

TP 5

PL/PSQL : encore des fonctions

Ce TP est la continuation du TP précédent. Il porte sur les mêmes tables. Le texte du TP precedent est sur Didel.

Il est à noter que la plupart des fonction PLSQL que l'on vous demande de coder sont faisables de maniere plus efficace avec une requête SQL, (cf. exercice 4 et 5 du TP4).

1 Une fonction qu'on ne sait pas faire en SQL

Ecrivez une fonction qui prend en argument un `trip_id` et un nombre entier `n` de minutes et calcule sur le trip dont le numero est fourni, la derniere station a moins de `n` minutes du debut, puis la station a moins de `n` minutes de la precedente... Le resultat sera retourne sous forme d'une table contenant le numero de la station par rapport au trajet (`stop_sequence`) et le nom de la station.

Attention: `trip_id` est de type `bigint`.

conversion de date : pour convertir une chaîne de caracteres du type `\19:20:00` en heure (et jour) on utilisera : `to_timestamp(chaine, 'HH24:MI:SS')`. Il n'y a pas de fonction de conversion en heure seule.

Pour obtenir un intervalle de 1 minutes on écrira : `interval '1 minutes'`

2 Boucles sur des résultats

Ecrivez une fonction qui verifie que les dates de `calendar_dates` pour un service donne sont toutes dans l'intervalle correspondant de `calendar`.

A chaque fois qu'une erreur est detectee vous devez ecrire une ligne d'avertissement avec `RAISE NOTICE` indiquant le numero de service, la date et l'intervalle trouve. Il y a trois erreurs.

Les fonctions de conversions de date utiles sont decrites dans le TP4.

3 Curseurs

Re-ecrivez la fonction precedente avec un ou des curseurs.

4 Execute

Lorsque les requêtes ne sont pas toutes connues à l'avance, on peut les créer en cours d'exécution puis les exécuter à l'aide de la commande `EXECUTE`. Un exemple typique sont les requêtes avec un nombre variable de clauses `WHERE`.

Dans cet exercice, on vous demande d'écrire une fonction qui prend en entrée les arguments `dateDebut DATE`, `dateFin DATE`, et construit la requête qui retourne le nombre de `service_id` dans la table `calendar_dates`. Si l'argument `dateDebut` n'est pas `NULL`, il faut retourner le nombre de `service_id` après `dateDebut` ; si l'argument `dateFin` n'est pas `NULL`, il faut exclure les `service_id` qui ont lieu après `dateFin`. Votre requête aura donc entre 0 et 2 clauses `WHERE`.