

# 프로젝트 문서

## Sabang - DB 정의서

2019-04-16

..

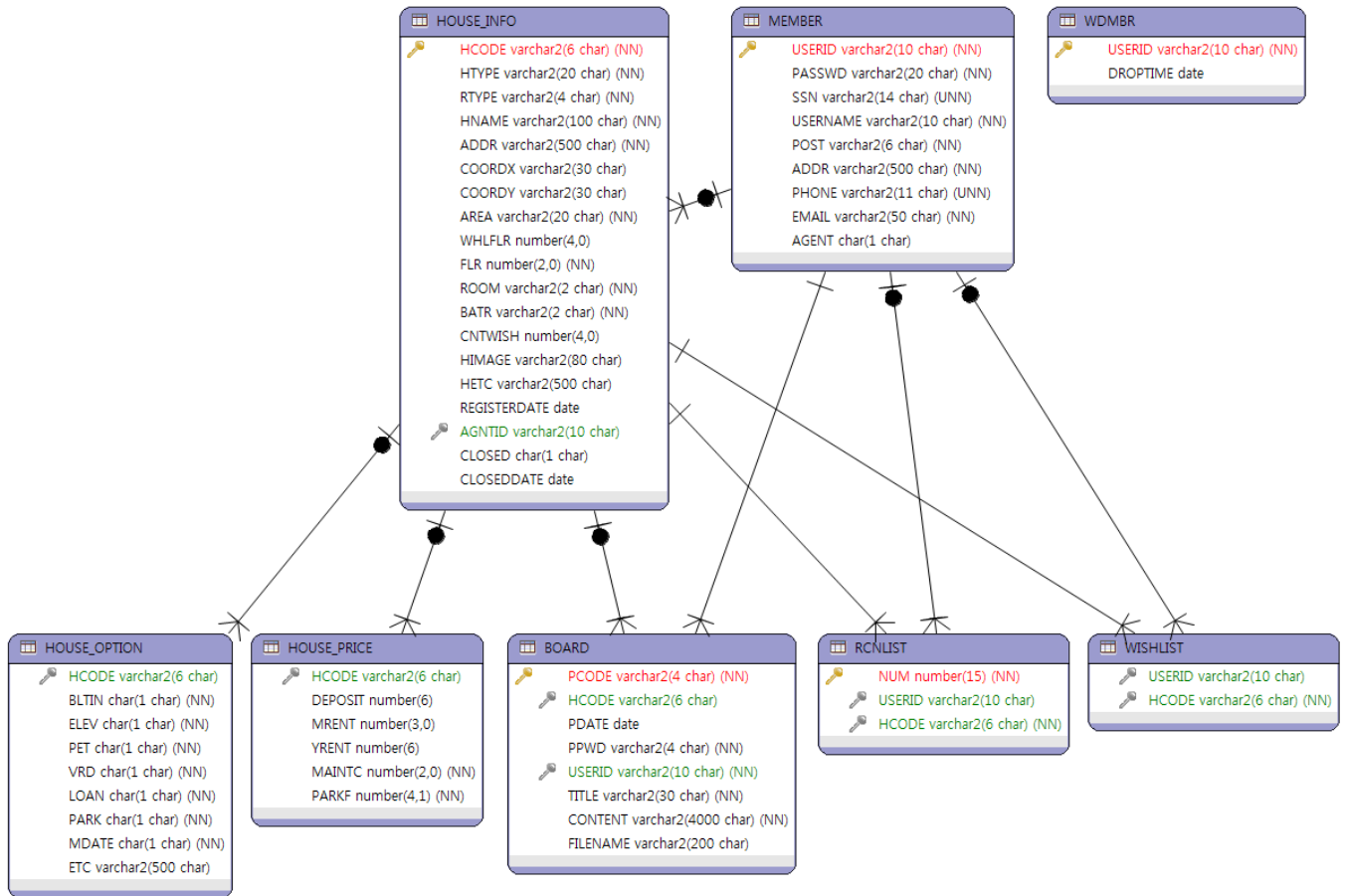
팀 명 : 1 조 (Sabang)

민희정 박태서 함효정

# 목차

<b>1. ER Diagram .....</b>	<b>3</b>
<b>2. Table List .....</b>	<b>4</b>
<b>3. 테이블 스크립트 .....</b>	<b>5</b>
i. MEMBER 테이블 .....	5
ii. WDMBR 테이블 .....	6
iii. BOARD 테이블 .....	7
iv. HOUSE_INFO 테이블 .....	8
v. HOUSE_OPTION 테이블 .....	9
vi. HOUSE_PRICE 테이블 .....	10
vii. RCNLIST 테이블 .....	11
viii. WISHLIST 테이블 .....	12
<b>4. SQL 정의서 .....</b>	<b>13</b>
i. MEMBER Mapper .....	13
ii. BOARD Mapper .....	14
iii. HOUSE Mapper .....	15

# 1. ER Diagram



## 2. Table List

테이블명	테이블 리스트	작성일	2019.04.16
DB	Oracle	작성자	민희정, 박태서 함효정

테이블명	설명
<b>MEMBER</b>	공인중개사와 일반회원의 정보를 저장하고 있는 테이블 네이버 회원일 경우 네이버에서 넘겨주는 unique id, name, email 정보로 저장
<b>WDMBR</b>	탈퇴 회원의 정보를 저장하고있는 테이블로 탈퇴 날짜와 탈퇴회원의 아이디를 저장하고 있다.
<b>BOARD</b>	매물과 관련된 문의 사항을 저장하고있는 테이블로, 문의글코드와 매물코드, 포스팅 날짜, 포스트 내용 등을 저장하고 있다.
<b>HOUSE_INFO</b>	매물과 관련된 일반적인 정보를 저장하고 있는 테이블로 매물의 가격 / 부가 옵션에 대한 정보를 제외한 매물명, 매물설명, 매물 주소, 등록일자 등등을 가지고 있다.
<b>HOUSE_OPTION</b>	매물과 관련된 부가정보를 가지고 있는 테이블로, 엘리베이터 보유여부, 애완동물 가능 여부 등과 같은 부가정보와 기타입력란에 들어가는 설명을 저장하고 있다.
<b>HOUSE_PRICE</b>	매물과 관련된 가격정보를 가지고 있는 테이블로, 전/월세 가격정보, 주차비, 관리비 등과 같은 가격정보를 가지고 있다.
<b>RCNLIST</b>	태서
<b>WISHLIST</b>	태서

### 3. 테이블 스크립트

#### i. MEMBER 테이블

테이블 스크립트	제정일자	2019.04.16
	REV.NO	1
	PAGE	1/1

테이블명	MEMBER			관련업무 DB 계정	회원 정보저장		
					sabang		
컬럼명	제약조건	NULL 여부	FK 테이블	FK 컬럼	DATA 타입		비고
USERID	PK	NOT NULL			VARCHAR2	10	아이디
PASSWD		NOT NULL			VARCHAR2	20	비밀번호
SSN	UK	NOT NULL			VARCHAR2	14	주민등록번호
USERNAME		NOT NULL			VARCHAR2	10	이름
POST		NOT NULL			VARCHAR2	6	우편번호
ADDR		NOT NULL			VARCHAR2	500	도로명주소
PHONE	UK	NOT NULL			VARCHAR2	11	핸드폰번호
EMAIL		NOT NULL			VARCHAR2	50	이메일주소
AGENT		DEFAULT 'N'			CHAR	1	공인중개사 여부

## ii. WDMBR 테이블

테이블 스크립트	제정일자	2019.04.16
	REV.NO	1
	PAGE	1/1

테이블명	WDMBR			관련업무 DB 계정	탈퇴회원 정보저장		
					sabang		
컬럼명	제약조건	NULL 여부	FK 테이블	FK 컬럼	DATA 타입		비고
USERID	PK	NOT NULL			VARCHAR2	10	탈퇴회원 아이디
DROPTIME		DEFAULT SYSDATE			DATE		탈퇴 시간

## iii. BOARD 테이블

테이블 스크립트	제정일자	2019.04.16
	REV.NO	1
	PAGE	1/1

테이블명	BOARD			관련업무 DB 계정	매물관련 문의글 정보 저장		
					sabang		
컬럼명	제약조건	NULL 여부	FK 테이블	FK 컬럼	DATA 타입		비고
PCODE	PK	NOT NULL			VARCHAR2	4	문의글 코드
HCODE			HOUSE_INFO	HCODE	VARCHAR2	6	문의 매물 코드
PDATE					DATE		문의글 작성 날짜
PPWD		NOT NULL			VARCHAR2	4	문의글 비밀번호
USERID		NOT NULL	MEMBER	USERID	VARCHAR2	10	글쓴이 아이디
TITLE		NOT NULL			VARCHAR2	30	문의글 제목
CONTENT		NOT NULL			VARCHAR2	4000	문의글 내용 저장
FILENAME					VARCHAR2	200	첨부 파일명

## iv. HOUSE\_INFO 테이블

테이블 스크립트	제정일자	2019.04.16
	REV.NO	1
	PAGE	1/1

테이블명(영문)	HOUSE_INFO			관련업무 DB 계정	매물 일반 정보 관리		
					sabang		
영문컬럼(영문)	제약조건	NULL 여부	FK 테이블	FK 컬럼	DATA 타입		비고
HCODE	PK				VARCHAR2	6	매물코드
HTYPE	CHECK				VARCHAR2	20	매물타입
RTYPE		NOT NULL			VARCHAR2	4	전세/월세
HNAME		NOT NULL			VARCHAR2	100	매물명
ADDR		NOT NULL			VARCHAR2	500	매물 주소
COORDX					VARCHAR2	30	좌표 X
COORDY					VARCHAR2	30	좌표 Y
AREA		NOT NULL			VARCHAR2	20	면적
WHLFLR					NUMBER	4,0	전체층
FLR		NOT NULL			NUMBER	2,0	매물층
ROOM		NOT NULL			VARCHAR2	2	방개수
BATR		NOT NULL			VARCHAR2	2	화장실 개수
CNTWISH					NUMBER	4,0	좋아요 수
HIMAGE					VARCHAR2	80	매물 이미지명
HETC					VARCHAR2	500	기타사항
REGISTERDATE					DATE		등록일자
AGNTID	FK		MEMBER	USERID	VARCHAR2	10	담당 중개인아이디
CLOSED					CHAR	1	중개완료여부
CLOSEDDATE					DATE		중개완료날짜



## v. HOUSE\_OPTION 테이블

테이블 스크립트	제정일자	2019.04.16
	REV.NO	1
	PAGE	1/1

테이블명(영문)	HOUSE_OPTION			관련업무 DB 계정	매물 부가 정보 관리		
					sabang		
영문컬럼(영문)	제약조건	NULL 여부	FK 테이블	FK 컬럼	DATA 타입		비고
HCODE	FK		HOUSE_INFO	HCODE	VARCHAR2	20	매물코드
BLTIN	CHECK	NOT NULL			CHAR	1	발트인 여부
ELEV	CHECK	NOT NULL			CHAR	1	승강기 여부
PET	CHECK	NOT NULL			CHAR	1	애완동물 가능 여부
VRD	CHECK	NOT NULL			CHAR	1	발코니 여부
LOAN	CHECK	NOT NULL			CHAR	1	전세대출 가능여부
PARK	CHECK	NOT NULL			CHAR	1	주차 가능 여부
MDATE	CHECK	NOT NULL			CHAR	1	이주 일자 여부
ETC					VARCHAR2	500	기타 사항

## vi. HOUSE\_PRICE 테이블

테이블 스크립트	제정일자	2019.04.16
	REV.NO	1
	PAGE	1/1

테이블명(영문)	HOUSE_PRICE			관련업무 DB 계정	매물 가격 정보 관리		
					sabang		
영문컬럼(영문)	제약조건	NULL 여부	FK 테이블	FK 컬럼	DATA 타입		비고
HCODE	FK		HOUSE_INFO	HCODE	VARCHAR2	20	매물코드
DEPOSIT					NUMBER	6,0	보증금 가격
MRENT					NUMBER	3,0	월세 가격
YRENT					NUMBER	6,0	전세 가격
MAINTC		NOT NULL			NUMBER	2,0	관리비
PARKF		NOT NULL			NUMBER	4,1	주차비

## vii. RCNLIST 테이블

테이블 스크립트	제정일자	2019.04.16
	REV.NO	1
	PAGE	1/1

테이블명(영문)	RCNLIST			관련업무 DB 계정	회원의 최근 본 방을 저장한다.		
					sabang		
영문컬럼(영문)	제약조건	NULL 여부	FK 테이블	FK 컬럼	DATA 타입		비고
NUM					NUMBER	15	저장시간
USERID	FK		MEMBER	USERID	VARCHAR2	20	회원 아이디
HCODE	FK	NOT NULL	HOUSE_INFO	HCODE	VARCHAR2	20	매물번호

## viii. WISHLIST 테이블

테이블 스크립트	제정일자	2019.04.16
	REV.NO	1
	PAGE	

테이블명(영문)	WISHLIST			관련업무 DB 계정	회원의 찜한 방을 저장한다.		
					sabang		
영문컬럼(영문)	제약조건	NULL 여부	FK 테이블	FK 컬럼	DATA 타입		비고
USERID			MEMBER	USERID	VARCHAR2	10	회원 아이디
HCODE		NOT_NULL	HOUSE_INFO	HCODE	VARCHAR2	6	매물번호

## 4. SQL 정의서

### i. MEMBER Mapper

MemberMapper		멤버 관련 정보 관리
ID	SQL	Description
signMbr	insert into member (userid,passwd, ssn, username,post,addr, phone, email,agent) values( #{userid},#{passwd},#{ssn},#{username},#{post},#{addr}, #{phone},#{email},#{agent})	회원가입
idCheck	select count(*) from member where userid = #{userid}	아이디 중복 체크
ssnCheck	select count(*) from member where ssn = #{ssn}	주민등록번호 중복체크
phoneCheck	select count(*) from member where phone = #{phone}	핸드폰번호 중복체크
overDay	select TO_CHAR(DROPTIME + 1,'YYYYMMDDHH24') from wdMbr where userid = #{userid}	탈퇴일로부터 24시 간 이후를 조회
login	select userid,passwd, ssn, username,post,addr, phone, email, agent from member where userid = #{userid} and passwd = #{passwd}	로그인
delMbrId	delete agent where agntid = #{userid}	회원 탈퇴
checkMbrPw	select userid,passwd, ssn, username,post,addr, phone, email, agent from member where passwd = #{passwd}	회원 탈퇴 이전 비 밀번호 확인
addOutMbr	insert into wdmbr(userid,droptime) values(#{userid},sysdate)	탈퇴 회원 정보 저 장
naverUser	select count(*) from member where userid = #{uniqId}	네이버로 로그인시 유저 존재여부 확인
naverSignUp	insert into member (userid, passwd, ssn, username, post, addr, phone, email) values (#{uniqId}, 'naver', #{uniqId}, #{name}, 'na', 'na', #{uniqId}, #{email})	네이버로 로그인 시 사방에 정보가 없다면 회원가입처리
getNaverUser	select userid, passwd, ssn, username, post, addr, phone, email, agent from member where userid = #{uniqId}	네이버로 로그인

## ii. BOARD Mapper

BoardMapper		Board 관련 정보 관리
ID	SQL	Description
boardList	select pcode, TO_CHAR(pdate, 'YYYY.MM.DD HH:MI') pdate, ppwd, title, content, filename, hcode, userid from board where hcode = #{hcode}	매물 코드에 따른 문의글 조회
deleteBoard	delete from board where pcode = #{pcode}	문의글 삭제
updateBoard	update board set title = #{title}, content = #{content}, pdate = sysdate where pcode = #{pcode}	문의글 수정

## iii. HOUSE Mapper

HouseMapper		매물 관련 정보 관리
ID	SQL	Description
newItemCount	select to_char(max(registerdate)-7, 'yyyymmdd') from house_info	신매물 개수
retrieveNewItems	select hcode, htype, rtype, hname, area, coordx, coordy, flr, hetc, himage, deposit, mrent, yrent, maintc from house_info join house_price using(hcode) where registerdate >= #{maxSeven}	신매물 리스트
retrieveHotItems	select hcode, htype, rtype, hname, area, coordx, coordy, flr, hetc, himage, deposit, mrent, yrent, maintc from house_info join house_price using(hcode) where cntwish >= (select AVG(cntwish) from house_info)	핫매물 리스트
searchList	select hcode, htype, rtype, hname, himage, area, flr, hetc, deposit, mrent, yrent, maintc from house_info join house_price using(hcode) where addr like '%'  #{search}  '%' or hname like '%'  #{search}  '%'	검색 리스트
totalListBySearch	select count(*) from house_info join house_price using(hcode) where addr like '%'  #{search}  '%' or hname like '%'  #{search}  '%'	검색 리스트 총 개수
listByFilter	<pre> select hcode, htype, rtype, hname, himage, area, flr, hetc, himage, himage, deposit, mrent, yrent, maintc from house_info join house_price using(hcode) &lt;choose&gt;   &lt;when test="!(htype.isEmpty())"&gt;     where htype in     &lt;foreach item="item" collection="htype" open="(" separator=","     close=")"&gt;       #{item}     &lt;/foreach&gt;     &lt;if test="!(rtype.isEmpty())"&gt;       and rtype in       &lt;foreach item="item" collection="rtype" open="(" separator=","       close=")"&gt;         #{item}       &lt;/foreach&gt;     &lt;/if&gt;     &lt;if test="!(maintc.isEmpty())"&gt;       and maintc &lt;       &lt;foreach item="item" collection="maintc"&gt;         #{item}       &lt;/foreach&gt;     &lt;/if&gt;     &lt;if test="!(mrent.isEmpty())"&gt;       and mrent between       &lt;foreach item="item" collection="mrent" separator="and"&gt;         #{item}       &lt;/foreach&gt;     &lt;/if&gt;     &lt;if test="!(yrent.isEmpty())"&gt;       and yrent &lt;       &lt;foreach item="item" collection="yrent"&gt;         #{item}       &lt;/foreach&gt;     &lt;/if&gt;   &lt;/when&gt;   &lt;when test="!(rtype.isEmpty())"&gt;     where rtype in     &lt;foreach item="item" collection="rtype" open="(" separator=","     close=")"&gt;       #{item}     &lt;/foreach&gt;     &lt;if test="!(htype.isEmpty())"&gt;       and htype in       &lt;foreach item="item" collection="htype" open="(" separator=","       close=")"&gt;         #{item}       &lt;/foreach&gt;     &lt;/if&gt;     &lt;if test="!(maintc.isEmpty())"&gt;       and maintc &lt; </pre>	필터에 의한 리스트

```

    <foreach item="item" collection="maintc">
        #{item}
    </foreach>
</if>
<if test="!(mrent.isEmpty())">
    and mrent between
    <foreach item="item" collection="mrent" separator="and">
        #{item}
    </foreach>
</if>
<if test="!(yrent.isEmpty())">
    and yrent &lt;
    <foreach item="item" collection="yrent">
        #{item}
    </foreach>
</if>
</when>
<when test="!(maintc.isEmpty())">
    where maintc &lt;
    <foreach item="item" collection="maintc">
        #{item}
    </foreach>
    <if test="!(htype.isEmpty())">
        and htype in
        <foreach item="item" collection="htype" open="(" separator=","
            close=")">
            #{item}
        </foreach>
    </if>
    <if test="!(rtype.isEmpty())">
        and rtype in
        <foreach item="item" collection="rtype" open="(" separator=","
            close=")">
            #{item}
        </foreach>
    </if>
    <if test="!(mrent.isEmpty())">
        and mrent between
        <foreach item="item" collection="mrent" separator="and">
            #{item}
        </foreach>
    </if>
    <if test="!(yrent.isEmpty())">
        and yrent &lt;
        <foreach item="item" collection="yrent">
            #{item}
        </foreach>
    </if>
</when>
<when test="!(mrent.isEmpty())">
    where mrent between
    <foreach item="item" collection="mrent" separator="and">
        #{item}
    </foreach>
    <if test="!(htype.isEmpty())">
        and htype in
        <foreach item="item" collection="htype" open="(" separator=","
            close=")">
            #{item}
        </foreach>
    </if>
    <if test="!(rtype.isEmpty())">
        and rtype in
        <foreach item="item" collection="rtype" open="(" separator=","
            close=")">
            #{item}
        </foreach>
    </if>
    <if test="!(maintc.isEmpty())">
        and maintc &lt;
        <foreach item="item" collection="maintc">
            #{item}
        </foreach>
    </if>

```



	<pre> &lt;if test="!(yrent.isEmpty())"&gt;   and yrent &amp;lt;   &lt;foreach item="item" collection="yrent"&gt;     #{item}   &lt;/foreach&gt; &lt;/if&gt; &lt;/when&gt; &lt;when test="!(yrent.isEmpty())"&gt;   where yrent &amp;lt;   &lt;foreach item="item" collection="yrent"&gt;     #{item}   &lt;/foreach&gt; &lt;if test="!(htype.isEmpty())"&gt;   and htype in   &lt;foreach item="item" collection="htype" open="(" separator=","   close=")"&gt;     #{item}   &lt;/foreach&gt; &lt;/if&gt; &lt;if test="!(rtype.isEmpty())"&gt;   and rtype in   &lt;foreach item="item" collection="rtype" open="(" separator=","   close=")"&gt;     #{item}   &lt;/foreach&gt; &lt;/if&gt; &lt;if test="!(maintc.isEmpty())"&gt;   and maintc &amp;lt;   &lt;foreach item="item" collection="maintc"&gt;     #{item}   &lt;/foreach&gt; &lt;/if&gt; &lt;if test="!(mrent.isEmpty())"&gt;   and mrent between   &lt;foreach item="item" collection="mrent" separator="and"&gt;     #{item}   &lt;/foreach&gt; &lt;/if&gt; &lt;/when&gt; &lt;/choose&gt; </pre>	
totalListByFilter	<pre> select count(*) from house_info join house_price using(hcode) &lt;choose&gt;   &lt;when test="!(htype.isEmpty())"&gt;     where htype in     &lt;foreach item="item" collection="htype" open="(" separator=","     close=")"&gt;       #{item}     &lt;/foreach&gt;   &lt;if test="!(rtype.isEmpty())"&gt;     and rtype in     &lt;foreach item="item" collection="rtype" open="(" separator=","     close=")"&gt;       #{item}     &lt;/foreach&gt;   &lt;/if&gt;   &lt;if test="!(maintc.isEmpty())"&gt;     and maintc &amp;lt;     &lt;foreach item="item" collection="maintc"&gt;       #{item}     &lt;/foreach&gt;   &lt;/if&gt;   &lt;if test="!(mrent.isEmpty())"&gt;     and mrent between     &lt;foreach item="item" collection="mrent" separator="and"&gt;       #{item}     &lt;/foreach&gt;   &lt;/if&gt;   &lt;if test="!(yrent.isEmpty())"&gt;     and yrent &amp;lt;     &lt;foreach item="item" collection="yrent"&gt;       #{item}     &lt;/foreach&gt;   &lt;/if&gt; &lt;/when&gt; </pre>	필터에 의한 리스트 개수

```

<when test="!(rtype.isEmpty())">
  where rtype in
  <foreach item="item" collection="rtype" open="(" separator=","
  close=")">
    #{item}
  </foreach>
  <if test="!(htype.isEmpty())">
    and htype in
    <foreach item="item" collection="htype" open="(" separator=","
    close=")">
      #{item}
    </foreach>
  </if>
  <if test="!(maintc.isEmpty())">
    and maintc &lt;
    <foreach item="item" collection="maintc">
      #{item}
    </foreach>
  </if>
  <if test="!(mrent.isEmpty())">
    and mrent between
    <foreach item="item" collection="mrent" separator="and">
      #{item}
    </foreach>
  </if>
  <if test="!(yrent.isEmpty())">
    and yrent &lt;
    <foreach item="item" collection="yrent">
      #{item}
    </foreach>
  </if>
</when>
<when test="!(maintc.isEmpty())">
  where maintc &lt;
  <foreach item="item" collection="maintc">
    #{item}
  </foreach>
  <if test="!(htype.isEmpty())">
    and htype in
    <foreach item="item" collection="htype" open="(" separator=","
    close=")">
      #{item}
    </foreach>
  </if>
  <if test="!(rtype.isEmpty())">
    and rtype in
    <foreach item="item" collection="rtype" open="(" separator=","
    close=")">
      #{item}
    </foreach>
  </if>
  <if test="!(mrent.isEmpty())">
    and mrent between
    <foreach item="item" collection="mrent" separator="and">
      #{item}
    </foreach>
  </if>
  <if test="!(yrent.isEmpty())">
    and yrent &lt;
    <foreach item="item" collection="yrent">
      #{item}
    </foreach>
  </if>
</when>
<when test="!(mrent.isEmpty())">
  where mrent between
  <foreach item="item" collection="mrent" separator="and">
    #{item}
  </foreach>
  <if test="!(htype.isEmpty())">
    and htype in
    <foreach item="item" collection="htype" open="(" separator=","
    close=")">
      #{item}
    </foreach>
  </if>

```

# Sabang DB 정의서

	<pre> &lt;/foreach&gt; &lt;/if&gt; &lt;if test="!(rtype.isEmpty())"&gt;     and rtype in     &lt;foreach item="item" collection="rtype" open="(" separator=","     close=")"&gt;         #{item}     &lt;/foreach&gt; &lt;/if&gt; &lt;if test="!(maintc.isEmpty())"&gt;     and maintc &amp;lt;     &lt;foreach item="item" collection="maintc"&gt;         #{item}     &lt;/foreach&gt; &lt;/if&gt; &lt;if test="!(yrent.isEmpty())"&gt;     and yrent &amp;lt;     &lt;foreach item="item" collection="yrent"&gt;         #{item}     &lt;/foreach&gt; &lt;/if&gt; &lt;/when&gt; &lt;when test="!(yrent.isEmpty())"&gt;     where yrent &amp;lt;     &lt;foreach item="item" collection="yrent"&gt;         #{item}     &lt;/foreach&gt; &lt;if test="!(htype.isEmpty())"&gt;     and htype in     &lt;foreach item="item" collection="htype" open="(" separator=","     close=")"&gt;         #{item}     &lt;/foreach&gt; &lt;/if&gt; &lt;if test="!(rtype.isEmpty())"&gt;     and rtype in     &lt;foreach item="item" collection="rtype" open="(" separator=","     close=")"&gt;         #{item}     &lt;/foreach&gt; &lt;/if&gt; &lt;if test="!(maintc.isEmpty())"&gt;     and maintc &amp;lt;     &lt;foreach item="item" collection="maintc"&gt;         #{item}     &lt;/foreach&gt; &lt;/if&gt; &lt;if test="!(mrent.isEmpty())"&gt;     and mrent between     &lt;foreach item="item" collection="mrent" separator="and"&gt;         #{item}     &lt;/foreach&gt; &lt;/if&gt; &lt;/when&gt; &lt;/choose&gt; </pre>	
houseSoldByAgentCount	<pre> select count(*) count, TO_CHAR(closedDate, 'MM') closedDate from house_info where agntid = #{agntid} and closed='Y' GROUP BY TO_CHAR(closedDate, 'MM') </pre>	중개인 아이디 별 중개완료 매물 개수
houseByRegisterDate	<pre> select count(*) count, TO_CHAR(REGISTERDATE, 'MM') registerdate from house_info where agntid = #{agntid} GROUP BY TO_CHAR(REGISTERDATE, 'MM') </pre>	중개인 아이디 별 중개 매물 리스트
houseLikeByAgent	<pre> select hcode, rtype, addr, deposit, mrent, registerdate, himage, cntwish from house_info join house_price using(hcode) where agntid = #{agntid} and cntwish&gt;0 order by cntwish desc </pre>	중개인 아이디별 좋아요 많은 매물
houseSoldByAgent	<pre> select hcode, rtype, addr, deposit, mrent, registerdate, himage, TO_CHAR(closedDate, 'YY/MM/DD') closedDate from house_info join house_price using(hcode) where agntid = #{agntid} and house_info.closed = 'Y' </pre>	중개인 아이디별 중개완료 매물 리스트 리스트
houseByAgent	<pre> select hcode, rtype, addr, deposit, mrent, registerdate, himage from house_info join house_price using(hcode) where agntid = #{agntid} and house_info.closed = 'N' order by registerdate </pre>	중개인 아이디별 매 물 리스트

# Sabang DB 정의서

getLastCode	select max(hcode) from house_info where htype = #{htype}	매물등록시 매물 타입 선택 시 그 타입의 마지막 코드를 반환
houseRegister_option	<pre> insert into house_option (hcode, bltin, elev, pet, vrd, loan, park, mdate, etc) values ({hcode},     &lt;if test="bltin != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="bltin == 'Y'"&gt; #{bltin}, &lt;/if&gt;     &lt;if test="elev != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="elev == 'Y'"&gt; #{elev}, &lt;/if&gt;     &lt;if test="pet != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="pet == 'Y'"&gt; #{pet}, &lt;/if&gt;     &lt;if test="vrd != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="vrd == 'Y'"&gt; #{vrd}, &lt;/if&gt;     &lt;if test="loan != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="loan == 'Y'"&gt; #{loan}, &lt;/if&gt;     &lt;if test="park != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="park == 'Y'"&gt; #{park}, &lt;/if&gt;     &lt;if test="mdate != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="mdate == 'Y'"&gt; #{mdate}, &lt;/if&gt;     &lt;if test="etc != null"&gt; #{etc} &lt;/if&gt;     &lt;if test="etc == null"&gt; '' &lt;/if&gt; ) </pre>	house_option 테이블에 매물 등록
houseRegister_price	insert into house_price (hcode, deposit, mrent, yrent, maintc, parkf) values ({hcode}, #{deposit}, #{mrent}, #{yrent}, #{maintc}, #{parkf})	House_price에 매물 등록
houseRegister_info	insert into house_info (hcode, htype, rtype, hname, addr, coordx, coordy, area, whlflr, flr, room, batr, cntwish, himage, hetc, registerdate, agntid) values ({hcode}, #{htype}, #{rtype}, #{hname}, #{addr}, 'null', 'null', #{area}, #{whlflr}, #{flr}, #{room}, #{batr}, 0, #{himage}, #{hetc}, sysdate, #{agntid})	House_info에 매물 등록
houseUpdate_option	<pre> update house_option set bltin = &lt;if test="bltin != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="bltin == 'Y'"&gt; #{bltin}, &lt;/if&gt;     elev = &lt;if test="elev != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="elev == 'Y'"&gt; #{elev}, &lt;/if&gt;     pet = &lt;if test="pet != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="pet == 'Y'"&gt; #{pet}, &lt;/if&gt;     vrd = &lt;if test="vrd != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="vrd == 'Y'"&gt; #{vrd}, &lt;/if&gt;     loan = &lt;if test="loan != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="loan == 'Y'"&gt; #{loan}, &lt;/if&gt;     park = &lt;if test="park != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="park == 'Y'"&gt; #{park}, &lt;/if&gt;     mdate = &lt;if test="mdate != 'Y'"&gt; 'N', &lt;/if&gt;     &lt;if test="mdate == 'Y'"&gt; #{mdate}, &lt;/if&gt;     etc = &lt;if test="etc != null"&gt; #{etc} &lt;/if&gt;     &lt;if test="etc == null"&gt; '' &lt;/if&gt; where hcode = #{hcode} </pre>	house_option 테이블에 매물 수정
houseUpdate_price	update house_price set deposit = #{deposit}, mrent = #{mrent}, yrent = #{yrent}, maintc = #{maintc}, parkf = #{parkf} where hcode = #{hcode}	House_price 테이블에 매물 수정
houseUpdate_info	<pre> update house_info set rtype = #{rtype}, hname = #{hname}, addr = #{addr}, area = #{area}, whlflr = #{whlflr}, flr = #{flr}, room = #{room}, batr = #{batr}, &lt;if test="himage != null"&gt;     himage = #{himage}, &lt;/if&gt; hetc = #{hetc} where hcode = #{hcode} </pre>	House_info 테이블에 매물 수정
houseDel_info	<pre> delete from house_info where hcode in &lt;foreach item="item" collection="list" open="(" separator="," close=")"&gt;     #{item} &lt;/foreach&gt; </pre>	House_info 테이블에 매물 삭제
getCntWish	select cntwish from house_info where hcode = #{hcode}	매물별 좋아요 수 반환
updateCntWish	update house_info set cntwish = #{cntwish}+1 where hcode = #{hcode}	브라우저에서 찜하기 누르면 매물 좋아요 수 업데이트

## Sabang DB 정의서

duplicateHouseCheck	<pre>select count(*) from wishlist where hcode = #{hcode} and userid = #{userid}</pre>	브라우저에서 찜하기 누르면 이미 누른 매물인지 확인
---------------------	--	------------------------------