

## TP3 – fonctions et triggers

Dans cette séance vous devrez écrire des fonctions simples puis des triggers. Merci de respecter les deux consignes suivantes :

- Pour chaque fonction vous devez écrire un nombre suffisant de tests unitaires (autant que nécessaire pour valider tous les cas) pour montrer que votre fonction a bien le résultat attendu.
- A chaque fois que vous avez terminé une fonction, appelez l'enseignant pour qu'il note l'exercice correspondant. Vous devrez également déposer sur moodle un fichier avec tout ce que vous avez fait mais la correction se fera durant la séance !

Les étoiles à côté de chaque exercice correspondent au temps estimé : \* = 10mn.

### 1. Ecriture de fonctions simples

Exercice 1. (\*) Ecrire une fonction qui prend deux chaînes et retourne la plus longue. La fonction retourne la première chaîne si elles ont la même longueur. Voir slide 13.

Exercice 2. (\*) Ecrire une fonction qui prend un texte et affiche (notice) tous les mots un par un. N'oubliez pas que les notices s'affichent dans la fenêtre messages. On considère que les mots sont séparés par des espaces et rien d'autre. Voir slide 13.

Exercice 3. (\*) Ecrire une version itérative de la fonction factorielle. Votre fonction doit avoir le même comportement que la version récursive du slide 11 du cours.

### 2. Fonctions sur tables

D'après la base ci-dessous (création + peuplement disponible sur moodle) :

```
locations (NoDVD#, DateLocation, NoClient#, Dureelocation)
dvds (NoDVD, PrixAchat, Titre#)
films (Titre, Duree, NomEditeur#, Realisateur, AnneeSortie, Genre)
editeurs (NomEditeur, AdresseEditeur)
clients (NoClient, NomClient, AdresseClient)
```

Exercice 4. (\*\*) Calculer la somme des durées de tous les films en écrivant une requête SQL. Puis faire la même chose avec une fonction parcourant tous les enregistrements un par un (avec for x in et un RECORD, voir slide 34) et retournant le même résultat. Assurez-vous que les deux méthodes retournent le même résultat.

Exercice 5. (\*\*) Calculer le prix moyen d'achat des films en écrivant une requête SQL puis refaites de même avec une fonction parcourant les enregistrements avec un curseur (CURSOR, voir slide 38) et une boucle. La fonction devra lever une exception s'il n'y a aucun élément dans la table. Assurez-vous que les deux méthodes retournent le même résultat et testez l'exception en vidant la table (TRUNCATE TABLE).

Exercice 6. (\*\*\*) Les NoDVD dans la table dvd sont complètement chamboulés. Ecrire une fonction pour les renuméroter à partir de 1 et de manière consécutive. Faites ce qu'il faut pour que vos modifications soient répercutées automatiquement dans toute la base, par exemple avec un ON UPDATE CASCADE.

### 3. Triggers

Exercice 7. (\*) Ecrire un trigger (avec la fonction associée) qui lève une exception si on tente d'insérer ou de modifier un film en mettant une année de sortie inférieure à 1891. L'exception doit afficher l'année que l'on tente d'insérer.

Exercice 8. (\*) Ecrire un trigger (avec la fonction associée) pour que si un dvd est inséré dans la table sans prix d'achat indiqué (par exemple « INSERT INTO dvds (NoDVD, Titre) VALUES (1001, '8 mm'); ») alors le prix d'achat est fixé au prix d'achat moyen des dvds correspondant au même film (possédant le même titre). Si le prix est indiqué il faut juste faire l'insertion normalement.

Exercice 9. (\*\*) Choisir une clé primaire dans une table quelconque et écrivez un trigger qui va faire la vérification de clé primaire, c'est-à-dire qu'à chaque insertion ou modification dans la table choisie, le trigger va empêcher toute duplication.

Exercice 10. (\*\*\*\*) Choisir une clé étrangère dans une table quelconque et écrivez deux triggers qui vont faire la vérification de clé étrangère avec modifications en cascade, c'est-à-dire :

- Dans la table où la clé étrangère est créée, tout ajout ou modification va vérifier que la cible existe bien dans la table référencée. En cas d'erreur vous devez afficher une erreur similaire à celle de postgres.
- Dans la table référencée, toute modification de la clé unique (ou primaire) référencée, va modifier automatiquement la table référençante.

### 4. Exercices bonus (à la maison)

Déposez sur le dépôt « bonus » deux triggers pertinents en lien avec la base. Ce dépôt sera utilisé pour la deuxième chance si nécessaire.