

TP9

PL-PgSQL et Triggers

1 Fonctions

Hello World

Fonctions traitant des chaînes de caractères:

1. Écrire une fonction `hello` qui renvoie la chaîne de caractère "Hello World!" en **SQL**.
2. Écrire une fonction `hello_pl` qui renvoie la chaîne de caractère "Hello World!" en **PL/pgSQL**.
3. Comparer les coûts des deux plans d'exécutions de ces requêtes. Expliquer les coûts.

Arithmétique

1. Écrire en PL/pgSQL une fonction de division appelée `division`. Elle acceptera en entrée deux arguments de type entier et renverra un nombre réel (*numeric*).

2. Écrire cette même fonction en SQL.
3. Comment corriger le problème de la division par zéro ? Écrire cette nouvelle fonction dans les deux langages.
(Conseil : dans ce genre de calcul impossible, il est possible d'utiliser la constante `NaN` (*Not A Number*)).

Sélection

À partir de la base de données cave :

1. Écrivez une fonction `get_bordeaux` qui retournent les libelles tous les vins de Bordeaux.
2. Écrivez une fonction `vin_region_par_type` qui prend en paramètre un type de vin (*rouge, rose, blanc*) et une année et retourne la répartition des bouteilles en stock par région de production.
3. Écrivez une fonction `nb_par_recoltant_par_an` qui prend en paramètre une *annee* (int) et retourne le nombre de vins par récoltant de l'année en paramètre
4. Écrivez la fonction `nb_bouteilles_1` qui prend en paramètre d'entrée une année et un type de vin et retourne le nombre de bouteille de ce type produite.
5. Écrivez la fonction `nb_bouteilles_2` qui prend en paramètre d'entrée une liste variable d'années à traiter et un type de vin et retourne le nombre de bouteille du type produite par année.

2 Triggers

Dans cette partie, nous allons travailler avec la base de données cave

Triggers-1

Tracer dans une table toutes les modifications du champ *nombre* de la table *stock*.

On veut conserver l'ancienne et la nouvelle valeur. On veut aussi savoir qui a fait la modification et quand. Interdire la suppression des lignes dans stock. Afficher un message dans les logs dans ce cas.

Afficher aussi un message **NOTICE** quand nombre devient inférieur à 5, et **WARNING** quand il vaut 0.

Triggers-2

Interdire à tout le monde, sauf un compte admin, l'accès à la table des logs précédemment créée . En conséquence, le trigger fonctionne-t-il ? Le cas échéant, le modifier pour qu'il fonctionne.

Triggers-3

Lire toute la table *stock* avec un curseur.

Afficher dans les journaux applicatifs toutes les paires (*vin_id*, *contenant_id*) pour chaque nombre supérieur à l'argument de la fonction.