

TP n° 1 : Fonctions et triggers

Evaluation du TP – rappel des principes :

- **Tous** les TP seront **évalués**, à la fois
 - o En **présentiel**, en direct durant la séance (points de validation)
 - o **Dépôts** des TP sur **Moodle** quelques heures après la fin des TP
 - o Le dépôt sur Moodle ne dispense pas de la validation en séance... et la validation en séance comptera plus dans la note que le dépôt Moodle après la fin du TP...
 - o Donc, si vous avez des aménagements particuliers ou un empêchement ponctuel pour un TP donné, envoyez-moi un mail en avance à mohammed.naas@univ-lr.fr
- Pour la note finale, nous nous réservons le droit de ne compter que certains des TP (en cas de problèmes techniques lors d'un TP par exemple, celui-ci ne serait pas décompté dans la note finale)
- Les étudiants faisant partie d'un même binôme pourraient ne pas avoir la même note pour un TP donné, en fonction de leur implication en séance

Evaluation du TP – en pratique :

- A chaque fois que vous atteignez un point de validation (en rouge et encadré), appelez un intervenant pour qu'il note l'exercice correspondant. Ce sera la partie « présentiel » de la note de TP, qui ne sera pas forcément la même pour les 2 membres d'un binôme en fonction de l'implication de chaque étudiant.
- Si vous êtes parmi les 2-3 premiers binômes à atteindre un point de validation donné, et/ou que vous avez trouvé une solution particulièrement élégante, vous pourrez bénéficier d'un petit bonus sur la notation 😊
- Si vous levez la main mais que les deux intervenants sont occupés, continuez à vous signaler aux intervenants, mais continuez aussi à travailler sur le point de validation suivant. Vous pourrez toujours valider deux points de validation à la fois.
- Pour le rendu du TP sur Moodle, merci de déposer un unique fichier par binôme, nommé avec les noms des deux membres du binôme. Le dépôt doit se faire avec **un fichier de type .txt** (texte sans formatage). **Tous les autres types de fichiers seront interdits.**
- Pensez à bien numéroter les questions pour le rendu Moodle. Si vous les numérotez mal, ou que vous changez le nom des tables, attributs, fonctions, etc par rapport à l'énoncé, nous nous réservons le droit de vous mettre 0. Imaginez que 120 étudiants nous rendent des questions en vrac, ou avec des noms de tables dans lesquels on ne se retrouve pas !!!

Contenu du TP :

Dans cette séance vous devrez écrire des **fonctions simples** puis des **triggers** (voir le CM2 sur Moodle). Merci de respecter la consigne suivante :

- **Pour chaque fonction, avant d'appeler l'intervenant pour valider un point de validation, vous devez écrire un nombre suffisant de tests unitaires (autant que nécessaire pour valider tous les cas possible), pour nous montrer que votre fonction retourne bien le résultat attendu.**

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

1. Ecriture de fonctions simples

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

Exercice 2. (*) Ecrire une fonction qui prend un texte et affiche (notice) tous les mots un par un, en utilisant un tableau de mots. On considère que les mots sont séparés par des espaces et rien d'autre.
N.B. N'oubliez pas que les notices s'affichent dans la fenêtre « Messages » 😊.

Exercice 3. (*) Ecrire une version itérative de la fonction factorielle. Votre fonction doit avoir le même comportement que la version récursive du slide 7 du cours CM2.

Point de validation V1, V2, V3

2. Fonctions sur tables

On travaille maintenant sur la base dont le schéma est 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.
- Faire la même chose avec une fonction parcourant tous les enregistrements un par un (avec FOR x IN et un RECORD).
- Vérifiez bien que les deux méthodes retournent le même résultat.

Point de validation V4

Exercice 5. (**)

- a- Calculer le prix moyen d'achat des films en écrivant une requête SQL.
- b- Refaites de même avec une fonction parcourant les enregistrements avec un curseur (**CURSOR**) et une boucle. La fonction devra lever une **exception** s'il n'y a aucun élément dans la table. Testez l'exception en vidant la table (**TRUNCATE TABLE**), puis (évidemment !) re-peuplez la table avec le code de peuplement fourni sur Moodle.
- c- Assurez-vous que les deux méthodes retournent le même résultat.

Point de validation V5

Exercice 6. (*)** Les NoDVD dans la table DVD sont complètement dans le désordre, et en plus leur numérotation fait des gros « sauts ». Par exemple, si on les ordonne par nodvd (select nodvd from dvds order by nodvd;), on passe de 2 à 2638 direct. C'est loin d'être optimal en termes de stockage. Ecrire une fonction pour les renuméroter à partir de 1 et de manière consécutive. Il s'agit de les ordonner par nodvd du plus petit au plus grand, mais ensuite de renuméroter leurs nodvd pour qu'ils soient 1, 2, 3, etc.

Faites ce qu'il faut pour que vos modifications soient répercutées automatiquement dans toute la base, par exemple avec un **ON UPDATE CASCADE**.

N.B. Dans une base opérationnelle sans backup et à moins de vraiment maîtriser ce que l'on fait, je ne recommande pas l'utilisation des CASCADE. Mais ici, c'est un TP et vous avez les fichiers de peuplement, donc allez-y ;-)

Point de validation V6

Note : normalement, vous devriez être à peu près à 1h45 depuis le début de la séance...

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 (entre autres) l'année que l'on tente d'insérer.

Point de validation V7

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 dvd (NoDVD, Titre) VALUES (1001, '8 mm'); ») alors le prix d'achat est fixé au prix d'achat moyen des DVDs possédant le même titre (car, si les titres sont les mêmes, on fait l'hypothèse qu'il s'agit du même film, d'ailleurs le titre est la clé primaire de la table films...). Attention ! Le prix d'achat est un entier, mais la moyenne n'est pas forcément un nombre entier...

Si le prix est indiqué, il faut juste faire l'insertion normalement.

Point de validation V8

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 insertion d'une valeur doublon de la clé primaire, et lever une exception avec un message personnalisé (un peu à la manière du slide 21 du CM2).

Point de validation V9

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 référençante (celle où la clé étrangère est déclarée), tout ajout ou modification va vérifier que la nouvelle valeur de clé existe bien dans la table référencée. Si ce n'est pas le cas, vous devez retourner l'erreur « Ce DVD n'existe pas dans la base, impossible de le louer » (avec l'apostrophe dans le message 😊).

Point de validation V10a

- Dans la table référencée, toute modification de la clé unique (ou primaire) référencée, doit modifier automatiquement la table référençante. **N.B.** Ici, n'utilisez pas de CASCADE, c'est le rôle du trigger de faire cela ! Levez une « notice » avec la liste des modifications effectuées (ce que ne fait pas l'option CASCADE malheureusement).

Point de validation V10b