

정기수행평가 3회

JSP / DB

최준혁

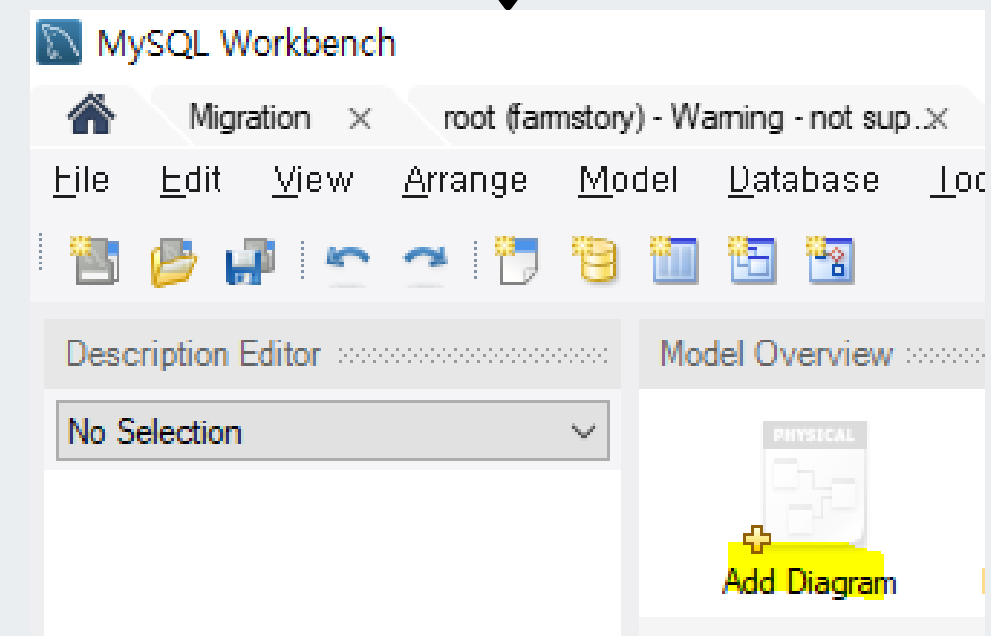
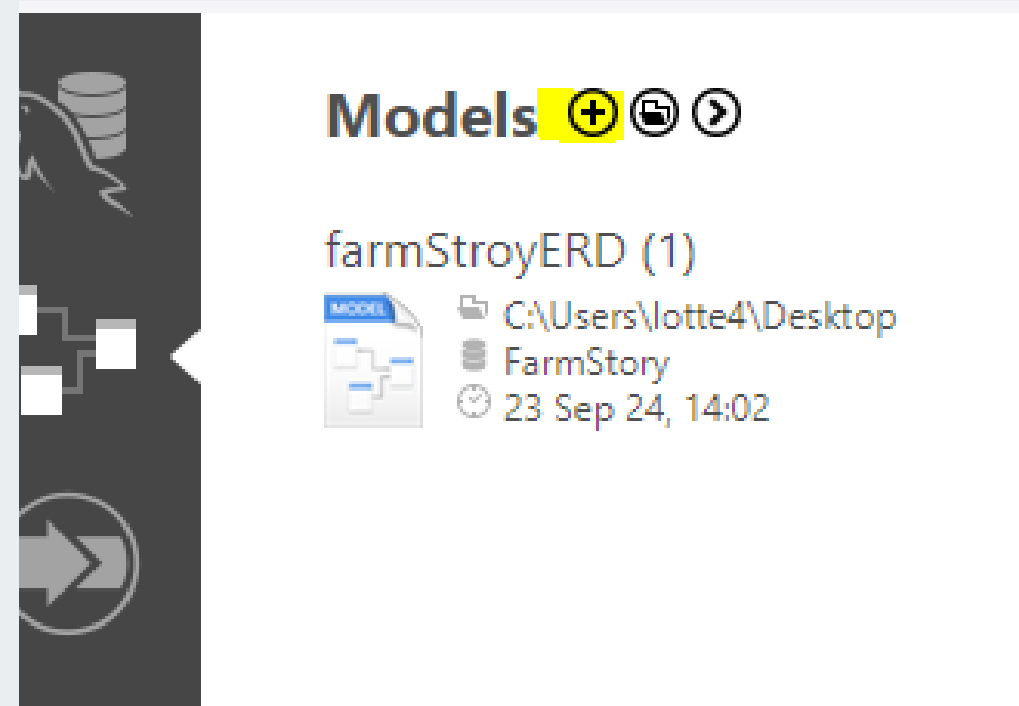
2024/09/23

loveu991111@gmail.com

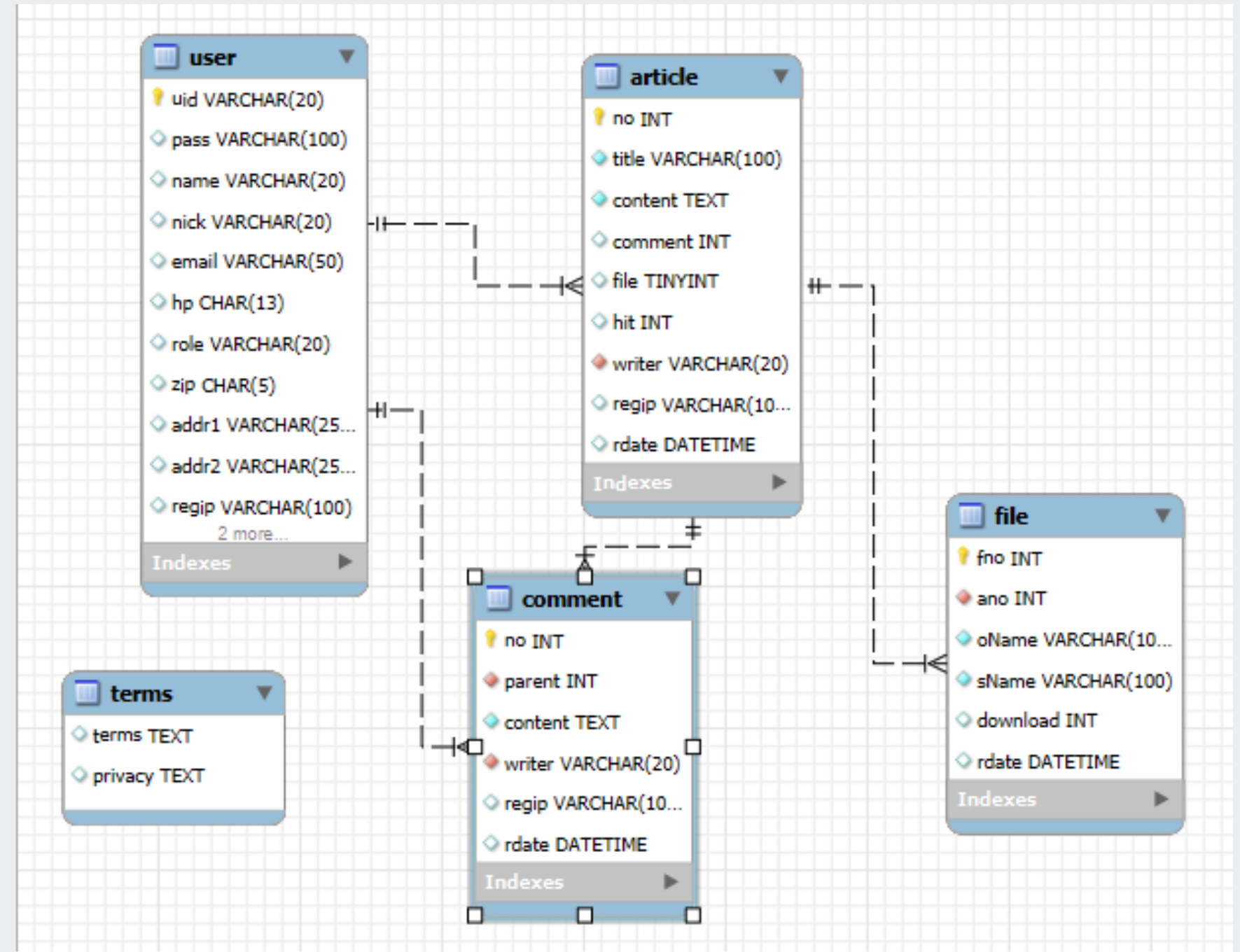
01	[DB] ERD 작성 및 테이블 생성	[DB] 작업1, 작업2
02	[DB] DB 샘플데이터 저장, SQL실행	[DB] 작업3, 작업4
03	[JSP] 로그인/회원가입 기능 구현	[JSP] 작업1
04	[JSP] 글쓰기 기능 구현	[JSP] 작업2
05	[JSP] 글목록 기능 구현	[JSP] 작업3
06	[JSP] 글보기 기능 구현	[JSP] 작업4

[DB] ERD 작성 및 테이블 생성

ERD 작성



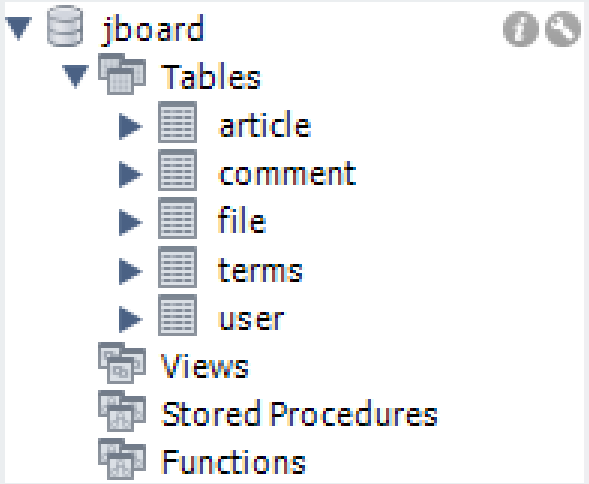
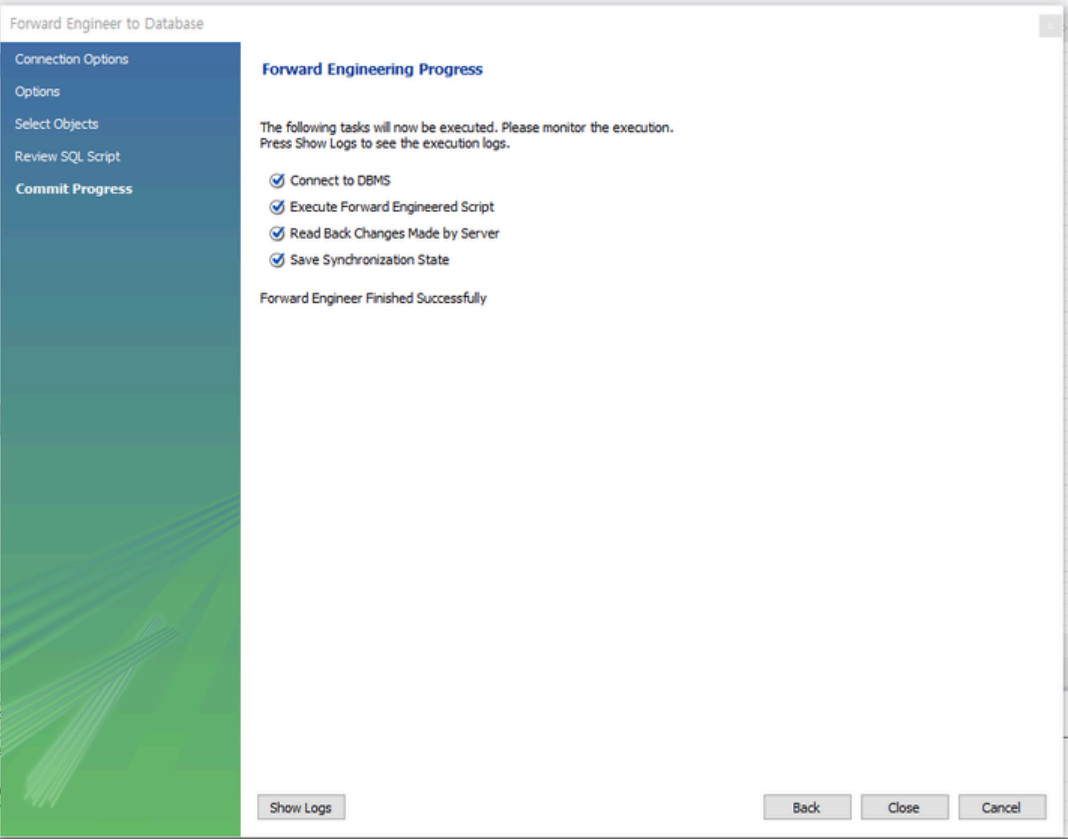
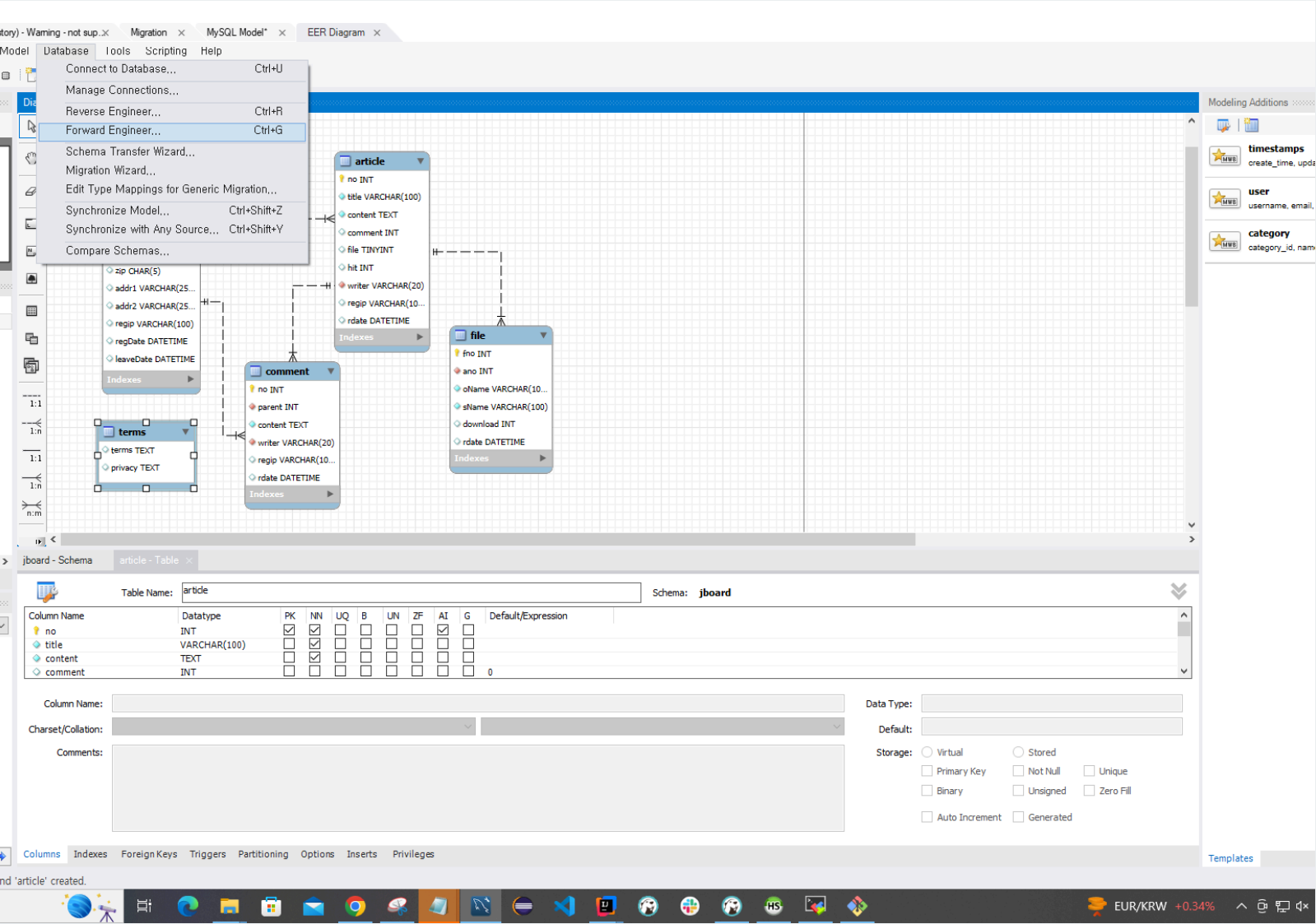
완성 ERD



[DB] ERD 작성 및 테이블 생성

DATABASE -> FORWARD ENGINEER

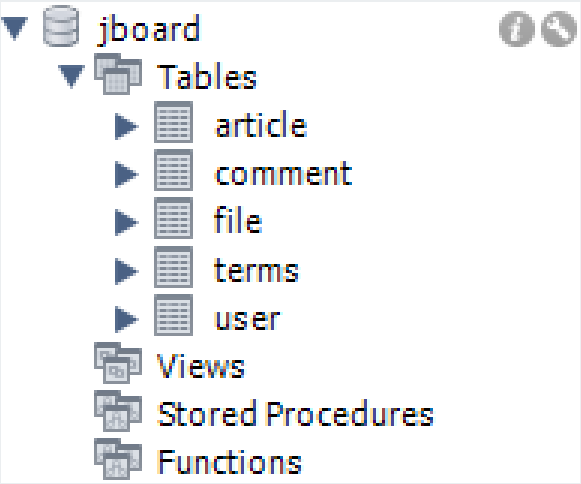
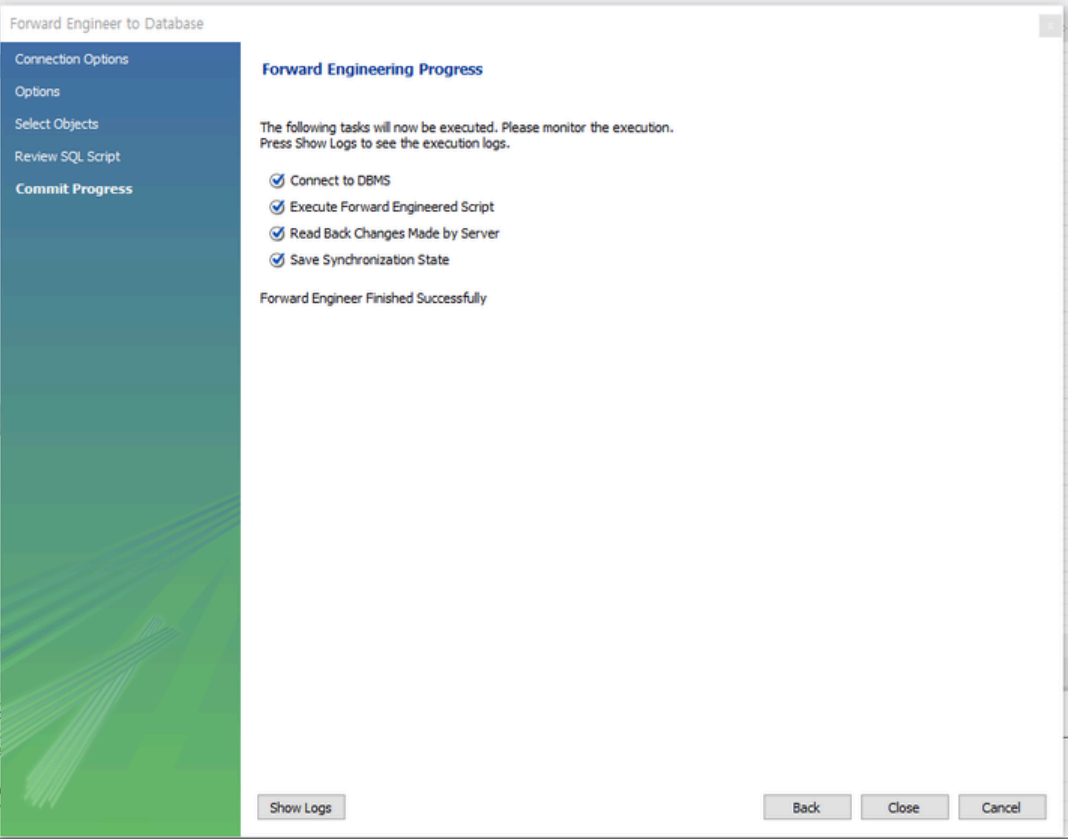
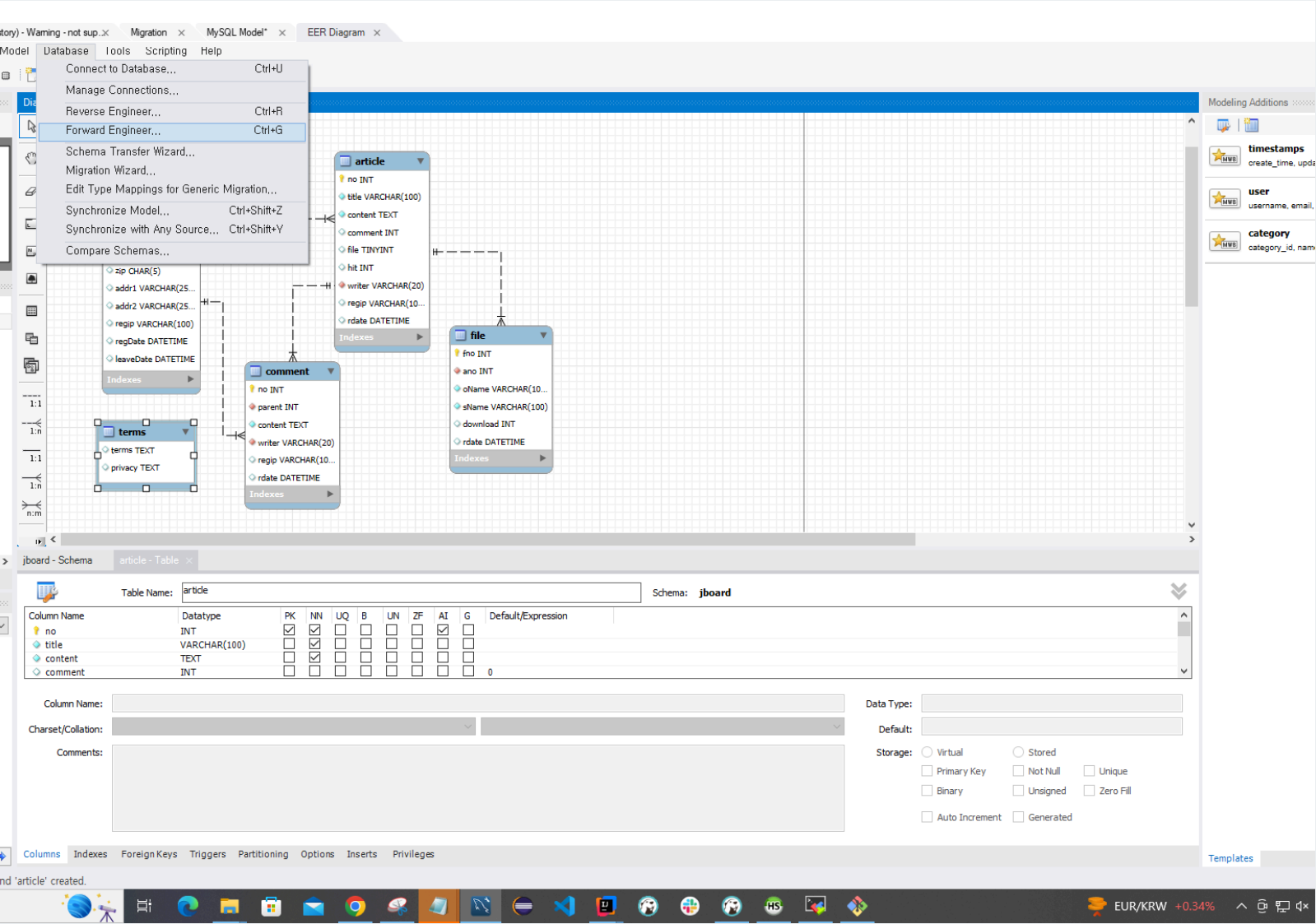
테이블 생성 완료 후 데이터베이스 생성 확인



[DB] ERD 작성 및 테이블 생성

DATABASE -> FORWARD ENGINEER

테이블 생성 완료 후 데이터베이스 생성 확인



[DB] DB 샘플데이터 저장, SQL실행

해당 쿼리를 이용하여 샘플 데이터 저장

```
INSERT INTO USER VALUES ('A102','A102','김춘추','춘추','ch@naver.com','010-1234-1112','user','12346','서울특별시','205호','0.0.0.1',now(),now());
INSERT INTO USER VALUES ('A103','A103','김수신','수신','su@naver.com','010-1234-1113','user','12347','인천광역시','205호','0.0.0.1',now(),now());
INSERT INTO USER VALUES ('A104','A104','장보고','보고','bo@naver.com','010-1234-1114','user','12348','광주광역시','205호','0.0.0.1',now(),now());
INSERT INTO USER VALUES ('A105','A105','이산','산','san@naver.com','010-1234-1115','user','12349','대구광역시','205호','0.0.0.1',now(),now());
```

```
INSERT INTO TERMS VALUES ('이용약관입니다', '개인정보처리방침입니다');
```

```
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST1', 'TEST1', 0, 'A101', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST2', 'TEST2', 0, 'A102', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST3', 'TEST3', 0, 'A103', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST4', 'TEST4', 0, 'A104', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST5', 'TEST5', 0, 'A105', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST6', 'TEST6', 0, 'A101', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST7', 'TEST7', 0, 'A102', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST8', 'TEST8', 0, 'A103', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST9', 'TEST9', 0, 'A104', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST10', 'TEST10', 0, 'A105', '0.0.0.1',now());
INSERT INTO ARTICLE (CATE, TITLE, CONTENT, HIT, WRITER, REGIP, RDATE) VALUES ('FREE', 'TEST11', 'TEST11', 0, 'A101', '0.0.0.1',now());
```

```
INSERT INTO COMMENT (PARENT, CONTENT, WRITER, REGIP, RDATE) VALUES (1, '댓글1', 'A101', '0.0.0.1',now());
INSERT INTO COMMENT (PARENT, CONTENT, WRITER, REGIP, RDATE) VALUES (1, '댓글2', 'A102', '0.0.0.1',now());
INSERT INTO COMMENT (PARENT, CONTENT, WRITER, REGIP, RDATE) VALUES (2, '댓글1', 'A103', '0.0.0.1',now());
INSERT INTO COMMENT (PARENT, CONTENT, WRITER, REGIP, RDATE) VALUES (2, '댓글2', 'A104', '0.0.0.1',now());
INSERT INTO COMMENT (PARENT, CONTENT, WRITER, REGIP, RDATE) VALUES (3, '댓글1', 'A105', '0.0.0.1',now());
INSERT INTO COMMENT (PARENT, CONTENT, WRITER, REGIP, RDATE) VALUES (3, '댓글2', 'A103', '0.0.0.1',now());
```

```
INSERT INTO FILE (ANO, ONAME, SNAME, RDATE) VALUES (1, '파일 1', 'ANSEKJSEKSJEJS', NOW());
INSERT INTO FILE (ANO, ONAME, SNAME, RDATE) VALUES (1, '파일 2', 'AAFSDASEKJSEKSJEJS', NOW());
INSERT INTO FILE (ANO, ONAME, SNAME, RDATE) VALUES (2, '파일 3', 'ANSERTYQWERYSEKSJEJS', NOW());
INSERT INTO FILE (ANO, ONAME, SNAME, RDATE) VALUES (2, '파일 4', 'ANSEKJSEKWETQTJS', NOW());
INSERT INTO FILE (ANO, ONAME, SNAME, RDATE) VALUES (3, '파일 5', 'ANSEKJTWETWEJS', NOW());
```

저장 확인

USER TABLE

uid	pass	name	nick	email	hp	role	zip	addr1	addr2	regip	regd
A101	A101	김유신	유신	yu@naver.com	010-1234-1111	user	12345	부산광역시	205호	0.0.0.1	2024
A102	A102	김춘추	춘추	ch@naver.com	010-1234-1112	user	12346	서울특별시	205호	0.0.0.1	2024
A103	A103	김수신	수신	su@naver.com	010-1234-1113	user	12347	인천광역시	205호	0.0.0.1	2024
A104	A104	장보고	보고	bo@naver.com	010-1234-1114	user	12348	광주광역시	205호	0.0.0.1	2024
A105	A105	이산	산	san@naver.com	010-1234-1115	user	12349	대구광역시	205호	0.0.0.1	2024

ARTICLE TABLE

no	cate	title	content	comment	file	hit	writer	regip	rdate
1	FREE	TEST1	TEST1	0	0	0	A101	0.0.0.1	2024-09-23 14:41:03
2	FREE	TEST2	TEST2	0	0	0	A102	0.0.0.1	2024-09-23 14:43:04
3	FREE	TEST3	TEST3	0	0	0	A103	0.0.0.1	2024-09-23 14:43:05
4	FREE	TEST4	TEST4	0	0	0	A104	0.0.0.1	2024-09-23 14:43:06
5	FREE	TEST5	TEST5	0	0	0	A105	0.0.0.1	2024-09-23 14:43:06
6	FREE	TEST6	TEST6	0	0	0	A101	0.0.0.1	2024-09-23 14:43:07
7	FREE	TEST7	TEST7	0	0	0	A102	0.0.0.1	2024-09-23 14:43:08
8	FREE	TEST8	TEST8	0	0	0	A103	0.0.0.1	2024-09-23 14:43:08
9	FREE	TEST9	TEST9	0	0	0	A104	0.0.0.1	2024-09-23 14:43:09
10	FREE	TEST10	TEST10	0	0	0	A105	0.0.0.1	2024-09-23 14:43:10
11	FREE	TEST11	TEST11	0	0	0	A101	0.0.0.1	2024-09-23 14:43:10
12	FREE	TEST12	TEST12	0	0	0	A102	0.0.0.1	2024-09-23 14:43:11
13	FREE	TEST13	TEST13	0	0	0	A103	0.0.0.1	2024-09-23 14:43:12

FILE TABLE

fno	ano	oName	sName	download	rdate
1	1	파일 1	ANSEKJSEKSJEJS	0	2024-09-23 14:47:26
2	1	파일 2	AAFSDASEKJSEKSJEJS	0	2024-09-23 14:48:05
3	2	파일 3	ANSERTYQWERYSEKSJEJS	0	2024-09-23 14:48:06
4	2	파일 4	ANSEKJSEKWETQTJS	0	2024-09-23 14:48:07
5	3	파일 5	ANSEKJTWETWEJS	0	2024-09-23 14:48:07
6	3	파일 6	ANSOQTEWQTQWTEKJSEKSJEJS	0	2024-09-23 14:48:08

COMMENT TABLE

no	parent	content	writer	regip	rdate
1	1	댓글1	A101	0.0.0.1	2024-09-23 14:45:32
2	1	댓글2	A102	0.0.0.1	2024-09-23 14:46:08
3	2	댓글1	A103	0.0.0.1	2024-09-23 14:46:09
4	2	댓글2	A104	0.0.0.1	2024-09-23 14:46:10
5	3	댓글1	A105	0.0.0.1	2024-09-23 14:46:10
6	3	댓글2	A103	0.0.0.1	2024-09-23 14:46:11

TERMS TABLE

terms	privacy
여러분을 환영합니다. 네이버 서비스 및 제품(...) 개인정보보호법에 따라 네이버에 회	

[DB] DB 샘플데이터 저장, SQL실행

1.전체 사용자를 조회하시오

```
SELECT * FROM USER;
```

uid	pass	name	nick	email	hp	role	zip	addr1	addr2	regip	regDate	leaveDate
A101	A101	김유신	유신	yu@naver.com	010-1234-1111	user	12345	부산광역시	205호	0.0.0.1	2024-09-23 14:36:15	2024-09-23 14:36:15
A102	A102	김춘추	춘추	ch@naver.com	010-1234-1112	user	12346	서울특별시	205호	0.0.0.1	2024-09-23 14:37:51	2024-09-23 14:37:51
A103	A103	김수신	수신	su@naver.com	010-1234-1113	user	12347	인천광역시	205호	0.0.0.1	2024-09-23 14:37:52	2024-09-23 14:37:52
A104	A104	장보고	보고	bo@naver.com	010-1234-1114	user	12348	광주광역시	205호	0.0.0.1	2024-09-23 14:37:53	2024-09-23 14:37:53
A105	A105	이산	산	san@naver.com	010-1234-1115	user	12349	대구광역시	205호	0.0.0.1	2024-09-23 14:37:54	2024-09-23 14:37:54

2. 글 목록 3페이지에 해당하는 전체 글을 조회하시오 (1페이지당 10개로 가정)

```
SELECT * FROM ARTICLE WHERE NO BETWEEN 21 AND 30;
```

no	cate	title	content	comment	file	hit	writer	regip	rdate
21	FREE	TEST21	TEST21	0	0	0	A101	0.0.0.1	2024-09-23 14:43:17
22	FREE	TEST22	TEST22	0	0	0	A102	0.0.0.1	2024-09-23 14:43:18
23	FREE	TEST23	TEST23	0	0	0	A103	0.0.0.1	2024-09-23 14:43:19
24	FREE	TEST24	TEST24	0	0	0	A104	0.0.0.1	2024-09-23 14:43:19
25	FREE	TEST25	TEST25	0	0	0	A105	0.0.0.1	2024-09-23 14:43:20
26	FREE	TEST26	TEST26	0	0	0	A101	0.0.0.1	2024-09-23 14:43:21
27	FREE	TEST27	TEST27	0	0	0	A102	0.0.0.1	2024-09-23 14:43:21
28	FREE	TEST28	TEST28	0	0	0	A103	0.0.0.1	2024-09-23 14:43:22
29	FREE	TEST29	TEST29	0	0	0	A104	0.0.0.1	2024-09-23 14:43:22
30	FREE	TEST30	TEST30	0	0	0	A105	0.0.0.1	2024-09-23 14:43:23

[DB] DB 샘플데이터 저장, SQL실행

3. 특정 글에 해당하는 모든 댓글을 조회하시오 (1번 글로 가정)

```
SELECT * FROM COMMENT AS C JOIN ARTICLE AS A ON C.PARENT = A.NO WHERE A.NO = 1;
```

no	parent	content	writer	regip	rdate	no	cate	title	content	comment	file	hit	writer	regip	rdate
1	1	댓글1	A101	0.0.0.1	2024-09-23 14:45:32	1	FREE	TEST1	TEST1	0	0	0	A101	0.0.0.1	2024-09-23 14:41:03
2	1	댓글2	A102	0.0.0.1	2024-09-23 14:46:08	1	FREE	TEST1	TEST1	0	0	0	A101	0.0.0.1	2024-09-23 14:41:03

4. 모든 전체글을 조회하시오. (단 사용자 이름과 닉네임이 조회 될 것)

```
SELECT A.*, U.NAME, U.NICK FROM ARTICLE AS A JOIN USER AS U ON A.WRITER = U.UID;
```

no	cate	title	content	comment	file	hit	writer	regip	rdate	NAME	NICK
1	FREE	TEST1	TEST1	0	0	0	A101	0.0.0.1	2024-09-23 14:41:03	김유신	유신
2	FREE	TEST2	TEST2	0	0	0	A102	0.0.0.1	2024-09-23 14:43:04	김춘추	춘추
3	FREE	TEST3	TEST3	0	0	0	A103	0.0.0.1	2024-09-23 14:43:05	김수신	수신
4	FREE	TEST4	TEST4	0	0	0	A104	0.0.0.1	2024-09-23 14:43:06	장보고	보고
5	FREE	TEST5	TEST5	0	0	0	A105	0.0.0.1	2024-09-23 14:43:06	이산	산
6	FREE	TEST6	TEST6	0	0	0	A101	0.0.0.1	2024-09-23 14:43:07	김유신	유신
7	FREE	TEST7	TEST7	0	0	0	A102	0.0.0.1	2024-09-23 14:43:08	김춘추	춘추
8	FREE	TEST8	TEST8	0	0	0	A103	0.0.0.1	2024-09-23 14:43:08	김수신	수신
9	FREE	TEST9	TEST9	0	0	0	A104	0.0.0.1	2024-09-23 14:43:09	장보고	보고
10	FREE	TEST10	TEST10	0	0	0	A105	0.0.0.1	2024-09-23 14:43:10	이산	산
11	FREE	TEST11	TEST11	0	0	0	A101	0.0.0.1	2024-09-23 14:43:10	김유신	유신
12	FREE	TEST12	TEST12	0	0	0	A102	0.0.0.1	2024-09-23 14:43:11	김춘추	춘추
13	FREE	TEST13	TEST13	0	0	0	A103	0.0.0.1	2024-09-23 14:43:12	김수신	수신

[DB] DB 샘플데이터 저장, SQL실행

5. 모든 전체 글을 조회하시오. (단 해당 글에 첨부된 원본 파일명과 댓글내용 그리고 최신글이 위로 정렬)

```
SELECT A.*, F.ONAME, C.CONTENT
FROM ARTICLE AS A
JOIN FILE AS F
ON A.NO = F.ANO
JOIN COMMENT AS C
ON A.NO = C.PARENT
ORDER BY A.RDATE DESC;
```

no	cate	title	content	comment	file	hit	writer	regip	rdate	ONAME	CONTENT
3	FREE	TEST3	TEST3	0	0	0	A103	0.0.0.1	2024-09-23 14:43:05	파일 5	댓글1
3	FREE	TEST3	TEST3	0	0	0	A103	0.0.0.1	2024-09-23 14:43:05	파일 5	댓글2
3	FREE	TEST3	TEST3	0	0	0	A103	0.0.0.1	2024-09-23 14:43:05	파일 6	댓글1
3	FREE	TEST3	TEST3	0	0	0	A103	0.0.0.1	2024-09-23 14:43:05	파일 6	댓글2
2	FREE	TEST2	TEST2	0	0	0	A102	0.0.0.1	2024-09-23 14:43:04	파일 3	댓글1
2	FREE	TEST2	TEST2	0	0	0	A102	0.0.0.1	2024-09-23 14:43:04	파일 3	댓글2
2	FREE	TEST2	TEST2	0	0	0	A102	0.0.0.1	2024-09-23 14:43:04	파일 4	댓글1
2	FREE	TEST2	TEST2	0	0	0	A102	0.0.0.1	2024-09-23 14:43:04	파일 4	댓글2
1	FREE	TEST1	TEST1	0	0	0	A101	0.0.0.1	2024-09-23 14:41:03	파일 1	댓글1
1	FREE	TEST1	TEST1	0	0	0	A101	0.0.0.1	2024-09-23 14:41:03	파일 1	댓글2
1	FREE	TEST1	TEST1	0	0	0	A101	0.0.0.1	2024-09-23 14:41:03	파일 2	댓글1
1	FREE	TEST1	TEST1	0	0	0	A101	0.0.0.1	2024-09-23 14:41:03	파일 2	댓글2

[JSP] 로그인/회원가입 기능 구현하시오

1.아이디/별명 중복 체크

핵심 코드

아이디/별명 중복체크

validation.js

```
const btnCheckUid = document.getElementById('btnCheckUid');
const btnCheckNick = document.getElementById('btnCheckNick');
```

```
// 1.아이디 유효성 검사
btnCheckUid.onclick = function(){

    const uid = registerForm.uid.value;
```

```
// 4.별명 유효성 검사
registerForm.nick.addEventListener('focusout', function(){

    const nick = registerForm.nick.value;
```

```
fetch('/jboard/user/checkUser.do?type=nick&value='+nick)
.then(response => response.json())
.then(data => {
    console.log(data);
    if(data.result > 0){
        resultNick.innerText = '이미 사용중인 별명입니다.';
        resultNick.style.color = 'red';
        isNickOk = false;
    }else{
        resultNick.innerText = '사용 가능한 별명입니다.';
        resultNick.style.color = 'green';
        isNickOk = true;
    }
})
.catch(err => {
    console.log(err);
});
```

```
// 중복체크
fetch('/jboard/user/checkUser.do?type=uid&value='+uid)
.then(resp => resp.json())
.then(data => {
    console.log(data);

    if(data.result > 0){
        resultId.innerText = '이미 사용중인 아이디 입니다.';
        resultId.style.color = 'red';
        isUidOk = false;
    }else{
        resultId.innerText = '사용 가능한 아이디 입니다.';
        resultId.style.color = 'green';
        isUidOk = true;
    }
})
.catch(err => {
    console.log(err);
});
```

PASSION

checkUserController.java

```
// 데이터 수신
String type = req.getParameter("type");
String value = req.getParameter("value");
logger.debug("type : " + type + ", value : " + value);

// 조회하기
int result = service.selectCountUser(type, value);
```

UserDAO.java

service 를 거쳐 dao로!

```
public int selectCountUser(String type, String value) {

    int result = 0;

    StringBuilder sql = new StringBuilder(SQL.SELECT_COUNT_USER);

    if(type.equals("uid")) {
        sql.append(SQL.WHERE_UID);
    }else if(type.equals("nick")) {
        sql.append(SQL.WHERE_NICK);
    }else if(type.equals("email")) {
        sql.append(SQL.WHERE_EMAIL);
    }else if(type.equals("hp")) {
        sql.append(SQL.WHERE_HP);
    }

    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(sql.toString());
        pstmt.setString(1, value);

        rs = pstmt.executeQuery();

        if(rs.next()) {
            result = rs.getInt(1);
        }
        closeAll();
    }catch (Exception e) {
        logger.error(e.getMessage());
    }
    return result;
}
```

```
// JSON 생성
JsonObject json = new JsonObject();
json.addProperty("result", result);

// JSON 출력
PrintWriter writer = resp.getWriter();
writer.print(json);
```

1. 중복 체크 버튼과 아이디, 별명 입력 필드값 문서객체 생성 후 해당 버튼에 클릭 메서드 생성
2. fetch 함수로 type, value값을 controller get 방식 전송
3. controller -> service -> dao 이동
4. type과 value값을 통해 해당 type 컬럼에 value이 존재하는지 select
5. 결과값 json생성 후 result property에 담아 전송
6. js 파일에서 해당 결과값이 존재한다면 '사용중인 아이디 입니다.' text 출력 후 해당 필드 red 변경
결과가 없다면 '사용 가능한 아이디 입니다.' 출력 후 해당 필드 green 변경

[JSP] 로그인/회원가입 기능 구현하시오

2. 이메일 인증

핵심 코드 validation.js

```
btnAuthEmail.onclick = function(){

    const code = registerForm.auth.value;

    fetch('/jboard/user/checkUser.do', {
        method: 'POST',
        body: JSON.stringify({"code":code})
    })
    .then(resp => resp.json())
    .then(data => {
        console.log(data);

        if(data.result > 0){
            resultEmail.innerText = '이메일이 인증되었습니다.';
            resultEmail.style.color = 'green';
            isEmailOk = true;
        }else{
            resultEmail.innerText = '유효한 인증코드가 없습니다.';
            resultEmail.style.color = 'red';
            isEmailOk = false;
        }
    })
    .catch(err => {
        console.log(err);
    });
};
```

1. type이 이메일이고 value값에 들어있는 이메일이 없다면
2. service에 sendEmailcode() 실행
3. sendEmailcode()
- ↳ 인증코드 랜덤 생성 -> 이메일 기본정보 입력 (본인 구글 이메일)
-> gmail SMTP 생성 -> 메일 발송
4. 해당 메일코드 세션 저장후 결과가 일치하면 인증처리

checkUserController.java

```
if(type.equals("email") && result == 0) {
    // 이메일 인증번호 발송하기
    String code = service.sendEmailCode(value);

    // 세션 저장
    HttpSession session = req.getSession();
    session.setAttribute("authCode", code);
}
```

UserService.java

```
public String sendEmailCode(String email) {
    // 인증코드 생성
    int code = ThreadLocalRandom.current().nextInt(1000000, 10000000);

    // 이메일 기본정보
    String title = "jboard 인증번호 입니다.";
    String content = "<h1>인증코드는 " + code + "입니다.</h1>";
    String sender = "chhak0503@gmail.com";
    String appPass = "lerr yafd wwnr codi"; // Google 앱 비밀번호

    // gmail SMTP 설정
    Properties props = new Properties();
    props.put("mail.smtp.host", "smtp.gmail.com");
    props.put("mail.smtp.port", "465");
    props.put("mail.smtp.auth", "true");
    props.put("mail.smtp.ssl.enable", "true");
    props.put("mail.smtp.ssl.trust", "smtp.gmail.com");

    // gmail session 생성
    Session gmailSession = Session.getInstance(props, new Authenticator(){
        @Override
        protected javax.mail.PasswordAuthentication getPasswordAuthentication(){
            return new PasswordAuthentication(sender, appPass);
        }
    });

    // 메일 발송
    Message message = new MimeMessage(gmailSession);

    try{
        message.setFrom(new InternetAddress(sender, "보내는사람", "UTF-8"));
        message.addRecipient(Message.RecipientType.TO, new InternetAddress(email));
        message.setSubject(title);
        message.setContent(content, "text/html;charset=utf-8");
        Transport.send(message);
    }catch(Exception e){
        e.printStackTrace();
    }

    return ""+code;
}
```

[JSP] 로그인/회원가입 기능 구현하시오

3. 결과 화면

사이트 이용정보 입력

아이디

qwer123

중복확인

사용 가능한 아이디 입니다.

비밀번호

.....

비밀번호 확인

.....

비밀번호가 일치합니다.

개인정보 입력

이름

최준혁

별명

공백없이 한글, 영문, 숫자만 입력가능
순수령

사용 가능한 별명입니다.

E-Mail

loveu9911111@gmail.com

인증번호 받기

이메일이 인증되었습니다.

JLF45v2f

확인

휴대폰

010-1111-1255

주소

13536

우편번호찾기

경기 성남시 분당구 판교역로 4

205

취소

회원가입

회원가입 후 DB 들어간것 확인

```
1 • SELECT * FROM jboard.user;
```

uid	pass	name	nick	email	hp	role	zip	addr1	addr2	regip	regDate	leaveDate
A101	A101	김유신	유신	yu@naver.com	010-1234-1111	user	12345	부산광역시	205호	0.0.0.1	2024-09-23 14:36:15	2024-09-23 14:36:15
A102	A102	김춘추	춘추	ch@naver.com	010-1234-1112	user	12346	서울특별시	205호	0.0.0.1	2024-09-23 14:37:51	2024-09-23 14:37:51
A103	A103	김수신	수신	su@naver.com	010-1234-1113	user	12347	인천광역시	205호	0.0.0.1	2024-09-23 14:37:52	2024-09-23 14:37:52
A104	A104	장보고	보고	bo@naver.com	010-1234-1114	user	12348	광주광역시	205호	0.0.0.1	2024-09-23 14:37:53	2024-09-23 14:37:53
A105	A105	이산	산	san@naver.com	010-1234-1115	user	12349	대구광역시	205호	0.0.0.1	2024-09-23 14:37:54	2024-09-23 14:37:54
qwer 123	\$2a\$10\$Y1ttd/3q4GybdI.LXzItVu278ecPxQDytp...	최준혁	순수령	loveu9911111@gmail.com	010-1111-1255	NULL	34672	남시 분당구 판교역로 5	205	NULL	NULL	NULL

로그인 후 글목록 이동

Board System

최준혁님 반갑습니다. [로그아웃]

글목록

제목 키워드, 글쓴이 검색

검색

번호	제목	글쓴이	날짜	조회
1	테스트 제목입니다. [3]	길동이	20-05-12	12

이전

1

2

3

다음

글쓰기

[JSP] 글쓰기 기능 구현하시오

1. 글쓰기 및 파일 첨부 구현

핵심코드

writeController.java

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    // 데이터 수신
    String title = req.getParameter("title");
    String content = req.getParameter("content");
    String writer = req.getParameter("writer");
    String regip = req.getRemoteAddr();

    // 파일 업로드
    List<FileDto> files = fileService.fileUpload(req);

    // 글 등록을 위한 DTO 생성
    ArticleDto dto = new ArticleDto();
    dto.setTitle(title);
    dto.setContent(content);
    dto.setFile(files.size());
    dto.setWriter(writer);
    dto.setRegip(regip);

    // 글 등록
    int no = articleService.insertArticle(dto);

    // 파일 등록
    for(FileDto fileDto : files) {
        fileDto.setAno(no);
        fileService.insertFile(fileDto);
    }

    resp.sendRedirect("/jboard/article/list.do");
}
```

데이터 수신 -> 파일 업로드 -> ArticleDTO 생성 ->
글등록 -> 파일등록 -> 리다이렉트

fileDAO.java

```
public void insertFile(FileDto dto) {
    try {
        conn = getConnection();
        psmt = conn.prepareStatement(SQL.INSERT_FILE);
        psmt.setInt(1, dto.getAno());
        psmt.setString(2, dto.getoName());
        psmt.setString(3, dto.getsName());
        psmt.executeUpdate();
        closeAll();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
}
```

파일 등록 메서드

fileService.java

```
public List<FileDto> fileUpload(HttpServletRequest req) {

    List<FileDto> files = new ArrayList<>();

    ServletContext ctx = req.getServletContext();
    String uploadPath = ctx.getRealPath("/uploads");
    logger.debug("uploadPath : " + uploadPath);

    try {
        // 첨부 파일 정보객체 가져오기
        Collection<Part> parts = req.getParts();

        for(Part part : parts) {
            // 파일명 추출
            String ofileName = part.getSubmittedFileName();

            // 파일을 첨부했으면
            if(ofileName != null && !ofileName.isEmpty()) {
                logger.debug("ofileName : " + ofileName);

                // 고유 파일명 생성
                int idx = ofileName.lastIndexOf(".");
                String ext = ofileName.substring(idx);

                String sfileName = UUID.randomUUID().toString() + ext;
                logger.debug("sfileName : " + sfileName);

                // 파일 저장
                part.write(uploadPath + File.separator + sfileName);

                // fileDTO 생성 후 리스트 저장
                FileDto fileDto = new FileDto();
                fileDto.setoName(ofileName);
                fileDto.setsName(sfileName);
                files.add(fileDto);
            }
        }

    } catch (Exception e) {
        e.printStackTrace();
    }

    return files;
}
```

파일 업로드 메서드

ArticleDAO.java

```
public int insertArticle(ArticleDto dto) {
    int no = 0;

    try {
        conn = getConnection();
        conn.setAutoCommit(false);

        stmt = conn.createStatement();
        psmt = conn.prepareStatement(SQL.INSERT_ARTICLE);
        psmt.setString(1, dto.getTitle());
        psmt.setString(2, dto.getContent());
        psmt.setInt(3, dto.getFile());
        psmt.setString(4, dto.getWriter());
        psmt.setString(5, dto.getRegip());
        psmt.executeUpdate();

        rs = stmt.executeQuery(SQL.SELECT_MAX_NO);
        if(rs.next()) {
            no = rs.getInt(1);
        }
        conn.commit();

    } catch (Exception e) {
        logger.error(e.getMessage());
        try {
            conn.rollback();
        } catch (SQLException e1) {
            logger.error(e1.getMessage());
        }
    } finally {
        closeAll();
    }

    return no;
}
```

글 등록 메서드

1. 컨트롤러 에서 데이터 수신
2. fileService에 업로드 메서드 실행
3. 글 등록을 위해 DTO 생성
4. articleService -> articleDAO 글등록 메서드 실행
5. fileService -> fileDAO 파일등록 메서드 실행
6. 리다이렉트

[JSP] 글쓰기 기능 구현하시오

2. 결과 화면

글 등록 화면

글쓰기

qwer123

제목	2024/09/03 JSP 테스트
내용	TEST1
파일	<div>최대 2개 파일 첨부 가능, 각 파일당 최대 10MB까지 가능</div> <div><div>파일 선택</div>logo.png</div> <div><div>파일 선택</div>선택된 파일 없음</div>

취소

작성완료

Article table

1

SELECT * FROM jboard.article;

no	cate	title	content	comment	file	hit	writer	regip	rdate
16	FREE	TEST16	TEST16	0	0	0	A101	0.0.0.1	2024-09-23 14:43:14
17	FREE	TEST17	TEST17	0	0	0	A102	0.0.0.1	2024-09-23 14:43:14
18	FREE	TEST18	TEST18	0	0	0	A103	0.0.0.1	2024-09-23 14:43:15
19	FREE	TEST19	TEST19	0	0	0	A104	0.0.0.1	2024-09-23 14:43:16
20	FREE	TEST20	TEST20	0	0	0	A105	0.0.0.1	2024-09-23 14:43:17
21	FREE	TEST21	TEST21	0	0	0	A101	0.0.0.1	2024-09-23 14:43:17
22	FREE	TEST22	TEST22	0	0	0	A102	0.0.0.1	2024-09-23 14:43:18
23	FREE	TEST23	TEST23	0	0	0	A103	0.0.0.1	2024-09-23 14:43:19
24	FREE	TEST24	TEST24	0	0	0	A104	0.0.0.1	2024-09-23 14:43:19
25	FREE	TEST25	TEST25	0	0	0	A105	0.0.0.1	2024-09-23 14:43:20
26	FREE	TEST26	TEST26	0	0	0	A101	0.0.0.1	2024-09-23 14:43:21
27	FREE	TEST27	TEST27	0	0	0	A102	0.0.0.1	2024-09-23 14:43:21
28	FREE	TEST28	TEST28	0	0	0	A103	0.0.0.1	2024-09-23 14:43:22
29	FREE	TEST29	TEST29	0	0	0	A104	0.0.0.1	2024-09-23 14:43:22
30	FREE	TEST30	TEST30	0	0	0	A105	0.0.0.1	2024-09-23 14:43:23
31	FREE	2024/09/03 JSP 테스트	TEST	0	1	0	qwer...	0.0.0.1	2024-09-23 16:30:01

fileTable

1

SELECT * FROM jboard.files;

fno	ano	oName	sName	download	rdate
1	1	파일 1	ANSEKJSEKSJEJS	0	2024-09-23 14:47:26
2	1	파일 2	AAFSDASEKJSEKSJEJS	0	2024-09-23 14:48:05
3	2	파일 3	ANSERTYQWERYSEKSJEJS	0	2024-09-23 14:48:06
4	2	파일 4	ANSEKJSEKWETQTJS	0	2024-09-23 14:48:07
5	3	파일 5	ANSEKJTWETWEJS	0	2024-09-23 14:48:07
6	3	파일 6	ANSOQTEWQTQWTEKJSEKSJEJS	0	2024-09-23 14:48:08
7	31	캡처.PNG	b5d78ce4-dea0-4974-81ff-be1...	0	2024-09-23 16:32:02
NULL	NULL	NULL	NULL	NULL	NULL

[JSP] 글목록 기능 구현하시오

1. 1페이지당 10개의 글과 글 목록 하단 페이지번호 구현

핵심코드

listController.java

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

    String pg = req.getParameter("pg");

    // 현재 페이지 번호 구하기
    int currentPage = service.getCurrentPage(pg);

    // 전체 게시물 갯수 구하기
    int total = service.selectCountTotal();

    // 마지막 페이지 번호 구하기
    int lastPageNum = service.getLastPageNum(total);

    // 현재 페이지 그룹 구하기
    PageGroupDto pageGroup = service.getCurrentPageGroup(currentPage, lastPageNum);

    // limit용 시작 번호 구하기
    int start = service.getStartNum(currentPage);

    // 페이지 시작 번호 구하기(목록에서 순서번호로 활용)
    int pageStartNum = service.getPageStartNum(total, currentPage);

    // 데이터 조회
    List<ArticleDto> articles = service.selectArticles(start);

    // 공유 참조
    req.setAttribute("articles", articles);
    req.setAttribute("lastPageNum", lastPageNum);
    req.setAttribute("pageGroup", pageGroup);
    req.setAttribute("pageStartNum", pageStartNum);
    req.setAttribute("currentPage", currentPage);

    // 포워드
    RequestDispatcher dispatcher = req.getRequestDispatcher("/WEB-INF/article/list.jsp");
    dispatcher.forward(req, resp);
}
```

- 전체 게시물 갯수를 구하여 해당 갯수로 페이지 번호 산출 후 공유참조

articleService.java

```
// 전체 게시물 갯수에서 마지막 페이지 번호 구하기
public int getLastPageNum(int total) {
    int lastPageNum = 0;

    if(total % 10 == 0) {
        lastPageNum = total / 10;
    }else {
        lastPageNum = total / 10 + 1;
    }
    return lastPageNum;
}

// 페이지 시작번호(limit용)
public int getStartNum(int currentPage) {
    return (currentPage - 1) * 10;
}

// 현재 페이지번호 구하기
public int getCurrentPage(String pg) {
    int currentPage = 1;

    if(pg != null) {
        currentPage = Integer.parseInt(pg);
    }

    return currentPage;
}

// 현재 페이지 그룹 구하기
public PageGroupDto getCurrentPageGroup(int currentPage, int lastPageNum) {

    int currentPageGroup = (int) Math.ceil(currentPage / 10.0);
    int pageGroupStart = (currentPageGroup - 1) * 10 + 1;
    int pageGroupEnd = currentPageGroup * 10;

    if(pageGroupEnd > lastPageNum){
        pageGroupEnd = lastPageNum;
    }

    return new PageGroupDto(pageGroupStart, pageGroupEnd);
}

// 페이지 시작번호
public int getPageStartNum(int total, int currentPage) {
    int start = (currentPage - 1) * 10;
    return total - start;
}
```

- 전체 게시물 갯수에서 마지막 페이지 번호 구하는 메서드
- 페이지 시작번호 산출 메서드 (limit용)
- 현재 페이지 번호 구하는 메서드
- 현재 페이지번호 그룹 구하는 메서드
- 페이지 시작번호 산출 메서드

[JSP] 글목록 기능 구현하시오

1. 1페이지당 10개의 글과 글 목록 하단 페이지번호 구현

핵심코드

ArticleDAO.java

```
public int selectCountTotal() {
    int total = 0;
    try {
        conn = getConnection();
        stmt = conn.createStatement();
        rs = stmt.executeQuery(SQL.SELECT_COUNT_TOTAL);
        if(rs.next()) {
            total = rs.getInt(1);
        }
    } catch (Exception e) {
        logger.error(e.getMessage());
    } finally {
        closeAll();
    }
    return total;
}
```

- 전체 게시물 구하는 메서드

```
public List<ArticleDto> selectArticles(int start) {
    List<ArticleDto> articles = new ArrayList<>();

    try {
        conn = getConnection();
        psmt = conn.prepareStatement(SQL.SELECT_ARTICLES);
        psmt.setInt(1, start);

        rs = psmt.executeQuery();

        while(rs.next()) {
            ArticleDto dto = new ArticleDto();
            dto.setNo(rs.getInt(1));
            dto.setCate(rs.getString(2));
            dto.setTitle(rs.getString(3));
            dto.setContent(rs.getString(4));
            dto.setComment(rs.getInt(5));
            dto.setFile(rs.getInt(6));
            dto.setHit(rs.getInt(7));
            dto.setWriter(rs.getString(8));
            dto.setRegip(rs.getString(9));
            dto.setRdateSubString(rs.getString(10));
            dto.setNick(rs.getString(11));
            articles.add(dto);
        }

    } catch (Exception e) {
        logger.error(e.getMessage());
    } finally {
        closeAll();
    }

    return articles;
}
```

- limit용 시작 번호에서 부터 게시물 select 메서드

[JSP] 글목록 기능 구현하시오

1. 1페이지당 10개의 글과 글 목록 하단 페이지번호 구현

글 목록

제목 키워드, 글쓴이 검색

검색

번호	제목	글쓴이	날짜	조회
1	TEXT2.[0]	김유신	24-09-03	12
2	TEXT3.[0]	김유신	24-09-03	12
3	TEXT4.[0]	김유신	24-09-03	12
4	TEXT5.[0]	김유신	24-09-03	12
5	TEXT1.[0]	김유신	24-09-03	12
6	TEXT6.[0]	김유신	24-09-03	12
7	TEXT7.[0]	김유신	24-09-03	12
8	TEXT8.[0]	김유신	24-09-03	12
9	TEXT9.[0]	김유신	24-09-03	12
10	TEXT10.[0]	김유신	24-09-03	12

이전

1

2

3

4

다음

글쓰기

글 목록

제목 키워드, 글쓴이 검색

검색

번호	제목	글쓴이	날짜	조회
31	2024/09/03 JSP 테스트.[3]	순수렁	24-09-23	12

이전

1

2

3

4

다음

글쓰기

전체 게시물 총 31개

1 • `ELECT * FROM jboard.article;`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	no	cate	title	content	comment	file	hit	writer	regip	rdate
	5	FREE	TEST5	TEST5	0	0	0	A105	0.0.0.1	2024-09-23 14:43:06
	6	FREE	TEST6	TEST6	0	0	0	A101	0.0.0.1	2024-09-23 14:43:07
	7	FREE	TEST7	TEST7	0	0	0	A102	0.0.0.1	2024-09-23 14:43:08
	8	FREE	TEST8	TEST8	0	0	0	A103	0.0.0.1	2024-09-23 14:43:08
	9	FREE	TEST9	TEST9	0	0	0	A104	0.0.0.1	2024-09-23 14:43:09
	10	FREE	TEST10	TEST10	0	0	0	A105	0.0.0.1	2024-09-23 14:43:10
	11	FREE	TEST11	TEST11	0	0	0	A101	0.0.0.1	2024-09-23 14:43:10
	12	FREE	TEST12	TEST12	0	0	0	A102	0.0.0.1	2024-09-23 14:43:11
	13	FREE	TEST13	TEST13	0	0	0	A103	0.0.0.1	2024-09-23 14:43:12
	14	FREE	TEST14	TEST14	0	0	0	A104	0.0.0.1	2024-09-23 14:43:12
	15	FREE	TEST15	TEST15	0	0	0	A105	0.0.0.1	2024-09-23 14:43:13
	16	FREE	TEST16	TEST16	0	0	0	A101	0.0.0.1	2024-09-23 14:43:14
	17	FREE	TEST17	TEST17	0	0	0	A102	0.0.0.1	2024-09-23 14:43:14
	18	FREE	TEST18	TEST18	0	0	0	A103	0.0.0.1	2024-09-23 14:43:15
	19	FREE	TEST19	TEST19	0	0	0	A104	0.0.0.1	2024-09-23 14:43:16
	20	FREE	TEST20	TEST20	0	0	0	A105	0.0.0.1	2024-09-23 14:43:17
	21	FREE	TEST21	TEST21	0	0	0	A101	0.0.0.1	2024-09-23 14:43:17
	22	FREE	TEST22	TEST22	0	0	0	A102	0.0.0.1	2024-09-23 14:43:18
	23	FREE	TEST23	TEST23	0	0	0	A103	0.0.0.1	2024-09-23 14:43:19
	24	FREE	TEST24	TEST24	0	0	0	A104	0.0.0.1	2024-09-23 14:43:19
	25	FREE	TEST25	TEST25	0	0	0	A105	0.0.0.1	2024-09-23 14:43:20
	26	FREE	TEST26	TEST26	0	0	0	A101	0.0.0.1	2024-09-23 14:43:21
	27	FREE	TEST27	TEST27	0	0	0	A102	0.0.0.1	2024-09-23 14:43:21
	28	FREE	TEST28	TEST28	0	0	0	A103	0.0.0.1	2024-09-23 14:43:22
	29	FREE	TEST29	TEST29	0	0	0	A104	0.0.0.1	2024-09-23 14:43:22
	30	FREE	TEST30	TEST30	0	0	0	A105	0.0.0.1	2024-09-23 14:43:23
	31	FREE	2024/09/03 JSP 테스트	TEST	0	1	0	qwer...	0.0.0.1	2024-09-23 16:30:01

4페이지

[JSP] 글보기 기능 구현하시오

1. 기본 글보기 기능, 첨부 파일 내용 출력 및 파일 다운로드 구현, 댓글 목록 및 쓰기 구현

핵심 코드

veiwController.java

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

    String no = req.getParameter("no");

    // 데이터 조회
    ArticleDto articleDto = service.selectArticle(no);

    // 댓글 조회
    List<CommentDto> comments = commentService.selectComments(no);

    // 공유 참조
    req.setAttribute("articleDto", articleDto);
    req.setAttribute("comments", comments);

    // 포워드(화면출력)
    RequestDispatcher dispatcher = req.getRequestDispatcher("/WEB-INF/article/view.jsp");
    dispatcher.forward(req, resp);
}
```

- view에서 글 번호 수신후 그 글번호를 통해 해당 글, 댓글을 조회하여 공유참조

articleDAO.java

```
public ArticleDto selectArticle(String no) {

    ArticleDto dto = null;
    List<FileDto> files = new ArrayList<FileDto>();

    try {
        conn = getConnection();
        psmt = conn.prepareStatement(SQL.SELECT_ARTICLE);
        psmt.setString(1, no);
        rs = psmt.executeQuery();

        while(rs.next()) {
            if(dto == null) {
                dto = new ArticleDto();
                dto.setNo(rs.getInt(1));
                dto.setCate(rs.getString(2));
                dto.setTitle(rs.getString(3));
                dto.setContent(rs.getString(4));
                dto.setComment(rs.getInt(5));
                dto.setFile(rs.getInt(6));
                dto.setHit(rs.getInt(7));
                dto.setWriter(rs.getString(8));
                dto.setRegip(rs.getString(9));
                dto.setRdate(rs.getString(10));
            }

            FileDto fileDto = new FileDto();
            fileDto.setFno(rs.getInt(11));
            fileDto.setAno(rs.getInt(12));
            fileDto.setoName(rs.getString(13));
            fileDto.setsName(rs.getString(14));
            fileDto.setDownload(rs.getInt(15));
            fileDto.setRdate(rs.getString(16));
            files.add(fileDto);
        }

        dto.setFiles(files);

    } catch (Exception e) {
        logger.error(e.getMessage());
    } finally {
        closeAll();
    }
    return dto;
}
```

- 특정 글번호의 글을 조회하는 메서드 (글의 해당되는 파일 포함)

commentDAO.java

```
public List<CommentDto> selectComments(String parent) {

    List<CommentDto> comments = new ArrayList<>();
    try {
        conn = getConnection();
        psmt = conn.prepareStatement(SQL.SELECT_COMMENTS);
        psmt.setString(1, parent);

        rs = psmt.executeQuery();

        while(rs.next()) {
            CommentDto dto = new CommentDto();
            dto.setNo(rs.getInt(1));
            dto.setParent(rs.getInt(2));
            dto.setContent(rs.getString(3));
            dto.setWriter(rs.getString(4));
            dto.setRegip(rs.getString(5));
            dto.setRdateSubstring(rs.getString(6));
            dto.setNick(rs.getString(7));
            comments.add(dto);
        }
    } catch (Exception e) {
        logger.error(e.getMessage());
    } finally {
        closeAll();
    }
    return comments;
}
```

- 특정 글번호에 포함되는 댓글 조회

[JSP] 글보기 기능 구현하시오

1. 기본 글보기 기능, 첨부 파일 내용 출력 및 파일 다운로드 구현, 댓글 목록 및 쓰기 구현

핵심 코드

writeController.java

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

    // 폼 데이터 수신
    String parent = req.getParameter("parent");
    String comment = req.getParameter("comment");
    String writer = req.getParameter("writer");
    String regip = req.getRemoteAddr();

    // DTO 생성
    CommentDto dto = new CommentDto();
    dto.setParent(parent);
    dto.setContent(comment);
    dto.setWriter(writer);
    dto.setRegip(regip);
    logger.debug(dto.toString());

    // 댓글 등록
    int pk = service.insertComment(dto);

    // 방금 등록한 댓글 조회
    CommentDto commentDto = service.selectComment(pk);

    // JSON 생성 및 출력
    Gson gson = new Gson();
    String json = gson.toJson(commentDto);

    PrintWriter printWriter = resp.getWriter();
    printWriter.print(json);
}
```

view.jsp (js)

```
// 댓글 등록
commentForm.onsubmit = function(e){
    e.preventDefault();

    // 사용자가 입력한 값 취함
    const parent = commentForm.parent.value;
    const writer = commentForm.writer.value;
    const comment = commentForm.comment.value;

    // 폼 데이터 생성 (JSON 데이터로 전송하게 되면 getParameter 수신 처리가 안되기 때문에 FormData로 전송)
    const formData = new FormData();
    formData.append("parent", parent);
    formData.append("writer", writer);
    formData.append("comment", comment);
    console.log(formData);

    fetch('/jboard/comment/write.do', {
        method: 'POST',
        body: formData
    })
    .then(resp => resp.json())
    .then(data => {
        console.log(data);

        if(data != null){
            alert('댓글이 등록되었습니다.');
```

- 폼에서 수신한 데이터를 수신
- 해당 데이터로 DTO 생성
- service를 통해 댓글 등록 후 등록한 댓글 바로 select
- 해당 댓글 json 형태로 수신

- 폼 데이터 controller로 수신
- controller에서 등록 후 해당 등록한 댓글 데이터 수신
- 해당 데이터를 동적 태그 생성으로 바로 삽입

[JSP] 글보기 기능 구현하시오

1. 기본 글보기 기능, 첨부 파일 내용 출력 및 파일 다운로드 구현, 댓글 목록 및 쓰기 구현

글 보기 화면

글보기

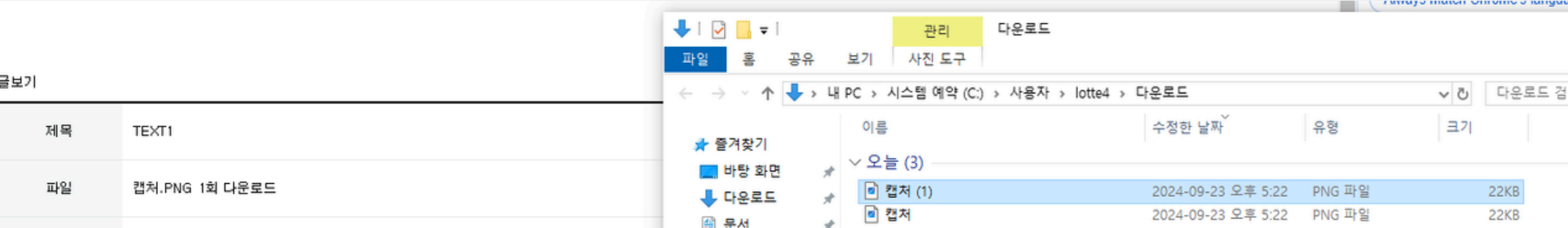
제목	TEXT1
파일	캡처.PNG 1회 다운로드
내용	TEXT1

삭제

수정

목록

다운로드 화면



댓글 목록 및 쓰기 구현

글보기

제목	TEXT1
파일	캡처.PNG 1회 다운로드
내용	TEXT1

삭제

수정

목록

댓글 목록

순수링 24-09-23
JSP 테스트

삭제 수정

THANK YOU

최준혁

2024/09/03

정기수행평가3회 (JSP/DB)