

Dossier de conception trigger procédure BD



I. Trigger	2
REDUCE_STOCK:	2
Procédure	3
DELETE_ALL_COMMANDS:	3
DELETE_ALL_PRODUCT	4
ADD_TO_CART	5

I. Trigger

REDUCE_STOCK:

```
create or replace TRIGGER reduce_stock
AFTER INSERT ON quantiteproduit
FOR EACH ROW
DECLARE
    v_min_stock NUMBER := 0;
BEGIN
    SELECT MIN(quantitestock) INTO v_min_stock FROM produit WHERE idproduit = :NEW.idproduit;

    IF v_min_stock - :NEW.nbproduit < 0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'La quantité de stock ne peut pas être négative');
    ELSE
        UPDATE produit
        SET quantitestock = quantitestock - :NEW.nbproduit
        WHERE idproduit = :NEW.idproduit;
    END IF;
END;
```

Ce déclencheur, appelé "reduce_stock", est déclenché après qu'une nouvelle ligne soit insérée dans la table "quantiteproduit". Il déclare une variable locale "v_min_stock" qui est initialisée à 0. Il sélectionne alors la quantité minimale de stock pour le produit spécifié dans la nouvelle ligne en utilisant l'identifiant de produit de cette ligne. Il vérifie ensuite si la quantité de stock minimale - la quantité de produit spécifiée dans la nouvelle ligne est inférieure à zéro. Si c'est le cas, il ne fait rien. Sinon, il met à jour la table "produit" en réduisant la quantité de stock pour le produit spécifié dans la nouvelle ligne en utilisant l'identifiant de produit de cette ligne.

Procédure

DELETE_ALL_COMMANDS:

```
create or replace PROCEDURE delete_info_user (  
    p_idclient IN NUMBER  
)  
AS  
    commands_count NUMBER;  
    paiement_count NUMBER;  
    panier_count NUMBER;  
BEGIN  
    -- Vérifiez s'il y a un client avec l'identifiant fourni  
    SELECT COUNT(*) INTO commands_count  
    FROM CLIENTS  
    WHERE idclient = p_idclient;  
  
    -- Si aucun client n'a été trouvé  
    IF commands_count = 0 THEN  
        -- Lancez une exception avec un message d'erreur  
        RAISE_APPLICATION_ERROR(-20002, 'Aucun client trouvé avec l''identifiant ' || p_idclient);  
    END IF;  
  
    -- Supprimez toutes les commandes du client  
    DELETE FROM COMMANDE WHERE idclient = p_idclient;  
    DELETE FROM DETAILPAIEMENT WHERE idclient = p_idclient;  
    DELETE FROM PANIER WHERE idclient = p_idclient;  
  
    -- Vérifiez s'il reste des commandes, détails de paiement ou paniers pour le client  
    SELECT COUNT(*) INTO commands_count  
    FROM COMMANDE  
    WHERE idclient = p_idclient;  
  
    SELECT COUNT(*) INTO paiement_count  
    FROM DETAILPAIEMENT  
    WHERE idclient = p_idclient;  
  
    SELECT COUNT(*) INTO panier_count  
    FROM PANIER  
    WHERE idclient = p_idclient;  
  
    -- Si des commandes, détails de paiement ou paniers sont toujours présents  
    IF commands_count > 0 OR paiement_count > 0 OR panier_count > 0 THEN  
        -- Lancez une exception avec un message d'erreur  
        RAISE_APPLICATION_ERROR(-20001, 'Il reste des choses à supprimer');  
    END IF;  
  
    -- Supprimez le client de la table CLIENTS  
    DELETE FROM CLIENTS WHERE idclient = p_idclient;  
END;
```

Cette procédure, appelée "delete_info_user", prend en paramètre un numéro d'identification d'un client (p_idclient). Elle utilise ensuite cet identifiant pour vérifier s'il existe un client correspondant dans la table "CLIENTS". Si aucun client n'est trouvé, elle ne fait rien. Sinon, elle supprime toutes les commandes, les détails de paiement et les paniers associés à ce client dans les tables "COMMANDE", "DETAILPAIEMENT" et "PANIER". Ensuite, elle vérifie s'il reste encore des commandes, des détails de paiement ou des paniers associés au client.

Si c'est le cas, elle ne fait rien. Sinon, elle supprime finalement le client de la table "CLIENTS".

DELETE_ALL_PRODUCT

```
create or replace PROCEDURE delete_all_product (  
    p_idproduit Produit.IDPRODUIT$type  
)  
AS  
    produit_count NUMBER;  
    panier_count NUMBER;  
    quantite_count NUMBER;  
BEGIN  
    -- Vérifiez s'il y a un produit avec l'identifiant fourni  
    SELECT COUNT(*) INTO produit_count  
    FROM PRODUIT  
    WHERE IDPRODUIT = p_idproduit;  
    -- Si aucun produit n'a été trouvé  
    IF produit_count = 0 THEN  
        -- Lancez une exception avec un message d'erreur  
        RAISE_APPLICATION_ERROR(-20002, 'Aucun produit trouvé avec l'identifiant ' || p_idproduit);  
    END IF;  
  
    -- Supprimez toutes les données du produit  
    DELETE FROM QUANTITEPRODUIT WHERE IDPRODUIT = p_idproduit;  
    DELETE FROM CONTENUPANIER WHERE IDPRODUIT = p_idproduit;  
    -- Vérifiez s'il reste des commandes, détails de paiement ou paniers pour le produit  
    SELECT COUNT(*) INTO quantite_count  
    FROM QUANTITEPRODUIT  
    WHERE IDPRODUIT = p_idproduit;  
  
    SELECT COUNT(*) INTO panier_count  
    FROM CONTENUPANIER  
    WHERE IDPRODUIT = p_idproduit;  
    -- Si des commandes, détails de paiement ou paniers sont toujours présents  
    IF quantite_count > 0 OR panier_count > 0 THEN  
        -- Lancez une exception avec un message d'erreur  
        RAISE_APPLICATION_ERROR(-20001, 'Il reste des choses à supprimer');  
        rollback;  
    END IF;  
    -- Supprimez le client de la table CLIENTS  
    DELETE FROM PRODUIT WHERE IDPRODUIT = p_idproduit;  
END;
```

Cette procédure, appelée "delete_all_product", prend en paramètre un numéro d'identification d'un produit (p_idproduit). Elle utilise ensuite cet identifiant pour vérifier s'il existe un produit correspondant dans la table "PRODUIT". Si aucun produit n'est trouvé, elle ne fait rien. Sinon, elle supprime toutes les quantités de produits et les produits associés dans les tables "QUANTITEPRODUIT" et "CONTENUPANIER". Ensuite, elle vérifie s'il reste encore des quantités de produits ou des paniers associés au produit. Si c'est le cas, elle ne fait rien. Sinon, elle supprime finalement le produit de la table "PRODUIT".

ADD_TO_CART

```
create or replace PROCEDURE add_to_cart(  
  p_id_panier IN PANIER.IDPANIER%TYPE,  
  p_id_client IN CLIENTS.IDCLIENT%TYPE,  
  p_id_produit IN PRODUIT.IDPRODUIT%TYPE  
) AS  
  v_count NUMBER;  
  v_prix_produit PRODUIT.PRIXPRODUIT%TYPE;  
  v_nb_produit NUMBER;  
BEGIN  
  -- Vérifier si le panier existe déjà  
  SELECT COUNT(*) INTO v_count  
  FROM panier  
  WHERE idPanier = p_id_panier;  
  IF v_count = 0 THEN  
    -- Insérer un nouveau panier  
    INSERT INTO panier (idPanier, idClient, prixPanier)  
    VALUES (p_id_panier, p_id_client, 0);  
  END IF;  
  -- Récupérer le prix du produit  
  BEGIN  
    SELECT prixProduit INTO v_prix_produit  
    FROM produit  
    WHERE idProduit = p_id_produit;  
  EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
      RAISE_APPLICATION_ERROR(-20201, 'Produit introuvable');  
  END;  
  -- Vérifier si le produit est déjà dans le panier  
  SELECT NBPRODUIT INTO v_nb_produit  
  FROM contenuPanier  
  WHERE idPanier = p_id_panier AND idProduit = p_id_produit;  
  -- Mettre à jour le panier en ajoutant le prix du produit  
  BEGIN  
    IF v_nb_produit = 0 THEN  
      -- Insérer le produit dans le panier  
      INSERT INTO contenuPanier (idPanier, idProduit, NBPRODUIT)  
      VALUES (p_id_panier, p_id_produit, 1);  
      UPDATE panier  
      SET prixPanier = prixPanier + v_prix_produit  
      WHERE idPanier = p_id_panier;  
    ELSE  
      -- Mettre à jour la quantité du produit dans le panier  
      UPDATE contenuPanier  
      SET NBPRODUIT = NBPRODUIT + 1  
      WHERE idPanier = p_id_panier AND idProduit = p_id_produit;  
      UPDATE panier
```

```

        SET NBPRODUIT = NBPRODUIT + 1
        WHERE idPanier = p_id_panier AND idProduit = p_id_produit;
    UPDATE panier
    SET prixPanier = prixPanier + v_prix_produit
    WHERE idPanier = p_id_panier;
END IF;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20301, 'Erreur lors de la mise à jour du panier');
END;
END;
```

Cette procédure, appelée "add_to_cart", prend en paramètres un identifiant de panier (p_id_panier), un identifiant de client (p_id_client) et un identifiant de produit (p_id_produit). Elle utilise ensuite ces identifiants pour vérifier si le panier existe déjà dans la table "panier", et si ce n'est pas le cas, elle crée un nouveau panier avec les informations fournies. Elle récupère ensuite le prix du produit dans la table "produit" en utilisant l'identifiant de produit fourni. Elle vérifie ensuite si le produit est déjà dans le panier en utilisant l'identifiant de panier et l'identifiant de produit fournis. Si le produit n'est pas encore dans le panier, elle l'ajoute en mettant à jour la quantité et le prix total du panier. Sinon, elle met à jour la quantité du produit dans le panier et met à jour le prix total du panier.