

Prise en main de PL/pgSQL

Consignes

Vous effectuerez ce TP dans votre base de données. Durant ce TP vous devrez utiliser le langage procédural pl/pgsql. Ce langage doit être « installé » dans toute base de données amenée à l'utiliser. Pour cela, vous devez taper la commande :

```
CREATE LANGUAGE plpgsql HANDLER plpgsql_call_handler VALIDATOR plpgsql_validator;
```

Pour toutes informations complémentaires, n'hésitez pas à consulter la doc postgresQL :

Langages de procédures : <http://docs.postgresql.fr/8.3/xplang.html>

Exercices

1. Écrire une fonction sans argument qui affiche l'information (notice) « Hello World ».
2. Écrire une fonction qui prend en entrée un prénom et renvoie (return) « Bonjour *Prénom* ». Pour plus de politesse, assurez vous que le prénom commence toujours par une majuscule suivie de minuscule (utilisez une fonction existante).
3. Écrire une fonction de comparaison de deux entiers passés en argument d'entrée. Cette fonction, nommée *compar*, affichera en information la valeur de chaque entier et retournera un record dont le premier champ contiendra 'min=*valeur_min*' et le second 'max=*valeur_max*'. En cas d'égalité, le record ne contiendra qu'un champ 'equal=*valeur*'.
Exemple :

```
nath=# select compar(4,3);
NOTICE: a vaut 4
NOTICE: b vaut 3
 compar
-----
(min=3,max=4)
(1 ligne)

nath=# select compar(3,3);
NOTICE: a vaut 3
NOTICE: b vaut 3
 compar
-----
(equal=3)
(1 ligne)

nath=# select compar(3,4);
NOTICE: a vaut 3
NOTICE: b vaut 4
 compar
-----
(min=3,max=4)
(1 ligne)
```

4. Écrire une fonction prenant en argument un entier et retournant son factorielle. Ne pas utiliser la fonction factorielle mais ré-implémenter en itératif et en récursif. Vous devez gérer les erreurs dues aux nombres négatifs (ex : « Merci de saisir un nombre positif »)

5. Écrire une fonction prenant en argument un entier et affichant sa table de multiplication. Trois solutions possibles, exposées de la plus simple à la plus complexe (mais la plus intéressante) :

- v1 - renvoyer le résultat dans un tableau
- v2 - créer une table et stocker le résultat dans la table
- v3 - renvoyer une table. *Indice : returns setof record*

```
nath=# select tablemul(5);
      tablemul
-----
{5,10,15,20,25,30,35,40,45,50}
(1 ligne)
```

v1 :

```
a | b | mul
---+---+---
(0 lignes)

nath=# select tablemul2(5);
      tablemul2
-----
(1 ligne)

nath=# select * from res;
 a | b | mul
---+---+---
 1 | 5 |  5
 2 | 5 | 10
 3 | 5 | 15
 4 | 5 | 20
 5 | 5 | 25
 6 | 5 | 30
 7 | 5 | 35
 8 | 5 | 40
 9 | 5 | 45
10 | 5 | 50
(10 lignes)

nath=# select tablemul2(6);
      tablemul2
-----
(1 ligne)

nath=# select * from res;
 a | b | mul
---+---+---
 1 | 6 |  6
 2 | 6 | 12
 3 | 6 | 18
 4 | 6 | 24
 5 | 6 | 30
 6 | 6 | 36
 7 | 6 | 42
 8 | 6 | 48
 9 | 6 | 54
10 | 6 | 60
(10 lignes)
```

v2 :

```
nath=# select * from tablemul3(4) as result(a integer, b integer, mul integer);
a | b | mul
-----+-----+-----
1 | 4 | 4
2 | 4 | 8
3 | 4 | 12
4 | 4 | 16
5 | 4 | 20
6 | 4 | 24
7 | 4 | 28
8 | 4 | 32
9 | 4 | 36
10 | 4 | 40
(10 lignes)
```

v3 :

6. Créer une fonction qui renvoie le nombre d'instruments utilisés par un interprète donné (nom et prénom en paramètre d'entrée).

7. Écrire une fonction permettant de tester la syntaxe d'une requête passée en argument. Cette fonction devra renvoyer les messages suivants selon la syntaxe de la requête:

- 'requête syntaxiquement correcte' ;
- 'une table de la requête n'existe pas' ;
- 'une colonne de la requête n'existe pas'

Dans le cas où plusieurs erreurs se cumuleraient, une seule (la première détectée) sera renvoyée.

Bonus : Nouvel affichage de la question 3.

```
nath=# select * from compar2(3,3) as tmp (min integer, max integer, equal integer);
NOTICE: a vaut 3
NOTICE: b vaut 3
min | max | equal
-----+-----+-----
0 | 0 | 3
(1 ligne)

nath=# select * from compar2(3,5) as tmp (min integer, max integer, equal integer);
NOTICE: a vaut 3
NOTICE: b vaut 5
min | max | equal
-----+-----+-----
3 | 5 | 0
(1 ligne)

nath=# select * from compar2(6,5) as tmp (min integer, max integer, equal integer);
NOTICE: a vaut 6
NOTICE: b vaut 5
min | max | equal
-----+-----+-----
5 | 6 | 0
(1 ligne)
```

Vidéo club

L'association de 10 magasins de vidéos club va ouvrir ! Implémenter le MR réalisé en TD.
Attention cette base de données servira pour le prochain TP... donc à réaliser AVANT le prochain TP.