QNA

- User는 글을 작성할 수 있고 자신이 작성한 글만 확인할 수 있다.
- 댓글은 관리자만 작성할 수 있다.

1	tb_use	er 1 🗶									
«Τ	select	* from tb_user	Enter a SC	QL expression to	filter results (use C	Ctrl+Space)					
비금		123 user_no 🔻	ABC gr_no 🕶	ABC user_id 🕶	ABC user_pwd ▼	ABC user_nm 🔻	ABC user_nicknm 🔻	123 user_gen 🔻	ABC user_ema 🕶	② reg_date ▼	явс admin 💌
미.	1	1	0	earth0629	0629earth!	노지구	나안관리자	0	earth0629@gma	5:48:41.372 +0900	N
	2	2	0	earth0630	0630earth!	나지구	나관리자	0	earth0629@gma	5:53:45.138 +0900	Υ

회원					User
©	회원 번호	user_no	Domain	BIGINT	NOT NULL
©7	등급 번호	gr_no	Domain	char(1)	NOT NULL
	아이디	user_id	Domain	VARCHAR(15)	NOT NULL
	비밀번호	user_pwd	Domain	VARCHAR(15)	NOT NULL
	이름	user_nm	Domain	VARCHAR(30)	NOT NULL
	닉네임	user_nicknm	Domain	VARCHAR(10)	NOTNULL
	성별	user_gen	Domain	CAHR(1)	NOT NULL
	이메일	user_email	Domain	VARCHAR(150)	NOT NULL
	가입일	reg_date	Domain	TIMESTAMPTZ	NOT NULL
	관리자여부	admin	Domain	CHAR(1)	NOT NULL
	회원 이미지	image	Domain	TEXT	NULL
	팔로잉	following	Domain	BIGINT	NOT NULL
	팔로워	followers	Domain	BIGINT	NOT NULL
	신고개수	report_cnt	Domain	INT	NOT NULL

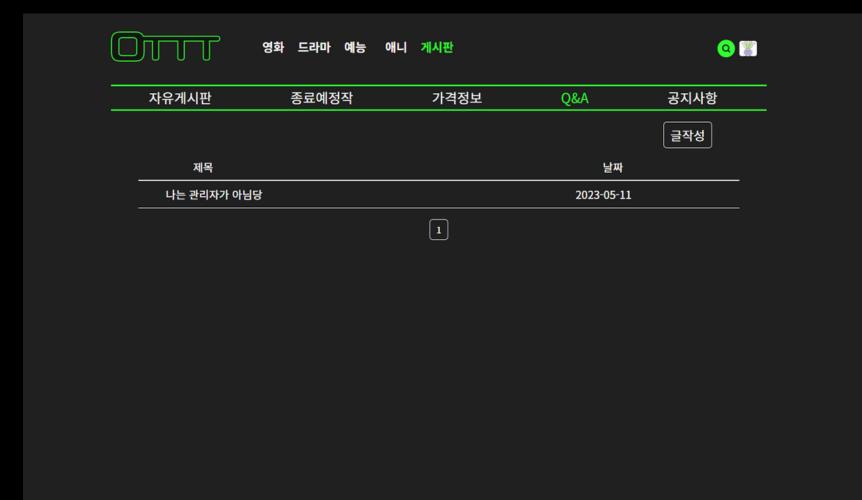
게시	글				Article
©=	게시글번호	article_no	Domain	BIGINT	NOT NULL
© ⊋	회원 번호	user_no	Domain	BIGINT	NOT NULL
©⊽	게시판종류	article_index_no	Domain	char(1)	NOT NULL
	제목	article_title		VARCHAR(255)	NULL
	내용	article_content		VARCHAR(65535)	NULL
	사진	article_image		TEXT	NULL
	등록날짜	article_create_dt		TIMESTAMP	NOT NULL
	작성자	article_create_user		VARCHAR(50)	NOT NULL
	수정일	article_mod_dt		TIMESTAMP	NULL
	좋아요	article_like_count		INT	NULL
	신고개수	report_cnt	Domain	INT	NULL

QNA

- User는 글을 작성할 수 있고 자신이 작성한 글만 확인할 수 있다.
- 댓글은 관리자만 작성할 수 있다.

=	tb_arti	icle 🕍 Enter a S	SQL	expression to fi	ilter	results (use Ctrl+Space)				
ш		123 article_no	•	123 user_no	•	ABC article_index_no	1	ABC article_content 🔻	RBC article_image 🔻	ABC article_title
	1		2		2	2		나관리자임당	[NULL]	나는 관리자임당
	2		1		1	2		노관리자임당	[NULL]	나는 관리자가 아님당

<user_no=1> 관리자가 아닐 경우



<user_no=2> 관리자일 경우



QNA

- User는 글을 작성할 수 있고 자신이 작성한 글만 확인할 수 있다.
- 댓글은 관리자만 작성할 수 있다.

로그인 CONTROLLER

```
@PostMapping("/login")
public String login(String user id, String user pwd, String toURL, boolean rememberId,
        HttpServletRequest request, HttpServletResponse response) throws UnsupportedEncodingException {
   if(!loginCheck(user id, user pwd)) {
   String msg = URLEncoder.encode("id 또는 pwd가 일치하지 않습니다", "utf-8");
    return "redirect:/login?msg="+msg;
    if(rememberId) {
   Cookie cookie = new Cookie("id", user_id);
    response.addCookie(cookie);
    else {
   Cookie cookie = new Cookie("id", user id);
    cookie.setMaxAge(0);
    response.addCookie(cookie);
   UserDTO userDTO = userDao.select(user id);
   HttpSession session = request.getSession();
   session.setAttribute("id", user_id);
   session.setAttribute("admin", userDTO.getAdmin());
   session.setAttribute("no", userDTO.getUser no());
    toURL = toURL == null || toURL.equals("") ? "/" : toURL;
    return "redirect:" +toURL;
```

QNA

- User는 글을 작성할 수 있고 자신이 작성한 글만 확인할 수 있다.
- 댓글은 관리자만 작성할 수 있다.

QNA CONTROLLER

```
@Controller
@RequestMapping("/community")
public class OnAController {
    @Autowired
    QnAServiceImpl serviceImpl;
    @Autowired
    LoginUserDao loginUserDao;
   //OnA
        @GetMapping(value = "/QnA")
        public String QnA(SearchItem sc, Model m, HttpServletRequest request, HttpSession session) {
            try {
                UserDTO userDTO = loginUserDao.select((String) session.getAttribute("id"));
                char admin = (char) session.getAttribute("admin");
                if(admin=='Y') {
                    int totalCnt = serviceImpl.getAllCount(sc);
                    m.addAttribute("totalCnt", totalCnt);
                    List<ArticleDTO> list = serviceImpl.getAllPage(sc);
                    PageResolver pageResolver = new PageResolver(totalCnt, sc);
                    m.addAttribute("list", list);
                    m.addAttribute("pr", pageResolver);
                }else {
                    int totalCnt = serviceImpl.getCount(sc, userDTO.getUser no());
                    m.addAttribute("totalCnt", totalCnt);
                    List<ArticleDTO> list = serviceImpl.getPage(sc, userDTO.getUser no());
                    PageResolver pageResolver = new PageResolver(totalCnt, sc);
                    m.addAttribute("list", list);
                    m.addAttribute("pr", pageResolver);
            } catch (Exception e) {e.printStackTrace();}
            return "/community/QnA/QnA";
```

QNA

- User는 글을 작성할 수 있고 자신이 작성한 글만 확인할 수 있다.
- 댓글은 관리자만 작성할 수 있다.

〈관리자일 경우〉

<QnA Controller>

```
if(admin=='Y') {
    int totalCnt = serviceImpl.getAllCount(sc);
    m.addAttribute("totalCnt", totalCnt);
    List<ArticleDTO> list = serviceImpl.getAllPage(sc);
    PageResolver pageResolver = new PageResolver(totalCnt, sc);
    m.addAttribute("list", list);
    m.addAttribute("pr", pageResolver);}
```

<ArticleQnADao>

```
@Repository
public class ArticleQnADao implements ArticleNoticeDao {
    @Autowired
    private SqlSession session;
    private static String namespace = "com.ottt.ottt.dao.community.qnaMapper.";

public List<ArticleDTO> selectAll(SearchItem sc) throws Exception{
        return session.selectList(namespace+"selectAll", sc);
    }

public int countAll(SearchItem sc) throws Exception {
        return session.selectOne(namespace+"countAll", sc);
    }
```

<QnAServiceImpl>

```
@Service
public class QnAServiceImpl implements ArticleService {
    @Autowired
    ArticleQnADao articleQnADao;

public List<ArticleDTO> getAllPage(SearchItem sc) throws Exception {
    return articleQnADao.selectAll(sc);
}

public int getAllCount(SearchItem sc) throws Exception {
    return articleQnADao.countAll(sc);
}
```

<QnA Mapper>

```
<select id="selectAll" parameterType="SearchItem" resultType="ArticleDTO">
    select *
    from tb_article
    where article_index_no = '2'
    order by article_no desc
    limit #{pageSize} offset #{offset}

</select>

<select id="countAll" parameterType="SearchItem" resultType="int">
    select count(*)
    from tb_article
    where article_index_no = '2'
</select>
```

QNA

- User는 글을 작성할 수 있고 자신이 작성한 글만 확인할 수 있다.
- 댓글은 관리자만 작성할 수 있다.

〈관리자가 아닐 경우〉

<QnA Controller>

```
else {
    int totalCnt = serviceImpl.getCount(sc, userDTO.getUser_no());
    m.addAttribute("totalCnt", totalCnt);
    List<ArticleDTO> list = serviceImpl.getPage(sc, userDTO.getUser_no());
    PageResolver pageResolver = new PageResolver(totalCnt, sc);
    m.addAttribute("list", list);
    m.addAttribute("pr", pageResolver);
}
```

<ArticleQnADao>

```
@Repository
public class ArticleQnADao implements ArticleNoticeDao {
    @Autowired
    private SqlSession session;
    private static String namespace = "com.ottt.ottt.dao.community.qnaMapper.";
    public List<ArticleDTO> selectPage(SearchItem sc, Integer user_no) throws Exception {
        Map map = new HashMap();
        map.put("searchItem", sc);
        map.put("user no", user no);
        map.put("pageSize", sc.getPageSize());
        map.put("offset", sc.getOffset());
        return session.selectList(namespace+"selectList", map);
    public int count(SearchItem sc, Integer user_no) throws Exception {
        Map map = new HashMap();
        map.put("SearchItem", sc);
        map.put("user no", user no);
        return session.selectOne(namespace+"count", map);
```

<QnAServiceImpl>

```
@Service
public class QnAServiceImpl implements ArticleService {
    @Autowired
    ArticleQnADao articleQnADao;

public List<ArticleDTO> getPage(SearchItem sc, Integer user_no) throws Exception {
    return articleQnADao.selectPage(sc, user_no);
    }

public int getCount(SearchItem sc, Integer user_no) throws Exception {
    return articleQnADao.count(sc, user_no);
}
```

<QnA Mapper>

```
<select id="selectList" parameterType="map" resultType="ArticleDTO">
    SELECT article_title, article_create_dt, article_no
    FROM tb_article
    where article_index_no = '2' and user_no = #{user_no}
    order by article_no desc
    limit #{pageSize} offset #{offset}

</select>

<select id="count" parameterType="map" resultType="int">
        select count(*)
        from tb_article
        where article_index_no = '2' and user_no = #{user_no}
</select>
```

공지사함

- 관리자만 글을 작성할 수 있으며, User는 글을 작성할 수 없다.
- 댓글도 작성할 수 없다.

	tb_use	er 1 🗙										
φT	select	* from tb_user	K	Enter a SQ	L expression to	filter results (use C	Ctrl+Space)					
를 le		123 user_no		ABC gr_no 🔻	ABC user_id 🕶	asc user_pwd ▼	ABC user_nm 🕶	ABC user_nicknm 🕶	123 user_gen 🔻	ABC user_ema 🕶	② reg_date ▼	ABC admin ▼
	1	1	(0	earth0629	0629earth!	노지구	나안관리자	0	earth0629@gma	5:48:41.372 +0900	N
	2	2	2 (0	earth0630	0630earth!	나지구	나관리자	0	earth0629@gma	5:53:45.138 +0900	Υ

회원					User
©	회원 번호	user_no	Domain	BIGINT	NOT NULL
©7	등급 번호	gr_no	Domain	char(1)	NOT NULL
	아이디	user_id	Domain	VARCHAR(15)	NOT NULL
	비밀번호	user_pwd	Domain	VARCHAR(15)	NOT NULL
	이름	user_nm	Domain	VARCHAR(30)	NOT NULL
	닉네임	user_nicknm	Domain	VARCHAR(10)	NOTNULL
	성별	user_gen	Domain	CAHR(1)	NOT NULL
	이메일	user_email	Domain	VARCHAR(150)	NOT NULL
	가입일	reg_date	Domain	TIMESTAMPTZ	NOT NULL
	관리자여부	admin	Domain	CHAR(1)	NOT NULL
	회원 이미지	image	Domain	TEXT	NULL
	팔로잉	following	Domain	BIGINT	NOT NULL
	팔로워	followers	Domain	BIGINT	NOT NULL
	신고개수	report_cnt	Domain	INT	NOT NULL

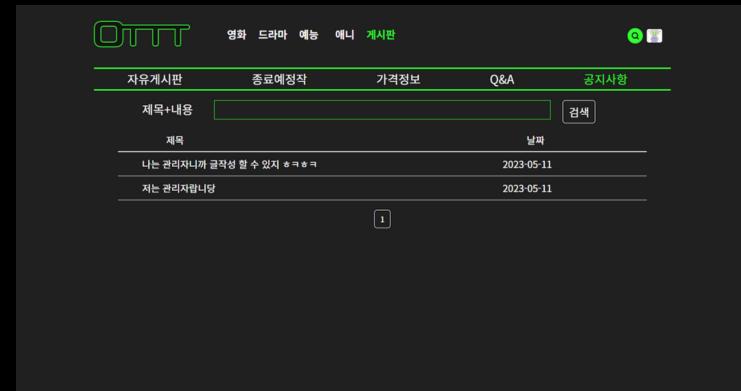
게시	글				Article
<u>Сэ</u>	게시글번호	article_no	Domain	BIGINT	NOT NULL
⊚≂	회원 번호	user_no	Domain	BIGINT	NOT NULL
⊚≂	게시판종류	article_index_no	Domain	char(1)	NOT NULL
	제목	article_title		VARCHAR(255)	NULL
	내용	article_content		VARCHAR(65535)	NULL
	사진	article_image		TEXT	NULL
	등록날짜	article_create_dt		TIMESTAMP	NOT NULL
	작성자	article_create_user		VARCHAR(50)	NOT NULL
	수정일	article_mod_dt		TIMESTAMP	NULL
	좋아요	article_like_count		INT	NULL
	신고개수	report_cnt	Domain	INT	NULL

공지사항

- 관리자만 글을 작성할 수 있으며, User는 글을 작성할 수 없다.
- 댓글도 작성할 수 없다.

> QnA.jsp

<user_no=1> 관리자가 아닐 경우



<user_no=2> 관리자일 경우

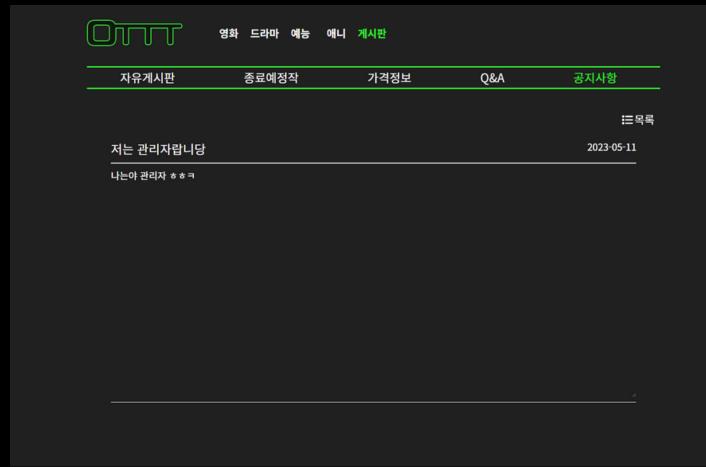
<u> </u>	Q&A 공지사항	검색 글작성	날짜	2023-05-11	2023-05-11	
	병보 Q&A			2023	2023	
	가격정					1
영화 드라마 예능	종료예정작			작성 할 수 있지 ㅎㅋㅎㅋ		
	자유게시판	제목+내용	제목	나는 관리자니까 글	저는 관리자랍니당	

공지사항

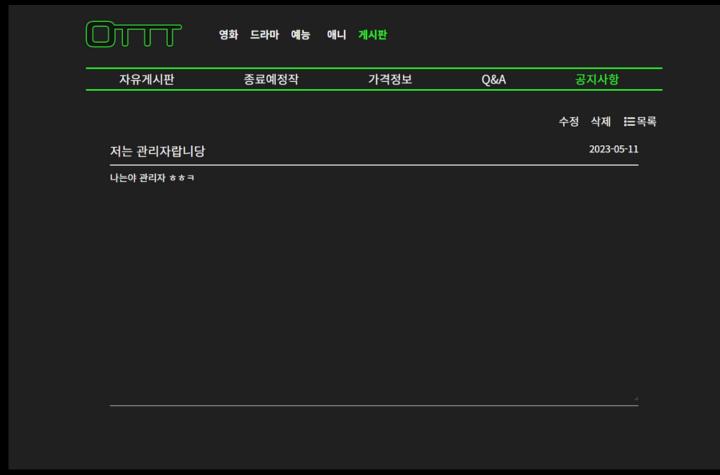
- 관리자만 글을 작성할 수 있으며, User는 글을 작성할 수 없다.
- 댓글도 작성할 수 없다.

> QnA.jsp

<user_no=1> 관리자가 아닐 경우



<user_no=2> 관리자일 경우





- 관리자만 글을 작성할 수 있으며, User는 글을 작성할 수 없다.
- 댓글도 작성할 수 없다.

notice.jsp

notice Mapper

