DECLARE

POINTS NUMBER(20);

INI\_POINTS NUMBER(20);

TOTAL NUMBER (20);

return\_value NUMBER(20);

count\_id number(3);

STAR NUMBER(38);

TOTAL\_POINT NUMBER(20);

BEGIN

-- Check retrieve point or not

IF :TOTAL\_S.TOTAL\_POINT IS NULL THEN

SET\_ALERT\_PROPERTY('REDEEM\_ALERT ', ALERT\_MESSAGE\_TEXT, 'Please retrieve point first.');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

--Check member ID

IF LENGTH(TRIM(TRANSLATE(:TOTAL\_S.MEMBER\_ID, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ', ' '))) is NULL THEN

SET\_ALERT\_PROPERTY('REDEEM\_ALERT', ALERT\_MESSAGE\_TEXT, 'Member ID must be number.');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

:TOTAL\_S.REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

IF LENGTH(:TOTAL\_S.MEMBER\_ID) <> LENGTH(TRIM(TRANSLATE(:TOTAL\_S.REDEEM, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ', ' '))) THEN

SET\_ALERT\_PROPERTY('REDEEM\_ALERT', ALERT\_MESSAGE\_TEXT, 'Member ID must be number.');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

END IF;

SELECT COUNT(\*)

INTO count\_id

FROM MEMBER

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

-- Check insert redeem amount

--NOT MEMBER

IF count\_id = 0 THEN

set\_item\_property('TOTAL\_S.REDEEEM',Update\_allowed,Property\_false);

SET\_ALERT\_PROPERTY('REDEEM\_ALERT', ALERT\_MESSAGE\_TEXT, 'Please enter another ID.');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

:TOTAL\_S.REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

elsif :TOTAL\_S.REDEEM IS NULL THEN

SET\_ALERT\_PROPERTY('ALERT\_MEM\_ID\_FOR\_STAR', ALERT\_MESSAGE\_TEXT, 'Please enter redeem amount.');

return\_value := SHOW\_ALERT('ALERT\_MEM\_ID\_FOR\_STAR');

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF LENGTH(TRIM(TRANSLATE(:TOTAL\_S.REDEEM, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ', ' '))) is NULL THEN

SET\_ALERT\_PROPERTY('REDEEM\_ALERT', ALERT\_MESSAGE\_TEXT, 'Redeemed points must be number');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

:TOTAL\_S.REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF LENGTH(:TOTAL\_S.REDEEM) <> LENGTH(TRIM(TRANSLATE(:TOTAL\_S.REDEEM, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ', ' '))) THEN

SET\_ALERT\_PROPERTY('REDEEM\_ALERT', ALERT\_MESSAGE\_TEXT, 'Redeemed points must be number');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

:TOTAL\_S.REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF FLOOR(:TOTAL\_S.REDEEM) <> CEIL(:TOTAL\_S.REDEEM) THEN

SET\_ALERT\_PROPERTY('REDEEM\_ALERT', ALERT\_MESSAGE\_TEXT, 'Redeemed points must be integer');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

:TOTAL\_S.REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF :TOTAL\_S.REDEEM <0 THEN

SET\_ALERT\_PROPERTY('REDEEM\_ALERT', ALERT\_MESSAGE\_TEXT, 'Redeemed points cannot be negative number');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

:TOTAL\_S.REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

elsif :TOTAL\_S.REDEEM >:TOTAL\_S.TOTAL\_POINT THEN

SET\_ALERT\_PROPERTY('REDEEM\_ALERT', ALERT\_MESSAGE\_TEXT, 'You do not have enough points.');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

:TOTAL\_S.REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF :TOTAL\_S.REDEEM > :TOTAL\_S.TOTAL THEN

SET\_ALERT\_PROPERTY('REDEEM\_ALERT', ALERT\_MESSAGE\_TEXT, 'Redeemed points exceed the total price.');

RETURN\_VALUE := SHOW\_ALERT('REDEEM\_ALERT');

:TOTAL\_S.REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

INI\_POINTS := :TOTAL\_S.TOTAL\_POINT;

POINTS := :TOTAL\_S.REDEEM;

TOTAL := :TOTAL\_S.TOTAL;

TOTAL := TOTAL - POINTS;

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

:TOTAL\_S.TOTAL\_POINT := INI\_POINTS - POINTS;

END;