**Member**

**M\_NO, M\_ID, M\_PASSWORD, M\_NAME, M\_ADDRESS, M\_CONTACT, M\_EMAIL**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| member\_insert | INSERT INTO member VALUES  (member\_no\_seq.NEXTVAL, ?, ?, ?, ?, ?, ?); | 회원등록 |
| member\_readOnebyId | SELECT m\_no, m\_id, m\_password, m\_name, m\_address, m\_contact, m\_email FROM member WHERE m\_id=?; | 아이디로 회원 1명 검색 |
| member\_readOnebyNo | SELECT m\_no, m\_id, m\_password, m\_name, m\_address, m\_contact, m\_email FROM member WHERE m\_no=?; | 회원번호로 회원 1명 검색 |
| member\_readAll | SELECT m\_no, m\_id, m\_password, m\_name, m\_address, m\_contact, m\_email FROM member; | 회원전체검색 |
| member\_update | UPDATE member SET m\_id=?, m\_password=?, m\_name=?, m\_address=?, m\_contact=?, m\_email=? WHERE m\_no=?;  > 이 기능 회원이 본인 정보 수정할 때 쓰는데 화면에서 회원번호를 안보여주기 때문에 WHERE 조건을 pk인 회원아이디(m\_id)로 할께요  UPDATE member SET m\_id=?, m\_password=?, m\_name=?, m\_address=?, m\_contact=?, m\_email=? WHERE m\_id=?; | 회원수정 |
| member\_delete | DELETE FROM member WHERE m\_id=?; | 회원삭제  (회원 탈퇴 기능을 만들 경우) |
| member\_existed | select count(\*) cnt from member where m\_id=?  (존재하면 1, 존재하지 않으면 0) | 회원존재여부 |

**PRODUCT**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| product\_insert | INSERT INTO PRODUCT VALUES (PRODUCT\_P\_NO\_SEQ.nextval,?,?,?,?,?,?,?,?,?,?); | 상품등록 |
| product\_readAllMain | select p\_no,p\_name,p\_price,p\_image from product order by p\_no DESC;  > 초기화면(최신등록순 - Descending)  select p\_no,p\_name,p\_price,p\_image from product order by p\_name;  select p\_no,p\_name,p\_price,p\_image from product order by p\_price (ASC or DESC); | 상품리스트  (초기화면-최신등록순)  (이름순)  (가격순) |
| product\_readAllAdmin | SELECT \* FROM PRODUCT; | 상품리스트  [상품관리 페이지 전체 리스트에 사용될 메소드] |
| product\_readOne | select \* from product where p\_no=?; | 상품 번호에 해당하는 게시물 |
| product\_update | UPDATE product SET p\_category=?,br\_name=?,p\_name=?,p\_volume=?,p\_price=?,p\_image=?,p\_keyword=?,p\_des=?,p\_count=?,t\_name=? WHERE p\_no=?; | 상품 수정 in Admin |
| product\_delete | delete product where p\_no=?; | 상품 삭제 in Admin |
| product\_search | select p\_no,p\_name,p\_price,p\_image from product where p\_name like ‘%’||?||’%’ ; | 상품 이름으로 검색 |
| product\_survey\_match | 1)SELECT \* from product p  LEFT OUTER JOIN gender g  ON p.p\_category=g.g\_pcategory  LEFT OUTER JOIN survey s  ON s.s\_gender=g.g\_sgender  LEFT OUTER JOIN accord a  ON a.acd\_name IN (s.s\_fav1, s.s\_fav2, s.s\_fav3)  WHERE s.m\_id='jenkim49' and p.p\_keyword LIKE '%ROSE%' and p.p\_keyword LIKE '%SWEET%' and p.p\_keyword LIKE '%CITRUS%' and p.p\_keyword LIKE '%10대%';    2)  SELECT \* from product p  LEFT OUTER JOIN gender g  ON p.p\_category=g.g\_pcategory  LEFT OUTER JOIN survey s  ON s.s\_gender=g.g\_sgender  WHERE s.m\_id='jenkim49' and p.p\_keyword LIKE '%ROSE%' and p.p\_keyword LIKE '%SWEET%' and p.p\_keyword LIKE '%CITRUS%' and p.p\_keyword LIKE '%10대%'; | 설문조사결과매칭 |
| product\_recommend | select p\_no,p\_name,p\_price,p\_image from product where t\_name='?' order by p\_count; | 유사제품추천  >인기순으로 정렬할 예정 |

**SURVEY**

**(설문결과 번호, 좋아하는 향1, 좋아하는 향2, 좋아하는 향3, 연령대, 성별, 회원 아이디)**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| survey\_insert | INSERT INTO survey VALUES  (survey\_no\_seq.NEXTVAL, ?, ?, ?, ?, ?, ?); | 설문지 결과값 등록 |
| survey\_readOneById | SELECT s\_no, s\_fav1, s\_fav2, s\_fav3, s\_age, s\_gender FROM survey WHERE m\_id=?; | 설문지 하나 읽기 |
| survey\_readAll | SELECT s\_no, s\_fav1, s\_fav2, s\_fav3, s\_age, s\_gender, m\_id FROM survey; | 설문지 전체 리스트 |
| survey\_update | UPDATE survey SET s\_fav1=?, s\_fav2=?, s\_fav3=?, s\_age=?,s\_gender=? WHERE s\_no=?;  > 관리자가 수정할 때는 s\_no 사용해도 되는데 보통 본인 이외에 설문지 결과를 수정하는 경우는 거의 없고 회원이 설문조사를 다시 하고 싶을 때 결과가 바뀌기 때문에 m\_id로 사용  UPDATE survey SET s\_fav1=?, s\_fav2=?, s\_fav3=?, s\_age=?,s\_gender=? WHERE m\_id=?; | 설문지 수정 |
| survey\_remove | DELETE FROM survey WHERE s\_no=?; | 설문지 삭제 |

**ACCORD**

**(향 이름, 향 설명, 향 이미지)**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| accord\_insert | INSERT INTO accord VALUES  (ACCORD\_no\_seq.nextval, ?, ?, ?); | 향 등록 |
| accord\_readOneByName | SELECT acd\_no, acd\_name, acd\_des, acd\_img FROM accord WHERE acd\_name=?; | 향 이름으로 향 하나 읽기 |
| accord\_readAll | SELECT acd\_no, acd\_name, acd\_des, acd\_img FROM accord; | 향 리스트 |
| accord\_update | UPDATE accord SET acd\_des=?,acd\_img=? WHERE acd\_name=?; | 향 수정 |
| accord\_remove | DELETE FROM accord WHERE acd\_name=?; | 향 삭제 |

**TAG**

**(태그 번호, 태그 이름, 태그 카운트)**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| tag\_insert | INSERT INTO tag VALUES(tag\_no\_seq.NEXTVAL,?,0);  >t\_count 0부터 시작 | 태그 등록 |
| tag\_readOneByNo | SELECT t\_no, t\_name, t\_count FROM tag WHERE t\_no=?; | 태그 번호로 태그 하나 읽기 |
| tag\_readAll | SELECT t\_no, t\_name, t\_count FROM tag; | 태그 리스트 |
| tag\_update | UPDATE tag SET t\_name=? WHERE t\_no=?; | 태그 수정 |
| tag\_remove | DELETE FROM tag WHERE t\_no=?; | 태그 삭제 |

**BRAND**

**(브랜드 번호, 브랜드명)**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| brand\_insert | INSERT INTO brand VALUES(brand\_no\_seq.NEXTVAL,?); | 브랜드 등록 |
| brand\_readOneByNo | SELECT br\_no, br\_name FROM brand WHERE br\_no=?; | 브랜드 번호로 태그 브랜드 읽기 |
| brand\_readAll | SELECT br\_no, br\_name FROM brand; | 브랜드 리스트 |
| brand\_update | UPDATE brand SET br\_name=? WHERE br\_no=?; | 브랜드 수정 |
| brand\_remove | DELETE FROM brand WHERE br\_no=?; | 브랜드 삭제 |

**JUMUN**

**(주문번호,주문이름,전체수량,전체금액,주문날짜,회원아이디)**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| Jumun\_insert | INSERT INTO JUMUN VALUES(JUMUN\_no\_seq.nextval,?,?,?,SYSDATE,?); | 주문생성 |
| Jumun\_readID | SELECT \* FROM JUMUN WHERE m\_id = ?; | 아이디로 주문찾기 |
| Jumun\_readAll | SELECT \* FROM JUMUN; | 주문리스트 |

**JUMUN\_DETAIL**

**(주문상품번호, 상품별수량, 상품별총가격, 주문번호, 상품번호)**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| JumunDetail\_insert | INSERT INTO JUMUN\_DETAIL VALUES(jumun\_detail\_no\_seq.NEXTVAL,?,?,?,?) | 주문상세생성 |
| JumunDetail\_readID | SELECT j.j\_no, p.p\_name, p.p\_volume, p.p\_price, p.p\_image, jd.jd\_item\_qty, jd.jd\_items\_price FROM member m JOIN jumun j ON m.m\_id = j.m\_id JOIN jumun\_detail jd ON jd.j\_no = j.j\_no AND jd.jd\_no=? JOIN product p ON p.p\_no = jd.p\_no; | 아이디로 주문상세찾기 |
| JumunDetail\_readAll | SELECT j.j\_no, p.p\_name, p.p\_volume, p.p\_price, p.p\_image, jd.jd\_item\_qty, jd.jd\_items\_price FROM member m JOIN jumun j ON m.m\_id = j.m\_id JOIN jumun\_detail jd ON jd.j\_no = j.j\_no JOIN product p ON p.p\_no = jd.p\_no; | 전체 주문상세 |
| JumunDetail\_update | UPDATE JUMUN\_DETAIL SET jd\_item\_qty = ?, jd\_items\_price = ? WHERE jd\_no=?; |  |
| JumunDetail\_delete | DELETE FROM JUMUN\_DETAIL WHERE jd\_no =?; | 주문상세삭제 |

**CART**

**(카트번호, 카트수량, 카트가격, 상품담은날짜, 회원아이디, 상품번호)**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| Cart\_insert | INSERT INTO cart values(cart\_no\_seq.nextval,?,?,SYSDATE,?,?) | 장바구니 추가 |
| *Cart\_viewById* | SELECT \* FROM cart WHERE m\_id=? | 장바구니보기(전체주문) |
| Cart\_updateQty | UPDATE cart SET cart\_qty=?, cart\_price=?, cart\_date=sysdate where cart\_no=? | 수량변경 |
| Cart\_deleteAll | DELETE FROM cart WHERE m\_id=? | 전체삭제 |
| Cart\_deletetOne | DELETE FROM cart WHERE cart\_no=?  : 상품 row 옆 ‘DELETE’, 하단 ‘선택상품삭제'에 사용됨 | 바로삭제/선택삭제 |
| Cart\_deleteDate | DELETE FROM cart WHERE ROUND(SYSDATE - cart\_date) > 7 (보류) | 날짜삭제(7일후) |
| *Cart\_viewBy*  *CartNo* | SELECT \* FROM cart WHERE cart\_no=?  : 상품 row 옆 ‘BUY NOW’, 하단 ‘선택상품주문'에 사용됨 | 선택주문/바로주문 |

**CHECKOUT**

**(결제 번호, 주문 전체수량, 주문 전체가격, 회원아이디, 주문번호, 결제날짜, 결제수단, 실배송주소)**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| Checkout\_insert | INSERT INTO checkout values(checkout\_no\_seq.nextval,?,?,?,?,SYSDATE,?,?) | 결제 추가 |
| *Checkout\_readAll* | SELECT \* FROM checkout; | 전체 결제내역 리스트 |
| Checkout\_readOneById | SELECT co.co\_no, j.j\_total\_qty, j.j\_total\_price, m.m\_id, j.j\_no, co.co\_date, co.co\_method, co.co\_address FROM checkout co JOIN member m ON co.m\_id = m.m\_id JOIN jumun j ON co.j\_no = j.j\_no WHERE m.m\_id=?; | 회원아이디로 결제내역 보기 |
| Checkout\_readOneByNo | SELECT \* FROM checkout WHERE co\_no=?; | 결제번호로 결제내역 보기 |
| Checkout\_delete | DELETE FROM checkout WHERE m\_id=?; | 결제내역 삭제 |
| *Checkout\_update* | UPDATE checkout SET j\_total\_qty=?, j\_total\_price=?,m\_id=?, j\_no=?, co\_date=?, co\_method=?, co\_address where co\_no=?; | 결제내역 수정 |

**Board\_category[]={’관리자공지사항’,’리뷰게시판’,’Q&A게시판’}**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| *Board\_insert* | INSERT INTO notice VALUES(notice\_sequence.nextval,?,?,?,?,SYSDATE) | 공지 등록  b\_no  b\_title  m\_id  b\_count  b\_content  b\_date |
| *Board\_readOne* | SELECT B\_NO, B\_TITLE,M\_ID, B\_COUNT, B\_CONTENT, B\_DATE  FROM notice  WHERE B\_NO=? | 공지읽기 |
| *Board\_update* | UPDATE board SET B\_TITLE = ? ,B\_CONTENT = ?  WHERE B\_NO = ? | 공지수정 |
| *Board\_delete* | DELETE board  WHERE B\_NO = ? | 공지삭제. |
| *Board\_readAll* | SELECT \*  FROM board | 게시물 리스트를 반환(게시물시작번호,게시물끝번호) |
| *Board*\_increaseReadCount | UPDATE board SET b\_count = b\_count + 1  WHERE b\_no = ?; | 게시물  조회수 증가하는  메쏘드 |
| *Board*\_getBoardCount | SELECT COUNT(\*)  FROM board; | 게시물 총 건수를 조회, 반환 |
| *Board\_readOne*  *ID* | SELECT B\_NO, B\_TITLE, B\_COUNT, B\_CONTENT, B\_DATE,M\_ID  FROM board  WHERE M\_ID=? | 사용안해도될듯 |

**Q&A**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| *Qna\_insert* | INSERT INTO QNA VALUES(qna\_sequence.nextval,?,?,?,?,?,SYSDATE,qna\_sequence.currval,?,?) | Q&A등록  q\_no  p\_no  q\_title  m\_id  q\_count  q\_content  q\_date  q\_groupno  q\_step  q\_depth |
| *Qna\_readOne*  *NO* | SELECT Q\_NO, P\_NO,Q\_TITLE, M\_ID ,Q\_COUNT, Q\_CONTENT, Q\_DATE, Q\_GROUPNO, Q\_STEP, Q\_DEPTH  FROM qna  WHERE Q\_NO=? | Q&A읽기 |
| *Qna\_update* | UPDATE qna SET Q\_TITLE = ? ,Q\_CONTENT = ?  WHERE Q\_NO = ? | Q&A수정 |
| *Qna\_delete* | DELETE qna  WHERE Q\_NO = ? | Q&A삭제. |
| *Qna\_create*  *Reply* | UPDATE qna SET Q\_STEP = Q\_STEP + 1  WHERE Q\_STEP > 1 AND Q\_GROUPNO= ?;  --------------------------------------------------------------------  INSERT INTO qna VALUES(qna\_sequence.nextval,?,?,?,?,?,SYSDATE,?,?,?) | Q&A답글쓰기  q\_no  p\_no  q\_title  m\_id  q\_count  q\_content .  q\_date  q\_groupno  q\_step  q\_depth |
| *Qna\_readAll* | SELECT \*  FROM qna | 게시물 리스트를 반환(게시물시작번호,게시물끝번호) |
| Qna\_countReplay | SELECT count(\*) cnt  FROM qna  WHERE Q\_GROUPNO = ?  AND Q\_depth >= ?  AND Q\_STEP >= ?  ORDER BY Q\_STEP ,Q\_depth ASC; | 답변게시물 존재여부확인메쏘드 |
| Qna\_increaseReadCount | UPDATE qna SET Q\_count = Q\_count + 1  WHERE Q\_no = ?; | 게시물  조회수 증가하는  메쏘드 |
| Qna\_getCount | SELECT COUNT(\*)  FROM qna ; | 게시물 총 건수를 조회, 반환 |
| *Qna\_readOne*  *ID* | SELECT Q\_NO,P\_NO, M\_ID ,Q\_TITLE, Q\_COUNT, Q\_CONTENT, Q\_DATE, Q\_GROUPNO, Q\_STEP, Q\_DEPTH  FROM qna  WHERE M\_ID=? |  |

**REVIEW**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| *Rv\_insert* | INSERT INTO Rv VALUES(Rv\_sequence.nextval,?,?,?,?,?,?,SYSDATE,Rv\_sequence.currval,?,?) | 리뷰등록  r\_no  r\_star  p\_no  r\_title  m\_id  r\_count  r\_content  r\_date  r\_groupno  r\_step  r\_depth |
| *Rv\_readOne*  *NO* | SELECT R\_NO, R\_STAR,P\_NO,R\_TITLE, M\_ID,R\_COUNT, R\_CONTENT, R\_DATE, R\_GROUPNO, R\_STEP, R\_DEPTH  FROM Rv  WHERE R\_NO=? | 리뷰읽기 |
| *Rv\_update* | UPDATE Rv SET R\_TITLE = ? ,R\_CONTENT = ?  WHERE R\_NO = ? | 리뷰수정 |
| *Rv\_delete* | DELETE RV  WHERE R\_NO = ? | 리뷰삭제. |
| *Rv\_create*  *Reply* | UPDATE Rv SET R\_STEP = R\_STEP + 1  WHERE R\_STEP > 1 AND R\_GROUPNO= ?;  --------------------------------------------------------------------  INSERT INTO Rv VALUES(Rv\_sequence.nextval,?,?,?,?,?,?,SYSDATE,?,?,?) | 리뷰쓰기  r\_no  r\_star  p\_no  r\_title  m\_id  r\_count  r\_content  r\_date  r\_groupno  r\_step  r\_depth |
| *Rv\_readAll* | SELECT \*  FROM Rv | 게시물 리스트를 반환(게시물시작번호,게시물끝번호) |
| Rv\_countReplay | SELECT count(\*) cnt  FROM Rv  WHERE R\_GROUPNO = ?  AND R\_depth >= ?  AND R\_STEP >= ?  ORDER BY R\_STEP ,R\_depth ASC; | 리뷰답변게시물 존재여부확인메쏘드 |
| Rv\_increaseReadCount | SELECT COUNT(\*) FROM Rv; | 게시물  조회수 증가하는  메쏘드 |
| Rv\_getCount | SELECT COUNT(\*)  FROM Rv ; | 게시물 총 건수를 조회, 반환 |
| *Rv\_readOne*  *ID* | SELECT R\_NO,R\_STAR,P\_NO, R\_TITLE,M\_ID,R\_COUNT, R\_CONTENT, R\_DATE, R\_GROUPNO, R\_STEP, R\_DEPTH  WHERE M\_ID=? |  |
|  |  | 별 |

**CHATROOM**

|  |  |  |
| --- | --- | --- |
| **메소드명** | **SQL** | **설명** |
| Chat\_sumit()  채팅보내기 | insert into CHAT  values(CHAT\_no\_seq.nextval,SYSDATE,?,?,?)  --------  insert into CHAT  values (NULL,SYSDATE,?,?,?) | chat\_roomNo  chat\_time  m\_id  chat\_admin  chat\_message |
| Chat\_getChatListById()  >>채팅메세지가져오기 | select \*  from chat  where  m\_id=?  and  chat\_roomNo >?  ORDER BY chat\_time;  --------  select \* from CHAT  WHERE m\_id= #{m\_id} and  and chat\_roomNo= #{chat\_roomNo} ; | chat\_roomNo  chat\_time  m\_id보낸  Chat\_admin  받는  chat\_message |
| Chat\_getChatListByIdRecent()  >>최근몇개만가져오기 >>max시퀀스로 가져오기 | select \*  from chat  where  ((fromId= ? and toid =?) OR (fromId= ? and toid =?)) and  chatid > (select MAX(chatId)-? FROM chat)  ORDER BY chat\_time; |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |