



Objectif : le but de ce TP est de tester des contraintes via des déclencheurs

Aide-mémoire : Définition d'un déclencheur

```
CREATE [OR REPLACE] TRIGGER nom_declencheur
BEFORE | AFTER
INSERT | DELETE | UPDATE          | INSERT [[OR] DELETE] [[OR] UPDATE]
ON nom_table
[FOR EACH ROW [WHEN <condition>]]
Bloc PL/SQL
```

Problème :

Soit la base de données 'SeancesSports' qui gère les séances de sports animées par des coachs sportifs. Une salle de sport fait appel à de nombreux coachs sportifs, et chacun des coachs sportifs peut animer plusieurs séances dans des salles de sport différentes. Une salle de sport à un budget annuel qui permet de payer entre autres les coachs. Chaque coach est payé par heure selon la durée réelle en nombre d'heures de séances effectuées, l'heure est tarifée à 1000 da. Voici le schéma relationnel relatif à cette gestion

Coach(NumC, nomc, prenomc, RevenuTot, etat); etat : 'actif', 'inactif'

Salle(NumS, DirS*, titre, budget, Nb_Coachs) ;

Séance(NumSe, NumS *, NumC *, Heure_Debut, Heure_Fin, Durée_Prévue, Durée_Réelle, nb_Participants);

A- Transformation et mise à jour automatique des données

- 1- Le nom du Coach doit être en majuscule. Transformer en majuscule la valeur du nom du coach quel que soit son format d'origine (utilisation de UPPER())
- 2- Gérer la clé primaire de NumS de manière automatique.

B- Propager des mises à jour sur des données

- 3- Le champ "Durée_Prévue" représente la durée en heures. Ce champ est égal à (Heure_Fin-Heure_Debut). Il est renseigné de manière automatique au moment de l'insertion dans la table séance. Cette valeur sera aussi attribuée au champ Durée_Réelle. À la fin de la séance du sport, si l'heure de fin n'est pas la même que celle prévue. Alors une modification sera apportée à la table séance en modifiant l'Heure_de_Fin. Le champ 'Durée_Réelle' sera alors mis à jour automatiquement à partir de la nouvelle valeur de 'Heure_de_Fin'.

Remarque : Les champs Heure_Fin et Heure_Debut sont des champs de type TIMESTAMP. La différence entre ces deux champs donne un résultat de type INTERVAL. Ce type contient le nbre de jours, heures et secondes. Pour pouvoir extraire

le nombre il faudra utiliser la fonction Extract(hour from *variable*). Pour extraire les minutes Extract(minute from *variable*)

- 4- Mettre à jour le **revenu total** d'un coach.
- 5- Mettre à jour le nombre de coachs d'une salle avec un déclencheur global

C- Empêcher les mises à jour de la base qui altèreraient l'intégrité des données.

- 6- Refuser de nommer un directeur d'une salle un coach qui est inactif.
- 7- Interdire au directeur d'une salle d'animer des séances dans son propre établissement (salle).
- 8- Limiter le nombre de séances d'un coach dans une même journée à 3

D- Historiser et auditer les opérations sur une table

- 9- On désire vérifier que la durée de la séance ne dépasse pas la durée réelle pour une séance. En cas de dépassement, on écrit un message dans la table TAB_LOG prévue pour signaler ce dépassement. Le schéma de la table TAB_LOG est **TAB_LOG**(Num_D, date_D, NumS*, message). Le champ **Num_D** est un champ qui doit se remplir et s'incrémenter automatiquement
 - Ecrire le déclencheur qui permet de répertorier l'information de dépassement
- 10- La TAB_LOG est une table qui ne doit pas être modifiée, en cas de modification de celle-ci, on veut connaître celui qui l'a modifié (l'utilisateur courant USER) et la date de cette modification. Ces informations seront automatiquement insérées dans la table **AuditTable_Log**(Num_Pb, Modifie_par, Date_Modif)
 - Ecrire le déclencheur qui permet d'inscrire ces modifications dans la table AuditTable_Log