

**TP N°8 (semaine 50) : Procédures – Fonctions et Packages****Exercice N°1 : Mise en place des données**

On donne les tables suivantes

PERSONNE (idPers, nom, solde)

SPECTACLE (idSpect, nbPlaces, nbPlacesLibres, tarif)

RESERVATION (idPers, idSpect, dateResa, nbPlaces)

- a) Lancer le script fourni **spectacle.sql** qui crée les deux tables et insère quelques données dans les tables. On donne également les procédures suivantes :
- infoPers (p\_idPers) qui affiche le nom et le solde de la personne correspondant ou qui affiche « *Personne inexistante* ». Elle liste également toutes les réservations de la personne.
  - infoSpect (p\_idSpect) qui affiche le titre, le nombre de places disponibles du spectacle correspondant ou qui affiche « *spectacle inexistant* ».
  - suppers (p\_idPers) qui supprime la personne dont l'id est passé en paramètre
- b) Modifier la procédure suppers pour qu'elle affiche « *Personne inexistante* » si tel est le cas.
- c) Tester (on utilise la commande EXECUTE : EXEC infoPers(1) par exemple) :
- Afficher les infos de la personne 1
  - Tenter de supprimer la personne 3
  - Afficher les informations du spectacle 2
  - Afficher les informations du spectacle 3

**Exercice N°2 : Création de fonctions/ procédures**

- a) Ecrire la fonction **genIdPers** qui renvoie un nouveau numéro de personne à partir du nb de personnes déjà présents dans la table (+1)
- b) Écrire les procédures suivantes :
- ajoutPers (p\_nom, p\_solde) qui ajoute (INSERT) une personne dans la table PERSONNE. On utilisera la fonction genIdPers().  
Afficher un message indiquant le nouvel identifiant utilisé : « *Personne 3 ajoutée* ».  
Tester en ajoutant la personne suivante : Lohan dont le solde est 200 puis supprimer la en utilisant la procédure suppers.
  - majSpect (p\_idSpect, p\_nbResa) qui met à jour (UPDATE) le nombre de places disponibles du spectacle passé en paramètre en le diminuant du nombre de places réservées passé en paramètre. Afficher le nombre de places restant.



La clause RETURNING INTO permet de renvoyer des valeurs de colonnes pour les lignes affectées par une instruction LMD (INSERT ou UPDATE)

```
UPDATE matable
SET chp = chp + valeur
WHERE ...
RETURNING chp INTO v_chp ;
```

c) Écrire la fonction suivante :

- `nbResa (p_idPers, p_idSpect)` qui renvoie le nombre total de places réservées par la personne pour le spectacle donné.

d) Écrire la procédure suivante :

- `majPers (p_idPers, p_idSpect, p_nbResa)` qui
  - Vérifie que le nombre de places disponibles du spectacle est suffisant (Exception avec affichage message « *nb de places disponibles insuffisant* »)
  - Vérifie que le solde de la personne est suffisant pour le nombre de places désiré en fonction du tarif du spectacle (Exception avec affichage message « *solde insuffisant* »)
  - Modifie, le cas échéant, le solde et insère une nouvelle réservation avec le nombre de places réservées pour la personne concernée à la date du jour.
  - Affiche le solde de la personne et le nombre total de places réservées pour le spectacle donné (utiliser la fonction précédemment créée).

**Remarque** : Effectuer une réservation complète consistera donc à appeler la procédure `majPers` puis `majSpect`. On sépare volontairement les deux traitements pour effectuer par la suite des tests sur l'exécution des transactions (TP09).

**Exemple** : La combinaison suivante :  
`EXEC majPers (1, 2, 4);`  
`EXEC majSpect (2, 4);`  
permet ainsi d'effectuer une réservation de 4 places pour la personne 1 au spectacle 2

a) Tester les procédures, pour cela :

- Ajouter la personne : Axel dont le solde est de 10€
- Afficher les infos de la personne 3
- Tenter de commencer une réservation pour la personne 3 au spectacle 1 de 2 places (solde insuffisant) : `EXEC majPers (p_idPers => 3, p_idSpect => 1, p_nb => 2)`
- Effectuer une réservation **complète** pour la personne 1 au spectacle 1 de 2 places
- Afficher les infos du spectacle 1 (reste 48 places) et de la personne 1 (César – 2 – 1900)
- Effectuer une réservation **complète** pour la personne 1 au spectacle 2 de 5 places
- Afficher les infos du spectacle 2 (reste 25 places) et de la personne 1 (César – 1750 – 7 places réservées)
- Valider les modifications (`COMMIT`)

### **Exercice N°3 : Les Packages**

a) **Création du package**

On désire créer un package pour regrouper certaines fonctions et procédures précédemment créées.

Créer un package en PL/SQL nommé *gestionSpec* contenant :

**Une fonction privée (uniquement dans le BODY) :**

- La fonction `genIdPers`

**Les procédures précédemment créées (à déclarer dans les Spécifications du package) :**

- infoPers (p\_idPers),
- infoSpect (p\_idSpect),
- suppPers (p\_idPers),
- ajoutPers (p\_nom, p\_solde),
- majSpect (p\_idSpect, p\_nbResa),
- majPers (p\_idPers, p\_idSpect, p\_nbResa)

**b) Utilisation du package**

Créer un bloc PL/SQL utilisant les procédures publiques du package créé :

- Ajout puis suppression d'un nouveau spectateur.

Appel de la procédure : *monPackage.maProc(param1, param2, ...)*

- Pour toutes les personnes, faire une réservation de 2 places pour le spectacle 1, afficher les informations de chaque personne.
- Afficher les informations du spectacle 1.