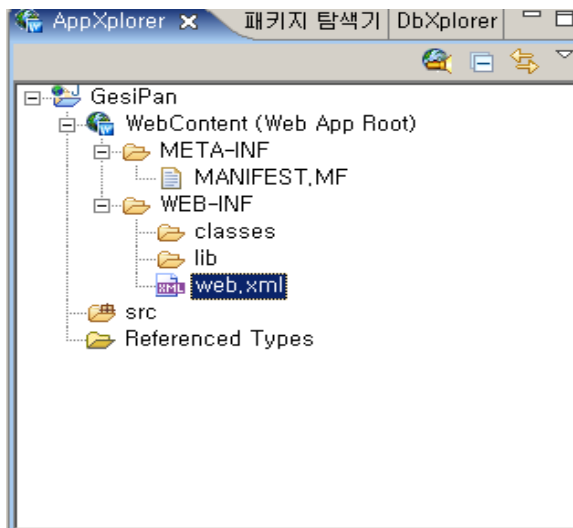
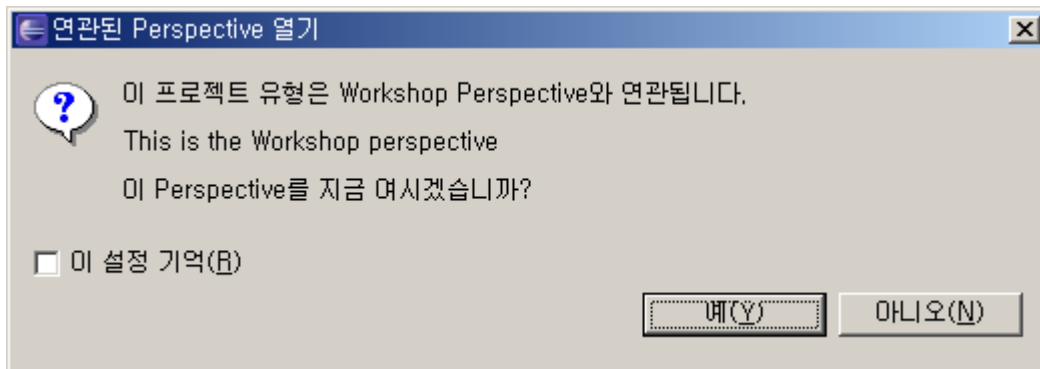
**Web Module**  
Configure web module settings.

Context Root: GesiPan

Content Directory: WebContent

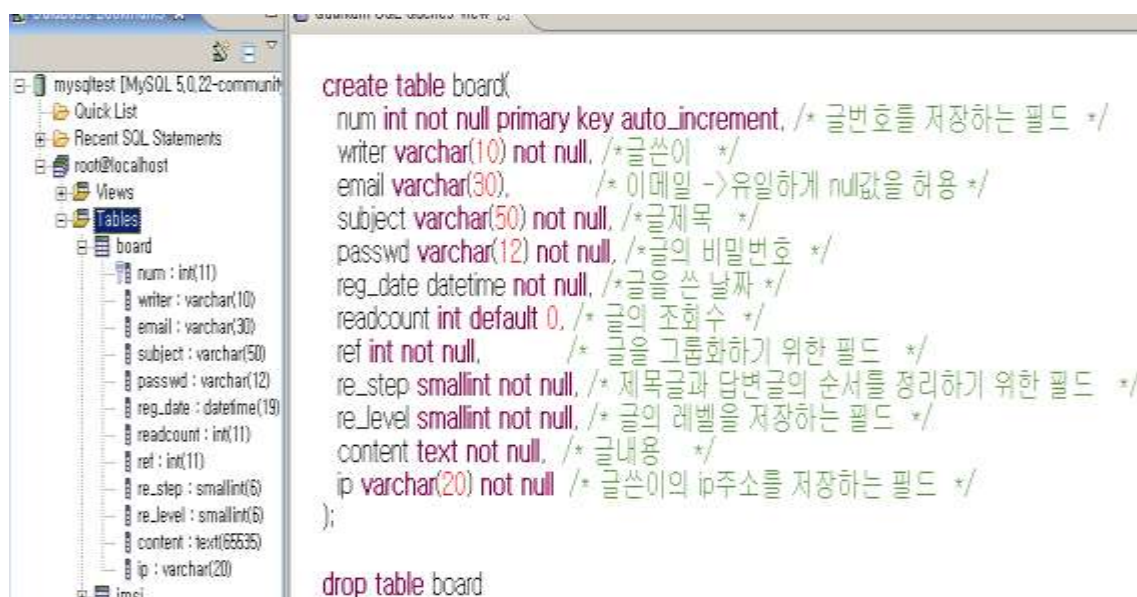
Java Source Directory: src

< 이전(B) 다음(N) > 완료(F) 취소



===게시판테이블 작성하기 위한 Quantum DB DB접속하기=====

board테이블을 저장



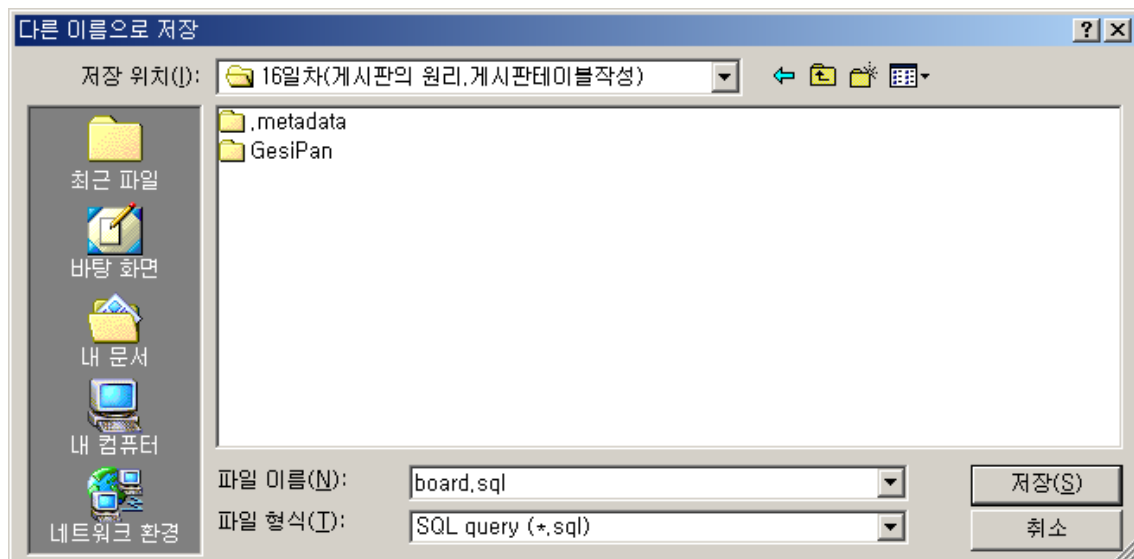
```

create table board(
num int not null primary key auto_increment, /* 글번호를 저장하는 필드 */
writer varchar(10) not null, /*글쓴이 */
email varchar(30), /* 이메일 ->유일하게 null값을 허용 */
subject varchar(50) not null, /*글제목 */
passwd varchar(12) not null, /*글의 비밀번호 */
reg_date datetime not null, /*글을 쓴 날짜 */
readcount int default 0, /* 글의 조회수 */
ref int not null, /* 글을 그룹화하기 위한 필드 */
re_step smallint not null, /* 제목글과 답변글의 순서를 정리*/
re_level smallint not null, /* 글의 레벨을 저장하는 필드 */
content text not null, /* 글내용 */
ip varchar(20) not null /* 글쓴이의 ip주소를 저장하는 필드 */
);

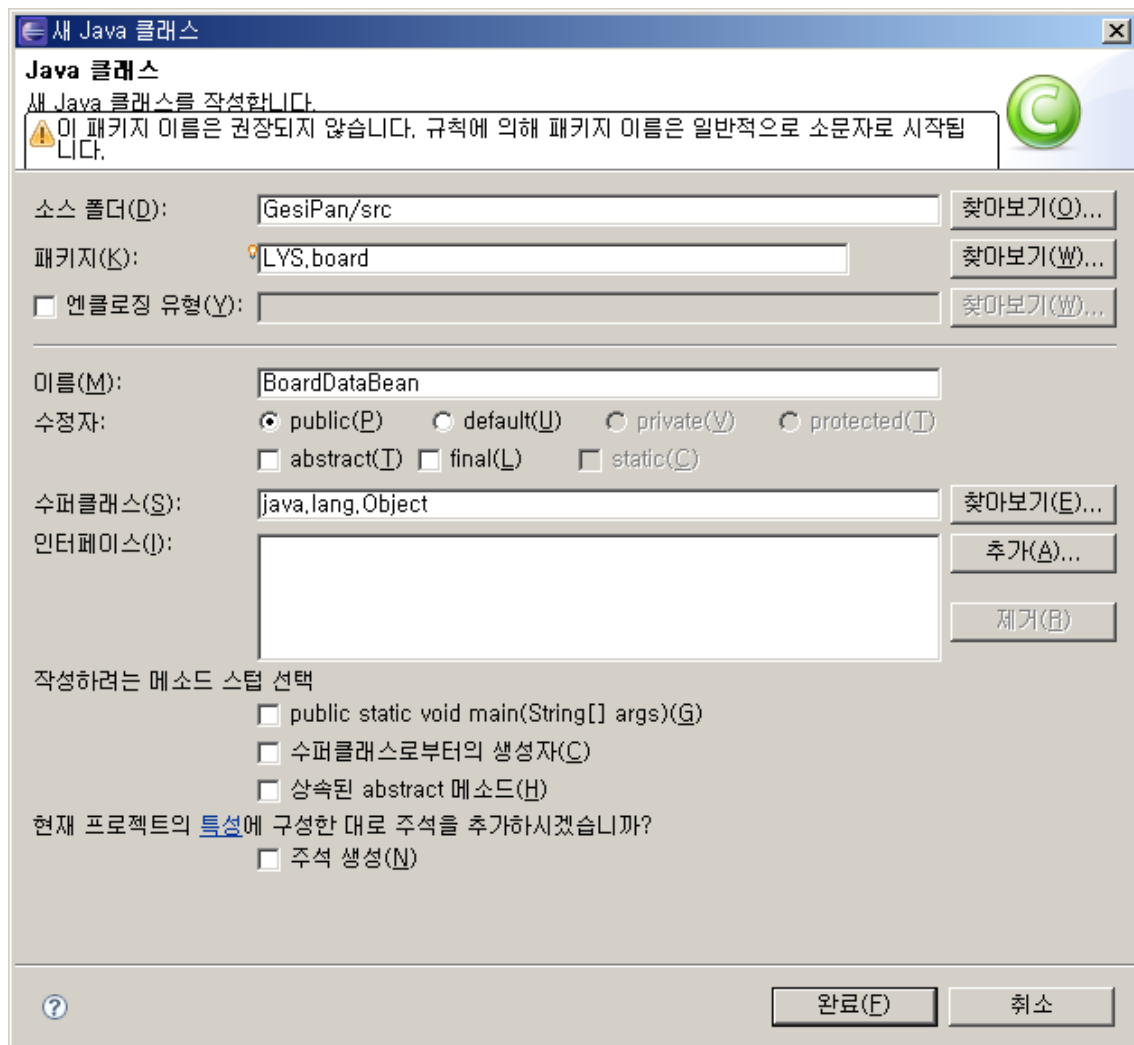
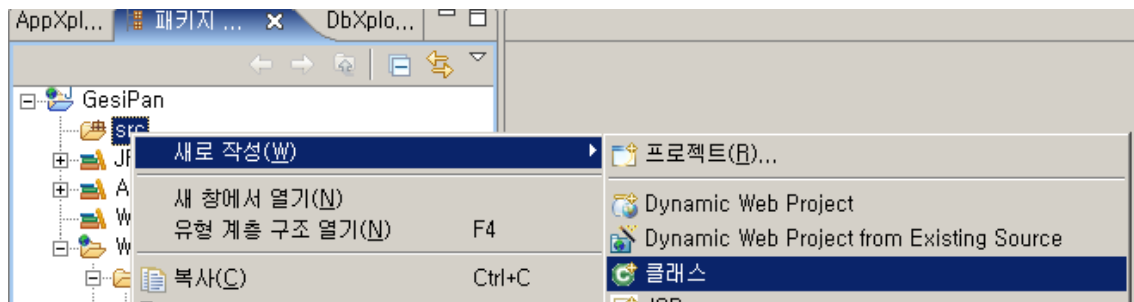
```

---

Board.sql파일로 저장



게시판의 자바빈 작성



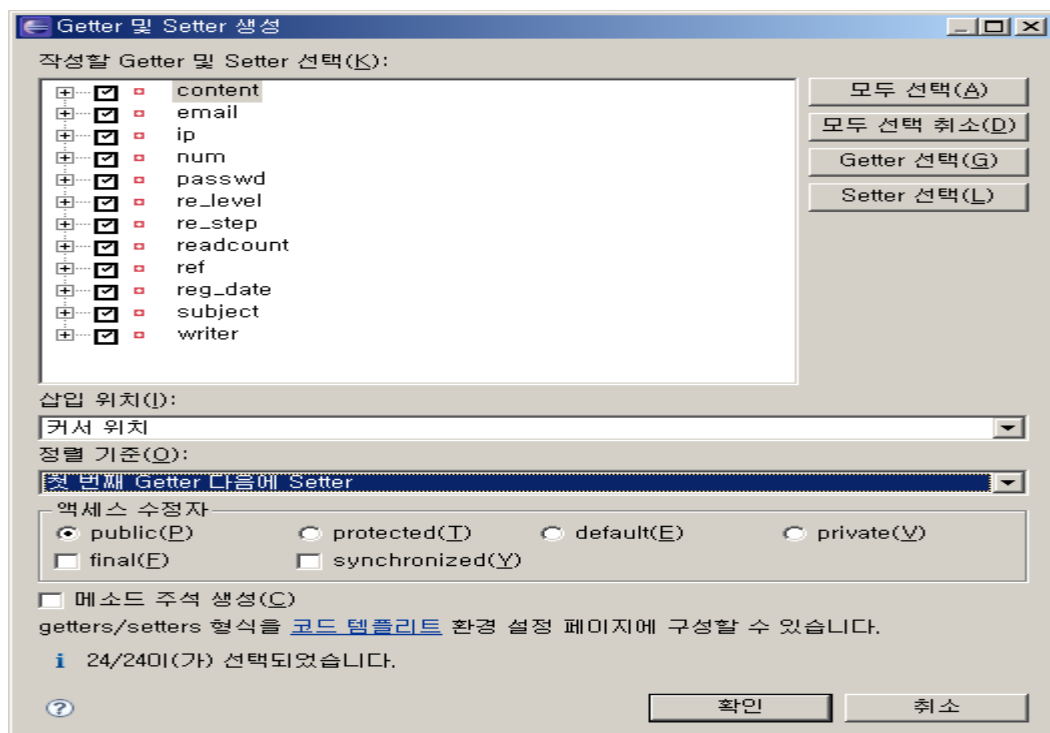
`package LYS.board;`

```
import java.sql.Timestamp;
```

```
public class BoardDataBean {
```

```
    private int num; //게시물 번호
    private String writer; //작성자
    private String subject; //글제목
    private String email; //이메일
    private String content; //글 내용
    private String passwd; //암호
    private Timestamp reg_date; //작성날짜
    private int readcount; //조회수
    private String ip; //작성자의 ip주소
    private int ref; //글 그룹번호
    private int re_step; //제목글과 답변글의 순서를 정리하기 위한 필드
    private int re_level; //글의 레벨을 저장하는 필드
```

소스/Getter 및 Setter 선택메뉴를 선택



```
package LYS.board;
```

```
import java.sql.Timestamp;
```

```
public class BoardDataBean {
```

```
    private int num; //게시물 번호
    private String writer; //작성자
    private String subject; //글제목
    private String email; //이메일
    private String content; //글 내용
    private String passwd; //암호
    private Timestamp reg_date; //작성날짜
    private int readcount; //조회수
    private String ip; //작성자의 ip주소
    private int ref; //글 그룹번호
    private int re_step; //제목글과 답변글의 순서를 정리하기 위한 필드
    private int re_level; //글의 레벨을 저장하는 필드
```

```
    public String getContent() {
        return content;
    }
```

```
    public String getEmail() {
        return email;
    }
```

```
    public String getIp() {
        return ip;
    }
```

```
    public int getNum() {
        return num;
    }
```

```
    public String getPasswd() {
        return passwd;
    }
```

```
    public int getRe_level() {
```

```

        return re_level;
    }
    public int getRe_step() {
        return re_step;
    }
    public int getReadcount() {
        return readcount;
    }
    public int getRef() {
        return ref;
    }
    public Timestamp getReg_date() {
        return reg_date;
    }
    public String getSubject() {
        return subject;
    }
    public String getWriter() {
        return writer;
    }

    public void setContent(String content) {
        this.content = content;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public void setIp(String ip) {
        this.ip = ip;
    }
    public void setNum(int num) {
        this.num = num;
    }
    public void setPasswd(String passwd) {
        this.passwd = passwd;
    }

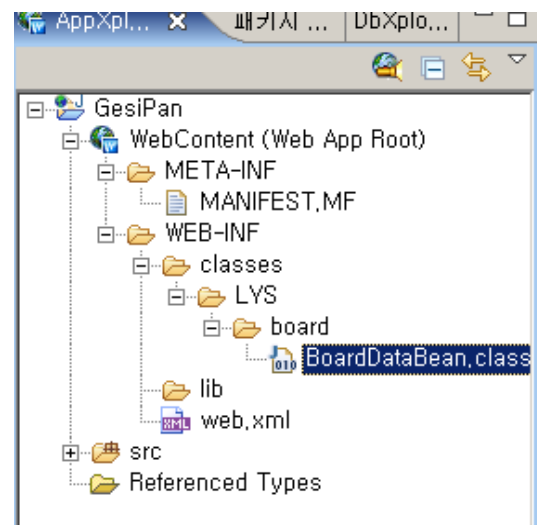
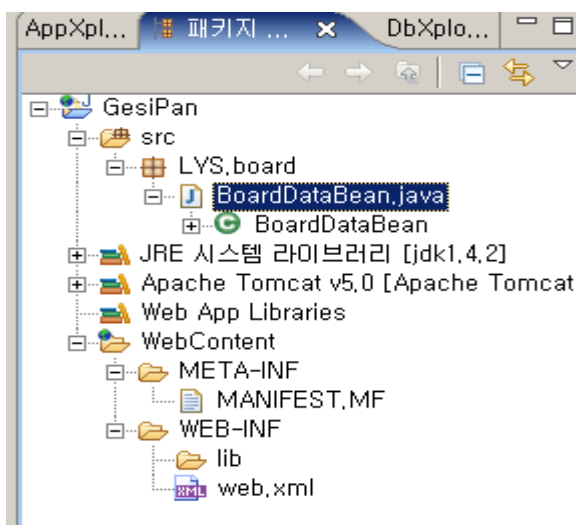
```



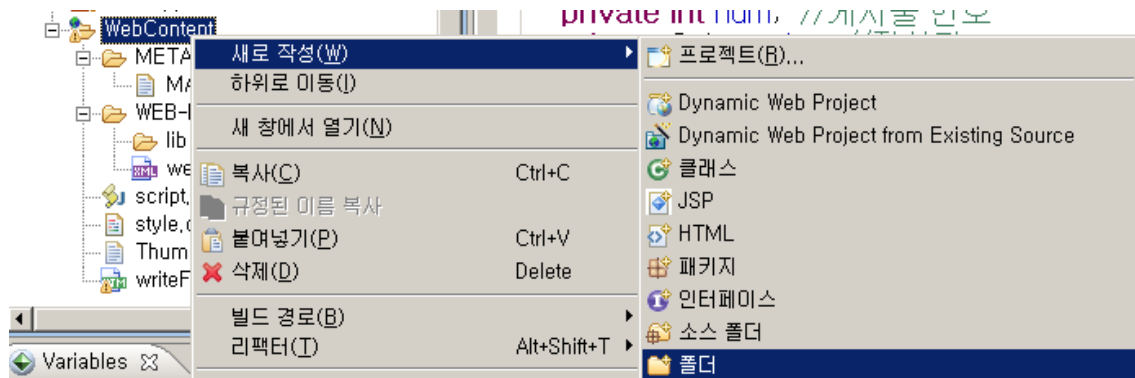
```

public void setRe_level(int re_level) {
    this.re_level = re_level;
}
public void setRe_step(int re_step) {
    this.re_step = re_step;
}
public void setReadcount(int readcount) {
    this.readcount = readcount;
}
public void setRef(int ref) {
    this.ref = ref;
}
public void setReg_date(Timestamp reg_date) {
    this.reg_date = reg_date;
}
public void setSubject(String subject) {
    this.subject = subject;
}
public void setWriter(String writer) {
    this.writer = writer;
}
}

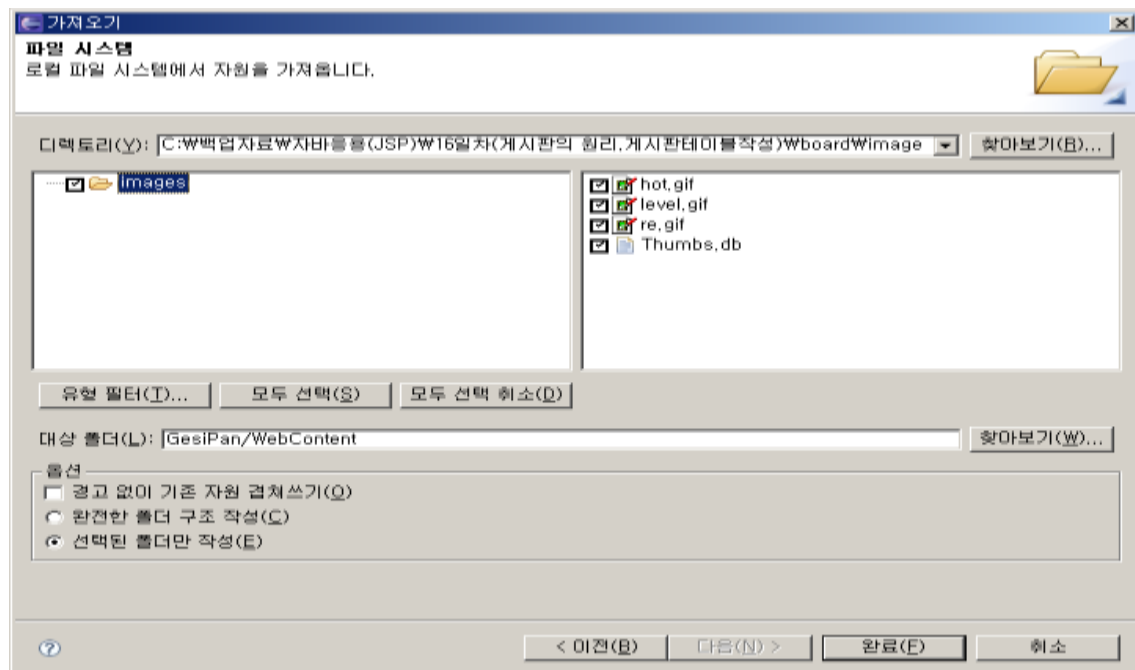
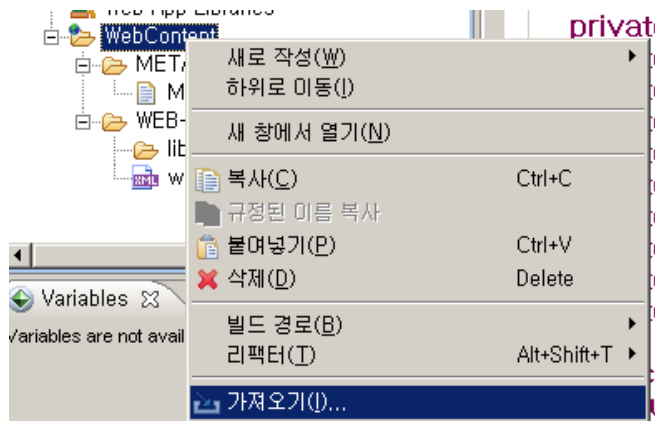
```



Script.js, style.css, writeForm.html, list.html 파일, 이미지 파일을 불러온다.



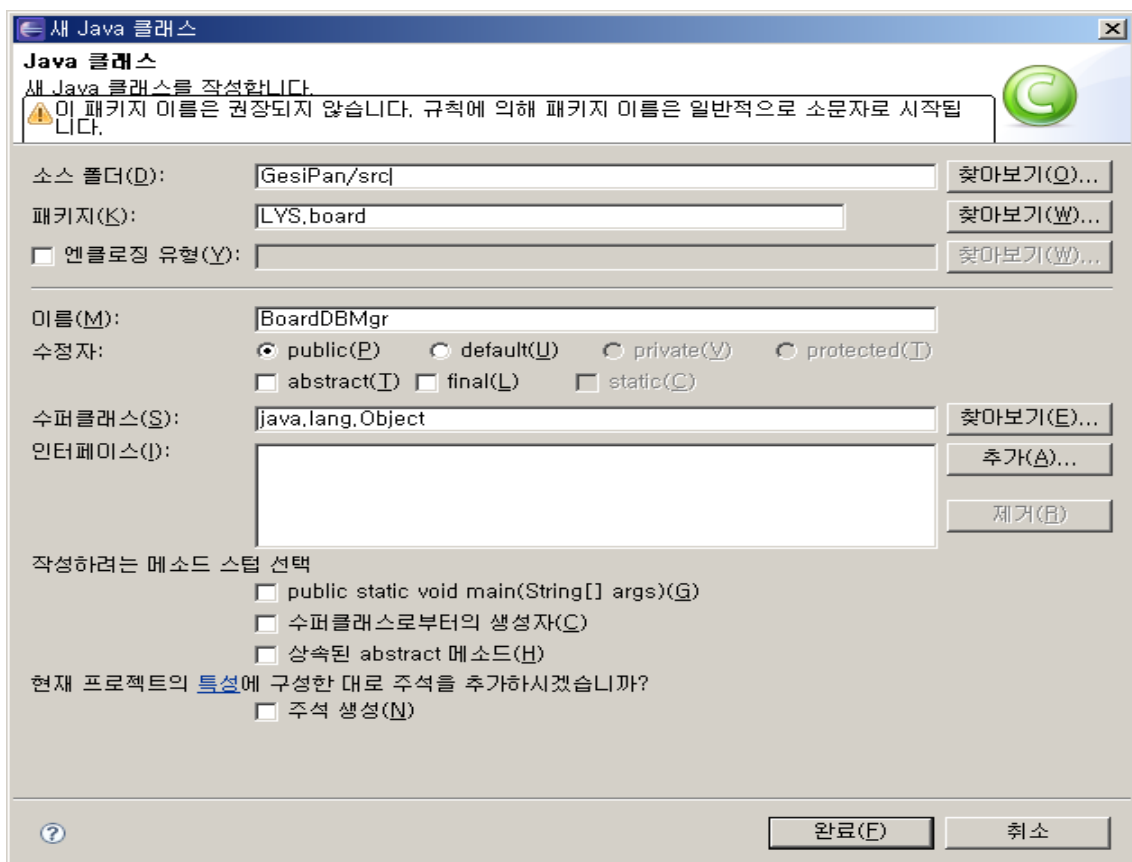
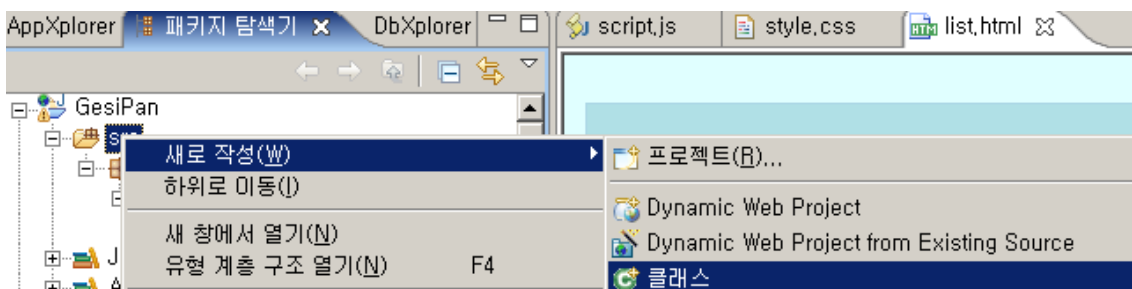
images폴더를 생성



List.html파일의 모습



=====글 목록보기 구현하기=====



=====BoardDBMgr.java작성=====

1. 먼저 DBConnectionMgr를 불러온다.(패키지를 LYS.board로 설정)



2. BoardDBMgr.java파일을 작성

```
=====
```

```
package LYS.board;
```

```
import java.sql.*;
```

```
import java.util.*;
```

```
public class BoardDBMgr {
```

```
    private DBConnectionMgr pool = null;
```

```
    public BoardDBMgr(){
```

```
        try{
```

```
            pool = DBConnectionMgr.getInstance(); //ctrl+enter
```

```
        }catch(Exception e){
```

```
            System.out.println("Error:커넥션 연결 실패");
```

```
        }
```

```
    } //BoardDBMgr
```

//전체 레코드수를 반환해주는 메서드 구현

```
public int getArticleCount(){

    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    int x =0;//반환할 레코드 수

    try{
        con = pool.getConnection();
        String sql="select count(*) from board";

        pstmt = con.prepareStatement(sql);
        rs = pstmt.executeQuery();

        if(rs.next()){ //데이터가 있다면
            x= rs.getInt(1);//레코드의 갯수를 반환
        }
    }catch(Exception ex){
        System.out.println("===getArticleCount()메서드 에러===");
        System.out.println("==라인번호29==");
        ex.printStackTrace();
    }finally{
        pool.freeConnection(con,pstmt,rs);
    }
    return x;
}
```

//글목록을 담고 있는 list를 반환해주는 메서드 구현

```

public List getArticles(int start , int end){

    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    List articleList = null;

    try{
        con = pool.getConnection();
String sql ="select * from board order by ref desc, re_step asc
            limit ?,?"; //한줄에 다 코딩

        pstmt = con.prepareStatement(sql);

        pstmt.setInt(1, start-1);
        pstmt.setInt(2, end);
        rs = pstmt.executeQuery();

        if(rs.next()){
            articleList = new ArrayList(end);
            do{
                BoardDataBean article = new BoardDataBean();
                article.setNum(rs.getInt("num"));
                article.setWriter(rs.getString("writer"));
                article.setEmail(rs.getString("email"));
                article.setSubject(rs.getString("subject"));
                article.setPasswd(rs.getString("passwd"));
                article.setReg_date(rs.getTimestamp("reg_date"));
                article.setReadcount(rs.getInt("readcount"));
                article.setRef(rs.getInt("ref"));
                article.setRe_step(rs.getInt("re_step"));
                article.setRe_level(rs.getInt("re_level"));
                article.setContent(rs.getString("content"));
                article.setIp(rs.getString("ip"));

                articleList.add(article);
            }while(rs.next());
        }
    }catch (Exception e){
        e.printStackTrace();
    }
}

```

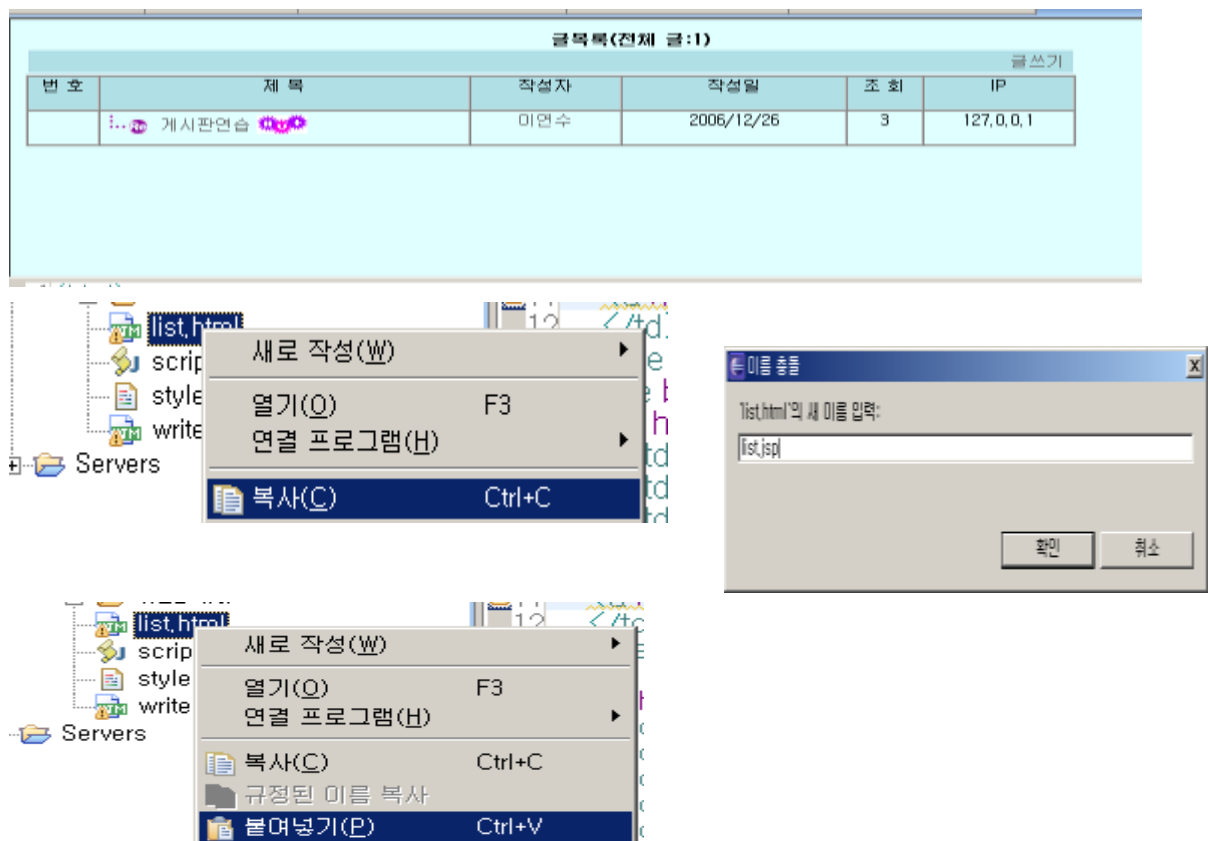
```

        }while(rs.next());
    }
}catch(Exception ex){
    System.out.println("===getArticles()메서드에러===");
    System.out.println("==라인번호57==");
    ex.printStackTrace();
}finally{
    pool.freeConnection(con,pstmt,rs);
}
return articleList;
}
}

```

=====

List.html→코드를 추가하고 list.jsp파일로 저장한다.



=====list.jsp파일을작성=====

```
<%@page contentType="text/html; charset=euc-kr" %>
<%@page import="LYS.board.*" %>
<%@page import="java.util.List" %>
<%@page import="java.text.SimpleDateFormat" %>

<%!
    int pageSize = 10;
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
HH:mm");
%>

<%
    String pageNum = request.getParameter("pageNum");//현재 페이지
    if (pageNum == null){
        pageNum = "1";
    }

    int currentPage = Integer.parseInt(pageNum);//현재페이지
    int startRow = (currentPage-1)*pageSize + 1;
        //현 페이지 글목록에 표시할 시작 글번호
    int endRow = currentPage*pageSize;
        //현 페이지 글목록에 표시할 마지막 글번호
    int count = 0; //전체 레코드수
    int number = 0;

    List articleList = null;

    BoardDBMgr bdpro = new BoardDBMgr();
    count = bdpro.getArticleCount();//총 레코드수를 구한다.

    if(count > 0) {
        articleList = bdpro.getArticles(startRow, pageSize);
    }
}
```



```
}
```

```
number = count-(currentPage-1)*pageSize;
```

```
%>
<html>
<head>
<title>게시판</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#e0fff">
<center><b>글목록</b>
<table width="700">
<tr>
<td align="right" bgcolor="#b0e0e6">
<a href="writeForm.jsp">글쓰기</a>
</td>
</tr>
</table>

<%
    if(count == 0){
%>
<table width="700" border="1" cellpadding="0" cellspacing="0">
<tr>
<td align="center">
게시판에 저장된 글이 없습니다.
</td>
</tr>
</table>

<% }else { %>

<table border="1" width="700" cellpadding="0" cellspacing="0"
align="center">
<tr height="30" bgcolor="#b0e0e6">
<td align="center" width="50" >번호</td>
```

```

<td align="center" width="250" >제 목</td>
<td align="center" width="100" >작성자 </td>
<td align="center" width="150" >작성일</td>
<td align="center" width="50" >조 회</td>
<td align="center" width="100" >IP</td>
</tr>

```

<%

```

for(int i = 0 ; i < articleList.size();i++){
    BoardDataBean article = (BoardDataBean)articleList.get(i);

```

%>

```

<tr height="30">
<td align="center" width="50" ><%=number--%></td>
<td width="250" >
    <%
        int wid = 0;
        if(article.getRe_level() > 0){
            wid=5*(article.getRe_level());
        %>
        " height="16">
        
        <% }else{ %>
            " height="16">
        <% } %>
        <a href="content.jsp?num=<%=article.getNum() %>&
            pageNum=<%=currentPage %>">
            <%=article.getSubject()%></a>
            <% if(article.getReadcount() >=20) { %>
                <%=}%>
    </td>
    <td align="center" width="100">
        <a href="mailto:<%=article.getEmail()%>">
        <%=article.getWriter()%></a></td>
    <td align="center" width="150">
        <%=sdf.format(article.getReg_date())%></td>

```

```

        <td align="center" width="50"><%=article.getReadcount()%></td>
        <td align="center" width="100"><%=article.getIp()%></td>
    </tr>
    <% } %>
</table>
<% } %>

<%

if(count > 0){

    int pageCount = count / pageSize + (count % pageSize==0?0:1);
    int startPage = currentPage/pageSize + 1;
    int endPage = startPage+10;

    if(endPage > pageCount) endPage = pageCount;

    if(startPage > 10) { %>
        <a href = "list.jsp?pageNum=<%=startPage-10 %>">[이전]</a>
    <% }

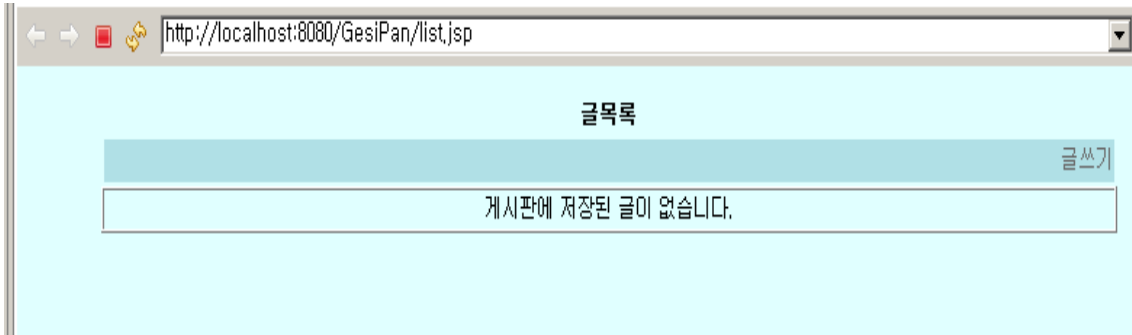
    for(int i = startPage; i<=endPage;i++) { %>
        <a href="list.jsp?pageNum=<%=i%>">[<%=i %>]</a>
    <%
    }
    if(endPage < pageCount) { %>
        <a href="list.jsp?pageNum=<%=startPage+10 %>">[다음]</a>
    <%
    }
}

%>

</center>
</body>
</html>

```

===최종실습(데이터가 없는 경우)=====



=====최종실습(데이터를 집어넣는 경우)=====

```
select * from board
```

```
insert into board
values(1,'이연수','nup49rok1@empal.com','[공지사항]게시판사용법',
      '1234','2006-12-26 17:23',3,0,0,0,'게시판 테스트','127.0.0.1');
```

```
insert into board values(2,'홍길동','hong@empal.com','답변게시판사용법',
      '1111','2006-12-26 17:23',1,0,1,1,'게시판 테스트2','127.0.0.1');
```

```
insert into board values(3,'이미자','lee@empal.com','답변의 답변게시판',
      '1111','2006-12-26 17:23',1,0,2,1,'게시판 테스트3','127.0.0.1');
```

```
insert into board values(4,'고길동','Go@empal.com','또 다른 게시판
테스트', '1111','2006-12-26 17:23',1,1,0,0,'답변형게시판
완성','127.0.0.1');
```

The screenshot shows a web browser window with the address bar displaying 'http://localhost:8080/GesiPan/list.jsp'. The page has a light blue background. At the top, there is a header '글목록' (Post List) and a button '글쓰기' (Write Post). Below the header, a table displays a list of posts. The table has 6 columns: 번호 (Number), 제목 (Subject), 작성자 (Author), 작성일 (Date), 조회 (Views), and IP. The table contains 4 rows of data.

번호	제 목	작성자	작성일	조 회	IP
4	또 다른 게시판 테스트	고길동	2006-12-26 17:23	1	127.0.0.1
3	[공지사항]게시판사용법	이연수	2006-12-26 17:23	3	127.0.0.1
2	!..답변게시판사용법	홍길동	2006-12-26 17:23	1	127.0.0.1
1	!..답변의 답변게시판	이미자	2006-12-26 17:23	1	127.0.0.1

[1]

//코드 추가한 후

```
insert into board values(5,'테스트','te@empal.com','연습',
                        '1234','2006-12-26 17:23',8,1,2,2,'연습','127.0.0.1');
```



번호	제 목	작성자	작성일	조 회	IP
5	또 다른 게시판 테스트	고길동	2006-12-26 17:23	1	127.0.0.1
4	연습	테스트	2006-12-26 17:23	8	127.0.0.1
3	[공지사항] 게시판사용법	이연수	2006-12-26 17:23	3	127.0.0.1
2	답변게시판사용법	홍길동	2006-12-26 17:23	1	127.0.0.1
1	답변의 답변게시판	이미자	2006-12-26 17:23	1	127.0.0.1