BNF101 Base de Données Relationnelles

Correction des Exercices sur les insertions

Mise à jour des données

LMD : INSERTION DES DONNEES

CORRECTION DES EXERCICES :

1/ fichier insert_ed_groupe_par_table_oracle_2024_2025.sql et

Fichier insert_ed_transactions_postgres_2024_2025.sql

a/ existe-t-il un ordre d'exécution des requêtes ?

Réponse :

On insère la totalité des données des 3 tables qui n'ont pas de clé étrangère.

CLIENT, VENDEUR, PRODUIT.

Ensuite, on insère les lignes de la table COMMANDE et les lignes de la table LIGNECOM.

LMD : INSERTION DES DONNEES

CORRECTION DES EXERCICES :

1/ fichier insert_ed_groupe_par_table_oracle_2024_2025.sql et Fichier insert_ed_transactions_postgres_2024_2025.sql

b/A quelle(s) étape(s) peut-on exécuter la directive « commit » ou « rollback ?

Autrement dit : quelle est la position des commit en cas de succès ou des rollback en cas d'échec d'au moins une des requêtes ?

Réponse :

1/Cas des tables qui n'ont pas de clé étrangère : CLIENT, VENDEUR PRODUIT

On exécute le commit après avoir inséré les lignes de chaque table.

Donc 3 commit au total. En cas d'erreur sur le groupe de requêtes qui concerne une table, on exécute un rollback.

2/Cas des tables qui ont une plusieurs clés étrangères : COMMANDE, LIGNECOM

On exécute le commit après avoir inséré la totalité des lignes des deux tables.

En effet une commande qui n'aurait pas toutes ses lignes de commandes ne serait pas complète.

Une ligne de commande qui référence qui commande non insérée ne serait pas intègre.

Donc un commit au total. En cas d'erreur sur le groupe de requêtes qui concerne une table, on exécute un rollback.

Remarque :

Il est déconseillé de désactiver les contraintes dans ce type d'opération.

LMD : INSERTION DES DONNEES

CORRECTION DES EXERCICES :

1/ fichier insert_ed_groupe_par_table_oracle_2024_2025.sql

Fichier insert_ed_transactions_postgres_2024_2025.sql

Exemple d'insertion d'une ligne (row) dans la table CLIENT2 dont la clé primaire (PK) peut être gérée en auto-incrément.

Dans ORACLE :

⇒Utilisation la séquence seq_numcli.

La valeur auto-incrémentée s'écrit seq_numcli.nextval.

Ci-dessous un exemple de requête d'insertion qui utilise une séquence :

INSERT INTO Client2 (numcli, nomcli, adressecli, codepos, ville, tel) VALUES (seq_numcli.NEXTVAL, 'weiz', '8 rue soufflot', 75005, 'paris', 0144447323);

LMD : INSERTION DES DONNEES

Dans POSTGRES :

⇒Utilisation la séquence client2_numcli_seq.

La valeur auto-incrémentée s'écrit NEXTVAL('client2_numcli_seq').

Le nom de la séquence est créé automatiquement au moment de la création de la table client2.

Ci-dessous un exemple de requête d'insertion qui utilise une séquence :

```
INSERT INTO Client2 (numcli, nomcli, adressecli, codepos, ville, tel) VALUES
(NEXTVAL('client2_numcli_seq'), 'weiz', '8 rue soufflot', 75005, 'paris',
0144447323);
```

LMD : INSERTION DES DONNEES

CORRECTION DES EXERCICES :

2/Fichier insert_ed_transactions_oracle_2024_2025.sql et Fichier insert_ed_transactions_postgres_2024_2025.sql

a/ une requête d'insertion d'un nouveau produit

+ une requête d'insertion d'une commande de ce produit pour un client existant

+ une requête d'insertion d'une ligne de commande pour cette commande et pour ce produit.

A quoi sert la ligne de commande ?

La ligne de commande sert à faire la relation entre la commande et le produit commandé.

Expliquez la position des commit ou des rollback ?

Autrement dit : expliquer la position des commit en cas de succès ou des rollback en cas d'échec d'au moins une des requêtes ?

On trouve un commit/rollback tout de suite après l'insertion du nouveau produit. En effet l'enregistrement d'un produit par un commerçant est indépendant de toute commande.

On trouve un commit/rollback après l'insertion de la commande et de la ligne de commande car une commande n'est intègre qu'avec sa ligne de commande associée et réciproquement.

LMD : INSERTION DES DONNEES

CORRECTION DES EXERCICES :

2/Fichier insert_ed_transactions_oracle_2024_2025.sql et Fichier insert_ed_transactions_postgres_2024_2025.sql

b/ **Insertion d'un nouveau client et insertion de sa 1ère commande.**

cas de la relation 1-n entre un nouveau CLIENT et une COMMANDE

Position des commit en cas de succès ou des rollback en cas d'échec d'au moins une des requêtes :

Il existe un commit/rollback unique car dans le cas d'une relation (1,n), un client ne peut pas être enregistré sans commande. Ensuite une commande ne peut pas être enregistrée sans ligne(s) de commande.

Lignes de commandes :

Il existe deux lignes de commande qui relient la commande à deux produits différents :

Un produit de numPro égal à 1 et de nomPro égal à « dvd brazil » et

Un produit de numPro égal à 5 et de nomPro égal à « stabylo »

LMD : INSERTION DES DONNEES

CORRECTION DES EXERCICES :

2/Fichier insert_ed_transactions_oracle_2024_2025.sql et Fichier insert_ed_transactions_postgres_2024_2025.sql

b/ Insertion d'un nouveau client et insertion de sa 1ère commande.

cas de la relation 0-n entre un nouveau CLIENT et une COMMANDE

Position des commit en cas de succès ou des rollback en cas d'échec d'au moins une des requêtes :

Il existe deux commit/rollback :

⇒ Un commit/rollback après l'enregistrement du client.

car dans le cas d'une relation (0,n), un client peut être enregistré sans commande.

⇒ Un commit/rollback après l'enregistrement de la ligne de commande.

Car une commande ne peut pas être enregistrée sans ligne de commande.

Lignes de commandes :

Il existe une ligne de commande qui relie la commande à un seul produit :

Un produit de numPro égal à 6 et de nomPro égal à « tablette nexus 7 »

LMD : INSERTION DES DONNEES

Remarque sur le format de la colonne tel de la table client :

Tel est déclarée avec un type numérique => Tel NUMBER(10).

Ainsi le SGBD (ORACLE ou POSTGRES) supprime le chiffre 0 qui préfixe les numéros de téléphone.

Pour conserver le 0, il faut déclarer la colonne tel avec un type alphanumérique (CHAR ou VARCHAR2).

Dans ORACLE :

ALTER TABLE CLIENT

MODIFY tel CHAR(10);

Dans POSTGRES :

ALTER TABLE CLIENT

ALTER COLUMN tel CHAR(10);

Nota Bene : Avant de modifier le type de la colonne, il faut supprimer les valeurs insérées dans la colonne.

LMD : Mise à jour des données

Modification des données par expression

Il arrive que l'on ait besoin de modifier le contenu d'une (ou plusieurs) colonne pour une occurence ou toute une table, cela peut se faire par SQL en utilisant l'instruction UPDATE avec la forme suivante:

UPDATE nom_de table **SET** nom_colonne= expression,...[**WHERE** condition];

Exemple: S'il s'agit de multiplier le prix unitaire de tous les produits par 1,5, on écrit la requête suivante :

UPDATE PRODUIT **set** PrixUni =PrixUni*1,5;

Remarque:

L'expression qui suit le signe = peut être une constante, une autre colonne ou une combinaison de colonnes et de constantes.

LMD : Mise à jour des données

Modification des données par expression

Autre exemple : Si on a besoin de mettre en majuscule le nom des clients de la table Client.

UPDATE Client **SET** NomCli=UPPER(NomCli);

Suppression des valeurs d'une colonne :

Exemple colonne « tel » de la table CLIENT.

Update CLIENT set TEL=NULL;