

Exercices PL/SQL

Description de la base de données Taxi

Pour les exercices suivants, la bases de données Taxi sera utilisée. Les tables de la bases de données sont les suivantes :

```
taxi      (id_taxi, marque, km, nb_place)
chauffeur (id_chf, nom_chf, prenom_chf)
reparation (id_rep, #id_taxi, desc_rep, km_cpt, prix)
course    (id_course, #id_taxi, #id_chf, adresse_dep, adresse_ar,
           nb_personne_course, duree_course, prix_course, date_course)
```

La table `taxi` contient un identifiant, la marque du véhicule, le kilométrage du véhicule ainsi que le nombre de places passager.

La table `chauffeur` contient un identifiant de chauffeur, le nom et le prénom du chauffeur.

La table `réparation` contient l'identifiant du taxi réparé, la description de la réparation, le kilométrage du taxi lors de la réparation et le prix de la réparation.

La table `course` comporte un identifiant, l'identifiant du taxi et du chauffeur de la course, l'adresse de départ et l'adresse d'arrivée, le nombre de personne de la course, sa durée, son prix et la sa date.

Procédures et fonctions

1. Écrire une fonction PL/SQL qui retourne le $n^{\text{ième}}$ nombre de Fibonacci.
Écrire une version récursive et une version itérative.
2. Définir une procédure PL/SQL qui modifie le nom des chauffeurs de taxi tel que la première lettre du nom soit en majuscule et le reste des lettres en minuscule. Les nom des chauffeurs corrects ne seront pas mis à jour.
La procédure affichera aussi le nombre de n-uplets modifiés s'il y en a et le message « Aucune ligne modifiée » s'il n'y en a pas.
3. Écrire une fonction qui retourne le nombre de trajets enregistrés de l'adresse `adresse_depart` à `adresse_arrivée` (la fonction a 2 paramètres).
4. Écrire une procédure qui ajoute une réparation de type « Révision » dont le coût est 60 euros pour tous les taxis n'ayant pas eu de réparation depuis au moins 30 000 km.
La procédure utilisera un curseur.
5. Définir une procédure qui prends en paramètre une requête écrite sous la forme d'une chaîne de caractère, qui exécute la requête et qui affiche le temps d'exécution.
Indice : `DBMS_UTILITY.GET_TIME` renvoie l'heure courante.
`EXECUTE IMMEDIATE ...` permet d'exécuter une requête dynamique (requête écrite sous la forme d'une chaîne de caractère).
6. Définir une procédure faisant N insertions ou N mises à jour.
Comparer N insertions standards et en mode bulk (`FORALL`).