DECLARE

PRICE NUMBER(20);

TOTAL\_AMT NUMBER(20);

STATUS VARCHAR2 (50);

REQUIRED\_Q NUMBER(10);

ING\_Q NUMBER(10);

INV\_Q NUMBER(10);

ING\_NUMBER NUMBER(10);

i NUMBER(10);

NEEDED\_Q NUMBER(10);

PRO\_AVAILIBLE NUMBER(10);

PRO\_NAME VARCHAR(100);

OCCUPIED\_Q NUMBER(10);

BEGIN

--CHECK FOR THE STOCK OUT PRODUCT (only handle for buying MORE THAN 1 product)

REQUIRED\_Q:=:QUANTITY\_S.DETAIL\_QUANTITY;

SELECT Count(ING\_ID) INTO ING\_NUMBER

FROM(

SELECT INGREDIENT.ING\_ID

FROM PRODUCT, INGREDIENT , INVENTORY

WHERE PRODUCT.PRODUCT\_ID = INGREDIENT.PRODUCT\_ID AND INVENTORY.INVENTORY\_ID = INGREDIENT.INVENTORY\_ID AND

PRODUCT\_NAME = :QUANTITY\_S.PRODUCT\_NAME

GROUP BY INGREDIENT.ING\_ID);

FOR i IN 1..ING\_NUMBER

LOOP

SELECT PRODUCT\_NAME, INVENTORY\_QUANTITY, ING\_QUANTITY

INTO PRO\_NAME, INV\_Q, ING\_Q

FROM(

SELECT PRODUCT\_NAME, INVENTORY\_QUANTITY, ING\_QUANTITY, ROWNUM RN

FROM PRODUCT, INGREDIENT , INVENTORY

WHERE PRODUCT.PRODUCT\_ID = INGREDIENT.PRODUCT\_ID

AND INVENTORY.INVENTORY\_ID = INGREDIENT.INVENTORY\_ID

AND PRODUCT\_NAME = :QUANTITY\_S.PRODUCT\_NAME

GROUP BY PRODUCT\_NAME,INVENTORY\_QUANTITY, ING\_QUANTITY, ROWNUM

ORDER BY RN)

WHERE RN = i;

forms\_ddl('Commit');

NEEDED\_Q := REQUIRED\_Q \* ING\_Q;

PRO\_AVAILIBLE := FLOOR(INV\_Q / ING\_Q);

MESSAGE( PRO\_AVAILIBLE );

IF PRO\_AVAILIBLE < REQUIRED\_Q THEN

MESSAGE(' You can only buy ' || PRO\_AVAILIBLE||' ' || PRO\_NAME || '.');

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

END LOOP;

IF :QUANTITY\_S.DETAIL\_TEMPERATURE = 'HOT' AND :QUANTITY\_S.DETAIL\_COLDNESS <> '0%' then

MESSAGE ('This is a hot drink! NO ICE!');

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

IF :QUANTITY\_S.PRODUCT\_NAME IS NULL

THEN MESSAGE('Please select product');

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF :QUANTITY\_S.DETAIL\_QUANTITY IS NULL

THEN MESSAGE('Please enter quantity');

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF :QUANTITY\_S.DETAIL\_TEMPERATURE IS NULL

THEN MESSAGE('Please select temperature');

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

GO\_BLOCK('QUANTITY\_S');

First\_record;

TOTAL\_AMT :=0;

STATUS := 'PENDING';

:TRANS\_COM\_S.STATUS := STATUS;

LOOP

GO\_BLOCK('SC\_S');

:SC\_S.PRODUCT\_NAME := :QUANTITY\_S.PRODUCT\_NAME;

:SC\_S.DETAIL\_TEMPERATURE := :QUANTITY\_S.DETAIL\_TEMPERATURE;

:SC\_S.DETAIL\_COLDNESS := :QUANTITY\_S.DETAIL\_COLDNESS;

:SC\_S.DETAIL\_SWEETNESS := :QUANTITY\_S.DETAIL\_SWEETNESS;

:SC\_S.DETAIL\_ADDPARTS := :QUANTITY\_S.DETAIL\_ADDPARTS;

:SC\_S.DETAIL\_QUANTITY := :QUANTITY\_S.DETAIL\_QUANTITY;

SELECT PRODUCT\_PRICE INTO PRICE FROM PRODUCT

WHERE PRODUCT\_NAME=LOWER(:SC\_S.PRODUCT\_NAME);

IF :SC\_S.DETAIL\_ADDPARTS IS NOT NULL THEN

PRICE := PRICE +4;

END IF;

:SC\_S.PRODUCT\_PRICE:= PRICE \* :SC\_S.DETAIL\_QUANTITY;

---SELECT PRICE INTO :SC\_S.PRODUCT\_PRICE FROM PRODUCT

---WHERE PRODUCT\_NAME=LOWER(:SC\_S.PRODUCT\_NAME);

---:SC\_S.PRODUCT\_PRICE := PRICE;

forms\_ddl('Commit');

TOTAL\_AMT := TOTAL\_AMT + :SC\_S.PRODUCT\_PRICE;

:TOTAL\_S.TOTAL := TOTAL\_AMT;

GO\_BLOCK('SC\_S');

NEXT\_RECORD;

GO\_BLOCK('QUANTITY\_S');

---EXIT WHEN :SYSTEM.LAST\_RECORD = 'TRUE';

NEXT\_RECORD;

EXIT WHEN :QUANTITY\_S.PRODUCT\_NAME IS NULL;

END LOOP;

FIRST\_RECORD;

GO\_BLOCK('SC\_S');

FIRST\_RECORD;

GO\_BLOCK('CONFIRM\_ORD\_S');

CLEAR\_BLOCK(no\_validate);

END;