

M1 MIAGE FC/FA – 2010/2011

Bases de Données Avancées

novembre 2010

PL/SQL avancé

Exercice 1 : SQL DYNAMIQUE

Question 1.1 : Créer une procédure stockée `menage1` qui prend en paramètre un nom de table et supprime cette table

Question 1.2 : Créer une procédure stockée `menage2` qui prend en paramètre un "masque" de nom d'objet, un type d'objet et supprime les objets de ce type, dont le nom correspond au masque. Par exemple `menage2('%PE%', 'TABLE')` supprime les tables dont le nom contient 'PE'. Vous utiliserez la vue `USER_OBJECTS` pour connaître les objets à supprimer.

Exercice 2 : CURSEURS "FOR UPDATE"

On considère une table contenant des équipes de football, et des tables (1 par journée de championnat) contenant les points remportés par chaque équipe lors d'une journée de championnat. Ces tables ont été créées de la façon suivante :

```
create table Equipe(  
    id number(2) constraint equipe_pkey primary key,  
    nom varchar2(50) not null,  
    total number(2) default 0 not null,  
    classement number(2) default 0 not null  
);  
  
create table Points_j1(  
    id_equ constraint pointsj1_equi_fkey references equipe,  
    points number(1)  
);  
  
create table Points_j2(  
    id_equ constraint pointsj2_equi_fkey references equipe,  
    points number(1)  
);  
  
create table Points_j3(  
    id_equ constraint pointsj3_equi_fkey references equipe,  
    points number(1)  
);
```

Question 2.1 : Créez ces tables et remplissez-les, à partir des tables `EQUIPE`, `POINTS_J1`, `POINTS_J2`, `POINTS_J3` de l'utilisateur `CARON`.

On souhaite calculer pour chaque équipe le total de ses points, à partir des tables `POINTS_J1`, `POINTS_J2`, `POINTS_J3`, c'est à dire mettre à jour la colonne `total` de la table `EQUIPE`.

Les trois questions suivantes vont fournir trois façons de calculer cette colonne **total** à partir des tables **POINTS_Ji**.

Question 2.2 : Ecrivez une instruction **update** qui permet de mettre à jour la colonne **total** à partir des tables **POINTS_J1**, **POINTS_J2**, **POINTS_J3**. Quel est le plan d'exécution de cette requête ? Combien de fois chaque table **POINTS_Ji** est-elle parcourue lors de cet **update** ?

Question 2.3 : Ecrire une procédure stockée **calcul_points** qui met à jour la colonne **total** de la table **EQUIPE** en parcourant un curseur sur les tables contenant les points. Combien de fois chaque table **POINTS_Ji** est-elle parcourue ? Combien d'instructions **update** sont exécutées ?

Question 2.4 : Transformer la procédure stockée précédente pour que l'instruction **update** profite du curseur. Le programme va parcourir un curseur sur toutes les tables (**Equipe** et les **POINTS_Ji**) et modifie via ce curseur la colonne **total** de la table **EQUIPE**.

On souhaite maintenant calculer le classement de chaque équipe.

Comparer les réponses aux deux questions suivantes, qui permettent de calculer les classements des équipes :

Question 2.5 : Ecrire une instruction **update** qui calcule tous les classements des équipes.

Question 2.6 : Ecrire une procédure stockée **calcul_classements** utilisant un curseur **for update** grâce auquel on peut mettre à jour la colonne **classement** de la table **EQUIPE**.

Exercice 3 :

On utilise les tables **Pays** et **Ville** du TP précédent.

Les questions suivantes vous permettent de tester l'acquisition des données par "paquets" lors de l'exécution de requêtes SQL en PL/SQL. Vue la faible quantité de données, il ne s'agit pas ici d'une optimisation !

Question 3.1 : Ecrire une procédure **afficher_villes** qui prend en paramètre un masque de nom de ville et affiche toutes les villes correspondant à ce masque. Vous utiliserez un **select ... bulk collect into ...** pour récupérer les lignes de **Ville**

Question 3.2 : Ecrire une procédure **afficher_villes_bis** qui a le même comportement que la procédure précédente, mais récupère les lignes de **Ville** par paquets de 25 lignes.

Question 3.3 : Ecrire une procédure **supprimer_villes** qui prend en paramètre un masque de nom de ville, supprime toutes les villes correspondant à ce masque et affiche les noms des villes supprimées.