--CHECK AVAILABILITY

DECLARE

Selected\_Product\_name varchar2 (50);

Selected\_Product\_status varchar2 (50);

return\_value number;

TOTALAMOUNT NUMBER(20);

PRICE NUMBER(20);

Begin

First\_record;

LOOP

GO\_BLOCK('SC\_S');

Selected\_Product\_name:= :SC\_S.PRODUCT\_NAME;

SELECT PRODUCT\_STATUS

into Selected\_Product\_status

FROM

PRODUCT

WHERE Product\_name= :SC\_S.PRODUCT\_NAME;

If Selected\_Product\_status <> 'available' Then

SET\_ALERT\_PROPERTY('ALERT\_CHECK\_AVAIL', alert\_message\_text,' Product(S) is/are no longer available.');

return\_value:=show\_alert('ALERT\_CHECK\_AVAIL');

--Add more orders

IF return\_value= alert\_button1 THEN

GO\_BLOCK('SC\_S');

CLEAR\_BLOCK(no\_validate);

FIRST\_RECORD;

GO\_BLOCK('TOTAL\_S');

CLEAR\_BLOCK(no\_validate);

GO\_BLOCK('QUANTITY\_S');

First\_record;

Loop

SELECT PRODUCT\_STATUS

into Selected\_Product\_status

FROM PRODUCT

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

IF Selected\_Product\_status = 'unavailable' then

Delete\_record;

ELSIF Selected\_Product\_status <> 'unavailable' THEN

NEXT\_RECORD;

END IF;

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

end loop;

:QUANTITY\_S.PRODUCT\_PRICE := PRICE;

--FOLLOWINGS ARE to calculate total amount

FIRST\_RECORD;

TOTALAMOUNT := 0;

LOOP

PRICE := :QUANTITY\_S.PRODUCT\_PRICE;

GO\_BLOCK('PLACE\_ORDER\_S');

TOTALAMOUNT := TOTALAMOUNT + PRICE;

GO\_BLOCK('QUANTITY\_S');

NEXT\_RECORD;

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

END LOOP;

GO\_BLOCK('PLACE\_ORDER\_S');

:PLACE\_ORDER\_S.TOTAL\_TXB := TOTALAMOUNT;

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

end if;

NEXT\_RECORD;

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

END LOOP;

end;

--UPDATE MEMBER\_STAR

DECLARE

STAR NUMBER(20);

CUS\_PHONE\_COUNT NUMBER(20);

return\_value number;

return\_value2 number;

-- MEMBER\_NAME VARCHAR2(20);

-- ID\_NUMBER NUMBER(20);

PTS NUMBER(20);

username\_count NUMBER(3);

TOTAL NUMBER(20);

REDEEM NUMBER(20);

BEGIN

SELECT COUNT(\*) INTO CUS\_PHONE\_COUNT FROM MEMBER

WHERE MEM\_PHONE =:TOTAL\_S.MEMBER\_PHONE;

-- IF CUS\_PHONE\_COUNT = 0 THEN

-- SET\_ALERT\_PROPERTY('ALERT\_CHECKOUT', alert\_message\_text,'Invalid member phone.');

-- return\_value:=show\_alert('ALERT\_CHECKOUT');

-- RAISE FORM\_TRIGGER\_FAILURE;

IF CUS\_PHONE\_COUNT =1 THEN

SELECT MEM\_STAR

INTO STAR

FROM MEMBER

WHERE MEM\_PHONE=:TOTAL\_S.MEMBER\_PHONE;

--CHECK WETHER RETRIVE OR NOT

IF STAR <> (:TOTAL\_S.TOTAL\_POINT + :TOTAL\_S.REDEEM) THEN

MESSAGE('PLEASE RETRIEVE POINTS FIRST');

MESSAGE('PLEASE RETRIEVE POINTS FIRST');

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

IF :TOTAL\_S.REDEEM <> (STAR - :TOTAL\_S.TOTAL\_POINT) THEN

SET\_ALERT\_PROPERTY('ALERT\_CHECKSTAR\_TEXT', alert\_message\_text,'Customer wants to redeem '||:TOTAL\_S.REDEEM||' more points?');

return\_value:=show\_alert('ALERT\_CHECKSTAR\_TEXT');

IF return\_value =alert\_BUTTON1 THEN

:TOTAL\_S.REDEEM := :TOTAL\_S.REDEEM + (STAR - :TOTAL\_S.TOTAL\_POINT);

:TOTAL\_S.TOTAL\_POINT := :TOTAL\_S.TOTAL\_POINT - :TOTAL\_S.REDEEM;

:TOTAL\_S.DISCOUNTED\_TOTAL := :TOTAL\_S.TOTAL - :TOTAL\_S.REDEEM;

END IF;

IF return\_value = alert\_BUTTON2 THEN

:TOTAL\_S.REDEEM := STAR - :TOTAL\_S.TOTAL\_POINT;

END IF;

END IF;

IF :TOTAL\_S.TOTAL >= 20 THEN

IF :TOTAL\_S.DISCOUNTED\_TOTAL >= 20 THEN

UPDATE MEMBER

SET MEM\_STAR = STAR - :TOTAL\_S.REDEEM + TRUNC(:TOTAL\_S.DISCOUNTED\_TOTAL/ 20)

WHERE MEM\_PHONE=:TOTAL\_S.MEMBER\_PHONE;

forms\_ddl ('Commit');

ELSIF :TOTAL\_S.DISCOUNTED\_TOTAL < 20 THEN

UPDATE MEMBER

SET MEM\_STAR = STAR - :TOTAL\_S.REDEEM

WHERE MEM\_PHONE = :TOTAL\_S.MEMBER\_PHONE;

forms\_ddl ('Commit');

END IF;

END IF;

END IF;

END;

DECLARE

CURRENT\_ORDER\_ID NUMBER(20);

NEW\_ORDER\_ID NUMBER(20);

MEMBER\_ID NUMBER(20);

CUS\_PHONE\_COUNT NUMBER(20);

BEGIN

SELECT COUNT(\*) INTO CUS\_PHONE\_COUNT FROM MEMBER

WHERE MEM\_PHONE =:TOTAL\_S.MEMBER\_PHONE;

IF CUS\_PHONE\_COUNT = 1 THEN

SELECT MEM\_ID INTO MEMBER\_ID FROM MEMBER WHERE MEM\_PHONE = :TOTAL\_S.MEMBER\_PHONE;

SELECT MAX(ORDERS\_ID) INTO CURRENT\_ORDER\_ID FROM ORDERS;

NEW\_ORDER\_ID:=CURRENT\_ORDER\_ID+1;

INSERT INTO ORDERS (ORDERS\_ID,ORDERS\_STATUS,ORDERS\_AMOUNT,MEM\_ID) VALUES (NEW\_ORDER\_ID,'PENDING',:TOTAL\_S.TOTAL,MEMBER\_ID);

forms\_ddl ('Commit');

ELSE

SELECT MAX(ORDERS\_ID) INTO CURRENT\_ORDER\_ID FROM ORDERS;

NEW\_ORDER\_ID:=CURRENT\_ORDER\_ID+1;

INSERT INTO ORDERS (ORDERS\_ID,ORDERS\_STATUS,ORDERS\_AMOUNT) VALUES (NEW\_ORDER\_ID,'PENDING',:TOTAL\_S.TOTAL);

forms\_ddl ('Commit');

END IF;

END;

DECLARE

Current\_DETAIL\_ID NUMBER (20);

New\_DETAIL\_ID NUMBER(20);

PROD\_ID NUMBER(20);

ORD\_ID NUMBER(20);

BEGIN

FIRST\_RECORD;

LOOP

GO\_BLOCK('SC\_S');

--Generate DETAIL\_ID

SELECT MAX (DETAIL\_ID) INTO Current\_DETAIL\_ID FROM DETAIL;

New\_DETAIL\_ID:=Current\_DETAIL\_ID+1;

SELECT PRODUCT\_ID INTO PROD\_ID FROM PRODUCT

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

SELECT MAX(ORDERS\_ID) INTO ORD\_ID FROM ORDERS;

INSERT INTO DETAIL (DETAIL\_ID,DETAIL\_TEMPERATURE,DETAIL\_COLDNESS,DETAIL\_SWEETNESS,

DETAIL\_ADDPARTS,DETAIL\_QUANTITY,DETAIL\_STATUS,PRODUCT\_ID,ORDERS\_ID)

VALUES (New\_DETAIL\_ID,:SC\_S.DETAIL\_TEMPERATURE,:SC\_S.DETAIL\_COLDNESS,

:SC\_S.DETAIL\_SWEETNESS,:SC\_S.DETAIL\_ADDPARTS,

:SC\_S.DETAIL\_QUANTITY,'PENDING',PROD\_ID,ORD\_ID);

forms\_ddl('Commit');

NEXT\_RECORD;

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

END LOOP;

GO\_BLOCK('QUANTITY\_S');

CLEAR\_BLOCK(NO\_VALIDATE);

GO\_BLOCK('TOTAL\_S');

CLEAR\_BLOCK(no\_validate);

GO\_BLOCK('SC\_S');

CLEAR\_BLOCK(no\_validate);

GO\_BLOCK('PLACE\_ORDER\_S');

CLEAR\_BLOCK(no\_validate);

GO\_BLOCK('TRANS\_COM\_S');

END;