**1) INSERT**

IF :STAFFINSERT.SI\_PRO\_NAME IS NULL THEN

MESSAGE('Please fill in all product information.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_BRAND IS NULL THEN

MESSAGE('Please fill in all product information.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_ORIGIN IS NULL THEN

MESSAGE('Please fill in all product information.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_SWEETNESS IS NULL THEN

MESSAGE('Please fill in all product information.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_DESCRIPTION IS NULL THEN

MESSAGE('Please fill in all product information.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_VOLUME IS NULL THEN

MESSAGE('Please fill in all product information.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_PRICE IS NULL THEN

MESSAGE('Please fill in all product information.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_STOCK IS NULL THEN

MESSAGE('Please fill in all product information.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF LENGTH (TRIM(TRANSLATE(:STAFFINSERT.SI\_PRO\_VOLUME, '+-.0123456789', ' '))) IS NOT NULL

THEN message('Volume should not contain any non-numeric value.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF LENGTH (TRIM(TRANSLATE(:STAFFINSERT.SI\_PRO\_PRICE, '+-.0123456789', ' '))) IS NOT NULL

THEN message('Price should not contain any non-numeric value.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF LENGTH (TRIM(TRANSLATE(:STAFFINSERT.SI\_PRO\_STOCK, '+-.0123456789', ' '))) IS NOT NULL

THEN message('Stock should not contain any non-numeric value.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_VOLUME <= 0

THEN message('Volume should be larger than 0');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_PRICE <= 0

THEN message('Price should be larger than 0.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_STOCK <= 0

THEN message('Stock should be larger than 0.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

IF :STAFFINSERT.SI\_PRO\_VOLUME between 0 and 9999999999

then null;

ELSE message('Volume should not be larger than 9999,999,999.99.');

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

IF :STAFFINSERT.SI\_PRO\_PRICE between 0 and 999999

THEN NULL;

ELSE message('Price should not be larger than HKD$999,999.99.');

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

IF :STAFFINSERT.SI\_PRO\_STOCK between 0 and 99999

THEN NULL;

ELSE message('Stock should not be larger than 99,999.99.');

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

DECLARE

PRODUCTNAME\_COUNT NUMBER(3,0);

BEGIN

SELECT COUNT(\*) INTO PRODUCTNAME\_COUNT FROM PRODUCT

WHERE LOWER(PROD\_NAME)=LOWER(:STAFFINSERT.SI\_PRO\_NAME);

IF PRODUCTNAME\_COUNT>0 THEN

MESSAGE('This product had already been inserted. Click [Go to Update] for updates.');

RAISE FORM\_TRIGGER\_FAILURE;

ELSE NULL;

END IF;

END;

Declare

newpro\_id number(11, 0);

largestid number(11, 0);

Begin

select Max(PROD\_ID) into largestid from PRODUCT;

newpro\_id := largestid + 1;

INSERT INTO PRODUCT

(PROD\_ID,

PROD\_NAME,

PROD\_REGION,

PROD\_COUNTRY,

PROD\_SWEET,

PROD\_DESC,

PROD\_VOLUME,

PROD\_PRICE,

PROD\_QUANTITY)VALUES

(newpro\_id,

:STAFFINSERT.SI\_PRO\_NAME,

:STAFFINSERT.SI\_PRO\_BRAND,

:STAFFINSERT.SI\_PRO\_ORIGIN,

:STAFFINSERT.SI\_PRO\_SWEETNESS,

:STAFFINSERT.SI\_PRO\_DESCRIPTION,

:STAFFINSERT.SI\_PRO\_VOLUME,

:STAFFINSERT.SI\_PRO\_PRICE,

FLOOR(:STAFFINSERT.SI\_PRO\_STOCK));

MESSAGE('Product successfully inserted!!!');

End;

GO\_BLOCK('STAFFINSERT');

CLEAR\_BLOCK(no\_validate);

forms\_ddl('Commit');

**2) CANCEL**

GO\_BLOCK('STAFFINSERT');

CLEAR\_BLOCK(no\_validate);

Go\_block ('STAFFPAGE');

UPDATE PRODUCT

SET

PROD\_NAME=:STAFFUPDATE.SU\_PRO\_NAME,

PROD\_REGION=:STAFFUPDATE.SU\_PRO\_BRAND,

PROD\_COUNTRY=:STAFFUPDATE.SU\_PRO\_ORIGIN,

PROD\_DESC=:STAFFUPDATE.SU\_PRO\_DESCRIPTION,

PROD\_SWEET=:STAFFUPDATE.SU\_PRO\_SWEETNESS,

PROD\_VOLUME=:STAFFUPDATE.SU\_PRO\_VOLUME,

PROD\_PRICE=:STAFFUPDATE.SU\_PRO\_PRICE,

PROD\_QUANTITY=:STAFFUPDATE.SU\_PRO\_STOCK

WHERE PROD\_ID=:STAFFUPDATE.SU\_PRO\_ID;

MESSAGE('Product successfully updated!!!');

forms\_ddl('commit');

**3) GO TO UPDATE**

GO\_BLOCK('STAFFINSERT');

CLEAR\_BLOCK(no\_validate);

GO\_BLOCK('STAFFUPDATE');