1. 답변글 달기 로직 리뷰

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | 제목 | ref | re\_step | re\_level |  | 1. num은 1씩 증가 |
| 1 | 글1 | 1 | 0 | 0 | 2. ref - 원글이면 num과 같고, |
| 2 | 글2 | 2 | 0 | 0 | 답변글이면 원글의 ref와 같다 |
| 3 | 글3 | 3 | 0 | 0 | 3. re\_step은 원글이면 0 |
| 4 | 글1-1 | 1 | 0+1 | 1 | 답변글이면 다음과 같이 진행한다 |
| 5 | 글2-1 | 2 | 0+1+1+1+1 | 1 | ⓐ ref가 같고 원글보다 re\_step가 |
| 6 | 글3-1 | 3 | 0+1+1 | 1 | 큰 값이 있으면 그 글의 re\_step을 re\_step+1 |
| 7 | 글2-2 | 2 | 0+1+1+1 | 1 | ⓑ 답변글 re\_step = 원글 re\_step + 1 |
| 8 | 글2-3 | 2 | 0+1 | 1 | 4. re\_level - 원글이면 0 |
| 9 | 글2-3-1 | 2 | 0+1+1 | 2 | 답변글이면 원글의 re\_level+1 |
| 10 | 글3-2 | 3 | 0+1 | 1 |  |

List.jsp를 조정하여 아래와 같이 출력(SELECT \* FROM board ORDER BY ref DESC, re\_step ASC;)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| num | 제목 | ref | re\_step | re\_level |
| 3 | 글 3 | 3 | 0 | 0 |
| 10 | 글 3-2 | 3 | 0+1 | 1 |
| 6 | 글 3-1 | 3 | 0+1+1 | 1 |
| 2 | 글 2 | 2 | 0 | 0 |
| 8 | 글 2-3 | 2 | 0+1 | 1 |
| 9 | 글 2-3-1 | 2 | 0+1+1 | 2 |
| 7 | 글 2-2 | 2 | 0+1+1+1 | 1 |
| 5 | 글 2-1 | 2 | 0+1+1+1+1 | 1 |
| 1 | 글 1 | 1 | 0 | 0 |
| 4 | 글 1-1 | 1 | 0+1 | 1 |

**board.sql**

**DROP** **TABLE** BOARD;

**CREATE** **TABLE** BOARD(

NUM NUMBER(5) **PRIMARY** **KEY**, -- 글번호

WRITER VARCHAR2(30) **NOT** **NULL**, -- 작성자

SUBJECT VARCHAR2(100) **NOT** **NULL**, -- 글제목

CONTENT VARCHAR2(1000) **NOT** **NULL**, -- 글본문

EMAIL VARCHAR2(30), -- 작성자 메일주소

HIT NUMBER(5) **DEFAULT** 0,

PW VARCHAR2(30) **NOT** **NULL**, -- 글삭제 시 쓸 비밀번호

REF NUMBER(5) **NOT** **NULL**, -- 글그룹

RE\_STEP NUMBER(5) **NOT** **NULL**, -- 같은 글 그룹끼리 출력 순서(원글:0)

RE\_LEVEL NUMBER(5) **NOT** **NULL**, -- 답변글일 경우 들여쓰기 정도(원글:0)

IP VARCHAR2(20) **NOT** **NULL**, -- 작성자의 컴퓨터 IP 주소

RDATE **DATE** **DEFAULT** SYSDATE); -- 글쓴 시점(날짜+시간)

-- 글갯수

**SELECT** COUNT(\*) **FROM** BOARD;

-- 글목록 (최신글이 위로)

**SELECT** \* **FROM** BOARD **ORDER** **BY** REF **DESC**;

-- 글쓰기

**SELECT** NVL(MAX(NUM),0)+1 **FROM** BOARD; -- 새로운 글에 할당될 글번호

**INSERT** **INTO** BOARD (NUM, WRITER, SUBJECT, CONTENT, EMAIL, PW, REF, RE\_STEP, RE\_LEVEL, IP)

**VALUES** ((**SELECT** NVL(MAX(NUM),0)+1 **FROM** BOARD), '홍길동','제목임다','본문임다',**NULL**,

'111', (**SELECT** NVL(MAX(NUM),0)+1 **FROM** BOARD), 0, 0,'192.168.20.41');

**SELECT** \* **FROM** BOARD **ORDER** **BY** NUM **DESC**;

-- 글번호로 글(DTO) 가져오기

**SELECT** \* **FROM** BOARD **WHERE** NUM=1;

-- HIT수 올리기

**UPDATE** BOARD **SET** HIT = HIT+1 **WHERE** NUM=1;

-- 글 수정

**UPDATE** BOARD **SET** WRITER = '김길동',

SUBJECT = '바뀐 제목',

CONTENT = '바뀐 내용',

EMAIL = 'yi@naver.com',

PW = '111',

IP = '192.168.20.41'

**WHERE** NUM=1;

-- 글삭제

**DELETE** **FROM** BOARD **WHERE** NUM=1 **AND** PW='111';

**COMMIT**;

**SELECT** \* **FROM** BOARD **ORDER** **BY** NUM **DESC**;

-- 조회수 조작

**UPDATE** BOARD **SET** HIT = 11 **WHERE** NUM=2;

**UPDATE** BOARD **SET** HIT = MOD(NUM, 12);

-- 페이징을 위한 top-N구문

**SELECT** \* **FROM** BOARD **ORDER** **BY** REF **DESC**, RE\_STEP **ASC**; -- 1단계

**SELECT** ROWNUM RN, A.\* **FROM** (**SELECT** \* **FROM** BOARD **ORDER** **BY** REF **DESC**, RE\_STEP **ASC**) A; -- 2단계

**SELECT** \* **FROM** (**SELECT** ROWNUM RN, A.\* **FROM** (**SELECT** \* **FROM** BOARD **ORDER** **BY** REF **DESC**, RE\_STEP **ASC**) A)

**WHERE** RN **BETWEEN** 1 **AND** 10;-- 3단계

-- 답변글을 위한 sql문

**UPDATE** BOARD **SET** RE\_STEP = RE\_STEP+1

**WHERE** REF=215 **AND** RE\_STEP>0;

truncate **table** board;

**BoardDto.java**

**package** com.tj.dto;

**import** java.sql.Timestamp;

**public** **class** BoardDto {

**private** **int** num;

**private** String writer;

**private** String subject;

**private** String content;

**private** String email;

**private** **int** hit;

**private** String pw;

**private** **int** ref;

**private** **int** re\_step;

**private** **int** re\_level;

**private** String ip;

**private** Timestamp rDate;

// 매개변수 없는 생성자, 매개변수 있는 생성자 getters & setters, toString()

**public** BoardDto() {}

**public** BoardDto(**int** num, String writer, String subject, String content, String email, **int** hit, String pw, **int** ref,

**int** re\_step, **int** re\_level, String ip, Timestamp rDate) {

**this**.num = num;

**this**.writer = writer;

**this**.subject = subject;

**this**.content = content;

**this**.email = email;

**this**.hit = hit;

**this**.pw = pw;

**this**.ref = ref;

**this**.re\_step = re\_step;

**this**.re\_level = re\_level;

**this**.ip = ip;

**this**.rDate = rDate;

}

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

**public** **void** setNum(**int** num) {**this**.num = num;}

**public** String getWriter() {**return** writer;}

**public** **void** setWriter(String writer) {**this**.writer = writer;}

**public** String getSubject() {**return** subject;}

**public** **void** setSubject(String subject) {**this**.subject = subject;}

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

**public** **void** setContent(String content) {**this**.content = content;}

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

**public** **void** setEmail(String email) {**this**.email = email;}

**public** **int** getHit() {**return** hit; }

**public** **void** setHit(**int** hit) {**this**.hit = hit;}

**public** String getPw() {**return** pw;}

**public** **void** setPw(String pw) {**this**.pw = pw;}

**public** **int** getRef() {**return** ref;}

**public** **void** setRef(**int** ref) {**this**.ref = ref;}

**public** **int** getRe\_step() {**return** re\_step;}

**public** **void** setRe\_step(**int** re\_step) {**this**.re\_step = re\_step;}

**public** **int** getRe\_level() {**return** re\_level;}

**public** **void** setRe\_level(**int** re\_level) {**this**.re\_level = re\_level;}

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

**public** **void** setIp(String ip) {**this**.ip = ip;}

**public** Timestamp getrDate() {**return** rDate;}

**public** **void** setrDate(Timestamp rDate) {**this**.rDate = rDate;}

@Override

**public** String toString() {

**return** num + " / " + writer + " / " + subject + " / " + content

+ " / " + email + " / " + hit + " / " + pw + ", ref=" + ref

+ ", re\_step=" + re\_step

+ ", re\_level=" + re\_level + ", ip=" + ip + ", rDate=" + rDate;

}

}

**BoardDao.java**

**package** com.tj.dao;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.sql.Timestamp;

**import** java.util.ArrayList;

**import** javax.naming.Context;

**import** javax.naming.InitialContext;

**import** javax.naming.NamingException;

**import** javax.sql.DataSource;

**import** com.tj.dto.BoardDto;

**public** **class** BoardDao {

// 글목록출력할 때 쓸 listBoard()

// 글 갯수 가져오는 getBoradTotalCnt()

// 글쓰기 insertBoard(dto)

//죄회수하나 올리기 hitUp(int num)

// 죄회수하나 올리기 hitUp(String num)

// 글번호로 글한줄 가져오기 BoardDto getBoardOneLine(int num)

// 글번호로 글한줄 가져오기 BoardDto getBoardOneLine(String num)

// 글 수정 updateBoard(dto)

// 글 삭제 deleteBoard(int num, String pwChk)

**public** **static** **final** **int** ***SUCCESS*** = 1;

**public** **static** **final** **int** ***FAIL*** = 0;

**private** **static** BoardDao *instance*;

**public** **static** BoardDao getInstance() {

**if**(*instance*==**null**) {

*instance* = **new** BoardDao();

}

**return** *instance*;

}

**private** BoardDao() {}

**private** Connection getConnection() **throws** SQLException {

Connection conn = **null**;

**try** {

Context ctx = **new** InitialContext();

DataSource ds = (DataSource) ctx.lookup("java:comp/env/jdbc/Oracle11g");

conn = ds.getConnection();

} **catch** (NamingException e) {

System.***out***.println(e.getMessage());

}

**return** conn;

}

// 글갯수 가져오는 getBoardTotalCnt()

**public** **int** getBoardTotalCnt() {

**int** totCnt=0;

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT COUNT(\*) FROM BOARD";

**try** {

conn = getConnection();

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);

rs.next();

totCnt = rs.getInt(1);

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(stmt!=**null**) stmt.close();

**if**(conn!=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

**return** totCnt;

}

// 글목록 출력할 때 쓸 listBoard()

**public** ArrayList<BoardDto> listBoard(){

ArrayList<BoardDto> dtos = **new** ArrayList<BoardDto>();

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM BOARD ORDER BY REF DESC";

**try** {

conn = getConnection();

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);

**while**(rs.next()) {

**int** num = rs.getInt("num");

String writer = rs.getString("writer");

String subject = rs.getString("subject");

String content = rs.getString("content");

String email = rs.getString("email");

**int** hit = rs.getInt("hit");

String pw = rs.getString("pw");

**int** ref = rs.getInt("ref");

**int** re\_step = rs.getInt("re\_step");

**int** re\_level= rs.getInt("re\_level");

String ip = rs.getString("ip");

Timestamp rDate= rs.getTimestamp("rDate");

dtos.add(**new** BoardDto(num, writer, subject, content, email,

hit, pw, ref, re\_step, re\_level, ip, rDate));

}

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(stmt!=**null**) stmt.close();

**if**(conn!=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

**return** dtos;

}

// 글목록 출력할 때 쓸 listBoard()

**public** ArrayList<BoardDto> listBoard(**int** startRow, **int** endRow){

ArrayList<BoardDto> dtos = **new** ArrayList<BoardDto>();

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM (SELECT ROWNUM RN, A.\* FROM "

+ "(SELECT \* FROM BOARD ORDER BY REF DESC, RE\_STEP ASC) A) WHERE RN BETWEEN ? AND ?";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(sql);

pstmt.setInt(1, startRow);

pstmt.setInt(2, endRow);

rs = pstmt.executeQuery();

**while**(rs.next()) {

**int** num = rs.getInt("num");

String writer = rs.getString("writer");

String subject = rs.getString("subject");

String content = rs.getString("content");

String email = rs.getString("email");

**int** hit = rs.getInt("hit");

String pw = rs.getString("pw");

**int** ref = rs.getInt("ref");

**int** re\_step = rs.getInt("re\_step");

**int** re\_level= rs.getInt("re\_level");

String ip = rs.getString("ip");

Timestamp rDate= rs.getTimestamp("rDate");

dtos.add(**new** BoardDto(num, writer, subject, content, email,

hit, pw, ref, re\_step, re\_level, ip, rDate));

}

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(conn!=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

**return** dtos;

}

// 글쓰기 insertBoard(dto)

**public** **int** insertBoard(BoardDto dto) {

**int** result = ***FAIL***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "INSERT INTO BOARD"

+ " (NUM, WRITER, SUBJECT, CONTENT, EMAIL, PW, REF, RE\_STEP, RE\_LEVEL, IP)"

+ " VALUES ((SELECT NVL(MAX(NUM),0)+1 FROM BOARD), ?,?,?,?," +

" ?, (SELECT NVL(MAX(NUM),0)+1 FROM BOARD), 0, 0,?)";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, dto.getWriter());

pstmt.setString(2, dto.getSubject());

pstmt.setString(3, dto.getContent());

pstmt.setString(4, dto.getEmail());

pstmt.setString(5, dto.getPw());

pstmt.setString(6, dto.getIp());

result = pstmt.executeUpdate();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

**return** result;

}

// 글번호로 글 dto가져오기 getBoardOneLine(int num)

**public** BoardDto getBoardOneLine(**int** num) {

BoardDto dto = **null**;

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM BOARD WHERE NUM="+num;

**try** {

conn = getConnection();

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);

**while**(rs.next()) {

String writer = rs.getString("writer");

String subject = rs.getString("subject");

String content = rs.getString("content");

String email = rs.getString("email");

**int** hit = rs.getInt("hit");

String pw = rs.getString("pw");

**int** ref = rs.getInt("ref");

**int** re\_step = rs.getInt("re\_step");

**int** re\_level= rs.getInt("re\_level");

String ip = rs.getString("ip");

Timestamp rDate= rs.getTimestamp("rDate");

dto = **new** BoardDto(num, writer, subject, content, email, hit, pw,

ref, re\_step, re\_level, ip, rDate);

}

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(stmt!=**null**) stmt.close();

**if**(conn!=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

**return** dto;

}

// 글번호로 글 dto가져오기 getBoardOneLine(String num)

**public** BoardDto getBoardOneLine(String num) {

BoardDto dto = **null**;

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM BOARD WHERE NUM="+num;

**try** {

conn = getConnection();

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);

**while**(rs.next()) {

String writer = rs.getString("writer");

String subject = rs.getString("subject");

String content = rs.getString("content");

String email = rs.getString("email");

**int** hit = rs.getInt("hit");

String pw = rs.getString("pw");

**int** ref = rs.getInt("ref");

**int** re\_step = rs.getInt("re\_step");

**int** re\_level= rs.getInt("re\_level");

String ip = rs.getString("ip");

Timestamp rDate= rs.getTimestamp("rDate");

dto = **new** BoardDto(Integer.*parseInt*(num), writer, subject, content,

email, hit, pw, ref, re\_step, re\_level, ip, rDate);

}

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(stmt!=**null**) stmt.close();

**if**(conn!=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

**return** dto;

}

// hit수 하나 올리기 hitUp(int num)

**public** **void** hitUp(**int** num) {

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "UPDATE BOARD SET HIT = HIT+1 WHERE NUM=?";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(sql);

pstmt.setInt(1, num);

pstmt.executeUpdate();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

}

// hit수 하나 올리기 hitUp(String num)

**public** **void** hitUp(String num) {

hitUp(Integer.*parseInt*(num));

}

// 글 수정 updateBoard(dto)

**public** **int** updateBoard(BoardDto dto) {

**int** result = ***FAIL***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "UPDATE BOARD SET WRITER = ?, " +

" SUBJECT = ?, " +

" CONTENT = ?, " +

" EMAIL = ?, " +

" PW = ?, " +

" IP = ? " +

" WHERE NUM=?";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, dto.getWriter());

pstmt.setString(2, dto.getSubject());

pstmt.setString(3, dto.getContent());

pstmt.setString(4, dto.getEmail());

pstmt.setString(5, dto.getPw());

pstmt.setString(6, dto.getIp());

pstmt.setInt(7, dto.getNum());

result = pstmt.executeUpdate();

System.***out***.println(result==***SUCCESS***? "글수정성공":"글수정실패");

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

**return** result;

}

// 글삭제 deleteBoard(num, pw)

**public** **int** deleteBoard(**int** num, String pw) {

**int** result = ***FAIL***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "DELETE FROM BOARD WHERE NUM=? AND PW=?";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(sql);

pstmt.setInt(1, num);

pstmt.setString(2, pw);

result = pstmt.executeUpdate();

System.***out***.println(result==***SUCCESS***? num+"번 글삭제 성공":num+"번글삭제실패");

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

**return** result;

}

**public** **int** reply(BoardDto dto) {

**int** result = ***FAIL***;

replyAstep(dto.getRef(), dto.getRe\_step());

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "INSERT INTO BOARD"

+ " (NUM, WRITER, SUBJECT, CONTENT, EMAIL, PW, REF, RE\_STEP, RE\_LEVEL, IP)"

+ " VALUES ((SELECT NVL(MAX(NUM),0)+1 FROM BOARD), ?,?,?,?," +

" ?, ?, ?, ?,?)";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, dto.getWriter());

pstmt.setString(2, dto.getSubject());

pstmt.setString(3, dto.getContent());

pstmt.setString(4, dto.getEmail());

pstmt.setString(5, dto.getPw());

pstmt.setInt(6, dto.getRef());

pstmt.setInt(7, dto.getRe\_step()+1);

pstmt.setInt(8, dto.getRe\_level()+1);

pstmt.setString(9, dto.getIp());

result = pstmt.executeUpdate();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

**return** result;

}

**private** **void** replyAstep(**int** ref, **int** re\_step) {

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "UPDATE BOARD SET RE\_STEP = RE\_STEP+1 " +

" WHERE REF=? AND RE\_STEP>?";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(sql);

pstmt.setInt(1, ref);

pstmt.setInt(2, re\_step);

pstmt.executeUpdate();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}

}

}

}

**style.css**

@CHARSET *"UTF-8"*;

**body** {

background-color: *#fff0B5*;

font-size: *12px*;

}

**table** {border: *2px* *solid* *pink*; width:*95%*; margin: *10px* *auto*;}

**table** **tr** { background-color: *#FFE271*; }

**table** **tr***:hover* { background-color: *orange*;

cursor: *pointer*;

}

**td,** **th** {text-align: *center*; padding:*5px*; }

**caption** {font-size: *25px*; padding:*10px*;}

**h2,** *.currentPage*{text-align: *center*; color:*red*;}

**a** { text-decoration: *none*; color:*black*}

*.left*{text-align: *left*;}

*.paging* {text-align: *center*;}

**input***:not*(*.btn*)**,** **textarea** {

width: *90%*;

}

**list.jsp**

<%@page import=*"com.tj.dao.BoardDao,com.tj.dto.BoardDto,java.util.ArrayList"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"../css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<table>

<caption>게시판</caption>

<tr><td><a href=*"writeForm.jsp"*>글쓰기</a></td></tr>

</table>

<table>

<tr><th>글번호</th><th>작성자</th><th>제목</th><th>메일</th>

<th>IP</th><th>작성일</th><th>조회수</th>

</tr>

<%String pageNum = request.getParameter("pageNum");

**if**(pageNum==**null**)

pageNum = "1";

**int** currentPage = Integer.parseInt(pageNum);

**final** **int** PAGESIZE = 10, BLOCKSIZE = 10;

**int** startRow = (currentPage-1) \* PAGESIZE +1 ;

**int** endRow = startRow + PAGESIZE -1;

BoardDao bDao = BoardDao.getInstance();

**int** totCnt = bDao.getBoardTotalCnt();

**if**(totCnt==0){

out.print("<tr><td colspan='7'>등록된 글이 없습니다</td></tr>");

}**else**{

ArrayList<BoardDto> dtos = bDao.listBoard(startRow, endRow);

**for**(BoardDto dto : dtos){

// 글번호, 작성자

out.println("<tr><td>"+dto.getNum()+"</td><td>"+dto.getWriter()+"</td>");

// 제목

out.println("<td class='left'>");

if(dto.getRe\_level()>0){

int width = dto.getRe\_level()\*10; //level.gif의 width

out.print("<img src=../img/level.gif' height='10' width='"+width+"'>");

out.print("<img src='../img/re.gif'>");

}

**if**(dto.getHit()>10){

out.println("<img src='../img/hot.gif'>");

}

out.print("<a href='content.jsp?num="+dto.getNum()+"&pageNum="+pageNum+"'>"+dto.getSubject()+"</a></td>");

// 메일

out.print("<td>"+(dto.getEmail()!=**null**? dto.getEmail():"-")+"</td>");

//ip, 작성일

out.print("<td>"+dto.getIp()+"</td><td>"+dto.getrDate()+"</td>");

//조회수

out.print("<td>"+dto.getHit()+"</td></tr>");

}

}

%>

</table>

<div class=*"paging"*>

<%

**int** pageCnt = (**int**)Math.ceil((**double**)totCnt/PAGESIZE);

**int** startPage = ((currentPage-1)/BLOCKSIZE)\*BLOCKSIZE +1;

**int** endPage = startPage + BLOCKSIZE -1;

**if**(endPage > pageCnt)

endPage = pageCnt;

**if**(startPage>BLOCKSIZE){

out.println(" [ <a href='list.jsp?pageNum="+(startPage-1)+"'>이전</a> ] ");

}

**for**(**int** i=startPage ; i<=endPage ; i++){

**if**(i==currentPage){

out.println(" [ <span class='currentPage'>" +i +"</span> ] ");

}**else**{

out.println(" [ <a href='list.jsp?pageNum="+i+"'>"+ +i +" </a> ] ");

}

}

**if**(endPage<pageCnt){

out.println(" [ <a href='list.jsp?pageNum="+(endPage+1)+"'>다음</a> ] ");

}

%>

</div>

</body>

</html>

**content.jsp**

<%@page import=*"com.tj.dao.BoardDao, com.tj.dto.BoardDto"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"../css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<%

/\* String num = request.getParameter("num");

BoardDao bDao = BoardDao.getInstance();

bDao.hitUp(num);

BoardDto dto = bDao.getBoardOneLine(num);

\*/

**int** num = Integer.parseInt(request.getParameter("num"));

BoardDao bDao = BoardDao.getInstance();

bDao.hitUp(num);

BoardDto dto = bDao.getBoardOneLine(num);

/\* pageNum 추가 \*/

String pageNum = request.getParameter("pageNum");

%>

<table>

<caption>글 내용 상세보기</caption>

<tr><th>글번호</th><td><%=dto.getNum() %></td></tr>

<tr><th>글제목</th><td><%=dto.getSubject() %></td></tr>

<tr><th>메일</th><td><%=dto.getEmail()==**null** ? " - ":dto.getEmail() %></td></tr>

<tr><th>조회수</th><td><%=dto.getHit() %></td></tr>

<tr><th>본문</th><td><pre><%=dto.getContent() %></pre></td></tr>

<tr><th>ip</th><td><%=dto.getIp() %></td></tr>

<tr><th>글쓴시점</th><td><%=dto.getrDate() %></td></tr>

<tr><td colspan=*"2"*>

<button onclick=*"location.href='updateForm.jsp?num=*<%=num%>*&pageNum=*<%=pageNum%>*'"*>수정</button>

<button onclick=*"location.href='deleteForm.jsp?num=*<%=num%>*&pageNum=*<%=pageNum%>*'"*>삭제</button>

<button onclick=*"location.href='writeForm.jsp?num=*<%=num%>*&pageNum=*<%=pageNum%>*'"*>답변</button>

<button onclick=*"location.href='list.jsp?pageNum=*<%=pageNum%>*'"*>목록</button>

</table>

</body>

</html>

**writeForm.jsp**

<%@page import=*"com.tj.dto.BoardDto, com.tj.dao.BoardDao"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"../css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<%

**int** num=0, ref=0, re\_level=0, re\_step=0; //답변글이 아닐 경우를 생각해 초기화

String pageNum = request.getParameter("pageNum");

**if**(pageNum==**null**) pageNum = "1";

BoardDto dto = **new** BoardDto();

**if**(request.getParameter("num")!=**null**){ // 답변글을 쓰려고 온 경우

num = Integer.parseInt(request.getParameter("num"));

BoardDao bDao = BoardDao.getInstance();

dto = bDao.getBoardOneLine(num); // 원글에 대한 정보

ref = dto.getRef();

re\_step = dto.getRe\_step();

re\_level = dto.getRe\_level();

}

%>

<form action=*"writePro.jsp"* method=*"post"*>

<input type=*"hidden"* name=*"pageNum"* value=*"*<%=pageNum %>*"*>

<input type=*"hidden"* name=*"num"* value=*"*<%=num %>*"*>

<input type=*"hidden"* name=*"ref"* value=*"*<%=ref %>*"*>

<input type=*"hidden"* name=*"re\_step"* value=*"*<%=re\_step %>*"*>

<input type=*"hidden"* name=*"re\_level"* value=*"*<%=re\_level %>*"*> <table>

<caption>게시판 글쓰기</caption>

<tr><th>작성자</th>

<td><input type=*"text"* name=*"writer"* required=*"required"* autofocus=*"autofocus"*></td>

</tr>

<tr><th>제목</th>

<td><input type=*"text"* name=*"subject"* required=*"required"* value=*"*<%

**if**(num!=0)

out.print("[답]"+dto.getSubject().substring(0, 1));

%>*"*></td>

</tr>

<tr><th>메일</th>

<td><input type=*"email"* name=*"email"*></td>

</tr>

<tr><th>본문</th>

<td><textarea rows=*"5"* cols=*"5"* name=*"content"*></textarea></td>

</tr>

<tr><th>비밀번호</th>

<td><input type=*"password"* name=*"pw"* required=*"required"*></td>

</tr>

<tr><td colspan=*"2"*>

<input type=*"submit"* value=*"글쓰기"* class=*"btn"*>

<input type=*"reset"* value=*"다시쓰기"* class=*"btn"*>

<input type=*"button"* value=*"목록"* class=*"btn"*

onclick=*"location.href='list.jsp?pageNum=*<%=pageNum%>*'"*>

</td>

</tr>

</table>

</form>

</body>

</html>

**writePro.jsp**

<%@page import=*"com.tj.dao.BoardDao"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<%request.setCharacterEncoding("utf-8");%>

<jsp:useBean id=*"dto"* class=*"com.tj.dto.BoardDto"*/>

<jsp:setProperty name=*"dto"* property=*"\*"*/>

<%String pageNum = request.getParameter("pageNum");

dto.setIp(request.getRemoteAddr());

BoardDao bDao = BoardDao.getInstance();

**int** result;

**if**(dto.getRef()==0){ // 원글일 경우

result = bDao.insertBoard(dto);

}**else**{ // 답변글 일 경우

System.out.println(dto.toString()+"의 답변 쓸 예정");

result = bDao.reply(dto);

}

**if**(result == BoardDao.SUCCESS){

%>

<script>

alert('글쓰기 완료');

location.href='list.jsp?pageNum=<%=pageNum%>';

</script>

<%}**else**{%>

<script>

alert('글쓰기 실패');

history.back();

</script>

<%}%>

</body>

</html>

**updateForm.jsp**

<%@page import=*"com.tj.dto.BoardDto, com.tj.dao.BoardDao"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"../css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<%

**int** num = Integer.parseInt(request.getParameter("num"));

BoardDao bDao = BoardDao.getInstance();

BoardDto dto = bDao.getBoardOneLine(num);

/\* pageNum 추가 \*/

String pageNum = request.getParameter("pageNum");

%>

<form action=*"updatePro.jsp"* method=*"post"*>

<input type=*"hidden"* name=*"pageNum"* value=*"*<%=pageNum%>*"*>

<input type=*"hidden"* name=*"num"* value=*"*<%=num%>*"*>

<table>

<caption><%=num %>번 글수정</caption>

<tr><th>작성자</th>

<td><input type=*"text"* name=*"writer"* required=*"required"* autofocus=*"autofocus"*

value=*"*<%=dto.getWriter()%>*"*></td>

</tr>

<tr><th>제목</th>

<td><input type=*"text"* name=*"subject"* required=*"required"*

value=*"*<%=dto.getSubject()%>*"*></td>

</tr>

<tr><th>메일</th>

<td><input type=*"email"* name=*"email"*

value=*"*<%=dto.getEmail()!=**null**? dto.getEmail():""%>*"*></td>

</tr>

<tr><th>본문</th>

<td><textarea rows=*"5"* cols=*"5"* name=*"content"*><%=dto.getContent() %></textarea></td>

</tr>

<tr><th>비밀번호</th>

<td><input type=*"password"* name=*"pw"* required=*"required"*

value=*"*<%=dto.getPw()%>*"*></td>

</tr>

<tr><td colspan=*"2"*>

<input type=*"submit"* value=*"수정"* class=*"btn"*>

<input type=*"reset"* value=*"수정취소"* class=*"btn"* onclick="history.back()">

<input type=*"button"* value=*"목록"* class=*"btn"*

onclick=*"location.href='list.jsp?pageNum=*<%=pageNum%>*'"*>

</td>

</tr>

</table>

</form>

</body>

</html>

**updatePro.jsp**

<%@page import=*"com.tj.dao.BoardDao"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<%request.setCharacterEncoding("utf-8"); %>

<jsp:useBean id=*"dto"* class=*"com.tj.dto.BoardDto"*/>

<jsp:setProperty name=*"dto"* property=*"\*"* />

<%/\* pageNum 추가 \*/

String pageNum = request.getParameter("pageNum");

dto.setIp(request.getRemoteAddr());

BoardDao bDao = BoardDao.getInstance();

**int** result = bDao.updateBoard(dto);

**if**(result==BoardDao.SUCCESS){

%>

<script>

alert('글 수정 성공');

location.href = 'list.jsp?pageNum=<%=pageNum%>';

//location.href = 'content.jsp?num=<%=dto.getNum()%>&pageNum=<%=pageNum%>';

</script>

<%}**else**{ %>

<script>

alert('글 수정 실패');

history.go(-1);

</script>

<%}%>

</body>

</html>

**deleteForm.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"../css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<%String num = request.getParameter("num");

/\* pageNum 추가 \*/

String pageNum = request.getParameter("pageNum");

%>

<table>

<caption><%=num %>번글 삭제</caption>

<tr><td>

<fieldset>

<legend>삭제하기 위한 암호 입력</legend>

<form action=*"deletePro.jsp"* method=*"post"*>

<input type=*"hidden"* name=*"pageNum"* value=*"*<%=pageNum%>*"*>

<input type=*"hidden"* name=*"num"* value=*"*<%=num %>*"*>

<p>암호 <input type=*"password"* name=*"pw"* required=*"required"* class=*"btn"*></p>

<p><input type=*"submit"* value=*"삭제"* class=*"btn"*>

<input type=*"reset"* value=*"삭제취소"* onclick="history.go(-1)" class=*"btn"*></p>

</form>

</fieldset>

</td></tr>

</table>

</body>

</html>

**deletePro.jsp**

<%@page import=*"com.tj.dao.BoardDao"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<%/\* pageNum 추가 \*/

String pageNum = request.getParameter("pageNum");

**int** num = Integer.parseInt(request.getParameter("num"));

String pw = request.getParameter("pw");

BoardDao bDao = BoardDao.getInstance();

**int** result = bDao.deleteBoard(num, pw);

**if**(result==BoardDao.SUCCESS){

%>

<script>

alert('<%=num%>번 글 삭제성공');

location.href = 'list.jsp?pageNum=<%=pageNum%>';

</script>

<%}**else**{%>

<script>

alert('<%=num%>번 글 삭제실패. 비밀번호 확인하세요');

history.go(-1);

</script>

<%}%>

</body>

</html>