2021.12.07

Portfolio

Seock Jeong Hyeon

Project Name :

#인생맛집

2021.03 - 2021.06

#인생맛집은 기초적인 자바교육만 받은 상태에서 제작된 페이지로 첫 시작은 jsp로 시작하였지만 servlet이라는 개념을 알아가면서 MVC1에서 **MVC2모델을 제작**하게 되었습니다.

모든 페이지는 UI, UX, 기능적인 것들을 **직접 만들면서 비효율적인 코드겠지만 웹 제작 과정의 이해와 언어의 특징 및 역할에 대한 이해도를 높이기 위해 노력**하였습니다.

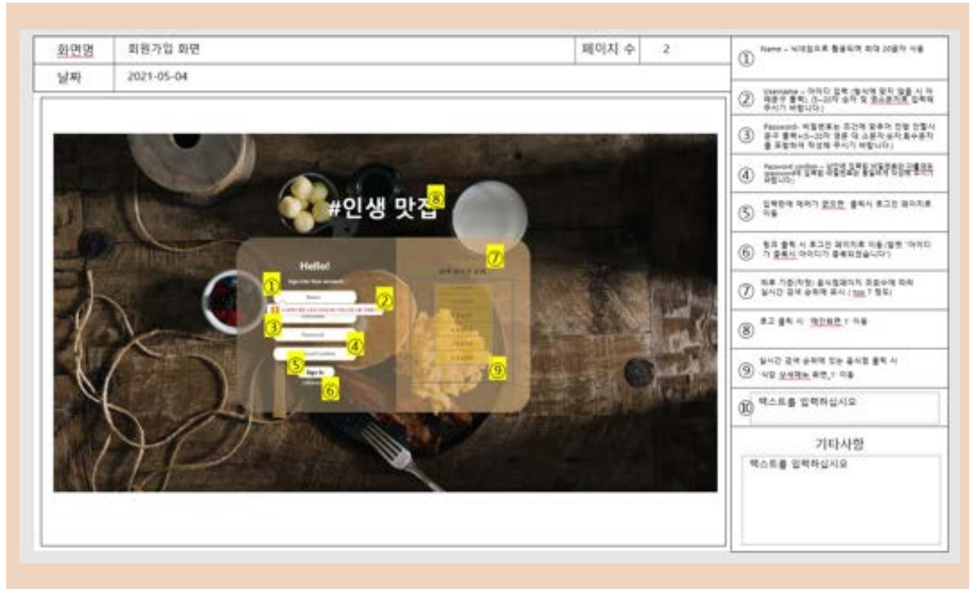
시작할 때는 데이터와의 공유 등 개발과정에 대해서 막연하였지만 개발을 완료한 지금은 각 **개발과정에서 유의점과 개발자의 역할에 대해서 인지**할 수 있게 되었습니다.

Process

#인생 맛집 계획표

수정일자	수정세부사항내역	수정자	기타사항
2021.04.24	- 색깔코드 안내 및 날짜, 페이지 수 입력 #인생 맛집 계획잡정 표 추가 로그아웃시 알맹 설정 추가 글자 크기 수정	석정현	
2021.04.26	- 로그인페이지 글자크기 및 세부사항 추가	석정현	
	- 메인 하단 footer부분 제작 (메인, 로그인, 프로젝트 소개버튼추가 / 커피라이터 추가)	김세희	
	- 메인 오늘의 메뉴 이미지 슬라이드 직접구현(mySlide) 시 처음 이미지에서 마지막 이미지로 못 넘어가는 오류 수정	이후건	
	- 메인 헤더 세로폭 크기 조정	윤재빈	
2021.04.27	- 하단 footer영역 폰트 디자인 수정	김세희	
	- 로그인 전/후 아이콘 변경, 로그인 전/후 페이지 연결	윤재빈	
2021.04.28	- 검색라이콘 위치 변경	윤재빈	
2021.04.29	- 프로젝트 2차 디자인 및 기획 수정 (1)	석정현	
2021.04.30	- 슬라이드 라디오 기능 구현	이후건	

No.	기능적 요구사항	회원	비회원	비고
1	회원가입 화면			
1.1	회원가입		회원가입	아이디, 패스워드, 패스워드 확인
2	로그인 화면			
2.1	로그인	로그인		아이디, 비밀번호
2.2	SNS 연동			SNS 링크 이동
2.3	실시간 검색 순위			실시간 검색 순위
3	메인화면			
3.1	메인 바로가기			메인화면 바로가기 링크
3.2	검색			음식점 검색 기능
3.3	검색유도 문구	검색유도		xx님, 무엇을 찾고 계신가요?
3.4	로그인		로그인	미로그인시 로그인 바로가기 링크
	로그아웃	로그아웃		로그인시 로그아웃 전환 링크
3.5	카테고리			한식, 중식, 일식, 양식
3.6	오늘의 메뉴			카테고리 탑 맛집 이미지 슬라이드
3.7	Footer - 메인바로가기			메인화면 바로가기 링크
3.8	Footer - 로그인		로그인	미로그인시 로그인 바로가기 링크
	Footer - 로그아웃	로그아웃		로그인시 로그아웃 전환 링크
3.9	Footer - 프로젝트 소개			프로젝트 소개 새창 링크
4	식당 화면			
4.1	카테고리 글자			메인에서 선택해서 접속한 카테고리 글자(한식, 중식, 일식, 양식)
4.2	전체보기			클릭시 등록된 업체 전체보기
4.3	식당미리보기			3개씩 카테고리별 등록된 업체 간략소개
4.4	식당 미리보기 라디오 버튼			등록된 식당 미리보기 하단의 라디오 버튼
5	전체보기			
5.1	줄여보기			전체보기 클릭시 줄여보기로 변경
5.2	식당 상세메뉴 이동			업체 이미지 클릭시 상세메뉴 페이지로 이동
5.3	탑 버튼			클릭시 화면 상단으로 이동
6	식당 상세메뉴			
6.1	별점			리뷰 평균 별점
6.2	위치			네이버 지도 연동 링크
6.3	상세정보			전화번호, 편의시설, 영업시간, 매장소개 등 업체 상세정보
6.4	대표메뉴			최대 8개의 업체 대표메뉴와 가격
6-1	리뷰			
6-1.1	리뷰 텍스트	리뷰 텍스트		텍스트 리뷰 작성 (300자 제한)
6-1.2	리뷰 별점	리뷰 별점		별 한 개 클릭시 0.5단위로 별 색 변동. 별 1개가 다 찼을 때 해당 별을 다시 누르면 별 삭제
6-1.3	리뷰 등록	리뷰 등록		저장버튼 클릭시 리뷰 등록(리뷰 등록 이후 리뷰가 기입된 칸은 =':;)
6-1.4	리뷰 삭제	리뷰 삭제		X버튼 클릭시 비밀번호를 요구하는 창을 팝업으로 안내. 본인이 작성한 리뷰가 아닌데 클릭시 "본인이 작성한 리뷰만 삭제 가능합니다" 알림



화면명	회원가입 화면	페이지 수	2
날짜	2021-05-04		

#인생 맛집

Hello!

① Name - 닉네임으로 활용되며 최대 20글자 이내

② Username - 아이디 입력 (형식에 맞지 않을 시 자동으로 출력) (5~20자 숫자 및 영소문자로 입력해 주시기 바랍니다.)

③ Password - 비밀번호는 조건에 맞추어 입력 안할시 글구 출력+5~20자 영문 대 소문자,숫자,특수문자을 포함하여 작성해 주시기 바랍니다.)

④ Password confirm - 입력한 비밀번호와 동일하게 입력해주세요.

⑤ 입력한에 맞추기 못하면 출력시 로그인 페이지로 이동

⑥ 정상 출력 시 로그인 페이지로 이동(정상 '아이디가 생성되어 아이디가 출력되었습니다.')

⑦ 회원 가입(작업) 출력(정상) 조합수에 따라 실시간 검색 순위에 표시 (100 개 정도)

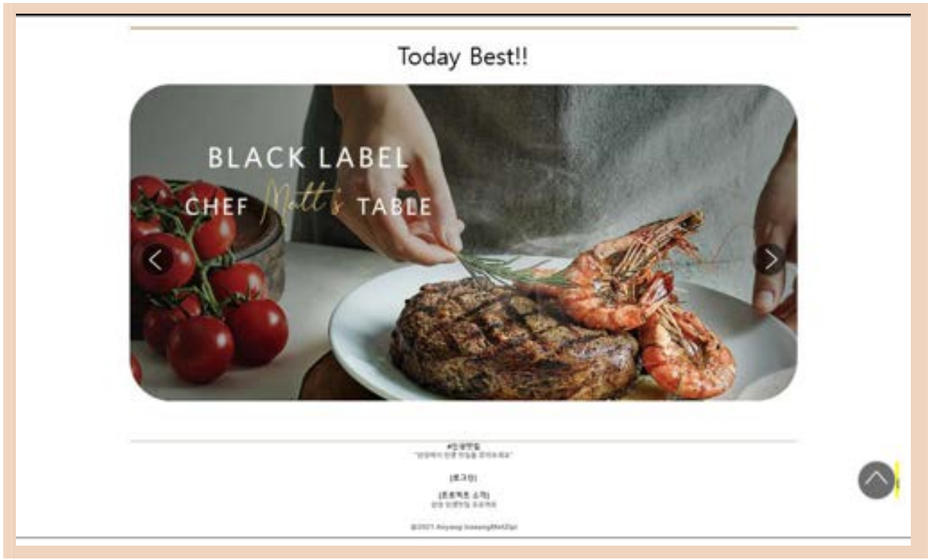
⑧ 로고 클릭 시 '메인화면 ' 이동

⑨ 실시간 검색 순위에 있는 음식점 클릭 시 '식당 상세메뉴 화면 ' 이동

⑩ 텍스트를 입력하십시오

기타사항

텍스트를 입력하십시오



Today Best!!

BLACK LABEL

CHEF Matt's TABLE

프로젝트 진행은 진행계획표와 요구명세서 등을 작성하였으며 **기획서는 ppt를 통해 각 기능별 페이지별 역할에 대해서 정의**를 하였습니다.

개발의 주요 기능으로는 **로그인, 회원가입, 실시간 순위, 리뷰, 파일 업로드, 데이터와 공유** 등을 중심으로 진행하였습니다.

각 프론트에서 진행할 수 있는 **부분은 html, CSS, js를 사용하였으며 js는 사진 슬라이드, 파일 더보기, 로그인, 회원가입에서의 오류안내 등에 사용**하였습니다.

이미지 슬라이드의 경우 img경로와 a태그를 사용해 해당 페이지 경로를 지정하였으며, **js를 이용, 3초마다 자동으로 슬라이드 기능과 각 이동버튼을 통해서도 이동할** 수 있도록 만들었습니다.

진행사항 및 계획표: ppt
요구명세서: 한글(워드)
디자인 페이지 구현: XD

#인생맛집

Process

MySQL을 통해 데이터 입력 및 노출과 각 div가 생성될 수 있도록 jstl을 사용해 만들었습니다.

더보기의 경우 js를 사용하여 기본 디폴트로 div 3개를 노출과 하단 번호를 노출하도록 하였으며 '더보기' 시에 노출될 정보는 모두 hidden치리를 통해 진행해 보았습니다.

세부메뉴에서는 네이버 링크를 사용해 지도를 표현하였습니다. 하단 footer에 간단한 프로젝트 소개설명을 위해 window.open을 통해 새창으로 만들어 보았습니다.

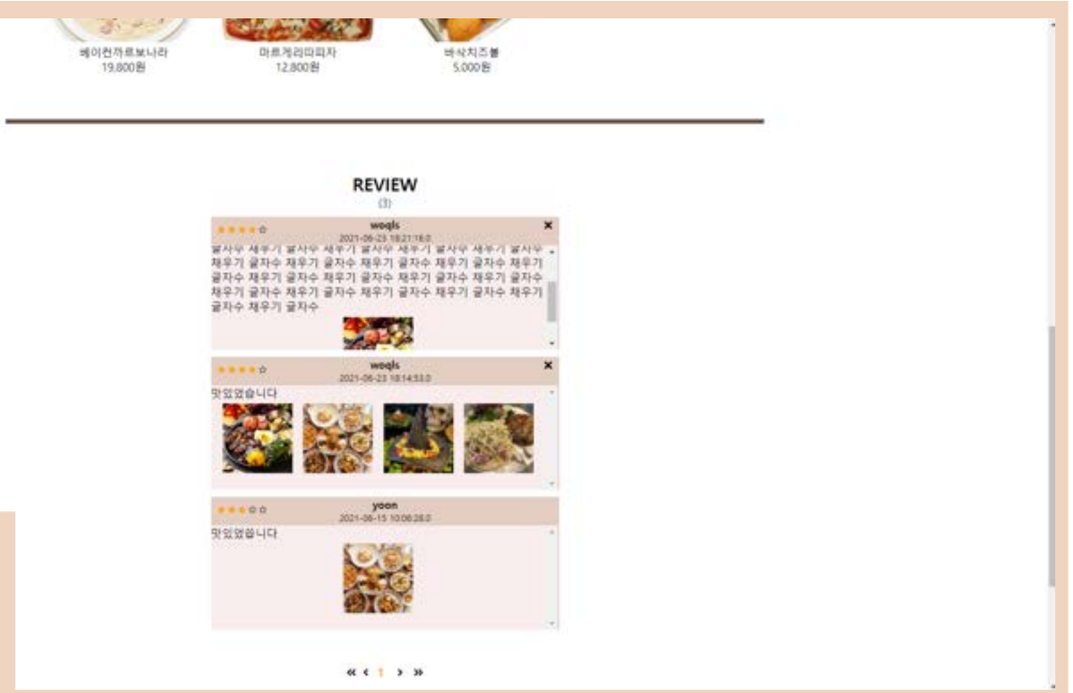
검색은 MySQL문을 사용해 식당이름과, 메뉴를 기준으로 검색결과를 도출할 수 있게 하였습니다.

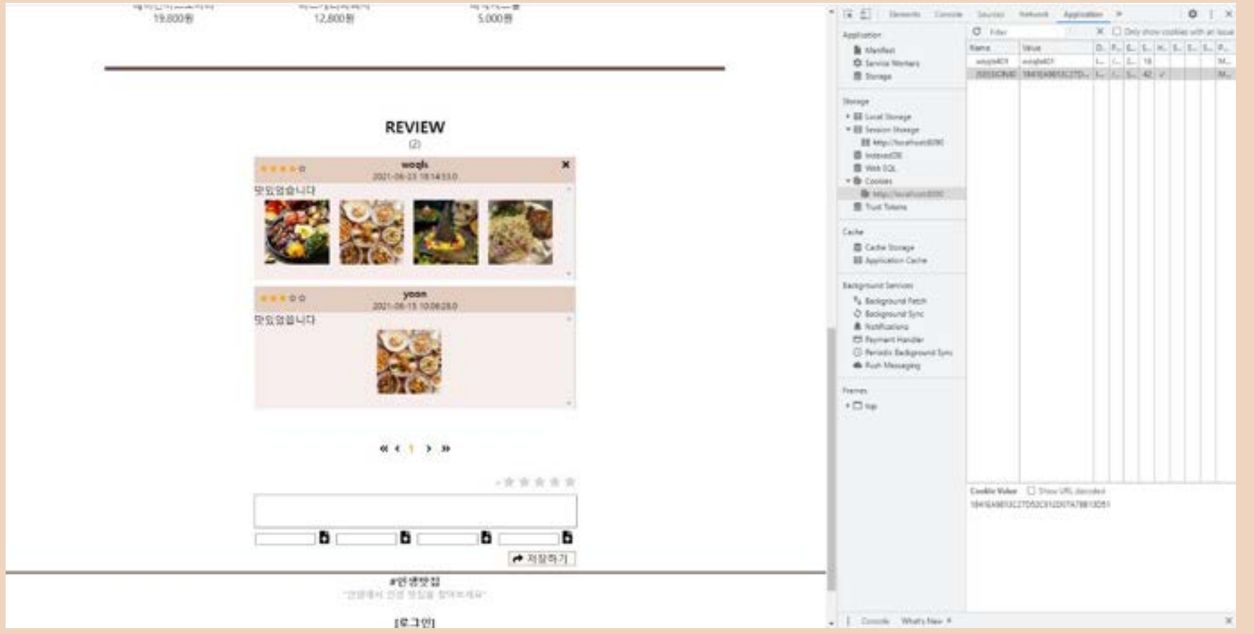
Process

로그인 상태에서 뒤로가기를 누르게 되면 로그아웃 되는 현상을 해결하기 위해 session을 사용하였습니다.

파일 업로드에 경우 @MultipartConfig어노테이션을 통해 fileSizeThreshold, maxFileSize, maxRequestSize로 전체용량, 첨부파일 최대크기, 첨부파일 개수를 지정하고 임시파일경로를 통해

데이터 저장의 경우 concat함수를 사용해 “,” 를 구분점으로 한컬럼에 strin형태로 데이터명과 경로를 합쳐서 저장하였습니다.

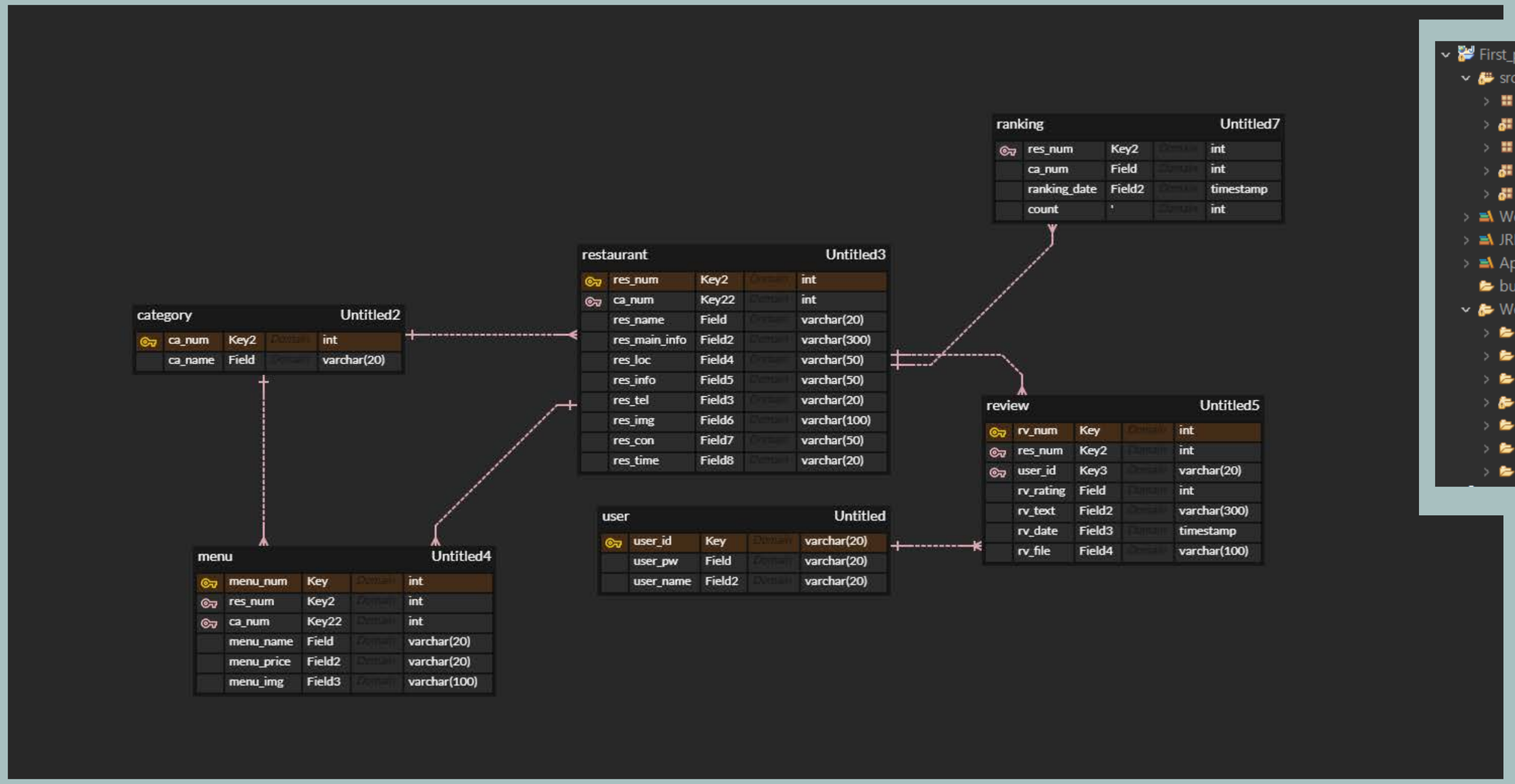


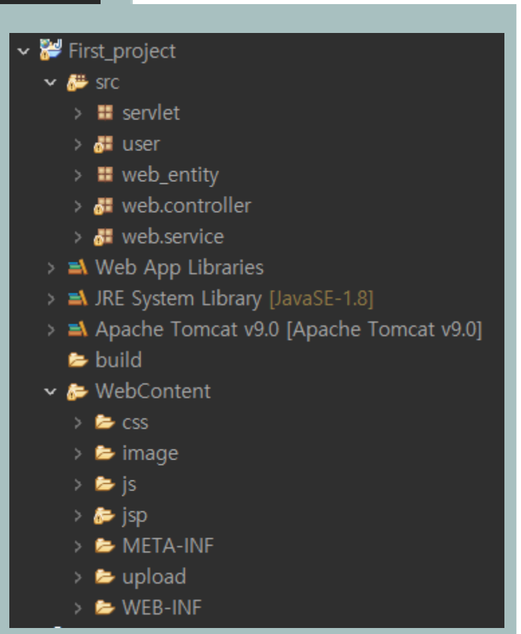


실시간 순위기능의 경우 쿠키를 통해 session에서 받아온 id와 number를 부여하여 중복을 제거하였으며 setMaxage를 통해 cookie를 하루동안 유지할 수 있도록 하였습니다.

MySQL상에서는 식당추가시 trigger을 사용해 자동으로 실시간순위 테이블에도 추가되도록 하였습니다.

#인생맛집 ERD





구성은 WebContent, controller, service, entity로 구분하여 MVC2모델이라 보기에는 부족하지만 근접하게 만들어 보고자 노력하였습니다.

MySQL 구문정리

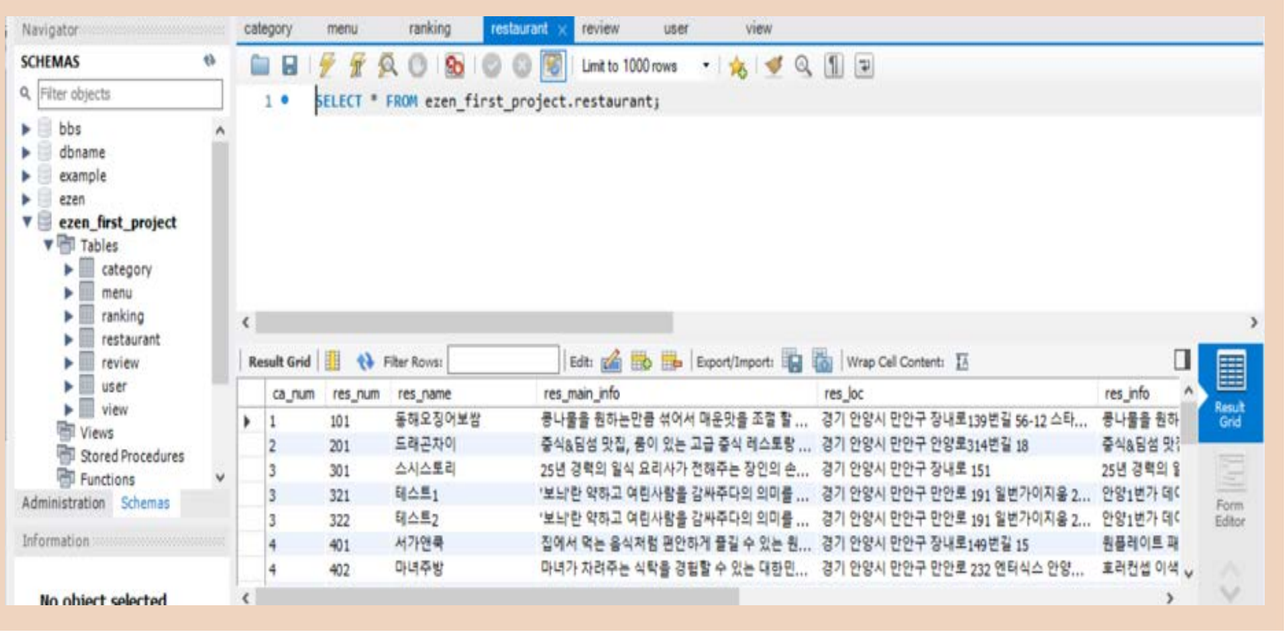
```
//카테고리 선택으로 페이지 접근했을경우 해당 카테고리에 포함되는 모든 음식점
  sql = "SELECT t.* FROM (SELECT @ROWNUM:=@rownum+1 as num, a.* FROM restaurant a, (SELECT @rownum:=0) tmp WHERE a.ca_num = ?) t WHERE t.num BETWEEN ? AND ?;";
  sql_rating = "SELECT a.res_num, avg(b.rv_rating) as avg FROM restaurant a, review b WHERE a.res_num = b.res_num AND a.ca_num = ? GROUP BY a.res_num ;";
```

```
if(caNum != 0) {
    sql = "SELECT a.* FROM restaurant a WHERE a.ca_num = ?;";
    sql_rating = "SELECT a.*, avg(b.rv_rating) as avg FROM restaurant a, review b WHERE a.res_num = b.res_num AND a.ca_num = ? GROUP BY a.res_num;";
}
  //사용자가 검색을 했을 경우 음식점,메뉴에 검색한 글자가 포함되는 데이터 찾기
else {
    sql = "SELECT a.* FROM restaurant a, (SELECT res_num FROM restaurant WHERE res_name LIKE ? UNION SELECT res_num FROM menu where menu_name LIKE ?) c WHERE a.r
    sql_rating = "SELECT a.res_num, avg(b.rv_rating) as avg FROM restaurant a, review b, (SELECT res_num FROM restaurant WHERE res_name LIKE ? UNION SELECT res_n
}
```

```
String sql = "SELECT b.res_name, b.res_img,b.res_num from ranking a, restaurant b where a.res_num = b.res_num order by a.count desc limit 5";
```

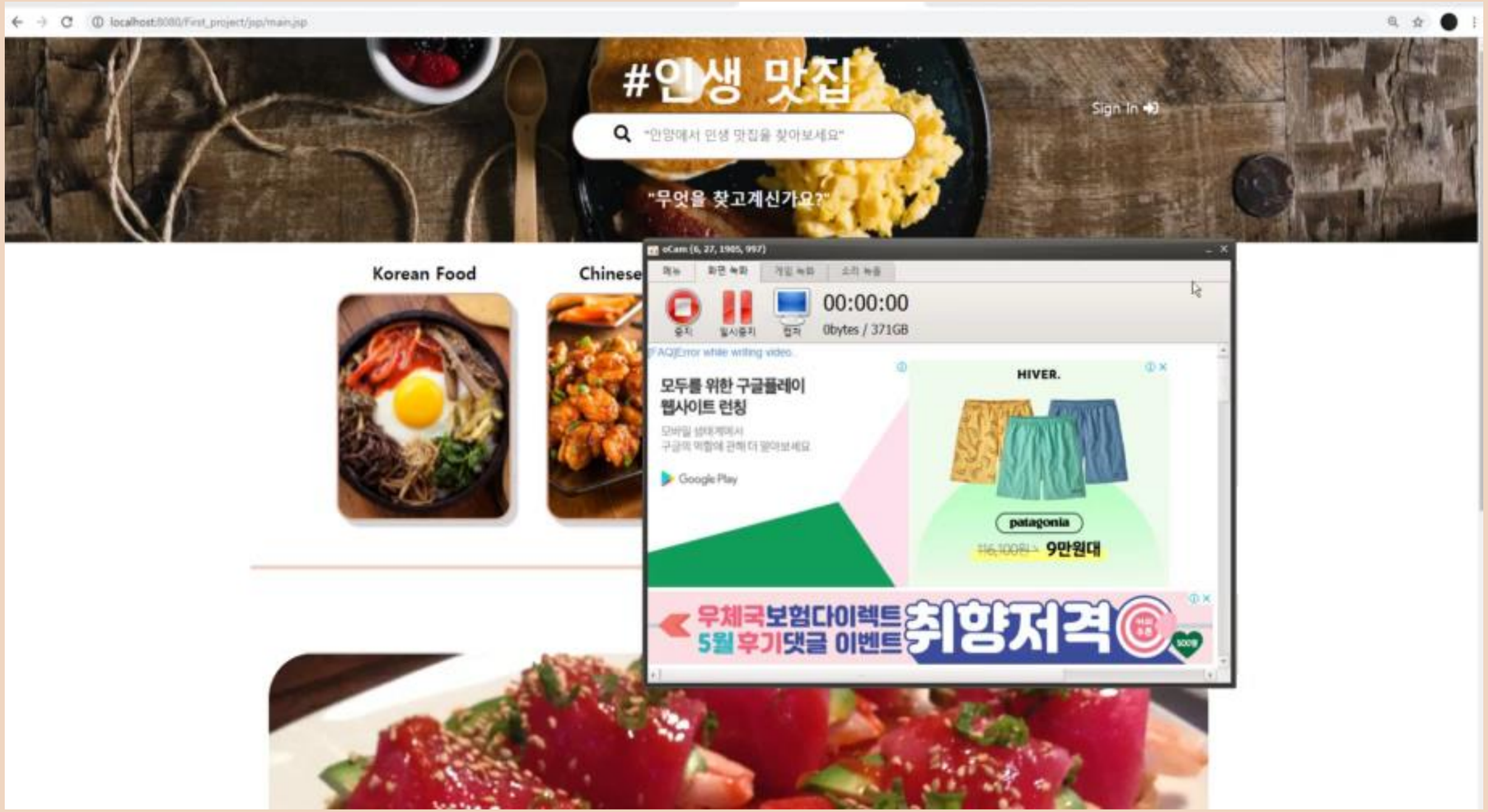
```
String sql="select t.* from("
      + " select @ROWNUM:=@rownum+1 as num, review.* from review, (select @rownum:=0) tmp  where res_num=? order by rv_date desc) t"
      + " where t.num between ? and ?";
```

DB (MySQL)



#인생맛집

영상자료


localhost:8080/First_project/jsp/main.jsp
#인생 맛집
"안양에서 인생 맛집을 찾아보세요"
"무엇을 찾고계신가요?"
Sign In
Korean Food
Chinese
oCam (6, 27, 1905, 997)
00:00:00
0bytes / 371GB
모두를 위한 구글플레이
웹사이트 런칭
Google Play
HIVER.
patagonia
9만원대
우체국보험다이렉트
5월 후기댓글 이벤트
취향저격

Project Name :

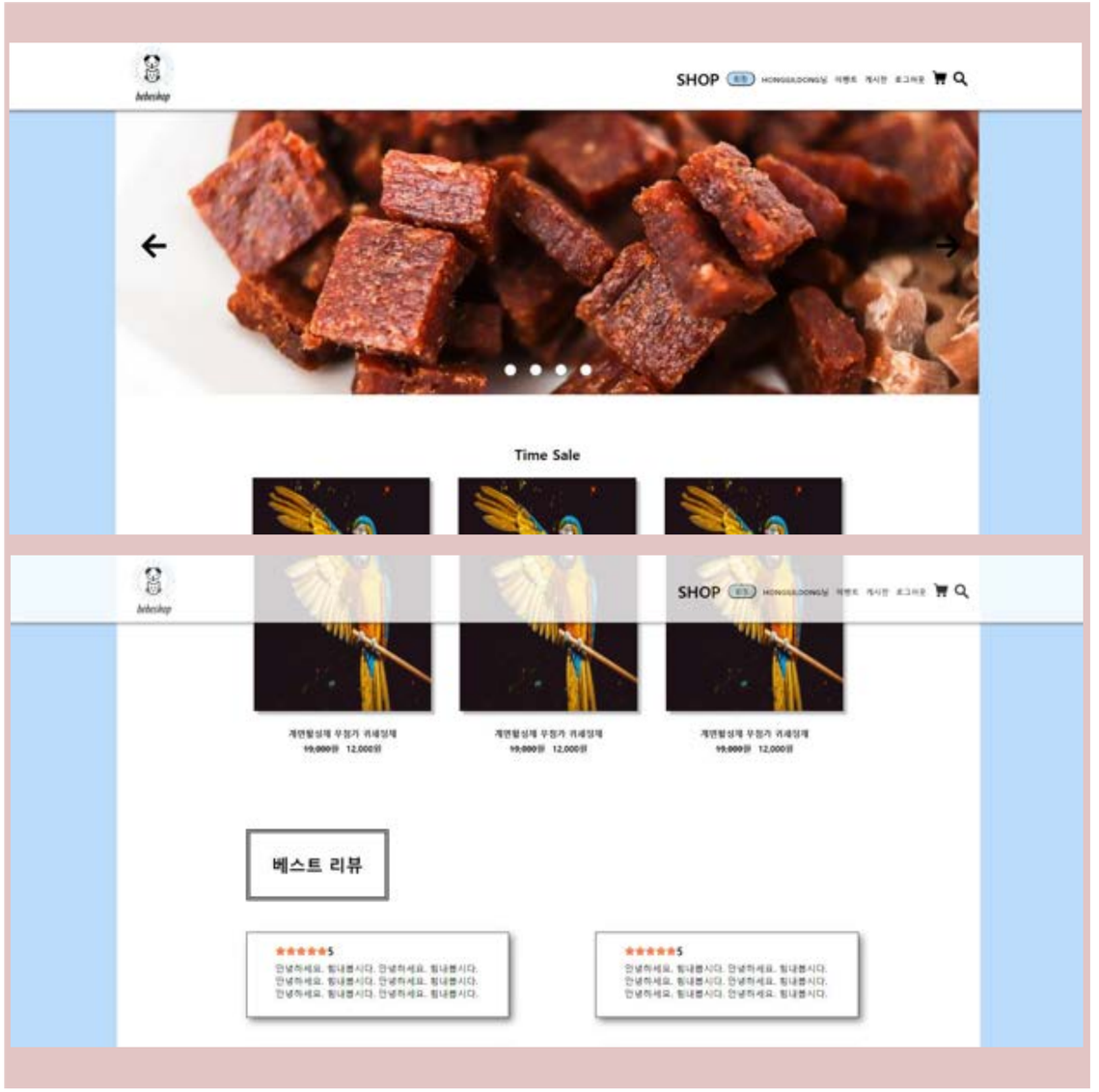
#bebeshop

2021.06 - 2021.08

#인생맛집을 개발하면서 개발과정의 유의점과 개발자의 역할에 대해서 인지하게 되었다면 **이번에는 효율적인 코드를 만들면서 더욱 폭넓은 언어사용 능력을 얻기 위해 노력하였습니다**.

언어의 경우 **백엔드는 Spring프레임워크를 직접 사용**해 보았습니다. 또한 데이터관리의 경우 **oracle을 사용해 보며 MySQL과의 차이점을 찾고 사용법을 숙지**해 보았습니다.

이번 프로젝트에서는 언어의 이해와 더불어 다양한 코드리뷰와 분석을 통해 효율적인 코드를 찾고 사용해보기 위해 노력하였습니다.



Process

진행사항 및 계획표: ppt
디자인 페이지 구현: XD
세부명세서: 엑셀(excel)

결제페이지의 경우 table을 활용했으며
주소는 경우 api와 연동하여 주소지를
oracle로 관리하는 것으로 기획했습니다.

상단 게시판은 10개 글만 노출, 하단은
게시판은 전부 노출하여 **성격이 다른
게시판을 한 페이지를 기획하였습니다.**

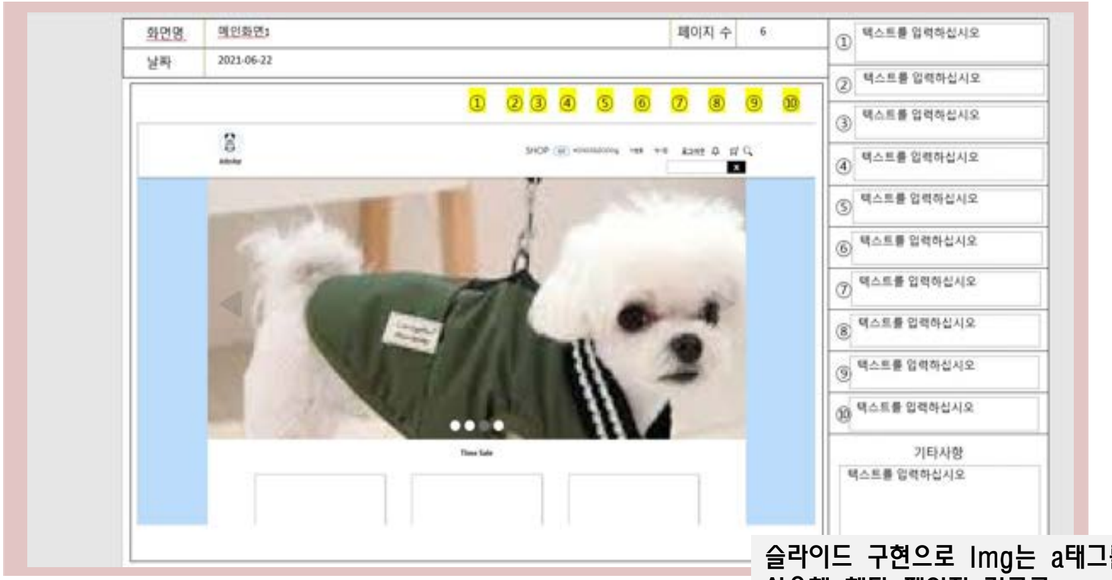
BEST FAQ

1	Testing	Testing ngrok
2	최신글 작성중입니다	최신글 작성중입니다.
3	최신글	최신글
4	555	555
5	123	별아리 123
6	10번째글	10번째글
7	test6	test6중입니다.
8	test5	test5중입니다.
9	test4	test4중입니다.
10	test3	test3중입니다.

공지사항

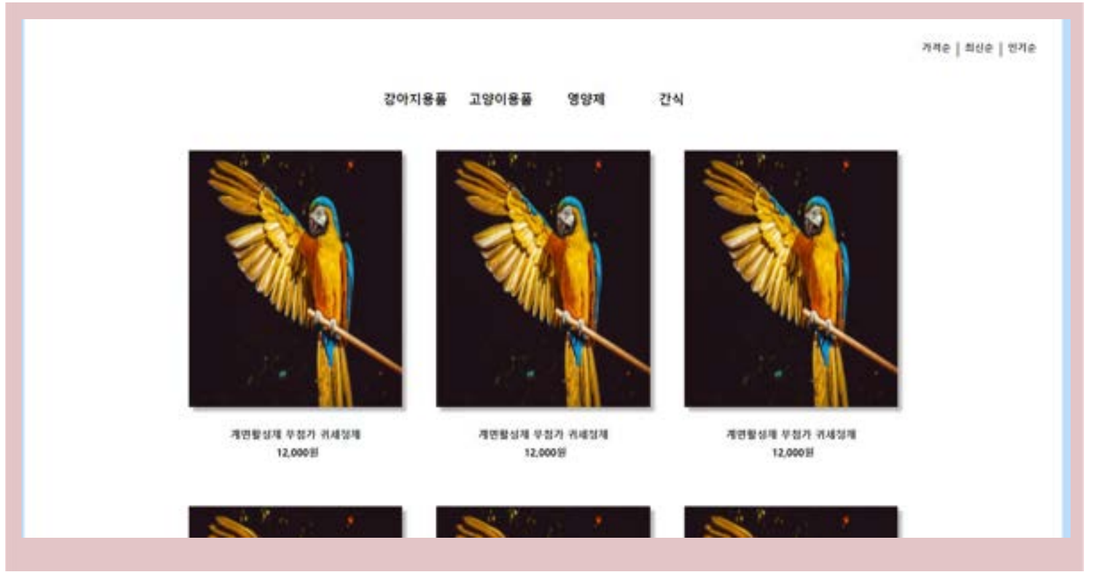
NO	공지사항	공지일자	공지제목
28	문의사항	2021. 7. 30.	테스트중
27	고객안내	2021. 7. 29.	테스트 중입니다
26	이벤트	2021. 7. 29.	ㅗㅗㅗㅗ
25	공지사항	2021. 7. 29.	ㅁㄴㅇㄹ
24	공지사항	2021. 7. 29.	ㅁㄴㅇㄹ

« ‹ [1] [2] [3] [4] [5] › »



슬라이드 구현으로 Img는 a태그를
사용해 해당 페이지 경로를
지정하였습니다.

Process



프론트 **개발은 HTML, CSS, JS를 사용해 개발** 하였으며 Vue를 활용해 완성된 프로젝트에서 개별적으로 진행해 보았습니다.

상단 **header**의 경우 **scrollTop을 통해 특정 값에서 투명화 하도록 만들었으며 검색창은 hidden, 제품은 transform을 통해 자연스럽게 동적 구현을 하였습니다.**

프론트 개발하면서는 HTML, CSS를 사용하면서 **position의 이해도를 높이기 위해 relative, absolute, static을 모두 사용**하여 개발해 보았습니다.

Process [notice.jsp]

```
<h2>BEST FAQ</h2>
</br>
<hr color="#000000" size="5px">
<form role="form" method="GET">
    <table>
        <c:forEach items="${faqList}" var="faqList" begin="0" end="9">
            <tr>
                <td><c:out value="${faqList.rnum}" /></td>
                <td><c:out value="${faqList.faqTitle}" /></td>
                <td><a href="/board/admin_write?faqNo=${faqList.faqNo}&b=2"><c:out value="${faqList.faqContent}" /><
            </tr>
        </c:forEach>
    </table>
```

게시글 제한

```
<div class="buttonMenu">
    <h2>공지사항</h2>
    <a href="/board/admin_write?b=0"><input type="button" value="등록" class="button"></a>
</div>
</br>
<hr color="#000000" size="5px">

<table>
    <tr>
        <th>NO</th>
        <th>공지사항</th>
        <th>공지일자</th>
        <th>공지제목</th>
    </tr>
    <c:forEach items="${noticeList}" var="noticeList" begin="0" end="4">
        <tr>
            <td><c:out value="${noticeList.noticeNo}" /></td>
            <td><c:out value="${noticeList.noticeCategory}" /></td>
            <td><fmt:formatDate value="${noticeList.noticeDate}" /></td>
            <td><a href="/board/admin_write?noticeNo=${noticeList.noticeNo}&b=1"><c:out
                        value="${noticeList.noticeTitle}" /></a></td>
        </tr>
    </c:forEach>
</table>
```

숫자(num) 부여

```
<li><a href="/board/admin_notice?page=1&perPageNum=5"><i class="fas fa-angle-double-left"></i></a></li>
<li><a href="/board/admin_notice${pageMaker.makeSearch(pageMaker.startPage-1)}"><i class="fas fa-angle-left"></i> </a></li>

<c:forEach begin="${pageMaker.startPage}" end="${pageMaker.endPage}" var="idx">
 <li <c:out value="${pageMaker.cri.page == idx?'class=active':''}"/>>
<a href="/board/admin_notice${pageMaker.makeSearch(idx)}" class="num">[${idx}]</a> </li>
</c:forEach>

<c:if test="${pageMaker.next == true}">
<li><a href="/board/admin_notice${pageMaker.makeSearch(pageMaker.endPage+1)}"><i class="fas fa-angle-right"></i> </a></li>
</c:if>
<c:if test="${pageMaker.next == false}">
<li><a href="/board/admin_notice${pageMaker.makeSearch(pageMaker.endPage)}"><i class="fas fa-angle-right"></i> </a></li>
</c:if>
<li><a href="/board/admin_notice?page=${pageMaker.last}&perPageNum=5"><i class="fas fa-angle-double-right"></i></a></li>
```

Fontawesome으로 구현
<,> = 페이지 이동
<<,>> = 가장 끝
페이지 이동 구현

Process [write.jsp]

```jsp
<!-- 제목 입력 사항 -->
<div class="postTitle">
    <h2>* 제목</h2>
    <c:if test="${b eq '0'}">
        <input type="text" class="title"
            placeholder="${rumcount .rnum}제목을 입력해 주세요." name="faqTitle">
    </c:if>
    <c:if test="${b eq '1'}">
        <input type="text" class="title" name="noticeTitle"
            placeholder="제목을 입력해 주세요." value="${readNotice.noticeTitle}">
        <input type="hidden" value="${readNotice.noticeNo}"
            name="noticeNo" />
    </c:if>
    <c:if test="${b eq '2'}">
        <input type="text" class="title" name="faqTitle"
            placeholder="제목을 입력해 주세요." value="${readFaq.faqTitle}">
        <input type="hidden" value="${readFaq.faqNo}" name="faqNo" />
    </c:if>
</div>


<!-- 내용 입력 사항 -->
<div class="postMain">
    <h2>* 내용</h2>
    <c:choose>
        <c:when test="${b eq 0}">
            <textarea class="content" placeholder="내용을 입력해 주세요."
                name="faqContent"></textarea>
        </c:when>
        <c:when test="${b eq 1}">
            <textarea class="content" placeholder="내용을 입력해 주세요."
                name="noticeContent">${readNotice.noticeContent}</textarea>
        </c:when>
        <c:otherwise>
            <textarea class="content" placeholder="내용을 입력해 주세요."
                name="faqContent">${readFaq.faqContent}</textarea>
        </c:otherwise>
```

숫자(num)에 따라
구현기능 및 화면 구분

Value에 따른 name의 변화

```jsp
function select() {
    let selectedChoice = document.querySelector('#choice').selectedIndex;
    if (document.getElementsByTagName('option')[selectedChoice].value == "Best FAQ") {
        document.querySelector('#secondChoice').style.display = 'none'
        document.querySelector('.title').setAttribute("name","faqTitle")
        document.querySelector('.content').setAttribute("name", "faqContent")
    }else{
        document.querySelector('#secondChoice').style.display = 'inline-block'
        document.querySelector('.title').setAttribute("name","noticeTitle")
        document.querySelector('.content').setAttribute("name", "noticeContent")
    }
};
```

Value에 따른 실행
차이 구분

```jsp
function btnU() {
    let selectedChoice = document.querySelector('#choice');
   if (!document.querySelector('.title').value || !document.querySelector('.content').value
           || selectedChoice.value == "공지분류") {
       alert("입력사항을 다시 확인해 주시기 바랍니다.")
   } else {
   let value = document.querySelector("#btnu").value;
   let answer = confirm(value + " 하시겠습니까?");
   let selectedChoice = document.querySelector('#choice').selectedIndex;
       if (value == "등록") {
           if (answer == true) {
               if (document.getElementsByTagName('option')[selectedChoice].value == "Best FAQ") {
                       frm.action = "/board/writeFaq"
                       frm.method = "post"
                       frm.submit()
                }else{
                   frm.action = "/board/writeNotice"
                   frm.method = "post"
                   frm.submit()
                }
           };
       }else if (value == "수정") {
           if (answer == true) {
               <c:if test="${b eq '1'}">
               frm.action = "/board/update"
               frm.method = "post"
               frm.submit()
               </c:if>
               <c:if test="${b eq '2'}">
               frm.action = "/board/update"
               frm.method = "post"
               frm.submit()
               </c:if>

               alert("수정되었습니다.")
           } else {location = "/board/admin_notice"}
       }
   }
```

Process [Controller]

```java
@RequestMapping(value = "delete", method = RequestMethod.GET)
public String delete(BoardVO boardVO, Model model, @RequestParam("b") int num) throws Exception {
    if (num == 1) {
        service.deleteNotice(boardVO.getNoticeNo());
    } else {
        service.deleteFaq(boardVO.getFaqNo());
    }
    System.out.println("삭 to the 제");
    return "redirect:/board/admin_notice";
}

@RequestMapping(value = "update", method = RequestMethod.POST)
public String update(BoardVO boardVO, Model model) throws Exception {


    if(boardVO.getFaqNo() == 0){
        service.updateNotice(boardVO);
    }
    else {
        service.updateFaq(boardVO);
    }

    System.out.println("업 to the 데");
    return "redirect:/board/admin_notice";
}
```

삭제-Delete, 등록-update
(BoardController)

- <%@ include>*를 이용 헤더 설정*
- *삭제 및 업데이트 경우 VO에서 게시판 번호 받아와 비교하여 삭제 및 수정*
- 숫자(num) 0과 1을 통해 2개의 게시판을 비교
- 등록, 수정 등 javascript를 사용해 관리자 이외 불가하도록 구현(hidden 처리)

```java
@Controller
@RequestMapping("/board/")
public class BoardController {

    // 객체타입이 일치하는 객체타입을 자동으로 주입한다.
    @Inject
    private BoardService service;

    @RequestMapping(value = "admin_notice", method = RequestMethod.GET)
//  jsp, servlet으로 값을 전달하는 request, session으로 받아오지만 spring은 model로 받아온다
    public String list(@ModelAttribute("cri") Criteria cri, Model model) throws Exception {
        model.addAttribute("noticeList", service.noticeList(cri));
        model.addAttribute("faqList",service.faqList());

        //페이징 관리
        PageMaker pageMaker = new PageMaker();
        pageMaker.setCri(cri);
        pageMaker.setTotalCount(service.noticeListCount(cri));
        model.addAttribute("pageMaker", pageMaker);

        System.out.println("admin_notice");
        return "board/admin_notice";
    }
```

목록-List (BoardController)

```java
@RequestMapping(value = "admin_write", method = RequestMethod.GET)
public String read(BoardVO boardVO, Model model, @RequestParam("b") int num, HttpServletResponse res)
        throws Exception {
//      model.addAttribute("key", 원하는 내용)
//      BoardVO에 있는 noticeNo를 넣어서 readNotice에 저장합니다.
        model.addAttribute("b", num);
        model.addAttribute("category", service.category(boardVO.getNoticeCategory()));

        if (num == 0) {
            boardVO.setRnum(service.rnumCount());
            if (boardVO.getRnum() >= 10) {
                res.setContentType("text/html; charset=utf-8");
                PrintWriter out = res.getWriter();
                out.println("<script>alert('Best FAQ는 최대10개까지 게시할 수 있습니다.'); </script>");
                out.flush();
            }
        } else if (num == 1) {
            model.addAttribute("readNotice", service.readNotice(boardVO.getNoticeNo()));

        } else if (num == 2) {
            model.addAttribute("readFaq", service.readFaq(boardVO.getFaqNo()));
        }

        System.out.println("admin_write");
        return "board/admin_write";
    }
```

세부내용-Read (BoardController)

- 오라클 Rownum을 이용해 채번을 구현
(채번: 중간에 빠진 숫자가 있을경우 자동으로 숫자가 채워지고 새로운 데이터는 가장 최신(상단)으로 노출)
- 상단 게시판은 10개로 등록할 수 있도록 지정하여 jsp에서 10개 넘어갈 경우 제재, 경고창은 자바로 구현
- Xml을 통해 받아온 data는 model에 넣어 출력
- Criteria를 통해 보여줄 페이지와 페이지 당 보여줄 게시글 개수 설정
- pageMaker에서 Criteria에서 주입 받아 버튼 구현

#인생맛집

Process [ServiceImpl, DaoImpl, xml]

(BoardServiceImpl)

```java
package com.bebe.spring.board.service;

import java.util.List;

@Service("service")
public class BoardServiceImpl implements BoardService {

    @Inject
    private BoardDao dao;

    @Override
    public List<BoardVO> noticeList(Criteria cri) throws Exception {
        return dao.noticeList(cri);
    }

    @Override
    public List<BoardVO> faqList() throws Exception {
        return dao.faqList();
    }

    @Override
    public int noticeListCount(Criteria cri) throws Exception {
        return dao.noticeListCount(cri);
    }

    @Override
    public BoardVO readNotice(int noticeNo) throws Exception {
        return dao.readNotice(noticeNo);
    }

    @Override
    public BoardVO readFaq(int faqNo) throws Exception {
        return dao.readFaq(faqNo);
    }

    @Override
    public void writeNotice(BoardVO boardVO) throws Exception {
        dao.writeNotice(boardVO);

    }

    @Override
    public void writeFaq(BoardVO boardVO) throws Exception {
```

```java
@Override
public List<BoardVO> category(String noticeCategory) throws Exception {
    return dao.category(noticeCategory);
}

@Override
public void deleteNotice(int noticeNo) throws Exception {
    dao.deleteNotice(noticeNo);
}

@Override
public void deleteFaq(int faqNo) throws Exception {
    dao.deleteFaq(faqNo);
}

@Override
public void updateNotice(BoardVO boardVO) throws Exception {
    dao.updateNotice(boardVO);
}

@Override
public void updateFaq(BoardVO boardVO) throws Exception {
    dao.updateFaq(boardVO);
}

@Override
public int rnumCount() throws Exception{
    return dao.rnumCount();
}
```

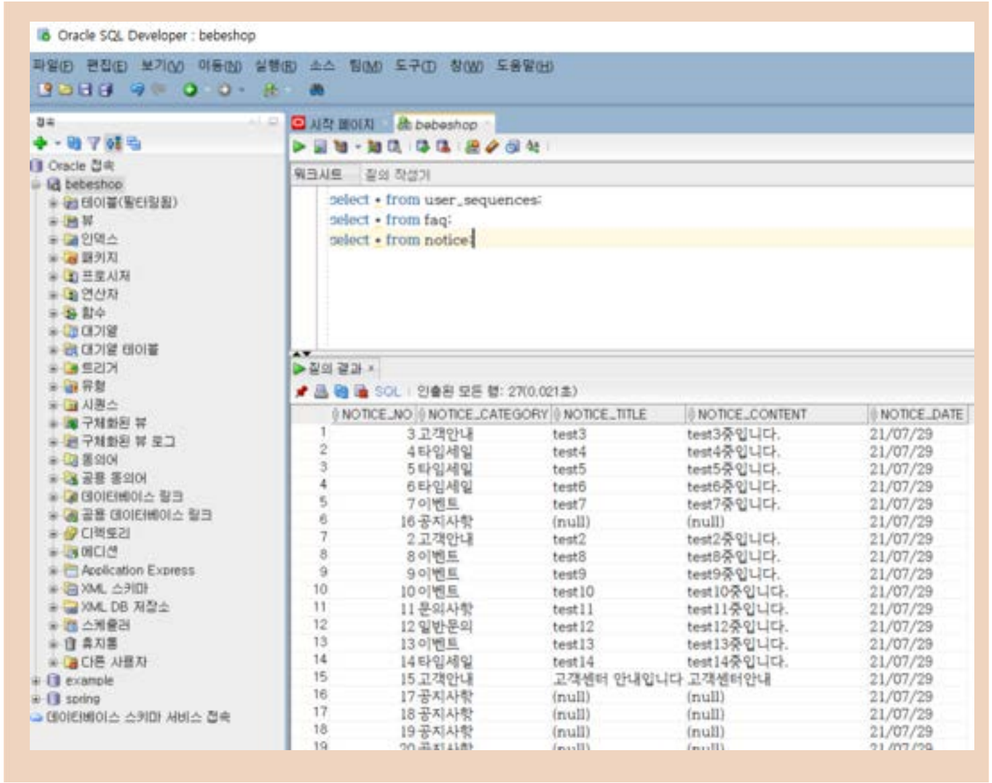
Process [ServiceImpl, DaoImpl, xml]

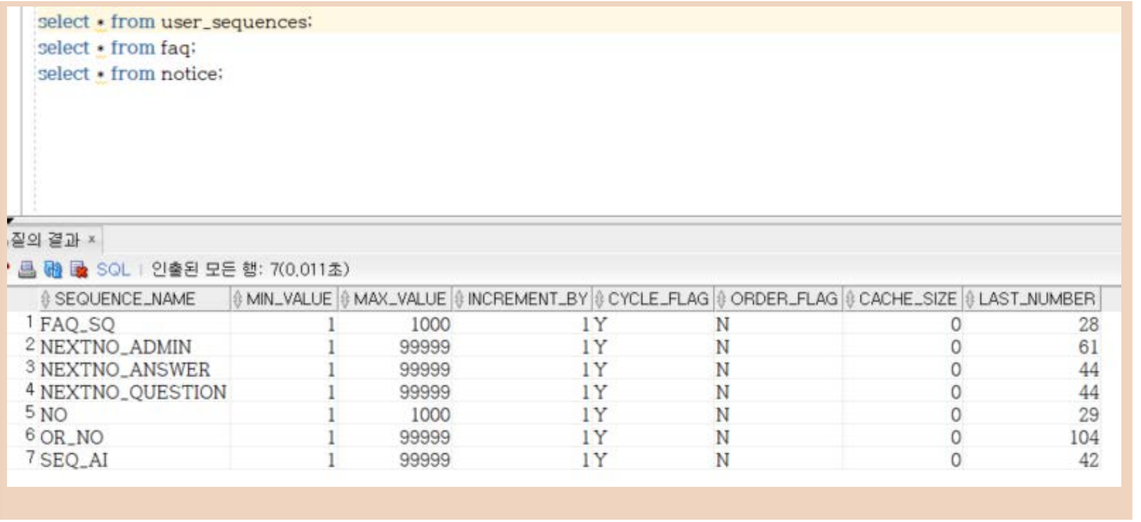
(BoardDaoImpl)

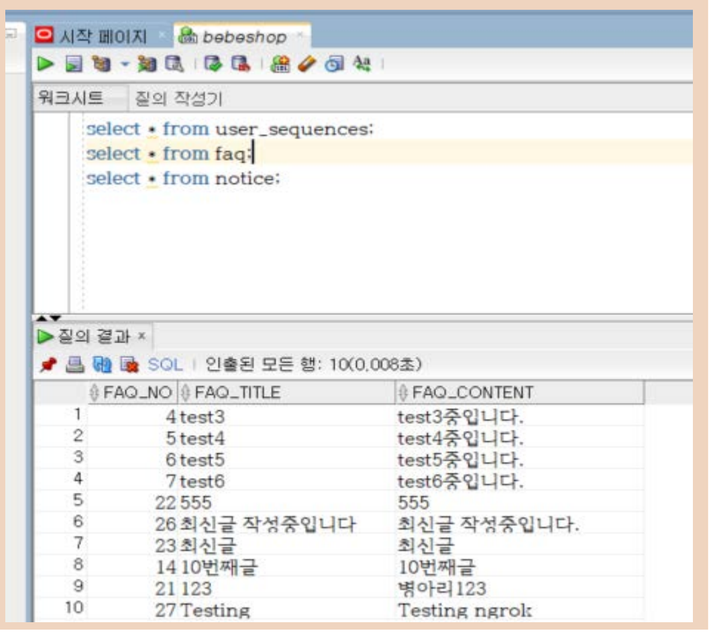
```java
public class BoardDaoImpl implements BoardDao {

    // @Inject는 객체타입이 일치하는 객체타입을 자동으로 주입한다.
    @Inject
    private SqlSession sqlSession;

    /* private static String namespace = "com.bebe.spring.mapper.Bo

    @Override
    public List<BoardVO> noticeList(Criteria cri) throws Exception
        return sqlSession.selectList("Board.noticeList");
        /* return sql.Session.selectList(namespace + ".list"); */
    }

    @Override
    public List<BoardVO> faqList() throws Exception {
        return sqlSession.selectList("Board.faqList");
    }

    @Override
    public int noticeListCount(Criteria cri) throws Exception {
        return sqlSession.selectOne("Board.noticeListCount");
    }

    @Override
    public BoardVO readNotice(int noticeNo) throws Exception {
//      xml을 실행시키는 역할로 Board에 있는 readNotice를 실행해라
        return sqlSession.selectOne("Board.readNotice");
    }

    @Override
    public BoardVO readFaq(int faqNo) throws Exception {
        return sqlSession.selectOne("Board.readFaq");
    }

    @Override
    public void writeNotice(BoardVO boardVO) throws Exception {
        sqlSession.insert("Board.writeNotice");
    }
```

```java
@Override
public void writeFaq(BoardVO boardVO) throws Exception {
    sqlSession.insert("Board.writeFaq");
}

@Override
public List<BoardVO> category(String noticeCategory) throws Exception {
    return sqlSession.selectList("Board.category");
}

@Override
public void deleteNotice(int noticeNo) throws Exception {
    sqlSession.delete("Board.deleteNotice");
}

@Override
public void deleteFaq(int faqNo) throws Exception {
    sqlSession.delete("Board.deleteFaq");
}

@Override
public void updateNotice(BoardVO boardVO) throws Exception {
    sqlSession.update("Board.updateNotice");
}

@Override
public void updateFaq(BoardVO boardVO) throws Exception {
    sqlSession.update("Board.updateFaq");
}

@Override
public int rnumCount() throws Exception{
    return sqlSession.selectOne("Board.rnumCount");
}
```

Oracle 테이블


Oracle SQL Developer : bebeshop
select * from user_sequences;
select * from faq;
select * from notice;
질의 결과
SQL | 인출된 모든 행: 27(0.021초)
NOTICE_NO NOTICE_CATEGORY NOTICE_TITLE NOTICE_CONTENT NOTICE_DATE
3 고객안내 test3 test3중입니다. 21/07/29
4 타임세일 test4 test4중입니다. 21/07/29
5 타임세일 test5 test5중입니다. 21/07/29
6 타임세일 test6 test6중입니다. 21/07/29
7 이벤트 test7 test7중입니다. 21/07/29
16 공지사항 (null) (null) 21/07/29
2 고객안내 test2 test2중입니다. 21/07/29
8 이벤트 test8 test8중입니다. 21/07/29
9 이벤트 test9 test9중입니다. 21/07/29
10 이벤트 test10 test10중입니다. 21/07/29
11 문의사항 test11 test11중입니다. 21/07/29
12 일반문의 test12 test12중입니다. 21/07/29
13 이벤트 test13 test13중입니다. 21/07/29
14 타임세일 test14 test14중입니다. 21/07/29
15 고객안내 고객센터 안내입니다 고객센터안내 21/07/29
17 공지사항 (null) (null) 21/07/29
18 공지사항 (null) (null) 21/07/29
19 공지사항 (null) (null) 21/07/29


select * from user_sequences;
select * from faq;
select * from notice;
질의 결과
SQL | 인출된 모든 행: 7(0.011초)
SEQUENCE_NAME MIN_VALUE MAX_VALUE INCREMENT_BY CYCLE_FLAG ORDER_FLAG CACHE_SIZE LAST_NUMBER
1 FAQ_SQ 1 1000 1 Y N 0 28
2 NEXTNO_ADMIN 1 99999 1 Y N 0 61
3 NEXTNO_ANSWER 1 99999 1 Y N 0 44
4 NEXTNO_QUESTION 1 99999 1 Y N 0 44
5 NO 1 1000 1 Y N 0 29
6 OR_NO 1 99999 1 Y N 0 104
7 SEQ_AI 1 99999 1 Y N 0 42


시작 페이지
bebeshop
워크시트
질의 작성기
select * from user_sequences;
select * from faq;
select * from notice;
질의 결과
SQL | 인출된 모든 행: 10(0.008초)
FAQ_NO FAQ_TITLE FAQ_CONTENT
4 test3 test3중입니다.
5 test4 test4중입니다.
6 test5 test5중입니다.
7 test6 test6중입니다.
22 555 555
26 최신글 작성중입니다 최신글 작성중입니다.
23 최신글 최신글
14 10번째글 10번째글
21 123 벙아리123
27 Testing Testing ngrok

Process[xml] (Oracle 구문정리)

목록, 세부내용 [xml]

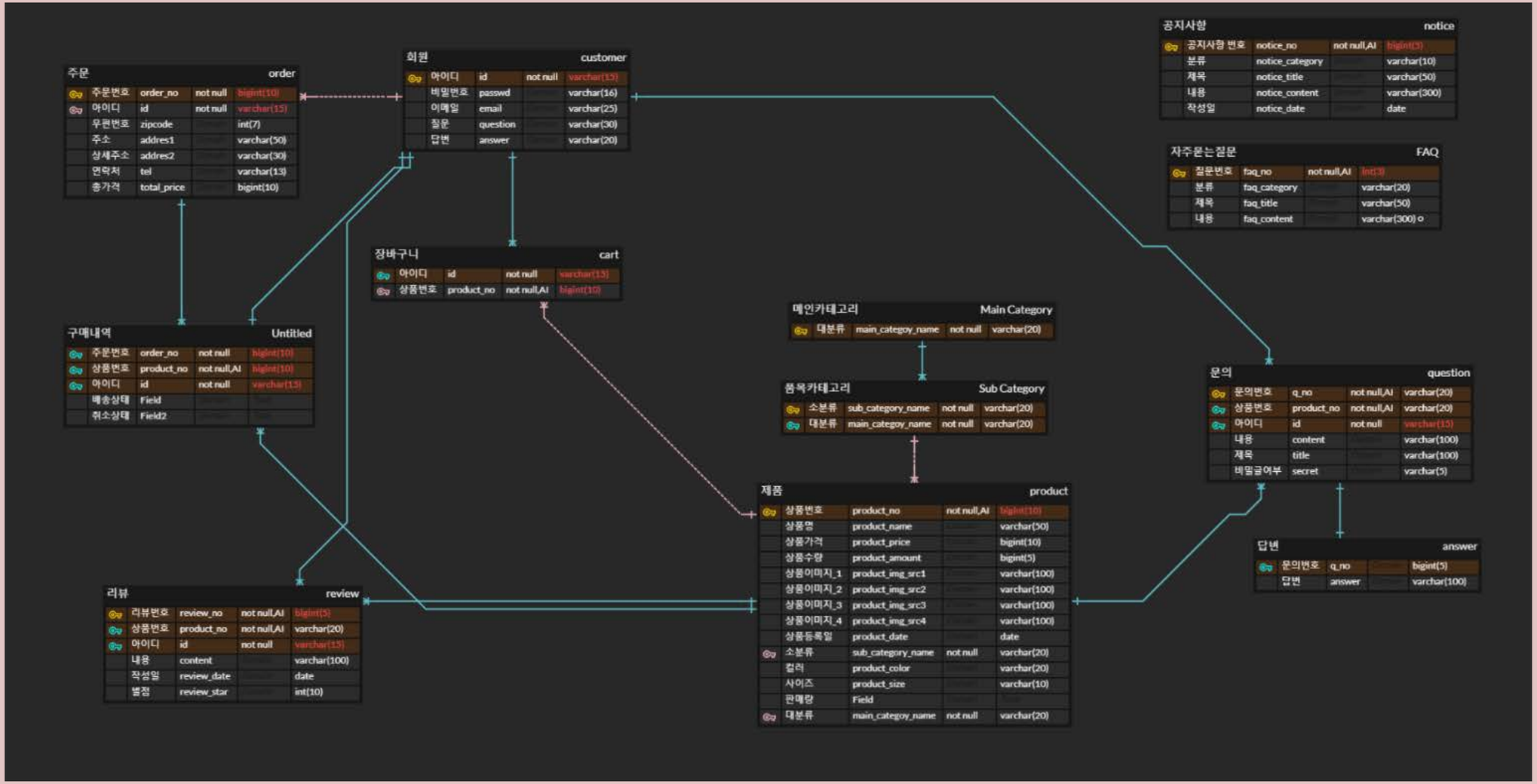
```xml
<select id="noticeList"
    resultType="com.bebe.spring.board.vo.BoardVO">
    <!-- SELECT * FROM notice ORDER BY notice_no desc -->
    SELECT * FROM (
    SELECT a.*, rownum rnum
    FROM (
    SELECT * FROM notice
    WHERE notice_no > 0
    ORDER BY notice_no DESC) a
    WHERE ((#{page} * #{perPageNum}) + 1 ) > rownum)
    WHERE rnum >= (((#{page}-1)
    * #{perPageNum}) + 1)
    ORDER BY notice_no DESC, notice_date DESC
</select>

<select id="faqList"
    resultType="com.bebe.spring.board.vo.BoardVO">
    select rownum as rnum, faq_no, faq_title, faq_content from faq order by faq_no  desc
</select>

<select id="readNotice"
    resultType="com.bebe.spring.board.vo.BoardVO">
    SELECT * FROM notice WHERE notice_no=#{noticeNo}
</select>

<select id="readFaq"
    resultType="com.bebe.spring.board.vo.BoardVO">
    SELECT * FROM faq WHERE faq_no=#{faqNo}
</select>
```

```xml
<select id="rnumCount" resultType="int">
    SELECT MAX(rownum) FROM faq
</select>
```

등록, 수정, 삭제 [xml]

```xml
<select id="writeNotice">
    INSERT INTO notice VALUES(no.NEXTVAL,
    #{noticeCategory},
    #{noticeTitle}, #{noticeContent}, SYSDATE)
</select>

<select id="writeFaq">
    INSERT INTO faq VALUES(faq_sq.NEXTVAL,#{faqTitle},
    #{faqContent})
</select>

<select id="category"
    resultType="com.bebe.spring.board.vo.BoardVO">
    SELECT notice_category FROM notice group by notice_category
</select>

<select id="deleteNotice">
    delete FROM notice WHERE notice_no=#{noticeNo}
</select>

<select id="deleteFaq">
    DELETE FROM faq WHERE faq_no=#{faqNo}
</select>

<select id="updateNotice">
    UPDATE notice SET notice_title = #{noticeTitle},
    notice_content = #{noticeContent},
    notice_date = SYSDATE WHERE
    notice_no = #{noticeNo}
</select>

<select id="updateFaq">
    UPDATE faq SET faq_title = #{faqTitle},
    faq_content =
    #{faqContent}
    WHERE faq_no = #{faqNo}
</select>

<select id="noticeListCount" resultType="int">
    <!-- SELECT * FROM ( SELECT ROWNUM num, n.* FROM ( SELECT *
        ORDER BY notice_no DESC ) n ) WHERE num between #{page}
    SELECT count(notice_no) FROM notice WHERE notice_no > 0
```

#인생맛집 ERD


주문 order
주문번호 order_no not null bigint(10)
아이디 id not null varchar(15)
우편번호 zipcode int(7)
주소 addres1 varchar(50)
상세주소 addres2 varchar(30)
연락처 tel varchar(13)
총가격 total_price bigint(10)
회원 customer
아이디 id not null varchar(15)
비밀번호 passwd varchar(16)
이메일 email varchar(25)
질문 question varchar(30)
답변 answer varchar(20)
공지사항 notice
공지사항 번호 notice_no not null,AI bigint(3)
분류 notice_category varchar(10)
제목 notice_title varchar(50)
내용 notice_content varchar(300)
작성일 notice_date date
자주묻는질문 FAQ
질문번호 faq_no not null,AI int(3)
분류 faq_category varchar(20)
제목 faq_title varchar(50)
내용 faq_content varchar(300) o
장바구니 cart
아이디 id not null varchar(15)
상품번호 product_no not null,AI bigint(10)
구매내역 Untitled
주문번호 order_no not null bigint(10)
상품번호 product_no not null,AI bigint(10)
아이디 id not null varchar(15)
배송상태 Field
취소상태 Field2
메인카테고리 Main Category
대분류 main_categoy_name not null varchar(20)
품목카테고리 Sub Category
소분류 sub_category_name not null varchar(20)
대분류 main_categoy_name not null varchar(20)
문의 question
문의번호 q_no not null,AI varchar(20)
상품번호 product_no not null,AI varchar(20)
아이디 id not null varchar(15)
내용 content varchar(100)
제목 title varchar(100)
비밀글여부 secret varchar(5)
답변 answer
문의번호 q_no bigint(5)
답변 answer varchar(100)
제품 product
상품번호 product_no not null,AI bigint(10)
상품명 product_name varchar(50)
상품가격 product_price bigint(10)
상품수량 product_amount bigint(5)
상품이미지_1 product_img_src1 varchar(100)
상품이미지_2 product_img_src2 varchar(100)
상품이미지_3 product_img_src3 varchar(100)
상품이미지_4 product_img_src4 varchar(100)
상품등록일 product_date date
소분류 sub_category_name not null varchar(20)
컬러 product_color varchar(20)
사이즈 product_size varchar(10)
판매량 Field
대분류 main_categoy_name not null varchar(20)
리뷰 review
리뷰번호 review_no not null,AI bigint(5)
상품번호 product_no not null,AI varchar(20)
아이디 id not null varchar(15)
내용 content varchar(100)
작성일 review_date date
별점 review_star int(10)