

T. P. 3 : PL/SQL et déclencheurs (triggers)

Rappel sur la base de données "vols-réservations"

ATTENTION : les tables sont déjà créées et remplies (utilisateur "TP1"), mais **vous devez les dupliquer sur votre compte** (en faisant un "clic droit" sur chaque table puis "copier" (avec le contenu)).

Exercice 1 : Le classique "Bonjour!"

Lancez SQLdeveloper et connectez vous au serveur Oracle de l'école.
Ecrivez un bloc PL/SQL anonyme qui affiche "Bonjour!" (voir ci-dessous).
Testez le.

```
SET serveroutput ON; -- pour activer les affichages
BEGIN
  DBMS_OUTPUT.PUT_LINE('Bonjour!');
END;
```

Exercice 2 : Bases du PL/SQL

1. Reprenez le bloc PL/SQL de l'exercice précédent. Définissez une variable "nbPlaces" de type entier, puis affectez lui le nombre total de places des réservations du vol V720. Affichez le contenu de cette variable dans un message de la forme "Nombre places V720 = ...". Testez.

2. Reprenez le bloc précédent et utilisez un test "IF THEN ELSE" pour savoir si le nombre total de places des réservations du vol V720 est inférieur ou égal à la capacité de l'avion 250. Si oui, afficher "capacite ok" sinon afficher "capacite insuffisante". Testez.

3. Modifiez le bloc précédent pour déclencher une exception si l'avion indiqué n'a pas de capacité renseignée. Le traitement de cette exception doit interrompre l'exécution (pour cela utiliser la procédure RAISE_APPLICATION_ERROR).

Testez avec l'avion 250 (cas "normal") et testez en indiquant un numéro d'avion n'ayant pas de capacité renseignée, par exemple le 320 (cas "exception").

Exercice 3 : Curseurs

Dans un bloc PL/SQL anonyme, déclarer un curseur permettant de lire les données suivantes : numéro de vol, numéro d'avion, date de départ, date d'arrivée des vols pour lesquels l'avion utilisé est le 101 ou le 720. Pour chaque vol lu par le curseur, calculer la durée du vol et l'afficher.

Exercice 4 : Procédures

Ecrivez une procédure stockée permettant de vérifier, pour un numéro de vol (NumVol) et un numéro d'avion (NumAv) transmis comme paramètres, si l'avion utilisé a une capacité au moins égale au nombre total de réservations sur le vol.

Si la capacité est suffisante, 1 sera retourné. Sinon, 0 sera retourné.

Ecrivez aussi un bloc anonyme pour tester cette procédure dans le cas où elle doit retourner 1 et dans le cas où elle doit retourner 0.

Remarque : la procédure doit être définie à l'aide de CREATE PROCEDURE dans la fenêtre de SQLdeveloper. Vous pouvez retourner 0/1 ou bien utiliser un **paramètre out**.

Exercice 5 : Déclencheurs (Triggers)

La base de données "vols-réservations" restera cohérente si elle vérifie les contraintes précisées dans cet exercice.

Rédigez un déclencheur (trigger) pour chacune d'entre elles.

Effectuez une **série de tests** pour chaque contrainte afin de vérifier le bon fonctionnement dans les cas "positifs" et les cas "négatifs".

1. Contrainte 1 : une réservation ne peut pas être passée sur un vol dont le départ a déjà eu lieu.
2. Contrainte 2 : il est impossible de supprimer une réservation relative à un vol en cours.
3. Contrainte 3 : pour chaque vol, la somme des coefficients CoeffPlace de chaque classe doit être inférieure ou égale à 1.
4. Contrainte 4 : à un instant donné, un pilote assure au plus un vol.
5. Contrainte 5 : à un instant donné, un avion est au plus utilisé pour un vol.