

## TP PL/SQL

### Tables de la base :

#### Avion :

Numav	Capacite	Type	Entrepot
14	25	A400	Garches
345	75	B200	Maubeuge

#### Pilote :

Matricule	Nom	Ville	Age	Salaire
1	Figue	Cannes	45	28004
2	Lavande	Touquet	24	11758

#### Passager :

Numab	Nomab
1	Nifance
8	Téarice

#### Vol :

Numvol	Heure_depart	Heure_arrivee	Ville_depart	Ville_arrivee
AF118	08:30	10:57	Paris	Athens
AF212	09:21	14:10	Paris	Moscow
AF178	12:56	14:15	Paris	London
TA215	08:00	10:10	Tunis	Paris
OA005	14:20	17:00	Athens	Paris
SA854	22:00	10:14	Singapore	Athens
AA111	15:45	21:10	Beijing	Singapore
AF218	21:12	09:16	Beijing	Paris
SA012	07:57	11:26	Sydney	Singapore
AF109	07:39	14:10	Tahiti	Sydney
AA517	23:57	07:12	Honolulu	Tokyo
JA014	15:35	19:00	Tokyo	Beijing
AF002	15:52	00:12	Tokyo	Paris
JA115	21:26	10:10	Los Angeles	Tokyo
AA015	20:50	07:00	New York	Lima
AA515	07:20	12:38	New York	Los Angeles
AF010	07:53	14:19	Paris	New York
AF012	07:58	20:10	Paris	Los Angeles
AA118	07:15	13:10	New York	Paris
AF001	22:10	12:00	Paris	Tahiti
PA022	10:12	23:55	Lima	Paris

#### Depart :

Numvol	Date_dep	Numav	Matricule
AF118	31-12-95	14	1
AF212	19-12-95	345	2

#### Reservation :

Numab	Numvol	Date_dep
1	AF118	31-12-95
8	AF118	31-12-95

### Escales :

Numescale	Ville_escale	Duree_escale
1	Moscou	5
2	Singapour	5
3	Sydney	4
4	Tahiti	4
5	Honolulu	4
6	Los Angeles	5
7	New York	4
8	Londres	3

## 1. Généralités SQL

Créez une table `VotreNom.Vol` en utilisant

```
CREATE TABLE Vol (Numvol VARCHAR2(8), Heure_depart DATE,
                   Heure_arrivee DATE, Ville_depart VARCHAR2(20),
                   Ville_arrivee VARCHAR2(20))
```

Ecrivez les instructions qui copient la table `Vol` dans la table `VotreNom.Vol` (les 2 tables ont la même structure).

Remarque : la fonction `TO_DATE` convertit des chaînes de caractères en dates, suivant le format (voir la documentation en ligne).

A partir de maintenant, pour éviter que d'éventuelles erreurs ne se répercutent sur l'ensemble du groupe, vous vous servirez exclusivement de vos tables.

Insérez une nouvelle ligne (valeurs de votre choix) dans la table `VotreNom.Vol` en utilisant :

```
COMMIT;
BEGIN
    INSERT INTO VotreNom.Vol (valeurs de votre choix, attention aux types);
END
/
```

Quittez SQL\*Plus en fermant brutalement la fenêtre. Reconnectez-vous et vérifiez si l'enregistrement que vous avez inséré est présent dans la table.

Refaites l'insertion et quittez ensuite SQL\*Plus avec `exit`. Reconnectez-vous et vérifiez si l'enregistrement que vous avez inséré est présent dans la table.

## 2. Fonctions et procédures

Expliquez les résultats obtenus par l'exécution du programme PL/SQL suivant :

```
DECLARE
    n NUMBER := 10;
    PROCEDURE etudeNocopy
        (n1 IN NUMBER, n2 IN OUT NUMBER, n3 IN OUT NOCOPY NUMBER) IS
    BEGIN
        n2 := 20;
        DBMS_OUTPUT.PUT_LINE(n1);
        n3 := 30;
        DBMS_OUTPUT.PUT_LINE(n1);
    END etudeNocopy;
BEGIN
    etudeNocopy(n, n, n);
```

```
        DBMS_OUTPUT.PUT_LINE(n) ;  
END ;
```

### 3. Paquetages

Reprendre les exercices 3 et 4 des ED PL/SQL (voir ci-dessous) et les écrire sous forme de procédures stockées avec comme paramètre la ville de départ.

1. Surcharger la procédure de l'exercice 4 en lui ajoutant les deux bornes supérieure et inférieure limitant le nombre d'escales.

2 Regrouper toutes les fonctions dans le même paquetage.

**Exercice 3 (ED PL/SQL) :** Ecrivez un programme PL/SQL qui calcule la moyenne des salaires des pilotes dont l'âge est entre 30 et 40 ans.

**Exercice 4 (ED PL/SQL) :** Ecrivez un programme PL/SQL qui propose les vols pour un tour du monde au départ de Paris avec des escales et des durées d'escale prédéfinies dans la table `Escales`. Le nombre d'escales à faire doit être demandé à l'utilisateur. Hypothèse de travail : pour chaque escale il existe plusieurs vols satisfaisant les contraintes.

### 4. Déclencheurs

Ecrivez un déclencheur qui, avant un `INSERT` dans la table `vol`, vérifie si un vol retour est présent dans la table (pour que la destination soit desservie dans les 2 sens) et, sinon, affiche un avertissement à la console.

Essayez de déclarer le déclencheur `AFTER` au lieu de `BEFORE`. Que se passe-t-il ? Pourquoi ?

Ecrivez un déclencheur qui, avant chaque insertion ou mise à jour du numéro de vol ou du numéro d'avion dans la table `Depart`, vérifie si l'avion employé a une capacité au moins égale au nombre de réservations et a comme entrepôt la ville de départ. Si les deux conditions sont satisfaites la modification est faite. Si la capacité est insuffisante et/ou l'entrepôt n'est pas la ville de départ, des messages spécifiques sont affichés.