

TD-TP DE BASES DE DONNEES

SEANCE 5 : PROGRAMMATION SQL – PL/SQL

L'objectif de cette séance est de se familiariser avec la programmation SGBD avec PL/SQL. Pour cela, vous allez créer des scripts PL/SQL (nommés avec une extension en « .sql ») que vous déclencherez avec la commande @nom_fichier dans SQL*Plus. Ces blocs PL/SQL peuvent à terme être intégrés au serveur sous forme de procédures stockées.

Q1 : Phase préliminaire et prise en main

Créer une table RESULTAT suivante : RESULTAT (CODE number, MESSAGE char(50)). Compléter le programme PL/SQL ci-dessous pour qu'il génère le résultat suivant.

PROGRAMME PL/SQL

```

DELETE FROM RESULTAT ;
PROMPT Nombre de lignes a produire
ACCEPT n
DECLARE
    x NUMBER:= ????????;
BEGIN
    FOR i IN 1..&n LOOP
        IF MOD(x, ??????) = 0 THEN -- x is even
            INSERT INTO RESULTAT VALUES (????????????);
        ELSE
            INSERT INTO RESULTAT VALUES (????????????);
        END IF;
        x := x ???????????? ;
    END LOOP;
    COMMIT;
END;
/
SELECT * FROM RESULTAT ;

```

Contenu table RESULTAT après exécution

CODE	MESSAGE
1	100 is even
2	101 is odd
3	102 is even
...	etc

Q2 : Bloc PL/SQL, expression conditionnelle

Charger le scripts CREATE-bis.sql et CREATE-bis-data.sql. Ecrire un programme PL/SQL qui :

- demande un numéro de client
- insère un tuple dans la table RESULTAT et le visualise. Ce tuple est tel que :
 - s'il n'y a pas de commandes pour ce client, il a comme valeur 'pas de commande'
 - sinon, il a comme valeur 'n commandes pour le client X', avec n le nombre de commandes pour X.

Q3 : Bloc PL/SQL, curseurs et boucles

Ecrire un programme qui permette d'afficher les nième et n+1ième commandes les plus récentes (afficher le numéro de la commande, la date, et le prix total) et d'insérer les informations correspondantes dans la table RESULTAT. Le prix total de la commande sera calculé à partir des détails et des prix des produits commandés. Le nombre n est un paramètre saisi par l'utilisateur.

Q4 : Bloc PL/SQL, curseurs paramétrés

Ecrire un programme qui utilisera un curseur paramétré et qui permette de vérifier que le montant payé (COM.Payement) pour une commande donnée (renseignée par l'utilisateur) est égal à la somme des lignes de commandes correspondantes. Pour la commande demandée, une ligne sera insérée dans la table résultat. Cette ligne aura la forme suivante : « NumCom : 100 - Payement : 5000 - Prix : 5000 ».

Q5 : Bloc PL/SQL, exceptions

Compléter le programme de la question 3 pour gérer l'erreur survenant dans le cas où le nombre N serait strictement supérieur au nombre de commandes dans la table Commandes. Dans ce cas, un message d'erreur sera inséré dans la table résultat.

Q6 : Bloc PL/SQL, packages

Ecrire le package comportant une procédure de tri des clients dans l'ordre décroissant de leur montant total de commandes qui fournit la liste triée des identifiants clients et du montant total de leurs commandes dans la table résultat, et une fonction qui renvoie l'identifiant du client ayant le montant de dépenses le plus élevé.