DECLARE

POINTS NUMBER(20);

INI\_POINTS NUMBER(20);

TOTAL NUMBER (20);

RETURN\_VALUE NUMBER(20);

username\_count number(3);

BEGIN

SELECT COUNT(\*) INTO username\_count from MEMBER

where MEM\_UNAME=:MEMBER\_LOGIN\_C.MEM\_USERNAME\_TXB;

IF username\_count = 0 THEN

set\_item\_property('REDEEM\_C.PTS\_WANT\_TO\_REDEEM',Update\_allowed,Property\_false);

SET\_ALERT\_PROPERTY('PTS\_ALERT', ALERT\_MESSAGE\_TEXT, 'Walk-in customer can not redeem points.');

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

:REDEEM\_C.PTS\_WANT\_TO\_REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF FLOOR(:REDEEM\_C.PTS\_WANT\_TO\_REDEEM) <> CEIL(:REDEEM\_C.PTS\_WANT\_TO\_REDEEM) THEN

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

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

:REDEEM\_C.PTS\_WANT\_TO\_REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF :REDEEM\_C.PTS\_WANT\_TO\_REDEEM <0 THEN

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

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

:REDEEM\_C.PTS\_WANT\_TO\_REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

elsif :REDEEM\_C.PTS\_WANT\_TO\_REDEEM >:REDEEM\_C.PTS\_EARNED THEN

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

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

:REDEEM\_C.PTS\_WANT\_TO\_REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

ELSIF :REDEEM\_C.PTS\_WANT\_TO\_REDEEM > :SC\_C.TOTAL THEN

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

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

:REDEEM\_C.PTS\_WANT\_TO\_REDEEM := 0;

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

INI\_POINTS := :REDEEM\_C.PTS\_EARNED;

POINTS := :REDEEM\_C.PTS\_WANT\_TO\_REDEEM;

TOTAL := :SC\_C.TOTAL;

TOTAL := TOTAL - POINTS;

:SC\_C.TOTAL := TOTAL;

:REDEEM\_C.PTS\_EARNED := INI\_POINTS - POINTS;

END;