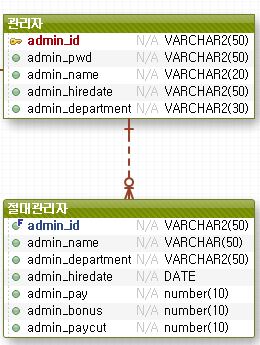
# Database 구조

**DB Schema**

## EMB00001d90074a

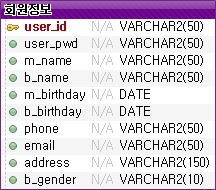
## DB와 주요 쿼리 설명

### 쇼핑몰

1. 사용자 권한(로그인)

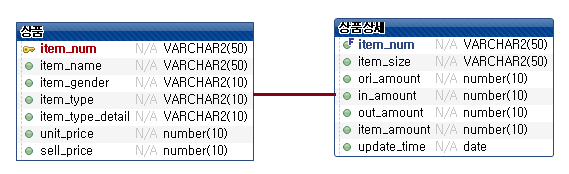
>> 회원가입

|  |
| --- |
| INSERT INTO USER\_ACCOUNT (USER\_ID, USER\_PWD, M\_NAME, B\_NAME, M\_BIRTHDAY,  B\_BIRTHDAY, PHONE, EMAIL, ADDRESS, B\_GENDER)  VALUES(#{USER\_ID}, #{USER\_PWD}, #{M\_NAME}, #{B\_NAME}, #{M\_BIRTHDAY},  #{B\_BIRTHDAY,JDBCTYPE=DATE}, #{PHONE}, #{EMAIL}, #{ADDRESS}, #{B\_GENDER}) |

1. 회원정보

>> 회원 정보 수정

|  |
| --- |
| UPDATE USER\_ACCOUNT SET USER\_PWD=#{USER\_PWD}, PHONE=#{PHONE},  EMAIL=#{EMAIL}, ADDRESS=#{ADDRESS} WHERE USER\_ID=#{USER\_ID} |

1. 상품

>> 상품 등록

|  |
| --- |
| INSERT INTO ITEM  VALUES (#{item\_num},#{item\_name},#{item\_gender},#{item\_type},  #{item\_type\_detail},#{unit\_price},#{sell\_price},#{pic\_url},#{description},#{contents}) |

>> 상품 내용수정

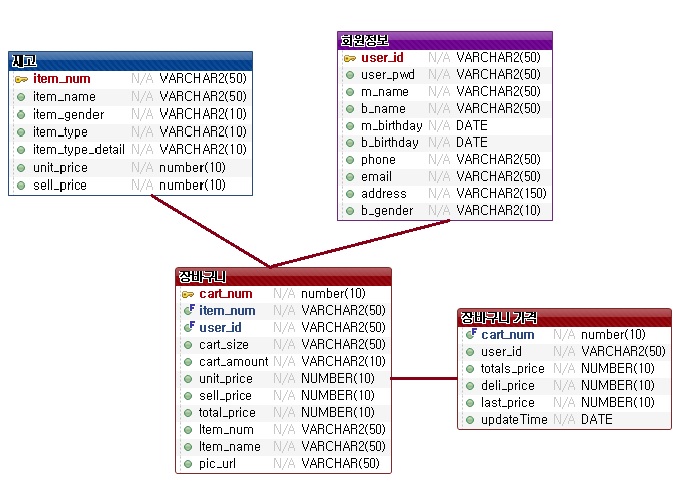
|  |
| --- |
| UPDATE ITEM SET PIC\_URL=#{pic\_url},CONTENTS=#{contents} WHERE ITEM\_NUM=#{item\_num} |

>> 상품 삭제

|  |
| --- |
| DELETE FROM ITEM WHERE ITEM\_NUM=#{item\_num} |

>> 큰 카테고리 ‘girl'선택 후, 작은 카테고리 ’top‘ 중 하나 선택 시 목록 출력

|  |
| --- |
| SELECT ITEM\_NUM,ITEM\_NAME,SELL\_PRICE,PIC\_URL,DESCRIPTION FROM ITEM  WHERE (ITEM\_TYPE='top'  <if test="itd.equals('tee')"> AND ITEM\_TYPE\_DETAIL='tee') </if>  <if test="itd.equals('blou')"> AND ITEM\_TYPE\_DETAIL='blou') </if>  <if test="itd.equals('knit')"> AND ITEM\_TYPE\_DETAIL='knit') </if>  AND ITEM\_GENDER='girl'  ORDER BY ITEM\_NUM DESC |

1. 장바구니

>> 장바구니에 추가

|  |
| --- |
| INSERT INTO CART  VALUES (seq\_cart\_num.NEXTVAL,#{user\_id},#{cart\_size},#{cart\_amount},#{unit\_price},#{sell\_price},  #{total\_price},#{item\_num},#{item\_name},#{pic\_url}) |

>> 장바구니에 이미 존재하는 상품의 수량만 추가

|  |
| --- |
| UPDATE CART SET CART\_AMOUNT=#{cart\_amount\_add},TOTAL\_PRICE=#{total\_price}  WHERE USER\_ID=#{user\_id} AND ITEM\_NUM=#{item\_num} |

>> 장바구니 목록보기

|  |
| --- |
| SELECT \* FROM CART WHERE USER\_ID=#{user\_id} ORDER BY CART\_NUM DESC |

>> 장바구니 가격 부분 추가

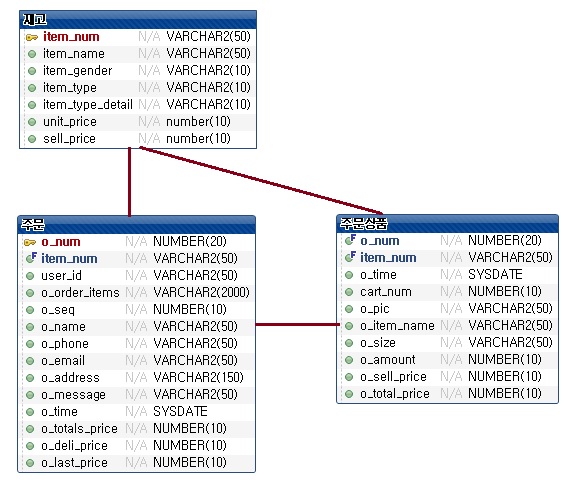
|  |  |
| --- | --- |
| INSERT INTO CART\_PRICE VALUES (#{user\_id}, #{totals\_price}, #{deli\_price}, #{last\_price}, SYSDATE) | [cs](http://colorscripter.com/info#e) |

>> 장바구니 총 결제가 출력

|  |
| --- |
| SELECT \* FROM (SELECT \* FROM CART\_PRICE ORDER BY UPDATETIME DESC)  WHERE ROWNUM=1 AND USER\_ID=#{user\_id} |

>> 장바구니 내에서 상품 수량 추가/삭제

|  |
| --- |
| UPDATE CART SET CART\_AMOUNT=#{cart\_amount},TOTAL\_PRICE=#{total\_price}  WHERE CART\_NUM=#{cart\_num} AND USER\_ID=#{user\_id} |
|  |

1. 주문

>> 주문 하기

* 주문 내역 추가

|  |
| --- |
| INSERT INTO ORDERS  VALUES (#{user\_id},#{o\_item\_name},seq\_o\_seq.NEXTVAL,#{o\_num},  #{o\_name},#{o\_phone},#{o\_email},#{o\_address},#{o\_message},  SYSDATE,#{o\_totals\_price},#{o\_deli\_price},#{o\_last\_price}) |

* 주문 상품 목록 추가

|  |
| --- |
| INSERT INTO ORDERSITEM VALUES  (#{o\_num},SYSDATE,#{cart\_num},#{item\_num},#{o\_pic},#{o\_item\_name},  #{o\_size},#{o\_amount},#{o\_sell\_price},#{o\_total\_price}) |

>> 주문한 상품 사이즈에 해당되는 재고 개수 가져오기

|  |
| --- |
| SELECT a.ITEM\_SIZE AS item\_size, s.ITEM\_AMOUNT AS item\_amount, a.UPDATE\_TIME  FROM (SELECT ITEM\_SIZE, MAX(UPDATE\_TIME) AS UPDATE\_TIME  FROM STOCKINFO  WHERE ITEM\_NUM=#{item\_num} AND ITEM\_SIZE=#{item\_size}  GROUP BY ITEM\_SIZE) a  INNER JOIN STOCKINFO s  ON s.ITEM\_SIZE=a.ITEM\_SIZE AND s.UPDATE\_TIME=a.UPDATE\_TIME |

>> 주문한 상품 사이즈에 해당되는 재고 개수 감소

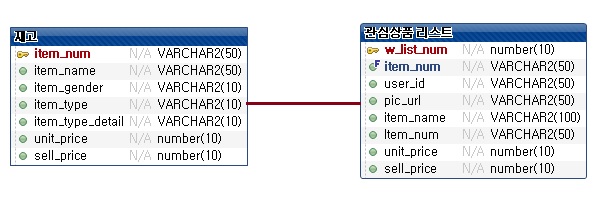
|  |
| --- |
| INSERT INTO STOCKINFO (ITEM\_NUM,ITEM\_SIZE,ORI\_AMOUNT,OUT\_AMOUNT,ITEM\_AMOUNT,UPDATE\_TIME)  VALUES (#{item\_num},#{item\_size},#{ori\_amount},#{out\_amount},#{item\_amount},SYSDATE) |

>> 주문 내역 목록

|  |
| --- |
| SELECT o\_num,o\_time,o\_order\_items,o\_last\_price FROM ORDERS  WHERE USER\_ID=#{user\_id} ORDER BY o\_time DESC |

>> 주문 상세내역 보기

|  |
| --- |
| SELECT \* FROM ORDERSITEM WHERE O\_NUM=#{o\_num} |

1. 관심상품

>> 관심상품 등록

|  |  |
| --- | --- |
| INSERT INTO WISHLIST  (W\_LIST\_NUM, USER\_ID, PIC\_URL, ITEM\_NAME, ITEM\_NUM, UNIT\_PRICE, SELL\_PRICE)  VALUES ((SELECT NVL(MAX(W\_LIST\_NUM), 0) + 1 FROM WISHLIST),  #{USER\_ID}, #{PIC\_URL},#{ITEM\_NAME},#{ITEM\_NUM},#{UNIT\_PRICE},#{SELL\_PRICE}) | [CS](http://colorscripter.com/info#e) |

>> 관심상품페이지에서 삭제

|  |
| --- |
| DELETE FROM WISHLIST WHERE W\_LIST\_NUM = #{W\_LIST\_NUM} |

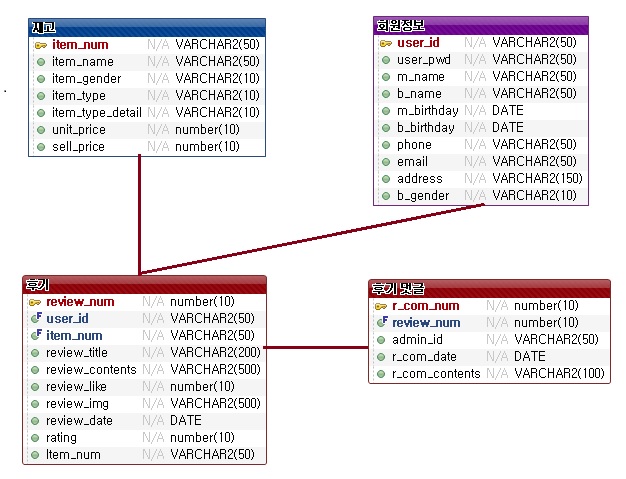
>> 상품 상세페이지에서 관심상품 삭제

|  |
| --- |
| DELETE FROM WISHLIST WHERE USER\_ID = #{USER\_ID} AND ITEM\_NUM = #{ITEM\_NUM} |

>> 관심상품 리스트 - 로그인한 유저가 등록한 관심 상품만 리스트로 가져옴

|  |
| --- |
| SELECT \*  FROM (SELECT A.\*, ROWNUM R  FROM (SELECT \* FROM WISHLIST WHERE USER\_ID=#{USER\_ID}  ORDER BY W\_LIST\_NUM DESC) A)  WHERE R >= #{START} AND R &LT;= #{END} |

1. 후기



>> 리뷰 등록

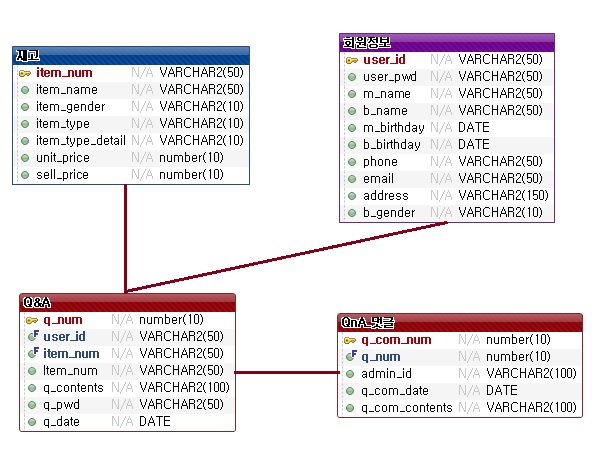
|  |  |
| --- | --- |
| INSERT INTO REVIEW (REVIEW\_NUM, USER\_ID, REVIEW\_TITLE, REVIEW\_CONTENTS,  REVIEW\_LIKE, REVIEW\_IMG, REVIEW\_DATE, RATING, ITEM\_NUM)  VALUES ((SELECT NVL(MAX(REVIEW\_NUM), 0) + 1 FROM REVIEW),  #{USER\_ID},#{REVIEW\_TITLE},#{REVIEW\_CONTENTS},0,0,SYSDATE,#{RATING},#{ITEM\_NUM}) | [CS](http://colorscripter.com/info#e) |

>> 리뷰 리스트

|  |
| --- |
| SELECT \*  FROM (SELECT A.\*, ROWNUM R  FROM (SELECT \* FROM REVIEW WHERE ITEM\_NUM=#{ITEM\_NUM}  ORDER BY REVIEW\_NUM DESC) A)  WHERE R >= #{START} AND R &LT;= #{END} |

>> 리뷰 삭제

|  |
| --- |
| DELETE FROM REVIEW WHERE REVIEW\_NUM = #{REVIEW\_NUM} |

1. Q&A

>> Question 등록

|  |
| --- |
| INSERT INTO QNA (Q\_NUM, USER\_ID, Q\_CONTENTS,  VALUES( (SELECT NVL(MAX(Q\_NUM),0)+1 FROM QNA),#{USER\_ID},#{Q\_CONTENTS}) |

>> Question 리스트

|  |
| --- |
| SELECT \* FROM  (SELECT A.\*, ROWNUM R FROM  (SELECT \* FROM QNA WHERE ITEM\_NUM=#{ITEM\_NUM} ORDER BY Q\_NUM DESC) A)  WHERE R >= #{START} AND R &LT;= #{END} |

>> Question 수정

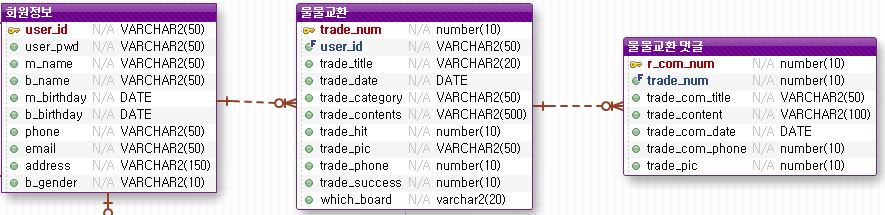
|  |
| --- |
| UPDATE QNA SET Q\_CONTENTS=#{Q\_CONTENTS} WHERE Q\_NUM = #{Q\_NUM} |

>> Question 삭제

|  |
| --- |
| DELETE FROM QNA WHERE Q\_NUM=#{Q\_NUM} |

>> Question 댓글 입력

|  |
| --- |
| INSERT INTO QNA\_COMMENT (Q\_COM\_NUM, ADMIN\_ID, Q\_NUM, Q\_COM\_CONTENTS, Q\_COM\_DATE)  VALUES ((SELECT NVL(MAX(Q\_COM\_NUM),0)+1 FROM QNA\_COMMENT)  ,#{ADMIN\_ID}, #{Q\_NUM}, #{Q\_COM\_CONTENTS}, SYSDATE) |

1. 물물교환

>> 거래가 완료될 시 trade\_success 값이 1로 바뀌며 게시글에 거래 완료라는 문구가 뜬다.

|  |
| --- |
| UPDATE TRADE SET TRADE\_SUCCESS = TRADE\_SUCCESS + 1  WHERE TRADE\_NUM = #{TRADE\_NUM} |

>> 물물교환 게시판 모든 글의 default 값은 TRADE로 선언한다. 물물교환 게시판 글 중 trade\_success = 0 이며 (거래가 완료 되지 않은) 현재 일 짜 로부터 7일 동안 거래가 되지 않은 글은 which\_board 변수가 ‘TRADE’에서 ‘DONATE’로 바뀌게 되고 리스트를 기부 게시판으로 이동한다.

|  |  |
| --- | --- |
| <![CDATA[ UPDATE  (SELECT \* FROM TRADE WHERE ((TRADE\_DATE <= (SELECT SYSDATE - INTERVAL '0' DAY FROM DUAL)  AND TRADE\_DATE >= (SELECT SYSDATE - INTERVAL '7' DAY FROM DUAL))  AND (TRADE\_SUCCESS = 0))) SET WHICH\_BOARD = 'DONATE' WHERE WHICH\_BOARD = 'TRADE']]> | [CS](http://colorscripter.com/info#e) |

### ERP

1. 인사관리

>> 관리자 회원 가입

|  |
| --- |
| INSERT INTO ADMIN  (ADMIN\_ID, ADMIN\_PWD, ADMIN\_NAME, ADMIN\_HIREDATE, ADMIN\_DEPARTMENT)  VALUES  (#{ADMIN\_ID},#{ADMIN\_PWD}, #{ADMIN\_NAME},#{ADMIN\_HIREDATE}, #{ADMIN\_DEPARTMENT}) |

>> 관리자 삭제 – 최상위 관리자는 회원가입 된 관리자들을 거절 할 수 있다

|  |
| --- |
| DELETE FROM ADMIN WHERE ADMIN\_ID=#{ADMIN\_ID} |

>> 하위 관리자 정보 출력

|  |
| --- |
| SELECT \* FROM HUMAN\_RESOURCE |

>> 하위 관리자 월급 관리

|  |
| --- |
| UPDATE HUMAN\_RESOURCE SET ADMIN\_PAY=#{ADMIN\_PAY}, ADMIN\_BONUS=#{ADMIN\_BONUS},  ADMIN\_PAYCUT=#{ADMIN\_PAYCUT} WHERE ADMIN\_ID=#{ADMIN\_ID} |

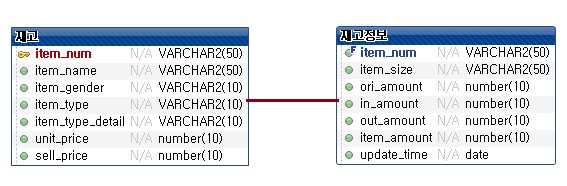
>> 모든 회원 정보 출력

|  |
| --- |
| SELECT\* FROM USER\_ACCOUNT |

>> 아이디에 따른 회원 정보 출력

|  |
| --- |
| SELECT\* FROM USER\_ACCOUNT WHERE USER\_ID=#{USER\_ID} |

1. 재고 관리



>> 재고 목록 출력

|  |
| --- |
| SELECT \* FROM (SELECT A.\*, ROWNUM R  FROM (SELECT \* FROM STOCK ORDER BY ITEM\_NUM DESC) A)  WHERE R >= #{start, jdbcType=NUMERIC} AND R &lt;= #{end, jdbcType=NUMERIC} |

>> 재고 검색 목록 출력

|  |
| --- |
| SELECT \* FROM  (SELECT A.\*, ROWNUM R FROM  (SELECT \* FROM STOCK ORDER BY ITEM\_NUM DESC) A)  WHERE (R >= #{start, jdbcType=NUMERIC} AND R &lt;= #{end, jdbcType=NUMERIC})  <if test="searchBox!=null and searchValue.equals('item\_num')">  AND ITEM\_NUM=#{searchBox} </if>  <if test="searchBox!=null and searchValue.equals('item\_name')">  AND ITEM\_NAME LIKE '%' || #{searchBox} || '%' </if>  <if test="searchBox!=null and searchValue.equals('item\_gender')">  AND ITEM\_GENDER=#{searchBox} </if>  <if test="searchBox!=null and searchValue.equals('item\_type')">  AND ITEM\_TYPE=#{searchBox}</if>  <if test="searchBox!=null and searchValue.equals('item\_type\_detail')">  AND ITEM\_TYPE\_DETAIL=#{searchBox}</if>  ORDER BY ITEM\_NUM DESC |

>> 상품 별 재고 개수 출력

|  |
| --- |
| SELECT DISTINCT s.ITEM\_SIZE, s.ITEM\_AMOUNT, a.UPDATE\_TIME  FROM (SELECT ITEM\_SIZE, MAX(UPDATE\_TIME) AS UPDATE\_TIME  FROM STOCKINFO  WHERE ITEM\_NUM=#{item\_num}  GROUP BY ITEM\_SIZE) a  INNER JOIN STOCKINFO s  ON s.ITEM\_SIZE=a.ITEM\_SIZE AND s.UPDATE\_TIME=a.UPDATE\_TIME  WHERE ITEM\_NUM=#{item\_num}  ORDER BY DECODE(ITEM\_SIZE,'xs',1,'s',2,'m',3,'l',4,'xl',5) |

>> 상품 별 품절 재고 개수 출력

|  |
| --- |
| SELECT COUNT(ITEM\_AMOUNT) AS COUNTNOSTOCK  FROM (SELECT DISTINCT s.ITEM\_SIZE, s.ITEM\_AMOUNT, a.UPDATE\_TIME  FROM (SELECT ITEM\_SIZE, MAX(UPDATE\_TIME) AS UPDATE\_TIME  FROM STOCKINFO  WHERE ITEM\_NUM=#{item\_num}  GROUP BY ITEM\_SIZE) a  INNER JOIN STOCKINFO s  ON s.ITEM\_SIZE=a.ITEM\_SIZE AND s.UPDATE\_TIME=a.UPDATE\_TIME  WHERE ITEM\_NUM=#{item\_num}  ORDER BY DECODE(ITEM\_SIZE,'xs',1,'s',2,'m',3,'l',4,'xl',5))  WHERE ITEM\_AMOUNT=0 |

>> 입고통계 검색된 재고 목록 출력

|  |
| --- |
| SELECT \*  FROM (SELECT A.\*,ROWNUM R  FROM (SELECT TO\_CHAR(i.UPDATE\_TIME) day, i.item\_num,s.item\_name,  i.item\_size,s.item\_gender,s.item\_type,s.item\_type\_detail,  sum(i.IN\_AMOUNT) in\_amount  FROM STOCKINFO i, STOCK s  WHERE i.item\_num = s.item\_num  <if test="day.length()==10">AND (TO\_CHAR(i.UPDATE\_TIME,'YYYY-MM-dd') = #{day})</if>  <if test="day.length()==7">AND (TO\_CHAR(i.UPDATE\_TIME,'YYYY-MM') = #{day})</if>  <if test="day.length()==4">AND (TO\_CHAR(i.UPDATE\_TIME,'YYYY') = #{day})</if>  AND i.IN\_AMOUNT IS NOT NULL  GROUP BY  TO\_CHAR(i.UPDATE\_TIME),i.item\_num,s.item\_name,i.item\_size,s.item\_gender,s.item\_type,s.item\_type\_detail  ORDER BY TO\_CHAR(i.UPDATE\_TIME)) A )  WHERE R >= #{start, jdbcType=NUMERIC} AND R &lt;= #{end, jdbcType=NUMERIC} |

>> 입고통계 검색 된 성별별 차트 출력

|  |
| --- |
| SELECT TO\_CHAR(i.UPDATE\_TIME) day, s.item\_gender AS item\_gender,  sum(i.IN\_AMOUNT) AS in\_amount  FROM STOCK s, STOCKINFO i  WHERE i.item\_num=s.item\_num  <if test="day.length()==10">AND (TO\_CHAR(i.UPDATE\_TIME,'YYYY-MM-dd') = #{day})</if>  <if test="day.length()==7">AND (TO\_CHAR(i.UPDATE\_TIME,'YYYY-MM') = #{day}) </if>  <if test="day.length()==4">AND (TO\_CHAR(i.UPDATE\_TIME,'YYYY') = #{day}) </if>  GROUP BY TO\_CHAR(i.UPDATE\_TIME), s.item\_gender |

1. 재무관리

>> 오늘 날짜 입금 현황 계산

|  |
| --- |
| SELECT TO\_CHAR (I.UPDATE\_TIME, 'YYYY-MM-DD') DAY, S.ITEM\_NUM, S.ITEM\_NAME,  S.ITEM\_GENDER, S.ITEM\_TYPE, S.ITEM\_TYPE\_DETAIL, S.UNIT\_PRICE UNIT\_PRICE,  SUM(I.IN\_AMOUNT) AMOUNT,SUM(I.IN\_AMOUNT) \* S.UNIT\_PRICE TOTAL  FROM STOCKINFO I, STOCK S  WHERE S.ITEM\_NUM = I.ITEM\_NUM  AND (TO\_CHAR(I.UPDATE\_TIME,'YYYY-MM-DD') = TO\_CHAR(SYSDATE,'YYYY-MM-DD'))  GROUP BY TO\_CHAR (I.UPDATE\_TIME, 'YYYY-MM-DD'), S.ITEM\_NUM, S.ITEM\_NAME, S.ITEM\_GENDER,  S.ITEM\_TYPE, S.ITEM\_TYPE\_DETAIL,S.UNIT\_PRICE  ORDER BY TO\_CHAR(I.UPDATE\_TIME, 'YYYY-MM-DD') |

>> 날짜 별 검색 시, 입금 현황 계산

|  |
| --- |
| SELECT TO\_CHAR (I.UPDATE\_TIME, 'YYYY-MM-DD') DAY, S.ITEM\_NUM, S.ITEM\_NAME,  S.ITEM\_GENDER, S.ITEM\_TYPE, S.ITEM\_TYPE\_DETAIL, S.UNIT\_PRICE UNIT\_PRICE,  SUM(I.IN\_AMOUNT) AMOUNT,SUM(I.IN\_AMOUNT)\*S.UNIT\_PRICE TOTAL  FROM STOCKINFO I, STOCK S  WHERE S.ITEM\_NUM = I.ITEM\_NUM AND (TO\_CHAR(I.UPDATE\_TIME,'YYYY-MM-DD') = #{DAY})  GROUP BY TO\_CHAR(I.UPDATE\_TIME, 'YYYY-MM-DD'), S.ITEM\_NUM, S.ITEM\_NAME, S.ITEM\_GENDER,  S.ITEM\_TYPE, S.ITEM\_TYPE\_DETAIL,S.UNIT\_PRICE  ORDER BY TO\_CHAR(I.UPDATE\_TIME, 'YYYY-MM-DD') |

>> 오늘 날짜 출금 현황 계산

|  |  |
| --- | --- |
| SELECT TO\_CHAR (O.O\_TIME, 'YYYY-MM-DD') DAY, I.ITEM\_NUM, I.ITEM\_NAME, I.ITEM\_GENDER, I.ITEM\_TYPE, ITEM\_TYPE\_DETAIL, O.O\_SELL\_PRICE SELL\_PRICE, SUM(O.O\_AMOUNT) AMOUNT, SUM(O.O\_AMOUNT)\*O.O\_SELL\_PRICE TOTAL  FROM ITEM I, ORDERSITEM O  WHERE I.ITEM\_NAME = O.O\_ITEM\_NAME  AND (TO\_CHAR(O\_TIME,'YYYY-MM-DD') = TO\_CHAR(SYSDATE, 'YYYY-MM-DD'))  GROUP BY TO\_CHAR(O.O\_TIME, 'YYYY-MM-DD'), I.ITEM\_NUM, I.ITEM\_NAME, O.O\_SELL\_PRICE, O\_ITEM\_NAME, I.ITEM\_GENDER, I.ITEM\_TYPE, ITEM\_TYPE\_DETAIL, O\_SELL\_PRICE  ORDER BY TO\_CHAR(O.O\_TIME, 'YYYY-MM-DD') | [CS](http://colorscripter.com/info#e) |

|  |  |
| --- | --- |
| >> 날짜 별 검색 시, 출금 현황 계산  SELECT TO\_CHAR (O.O\_TIME, 'YYYY-MM-DD') DAY, I.ITEM\_NUM, I.ITEM\_NAME, I.ITEM\_GENDER, I.ITEM\_TYPE, ITEM\_TYPE\_DETAIL, O.O\_SELL\_PRICE SELL\_PRICE, SUM(O.O\_AMOUNT) AMOUNT,  SUM(O.O\_AMOUNT)\*O.O\_SELL\_PRICE TOTAL  FROM ITEM I, ORDERSITEM O  WHERE I.ITEM\_NAME = O.O\_ITEM\_NAME AND (TO\_CHAR(O\_TIME,'YYYY-MM-DD') = #{DAY})  GROUP BY TO\_CHAR(O.O\_TIME, 'YYYY-MM-DD'), I.ITEM\_NUM, I.ITEM\_NAME, O.O\_SELL\_PRICE, O\_ITEM\_NAME, I.ITEM\_GENDER, I.ITEM\_TYPE, ITEM\_TYPE\_DETAIL, O\_SELL\_PRICE  ORDER BY TO\_CHAR(O.O\_TIME, 'YYYY-MM-DD') | [CS](http://colorscripter.com/info#e) |

>> 입출고관리의 출금 입력 쿼리 - 재고가 추가되면 자본금에서 그 차액만큼 빠져야 되기 때문에 BankBook에서 최근에 입력된 자본금 – 지출 금액을 하고 지출 내역과 금액을 DB에 저장

|  |  |
| --- | --- |
| INSERT INTO BANKBOOK  VALUES (SYSDATE, #{OUTCOME\_TITLE}, #{OUTCOME},' ',0,(SELECT REMAIN\_PRICE  FROM (SELECT A.REMAIN\_PRICE,ROWNUM R  FROM (SELECT \* FROM BANKBOOK ORDER BY BANKDATE DESC) A)WHERE R=1)-#{OUTCOME}) | [CS](http://colorscripter.com/info#e) |

>> 입출고관리의 입금 입력 쿼리 - 상품이 판매 되면 자본금이 증가해야 되기 때문에 BankBook에서 최근에 입력된 자본금 + 판매금액을 하고 판매 내역과 판매 금액을 DB에 저장

|  |  |
| --- | --- |
| INSERT INTO BANKBOOK  VALUES (SYSDATE, #{OUTCOME\_TITLE}, #{OUTCOME},' ',0,(SELECT REMAIN\_PRICE  FROM (SELECT A.REMAIN\_PRICE,ROWNUM R  FROM (SELECT \* FROM BANKBOOK ORDER BY BANKDATE DESC) A)WHERE R=1)-#{OUTCOME}) | [CS](http://colorscripter.com/info#e) |

>> 입출고관리에서 날짜 별 검색 쿼리 - 입출고관리에서 날짜 별로 검색을 해서 해당 날짜에 해당하는 리스트를 BankBook에서 출력

|  |
| --- |
| SELECT DAY, OUTCOME\_TITLE, OUTCOME, INCOME\_TITLE, INCOME, REMAIN\_PRICE  FROM (SELECT A.\*,ROWNUM R  FROM (SELECT TO\_CHAR(BANKDATE,'YYYY-MM-DD') DAY, OUTCOME\_TITLE, OUTCOME, INCOME\_TITLE, INCOME, REMAIN\_PRICE FROM BANKBOOK  WHERE (TO\_CHAR(BANKDATE,'YYYY-MM-DD') = #{DAY}) ORDER BY BANKDATE DESC) A)  WHERE R >= #{START} AND R &LT;= #{END} |