Documentation base de données

-

Développement de triggers et procédures

Sommaire:

I - Trigger de mise à jour des stocks	
A - Code	2
B - Résultats	3
II - Procédure de création de commande	4
A - Code	4
B - Résultat	5

I - Trigger de mise à jour des stocks

Ce trigger permet de mettre à jour la quantité en stock d'un produit après une commande.

A - Code

```
CREATE OR REPLACE TRIGGER t_i_quantite
BEFORE INSERT
ON QUANTITE
FOR EACH ROW
DECLARE
    lestock PRODUIT.STOCK%TYPE;
    SELECT STOCK INTO lestock FROM PRODUIT WHERE IDPRODUIT = :NEW.IDPRODUIT;
    IF INSERTING THEN
        IF (lestock - :NEW.quantite) > 0 THEN
            UPDATE Produit
            SET stock = stock - :NEW.quantite
            WHERE idproduit = :NEW.idproduit;
            UPDATE Produit
            SET stock = 0
            WHERE idproduit = :NEW.idproduit;
        END IF;
    END IF;
```

B - Résultats

Test avec quantité voulue > à quantité en stock

Avant:



Après:

```
insert into quantite values(21, 1, 15, 10);
```

 $21 \rightarrow ID$ du produit

 $1 \rightarrow ID$ de la commande

15 → quantité voulue

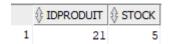
10 → prix unitaire au moment de l'achat



Un avertissement sera présent sur la page de commande pour informer sur ce cas.

Test avec quantité voulue < à quantité en stock

Avant:



Après:

```
insert into quantite
values(21, 1, 4, 10);
```

 $21 \rightarrow ID$ du produit

 $1 \rightarrow ID$ de la commande

4 → quantité voulue

 $10 \rightarrow prix$ unitaire au moment de l'achat



II - Procédure de création de commande

Cette procédure permet de créer une commande à partir d'un panier, elle prend en paramètre 5 valeurs, l'ID client, l'ID règlement, l'adresse, la ville et le code postal. Elle crée un tuple dans Quantite pour chaque produit dans le panier du client en question et supprime le panier du client.

A - Code

```
OR REPLACE PROCEDURE Commander
    p_idclient CLIENT.IDCLIENT%TYPE,
    p_idreglement REGLEMENT.IDREGLEMENT%TYPE,
    p_adresse COMMANDE.ADRESSE%TYPE,
    p_ville COMMANDE.VILLE%TYPE,
    p_codepostal COMMANDE.CODEPOSTAL%TYPE
)
IS
    CURSOR C_client_panier IS
        SELECT IDPRODUIT, QUANTITE FROM PANIER
        WHERE IDCLIENT = p_idclient;
    idcommande COMMANDE.IDCOMMANDE%TYPE;
    prixunitaire PRODUIT.PRIX%TYPE;
    v idclient CLIENT.IDCLIENT%TYPE;
    v_idreglement REGLEMENT.IDREGLEMENT%TYPE;
    SELECT IDCLIENT INTO v_idclient FROM CLIENT WHERE IDCLIENT = p_idclient;
    SELECT IDREGLEMENT INTO v_idreglement FROM REGLEMENT WHERE IDREGLEMENT = p_idreglement;
    idcommande := seq_commande.nextval;
    INSERT INTO COMMANDE
    VALUES(idcommande, p_idclient, p_idreglement, p_adresse, p_ville, p_codepostal);
    FOR elementpanier IN C_client_panier LOOP
         SELECT PRIX INTO prixunitaire FROM PRODUIT WHERE IDPRODUIT = elementpanier.idproduit;
        -- ajoute le produit à la commande

INSERT INTO QUANTITE

VALUES (elementpanier.idproduit, idcommande, elementpanier.quantite, prixunitaire);
        DELETE FROM PANIER WHERE IDCLIENT = p_idclient AND IDPRODUIT = elementpanier.idproduit;
    END LOOP;
    COMMIT;
```

B - Résultat

Test nominal:

Avant:

Le client 28 n'a pas de commande à son actif.

```
SELECT ★ FROM COMMANDE WHERE IDCLIENT = 28;

$\psi$ IDCOMMA... $\psi$ IDCLIENT $\psi$ IDREGLE... $\psi$ ADRESSE $\psi$ VILLE $\psi$ CODEPOS...
```

Mais il a un produit dans son panier.

```
SELECT * FROM PANIER WHERE IDCLIENT = 28;

$\instyle{\psi} \text{ IDCLIENT } \frac{\psi}{\psi} \text{ IDPRODUIT } \frac{\psi}{\psi} \text{ QUANTITE } \]

1 28 21 28
```

Stock du produit 21:



Après:

Le client commande.

```
EXEC Commander(28, 1, 'Rue de la vie', 'Toulouse', 31000);
```

 $28 \rightarrow ID du client$

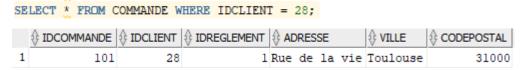
 $1 \rightarrow ID$ du moyen de paiement

'Rue de la vie' → adresse

'Toulouse' \rightarrow ville

 $31000 \rightarrow code postal$

La commande a été créée.



Et le panier est désormais vide.

```
SELECT * FROM PANIER WHERE IDCLIENT = 28;
```

Stock du produit 21 après commande :

		∜ STOCK
1	21	0

Test avec un client inexistant :

```
EXEC Commander(999, 101, 'Rue de la vie', 'Toulouse', 31000);

Erreur commençant à la ligne: 6 de la commande -

BEGIN Commander(999, 101, 'Rue de la vie', 'Toulouse', 31000); END;

Rapport d'erreur -

ORA-01403: aucune donnée trouvée

ORA-06512: à "SAEBD12.COMMANDER", ligne 26

ORA-06512: à ligne l

01403. 00000 - "no data found"

*Cause: No data was found from the objects.

*Action: There was no data from the objects which may be due to end of fetch.
```

Test avec un moyen de paiement inexistant :

```
EXEC Commander(28, 999, 'Rue de la vie', 'Toulouse', 31000);

Erreur commençant à la ligne: 4 de la commande -

BEGIN Commander(28, 999, 'Rue de la vie', 'Toulouse', 31000); END;

Rapport d'erreur -

ORA-01403: aucune donnée trouvée

ORA-06512: à "SAEBD12.COMMANDER", ligne 28

ORA-06512: à ligne 1

01403. 00000 - "no data found"

*Cause: No data was found from the objects.

*Action: There was no data from the objects which may be due to end of fetch.
```