

I. Transact-SQL

Pour accroître les possibilités de traitement des données au niveau du serveur, Transact-SQL apporte des possibilités de langage procédural au langage SQL qui est ensembliste.

Ainsi, Transact-SQL prend en compte des fonctionnalités procédurales telles que la gestion des variables, les structures de contrôle de flux, les curseurs et les lots d'instructions. C'est donc un langage complet qui comporte des instructions, qui manipule des objets SQL, qui admet la programmabilité et qui utilise des expressions.

A l'aide de Transact-SQL, il est possible de définir des fonctions et des procédures qui sont directement exécutées sur le serveur de base de données. Ce type de procédure et de fonction est particulièrement intéressant lorsque le traitement nécessite de manipuler un volume d'informations important pour produire le résultat. De même, le développement en Transact-SQL est parfaitement adapté dans un cadre de partage de fonctionnalités car les procédures et fonctions hébergées par le serveur peuvent être exécutées depuis un environnement client quelconque (.NET, Java, ...).

II. Base de données clicom2

Les questions de ce TP concernent la base de données **clicom2** qui a été sauvegardée sous SQL Server Management Studio dans le fichier **clicom2.bak**.

III. Les vues

1) Vue simplifiée de la table produit

a) Donner la requête SQL permettant de créer une vue simplifiée **v_prod(ref, libelle, prix)** de la table **produit** où **ref** correspond au code de la catégorie concaténé au numéro du produit et séparé par "-", **libelle** correspond aux 10 premières lettres du libellé et **prix** est le prix HT du produit.

Remarque : à partir de cette question, sauvegarder les requêtes SQL dans un fichier texte nommé **tp06_nomBinôme1-nomBinôme2.sql**.

b) Après avoir inséré le nouveau produit suivant : libellé = "Microsoft Sculpt Comfort Mouse", prix = 24.95, numéro = "SOU21", vérifier que la vue **v_prod** est bien mise à jour.

2) Créer une vue **habitude_achat** permettant d'afficher et de calculer par ville et par numéro de produit, le montant total commandé.

IV. Les fonctions utilisateur

1) Fonction d'affichage du nombre de commandes passées par un client.

Créer la fonction **nbre_cde()** qui retourne le nombre de commandes passées par un client dont le numéro est passé en paramètre.

2) Fonction d'affichage de la liste des produits ayant un prix inférieur à une valeur.

Créer la fonction table **produits_prix_inf()** qui permet de connaître la liste des produits ayant un prix inférieur à celui passé en paramètre.

3) Fonction d'affichage du nombre et du montant total des commandes passées par un client.

Créer la fonction **analyse_client()** qui retourne, sous forme d'une table à deux lignes (*cf. exemple ci-dessous*), le nombre et le montant total des commandes passées par un client dont le numéro est passé en paramètre.

Voici un exemple d'affichage de la fonction **analyse_client()** pour le client numéro 4 :

Libellé	Valeur
Nombre de commandes	4
Montant total des commandes	20787

V. Les procédures stockées

1) Procédure stockée de suppression d'un client.

a) Créer la procédure stockée **supp_cli()** qui supprime le client dont le numéro est passé en paramètre.

Cette procédure n'effectuera la suppression qu'après avoir vérifié que le numéro de client est valide et que le client n'a pas passé aucune commande.

b) Vérifier la procédure sur un client qui n'existe pas, sur un client qui a passé au moins une commande et sur un client qui n'en a passée aucune.

VI. Les déclencheurs

1) Déclencheur de saisie automatique de la date d'une commande.

Créer le déclencheur **ins_cde_date** qui ajoute la date courante à une commande venant d'être saisie.

2) Déclencheur de calcul du taux de remise suite à une nouvelle commande.

Créer le déclencheur **ins_cde_taux** qui met à jour le taux de remise d'une commande venant d'être saisie.

Ce taux sera de 5% si le client a passé au moins 10 commandes.

Remarque : Afin de simplifier l'algorithme du déclencheur, nous supposerons qu'une seule commande par client peut être saisie à la fois.

3) Vérification des déclencheurs **ins_cde_taux et **ins_cde_date****

Vérifier les déclencheurs **ins_cde_taux** et **ins_cde_date** en saisissant de nouvelles commandes passées par les clients numéro 2 et 3.